MICRO
#761

MICRO
#761

MICRO 2011 Submission #761. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

Cache Revive: Tuning Retention times of STT-RAM Caches for Enhanced Performance in CMPs.

Anonymous MICRO submission

Paper ID 761

Abstract

Spin-Transfer Torque RAM (STT-RAM) is a CMOS compatible emerging non-volatile memory (NVM) technology that has the potential to replace the conventional on-chip SRAM caches for designing a more efficient memory hierarchy for future multicore architectures. However, it's high write latency and dynamic write energy are major obstacles for being competitive with the SRAM-based cache hierarchy. On the other hand, STT-RAM technology has another adaptable feature that it is possible to reduce its write latency by reducing its retention time, thereby making it volatile. In this paper, we exploit this volatile property of the STT-RAM for designing an efficient L2 cache architecture. The paper addresses several critical design issues such as how do we decide a suitable retention time for last level cache, what is the relationship between retention time and write latency, and how do we architect the cache hierarchy with a volatile STT-RAM. Through an extensive execution driven analysis of the inter-write time of several PARSEC and SPEC 2006 banchmarks, we observe that retention time in the order of 10-40 ms is a good design point to handle most of the writes. Then for the rest of the cache blocks that have a higher inter-write time than the STT-RAM retention time, we propose an architectural solution to identify these blocks with a per block 2 bit counter, temporarily save a limited number of MRU blocks in a buffer, and writeback the rest of the dirty blocks to avoid any data loss. Our experiments with 4

1

MICRO
#761

MICRO 2011 Submission #761. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

MICRO
#761

and 8-core architectures with an SRAM-based L1 cache and STT-RAM-based L2 cache indicate that not only we can eliminate the high write overhead of an NVM STT-RAM, but can provide on an average 10-12% improvement in IPC compared to the traditional SRAM-based design, while reducing the energy consumption significantly

1. Introduction

Spin-Transfer Torque RAM (STT-RAM) is a promising memory technology that delivers on many aspects desirable of an universal memory. They exhibit high density, fast read times and low static power consumption. However, the high write latencies and write energy are key drawbacks of this technology. Consequently, recent efforts have focused on masking the effects of high write latencies and write energy at the architectural level. In contrast to these architectural approaches, a recent technique considers relaxing STT-RAM data retention times to reduce both write latencies and write energy. The focus of this paper is to tune this data retention time to closely match the required lifetime of cache line blocks to achieve significant performance and energy gains.

The non-volatile nature and non-destructive read ability of STT-RAM provides a key difference with regard to a comparably high-density DRAM memory. However, for many applications it is sufficient if the data stored in a cache hierarchy remains valid for a few tens of milliseconds. Consequently, the duration of data retention in STT-RAM is an obvious candidate for device optimization for cache design. First, we analyze how changes to the STT-RAM retention times influence the performance, power and area characteristics. A key distinction from prior efforts to relax data retention times is our consideration of device variability in our analysis (NEED SOME THING FROM CONG ON THIS IN HIS TEXT). This analysis reveals the granularities at which a device designer can reliably tune the data retention times.

Analyzing the lifetime of cache lines has been the focus of prior efforts to improve performance and reduce power consumption. In this work, we revisit this topic with the aim of identifying the suitable data retention times for STT-RAM caches. A key challenge in determining a suitable data retention

MICRO
#761

MICRO 2011 Submission #761. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

MICRO
#761

times for the STT-RAM is to balance the reduced write latency of cells with lower retention time with the overhead for data refresh or writeback of cache lines with longer lifetimes. Our analysis of the cache lifetimes performed for a multi-threaded workload demonstrates that a significant fraction of L2 cache lines can operate correctly without any additional support when the STT-RAM retention times are of the order of 50ms. However, architectural support is required to ensure that correct program state is maintained for the rest of the cache lines that have lifetimes exceeding 50ms. While a simple DRAM-style refresh has been proposed in [Sudhanva-et.al] to ensure correctness, it is possible to avoid many of these refresh by pursuing a life-time aware refresh strategy.

This work makes the following contributions

- We present a detailed device characterization of data retention tunability in STT-RAM Cells providing insight to the underlying principles enabling these tradeoffs

- We analyze the time between writes or replacements to a cache line for various multi-threaded and multi-programmed workloads. Our characterization augments the prior body of work that analyzes cache lifetimes mainly in single processor and single program configurations.

- We present a simple buffering mechanism to ensure integrity of programs given the volatile nature of our tuned STT-RAM cells.

- Finally, we show that our combined device-architecture life time tuning approach is better than recent efforts that attempt to address the long write latencies of STT-RAM.

The rest of this paper is as follows.

Next, we evaluate the lifetimes of cache lines when executing multi-threaded workloads. A cache line is no longer required, if the

for the duration of its lifetime in the program execution.

The cache hierarchy is a key component influencing both the

2. Understanding tradeoffs of Retention Time

In this section, we will understand the need of an ideal retention time which not only is suited for most applications but also satisfy the device limitations of fabricating low retention STT-RAM caches.
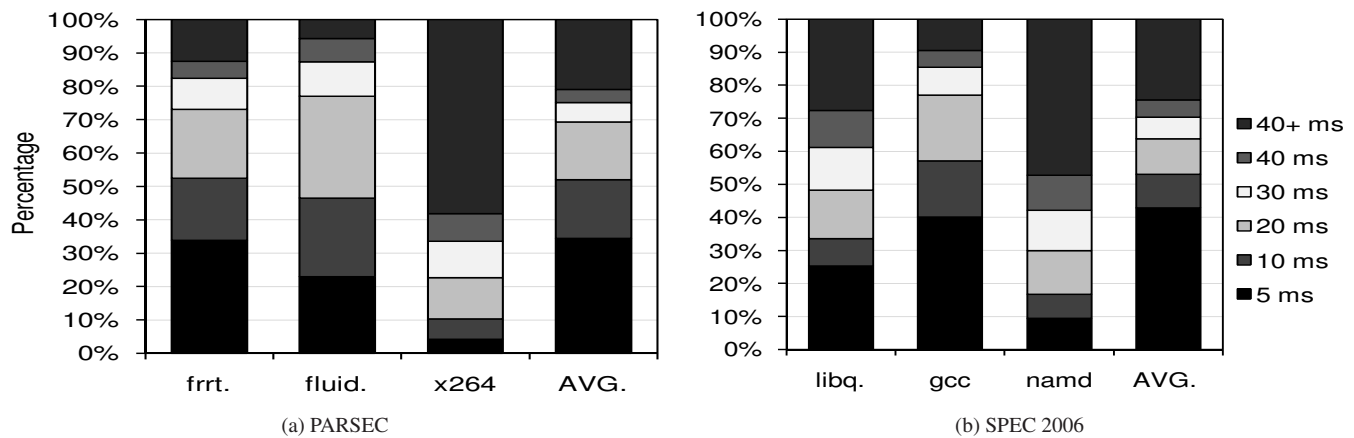
MICRO
#761

MICRO
#761

MICRO 2011 Submission #761. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

Figure 1. **Distribution of Blocks Showing Different Revival Times**

2.1. Relating Application Characteristics to Retention Time

Application characterization gives the basis for evaluating the impact of low retention time STT-RAM caches on the overall system performance. In order to do this characterization, the first step is to find the ideal time for which the cache block should retain the data. We see that, cache block is only refreshed when the block is written. Since we are using low retention time STT-RAM for L2 cache, we record intervals between two successive writes (refreshes) to the same L2 cache block. We define this interval to be *revival time*. While collecting these results, we ensure that if a block gets invalidated in between two consecutive writes, we don't consider the time in between the invalidation and the next write. Previous works [?] do similar type of revival time analysis but for L1 cache. Figure 1 shows the distribution of L2 cache blocks having different revival time intervals. These results are obtained by running multi-threaded (PARSEC [?]) and multi-programmed (SPEC 2006 [?]) applications on M5 Simulator [?] which models a 4GHz processor consisting of 4 cores, with 4MB STT-RAM L2 cache. Table ?? contains additional details of the system configuration. Figure 1 (a) and (b) shows results of three PARSEC and SPEC benchmarks along with the averages across 12 PARSEC and 14 SPEC benchmarks respectively. We observe from the figure that, on an average, approximately 50% of cache blocks get refreshed within 10 ms. About 20% of blocks remain in the cache for more than 40 ms and rest of the blocks have intermediate revival times. We conclude that blocks which stay longer than the retention time in the cache without being refreshed are going to affect the application performance the

MICRO
#761

MICRO 2011 Submission #761. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

MICRO
#761

Table 1. Retention and Write Latencies for STT-RAM L2 Cache

Retention Time	10years	1sec	500ms	100ms	10ms
Write Latency @4GHz	44 cycles	24 cycles	20 cycles	16 cycles	12 cycles

most. The opportunities lie with the computer architect to design micro-architecture schemes to make the unrefreshed block percentage as low as possible. 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 too volatile leading to degraded performance. In the next subsection, we will see how increasing the retention time, has negative impacts on write latency.

2.2. Low Retention STT-RAM Characteristics

Table 1 shows that there is significant reduction in write latency with reduction in retention time. Section 4 explains how these numbers originate. We want to clarify from device fabrication perspective that, these retention times are the most stable designs possible. As we lower the retention times of these STT-RAM cells in the range of *ms* it becomes much harder to precisely mark a STT-RAM cell with a fixed retention time. For the sake of correctness and preciseness we discuss these designs only in the paper. Later in the Section 7, it will be clear, that our design assumptions have no affect on the generality of the results.

To analyze the tradeoffs between retention time and overall system performance, lets consider an utopian cache with 10 year retention time having minimum write latency and energy. To bridge the gap between current and utopian cache, we need to reap the benefits of both: application characteristics and emerging device technology. From application side, it is best to choose a retention time which minimizes the unrefreshed blocks and from technology side it is ideal to choose the STT-RAM with minimum write latency and energy. We choose 10 ms retention time as optimal retention time which balances both the sides. In Section 7, we do a sensitivity analysis by choosing retention times 100 ms, 500 ms and 1sec. In Section 5, we propose micro-architecture techniques to deal with blocks having revival time greater than 10ms.

MICRO
#761

MICRO
#761

MICRO 2011 Submission #761. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

3. STT-RAM Design
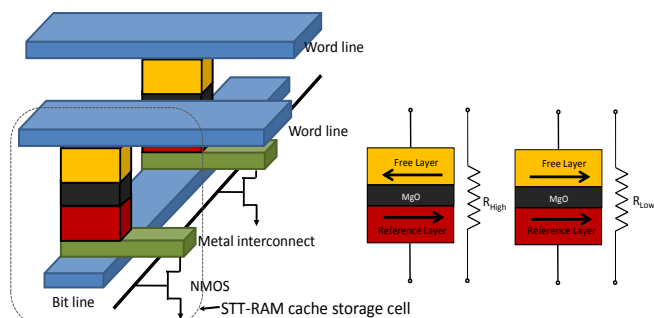
3.1. STT-RAM Cell Basics

Figure 2. **(a) Structural view of of STT-RAM Cache Cell (b) Anti Space Parallel (High Resistance, Indicating "1" state (c) Parallel (Low Resistance, Indicating "0" state**

STT-RAM uses Magnetic Tunnel Junction (MTJ) as the memory storage and leverages the difference in magnetic directions to represent the memory bit. As shown in Fig. 2, MTJ contains two ferromagnetic layers. One ferromagnetic layer is 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. If two ferromagnetic layers have the same directions, the resistance of MTJ is low, indicating a "1" state; if two layers have different directions, the resistance of MTJ is high, indicating a "0" state.

As shown in Fig. 2, when writing "0" state into STT-RAM cells, positive voltage difference is established between SL and BL; when writing "1" state, vice versa. The current amplitude required to reverse the direction of the free ferromagnetic layer is determined by the size and aspect ratio of MTJ and the write pulse duration.
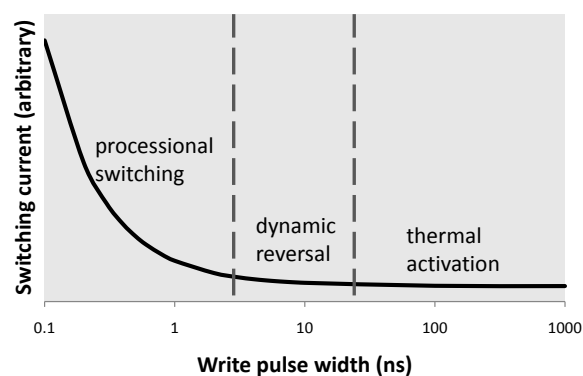
3.2. Write current versus write pulse width trade-off

Figure 3. **(a) Structural view of of STT-RAM Cache Cell (b) Anti Space Parallel (High Resistance, Indicating "1" state (c) Parallel (Low Resistance, Indicating "0" state**

The current amplitude required to reverse the direction of the free ferromagnetic layer is determined by a lot of factors such as material property, device geometry and importantly the write pulse duration. Generally, the longer the write pulse is applied, the less the switch-

6

MICRO
#761

MICRO
#761

MICRO 2011 Submission #761. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

ing current is needed to switch the MTJ state.

Three distinct switching modes were identified [?] according to the operating range of switching pulse width τ: thermal activation ($\tau > 20ns$), processional switching ($\tau < 3ns$) and dynamic reversal ($3ns < \tau < 20ns$).

The relationship between switching current density J_c and write pulse width τ was characterized by an analytical model in [?]. The equations are listed as follows,

$$
\begin{aligned}
J_{c,TA}(\tau) &= J_{c0}\{1 - (\frac{k_B T}{E_b})ln(\frac{\tau}{\tau_0})\} & (1) \\
J_{c,PS}(\tau) &= J_{c0} + \frac{C}{\tau^\gamma} & (2) \\
J_{c,DR}(\tau) &= \frac{J_{c,TA}(\tau) + J_{c,PS}(\tau)e^{-k(\tau-\tau_c)}}{1 + e^{-k(\tau-\tau_c)}} & (3)
\end{aligned}
$$

where $J_{c,TA}$, $J_{c,PS}$, $J_{c,DR}$ are the switching current densities for thermal activation, precessional switching and dynamic reversal respectively. J_{c0} is the critical switching current density, k_B is the Boltzmann constant, T is the temperature, E_b is the thermal barrier, and τ_0 is inverse of the attempt frequency. C, γ, k, and τ_c are fitting constants. Based on the observation from Fig. 3 and analysis of the analytical model, we found very different switching characteristics in the three switching modes. For example, in thermal activation mode, the required switching current increases very slowly even we decrease the write pulse width by orders of magnitude, thus short write pulse width is more favorable in this regime because reducing write pulse can reduce both write latency and energy without much penalty on read latency and energy. While in processional switching, write current goes up rapidly if we further reduce write pulse width, therefore minimum write energy of the MTJ is achieved at some particular write pulse width in this regime. Consequently, this paper will focus on the exploration of write pulse width in processional switching and dynamic reversal to optimize for different design goals.

MICRO
#761

MICRO
#761

MICRO 2011 Submission #761. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

3.3. STT-RAM Cell Area Modeling

To simulate the performance of STT-RAM cache, it is important to estimate its cell area first. As mentioned before, each 1T1J STT-RAM cell is composed of one NMOS and one MTJ. The NMOS access device is connected in series with the MTJ. The size of NMOS is constrained by both SET and RESET current, which are inversely proportional to the writing pulse width. In order to estimate the current driving ability of MOSFET devices, a small test circuit using HSPICE with PTM 45nm HP model [**?**] is simulated. The BL-to-SL current and SL-to-BL current are obtained by assuming typical TMR (120%) and LRS ($3k\Omega$) value [**?**] and bursting wordline voltage to be 1.5V (the optimal value is extracted from [**?**]). And we over size the access transistor width to guarantee enough write current provided to MTJ using the methodology discussed in [**?**]. To achieve high cell density, we model the STT-RAM cell area by referring to DRAM design rules [**?**]. As a result, the cell size of a STT-RAM cell is calculated as follows,

$$\text{Area}_{\text{cell}} = 3\,(W/L + 1)(F^2) \tag{4}$$
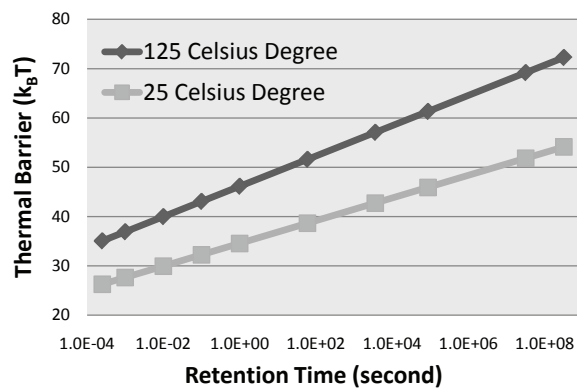
3.4. Impact of MTJ Retention Time on STT-RAM

Figure 4. **MTJ thermal stability requirement for different retention time**

The retention time of a MTJ is largely determined by the thermal stability of the MTJ. The relation between retention time and thermal barrier is captured in Figure 4, which can be modeled as $t = C \times e^{k\Delta}$, where t is the retention time and Δ is the thermal barrier while C and k are fitting constants. Thermal stability of the free layer in an MTJ does not only have impact on retention time of STT-RAM memory cell but also on the write current. It was found in **??** that the switching current of MTJ increases almost linearly with thermal barrier when thermal barrier is $< 70k_BT$, where k_B is the Boltzman constant and T is temperature. Here we combine this observation with the write current versus write time trade-off

8

MICRO
#761

MICRO
#761

MICRO 2011 Submission #761. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

Table 2. 16-way L2 Cache Simulation Results

			Area (mm^2)	Read Latency (ns)	Write Latency (ns)	Leakage Power (mW)
1MB SRAM			2.612	1.012	1.012	4542
4MB STT-RAM	$t = 10yr$	Leakage Opt.	2.628	2.434	4.919	1399
		Latency Opt.	3.003	0.998	10.61	2524
	$t = 1s$	Leakage Opt.	2.203	2.044	3.552	1388
		Latency Opt.	2.904	0.973	5.571	2235
	$t = 10ms$	Leakage Opt.	2.167	1.956	3.390	1151
		Latency Opt.	2.901	0.959	2.598	2227

described in Section 3.2, which essentially means that once the thermal barrier of a MTJ is lowered we are able to achieve faster write speed or/and smaller write current/energy. The most straightforward way to reduce thermal barrier is to tune device geometry such as planar area, thickness of free layer and aspect ratio of the elliptic MTJ.

3.5. STT-RAM Cache Simulation Setup

We simulate SRAM-based caches and STT-RAM-based caches with a tool called NVsim [**?**], which is a circuit-level performance, energy, and area simulator based on CACTI for emerging non-volatile memories. All the models described in this Section has been integrated in NVsim. The simulation results are listed in Table 2. We can see that the leakage-optimized $4MB$ non-volatile STT-RAM cache has almost the same area with $1MB$ SRAM. This is consistent with previous work [**?**]. By relaxing retention time of STT-RAM with lower thermal barrier, the leakage-optimized STT-RAM cache can have smaller area, faster write latency and less leaky peripheral circuity. However, the read latency of leakage-optimized $4MB$ STT-RAM cache is significantly larger than $1MB$ SRAM cache because sensing the state of STT-RAM cell takes longer and fast SRAM sensing. Thus, we reduce the array size to improve the latency of STT-RAM cache. As can be seen in Table 2, the latency-optimized STT-RAM cache has noticeable better delay with $14\% - 34\%$ area overhead compared to leakage-optimized STT-RAM cache with the same retention time.
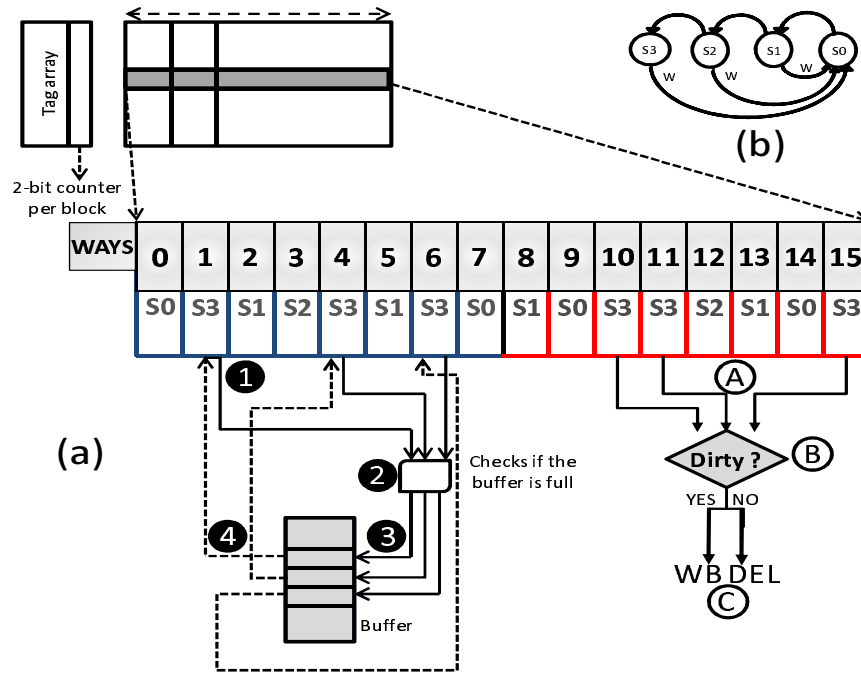
MICRO
#761

MICRO
#761

MICRO 2011 Submission #761. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

Figure 5. **Architecture**

4. Architecting Volatile STT-RAM

We observe from the figure 1 that approximately 50% of L2 cache blocks are refreshed within 10ms interval. The substantial number of blocks after this interval makes a case for strong architecture support. Figure **??** shows the schematic of overall architecture design. In the this section we will first describe the main components of our architecture and later in the section we discuss the impact of our schemes.

4.1. Main Components

Counter Design: We maintain a 2 bit counter per physical block similar to the one used in [**?**]. Figure **??** (b) shows the transition diagram for the counter. Each cache block can be in one of the four states. We divide the retention time of the STT-RAM cell into four equal parts and define each part as *transition time*. After every *transition time*, block goes from one state to another according to the transition diagram. The block goes back to *S0* state if the data is refreshed. The block in *S3* state indicates that it has attained its retention time and action needs to be taken. We calculated the overhead of 2 bit counters is 1.6% over L2 cache size.

Buffer Overflow detector: We keep a per bank small buffer with fixed number of entries made up of
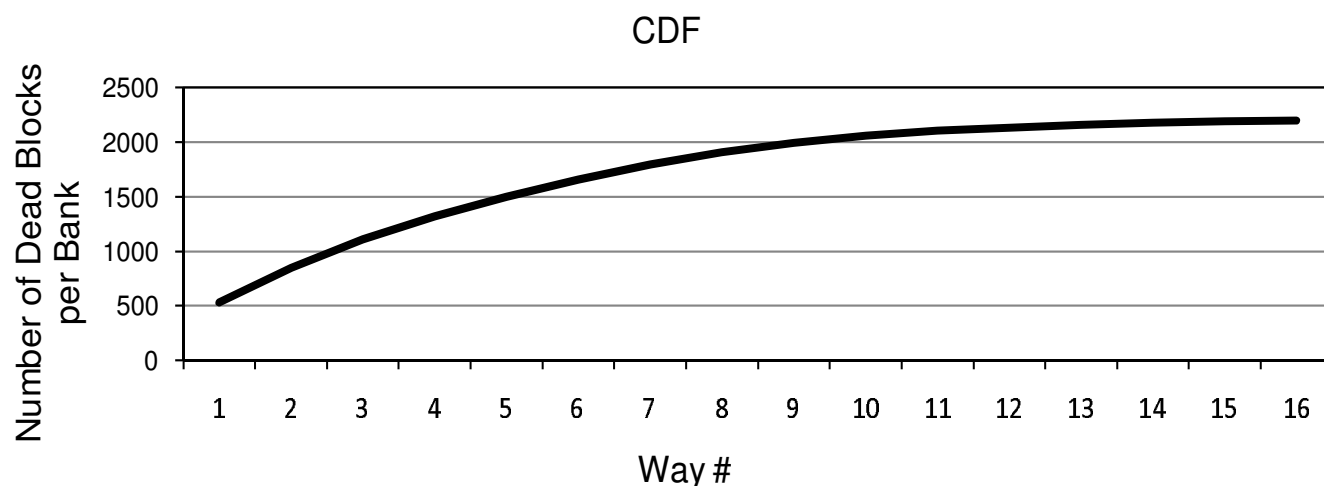
MICRO

#761

MICRO 2011 Submission #761. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

MICRO

#761

CDF

Figure 6. **CDF**

low-retention time STT-RAM cells, to temporarily store dead blocks. The optimal size of this buffer is

calculated later in the section.

Buffer Controller We maintain a 12 bit overflow detector per buffer and 12 bit block identifier associ-

ated with every buffer entry in the buffer controller. If a dead block is directed to the buffer, overflow

detector is checked to see it the buffer is already full or not. If it is not full then, block id is generated

by concatenating its set and way id and stored along with actual block in one of the empty buffer slots.

If the buffer is full, the blocks is checked whether it is dirty or not. If not, they are written back to the

main memory and invalidated from the cache.

4.2. Volatile STT-RAM design

If a block reaches S3 states, it needs to be dealt properly to make sure that no data is lost. In a naive

design, if a particular block is dirty and is in S3 state, then it is written back to the main memory. It has

negative impact on the performance for two reasons: 1) There will be lots of write backs to the main

memory after every interval. 2) The dying block, could have been frequently read and losing it will incur

additional expensive read misses.

4.3. Revived STT-RAM design

Revived STT-RAM Scheme is hereby proposed. We try to find the position of the block from MRU.

If the block is in certain slots from the MRU, we will use a small buffer bank to copy the dying block in

11

the buffer and again copy back to the same slot. If the block is not in chosen MRU slots we This method has multiple advantages: 1) It helps in preserving the frequently used block, which is most likely is frequently read. 2) It prevents unnecessary write backs from L2 cache to main memory.

Our scheme consists of following parameters to be tuned. 1) Number of MRU Slots 2) Number of Buffer Slots 3) Possible retention times. 9 suggests that if we consider first eight MRU slots for buffering, we cover x percentage of blocks.

5. Experimental Evaluation

Experimental Setup We evaluate our design and schemes on modified ALPHA M5 Simulator [] . We operate M5 Simulator in Full System (FS) mode for PARSEC applications and in System Emulation (SE) Mode for SPEC 2006 Multiprogammed mixes. We model a 4GHz processor with four out of order cores. We modified M5 simulator to model low retention time STT-RAM for L2 cache. The L2 cache is banked with different read and write latencies, with all the banks connected via a shared memory bus. We assume a fixed 400 cycles main memory latency for all our simulations. Table [] details our experimental system configuration.

Collection of Results We report results of 12 mulithreaded PARSEC applications and 14 SPEC 2006 multiprogrammed mixes. Table **??** shows the characterization of PARSEC applications and list of multiprogrammed mixes. We skip the initilization and termination phases of PARSEC benchmarks and report the results of only Region of Interest (ROI) for all PARSEC applications (except facesim, where we report results for only 2B instructions of ROI), after warming up caches for 100M Instructions in ROI. For SPEC multiprogrammed mixes, we fast forward 1B Instructions, warm up caches for 500M instructions and then report results for 1B instructions.

Performance Metrics For mulltithreaded PARSEC applications, we assume 4 threads are mapped to our modeled processor with four cores. We report normalized speedup for these applications, which is defined as the decrease in execution time of the slowest thread. We randomly choose 14 multiprogram mixes, each mix with four different applications and assign each every application to a core. We report Instruction throughput and Weighted Speedup for SPEC multiprogrammed mixes, which are defined as:
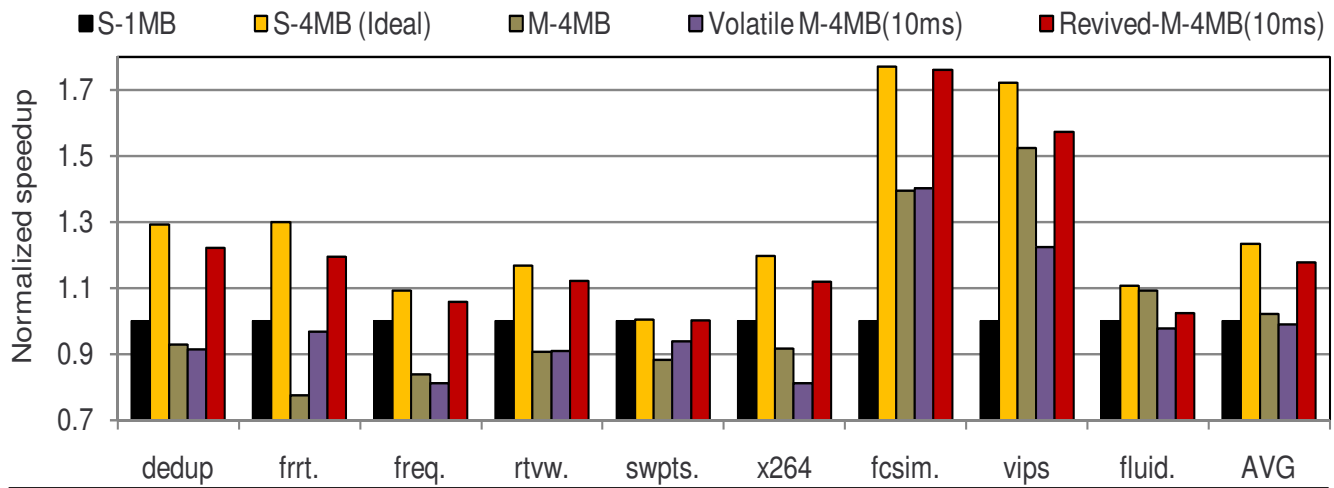
MICRO

#761

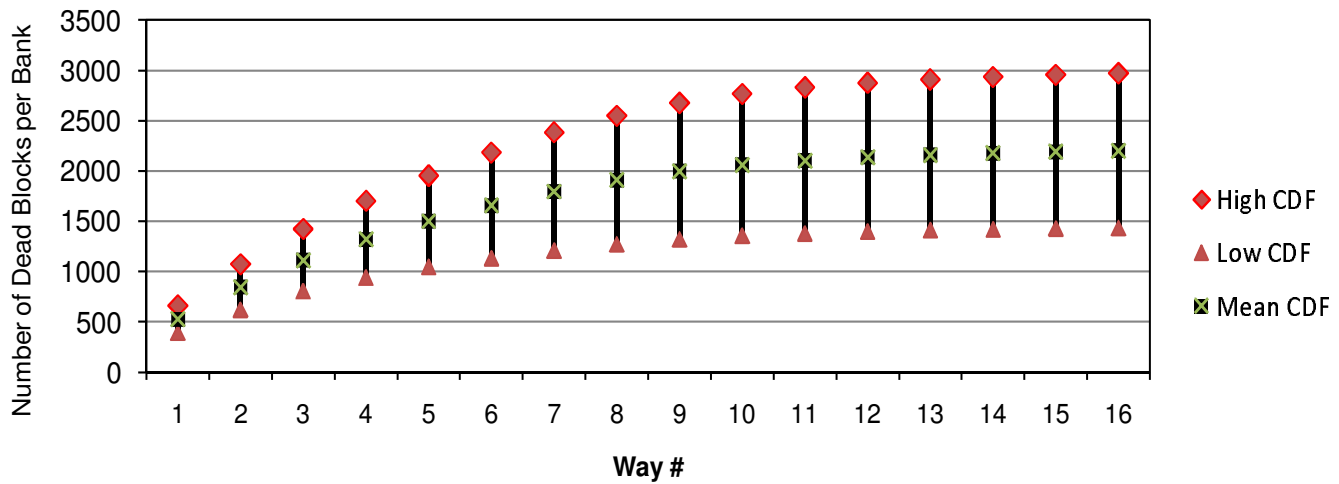MICRO 2011 Submission #761. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

MICRO

#761

Figure 7. **Normalized speedup**

Figure 8. **Confidence Intervals**
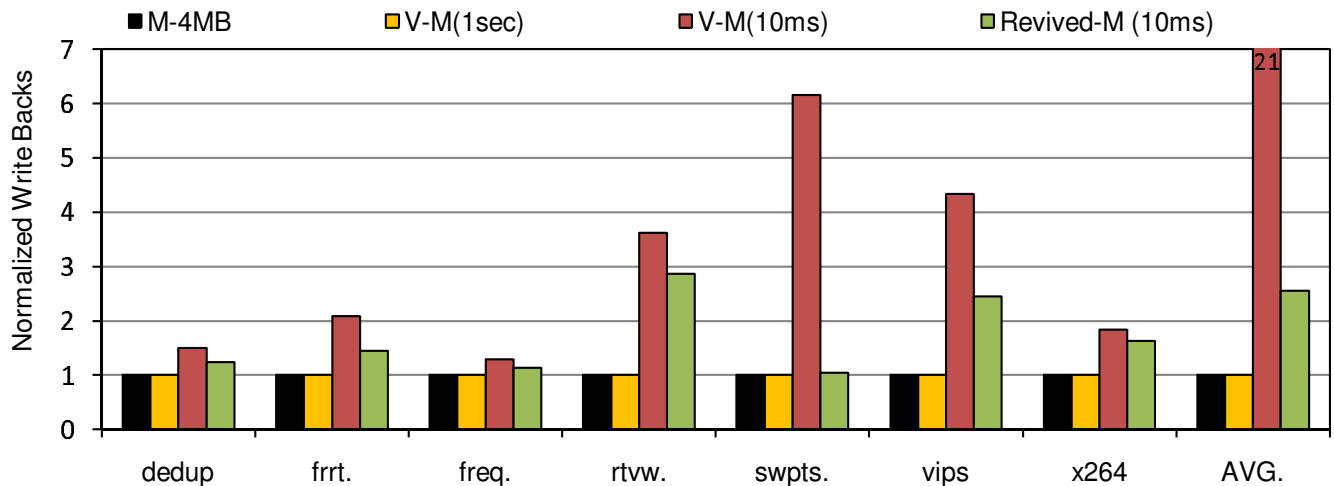
Figure 9. **Write backs**

6. Results

MICRO

#761

MICRO 2011 Submission #761. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

MICRO

#761
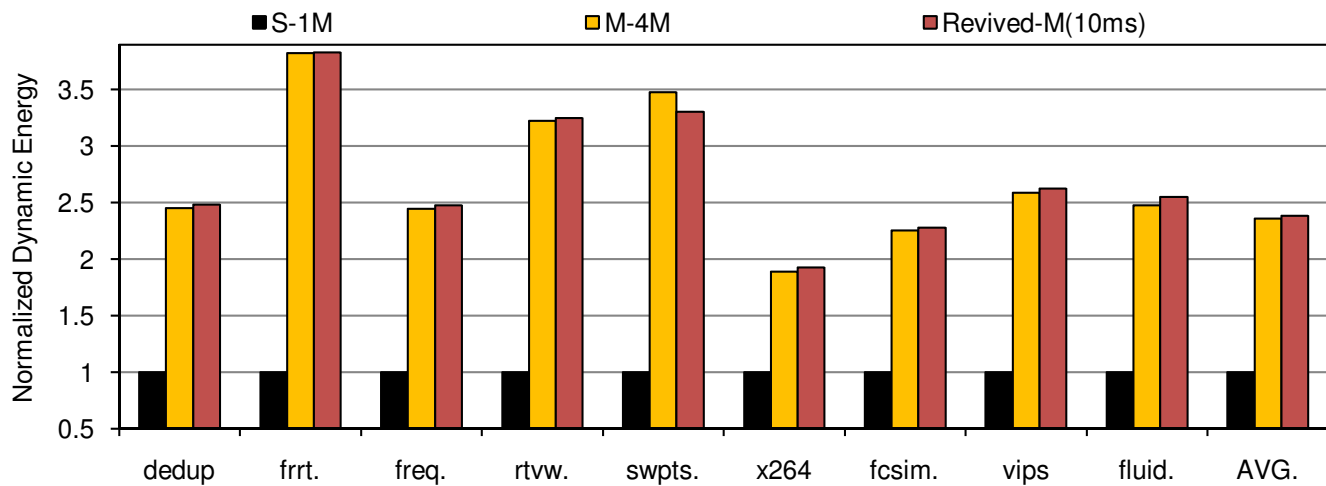
Figure 10. **Dynamic Energy**
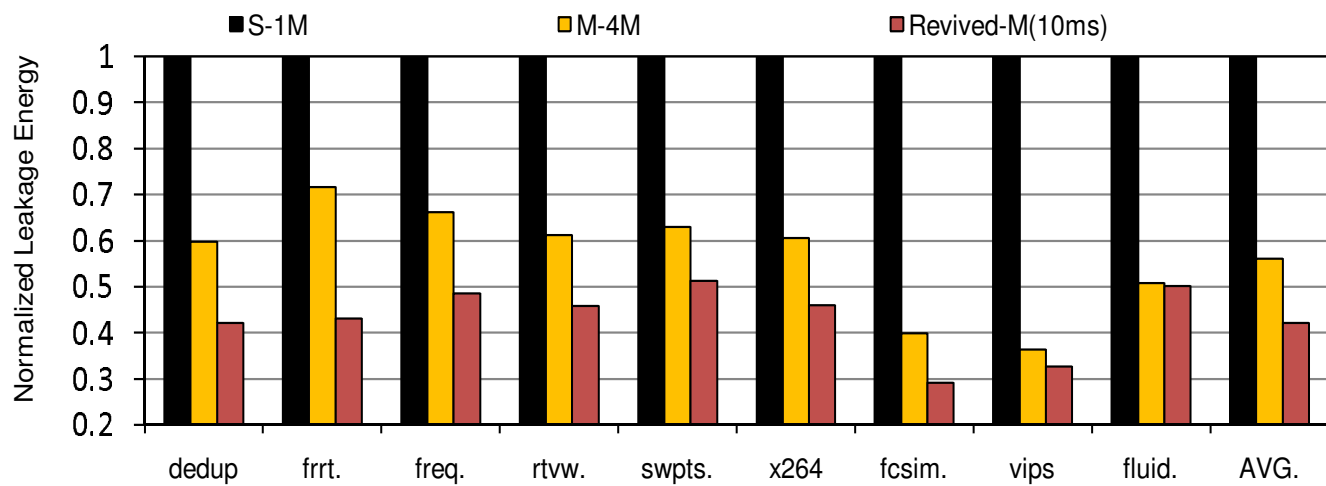
Figure 11. **Leakage**

Figure 12. **Total Energy**

MICRO
#761

MICRO
#761

MICRO 2011 Submission #761. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.
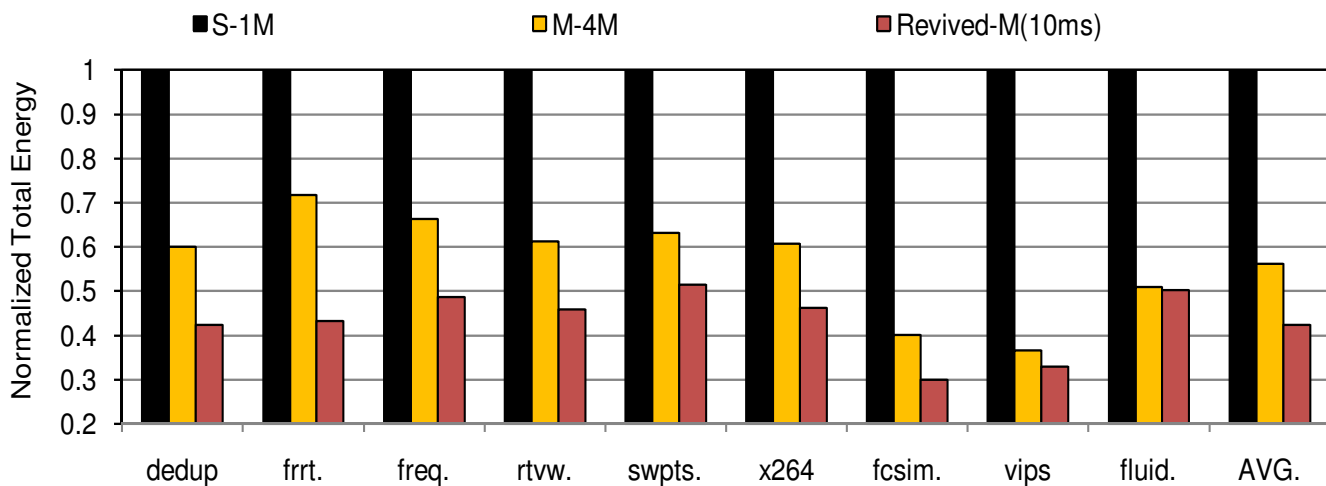
Normalized speedup 7 shows normalized speedup of PARSEC multithreaded applications. S-1MB stands for 1MB SRAM Bank, M-4MB non-volatile STT-RAM Bank, Volatile M-4MB for 4MB STT-RAM Bank with no buffers, and Revived M-4MB with buffers. We observe that our scheme is giving 11% IPC improvement over traditional

Weighted Speedup and Instruction Throughput

Normalized Energy

6.1. Sensitivity Analysis

Effect of variation in retention times

Choosing correct buffer slots

Number of bits for counter

6.2. Scaling Effects

7. Prior Work

8. Conclusions

Sample bibliography [1]".

References

[1] Authors. Frobnication tutorial, 2011. Supplied as additional material tr.pdf. 15