# Energy Reduction for STT-RAM Using Early Write Termination*

Ping Zhou, Bo Zhao, Jun Yang, Youtao Zhang‡
Electrical and Computer Engineering Department,‡ Department of Computer Science
University of Pittsburgh, Pittsburgh, PA 15261
{piz7, boz6, juy9}@pitt.edu, ‡zhangyt@cs.pitt.edu

## ABSTRACT

The emerging Spin Torque Transfer memory (STT-RAM) is a promising candidate for future on-chip caches due to STT-RAM's high density, low leakage, long endurance and high access speed. However, one of the major challenges of STT-RAM is its high write current, which is disadvantageous when used as an on-chip cache since the dynamic power generated is too high.

In this paper, we propose Early Write Termination (EWT), a novel technique to significantly reduce write energy with no performance penalty. EWT can be implemented with low complexity and low energy overhead. Our evaluation shows that up to 80% of write energy reduction can be achieved through EWT, resulting 33% less total energy consumption, and 34% reduction in $ED^2$. These results indicate that EWT is an effective and practical scheme to improve the energy efficiency of a STT-RAM cache.

## 1. INTRODUCTION

The demand of large, fast and low power on-chip cache has been increasing as more cores are integrated in a single die, i.e., Chip Multiprocessors (CMP). Traditional SRAM cache cannot meet this need since it is constrained by its low density and high leakage power. Recently, there is an increasing interest in using nonvolatile memory technologies in disk, main memory or even on-chip cache due to their high device density, near-zero cell leakage and resilience to single event upset. Among several new memory technologies, both Phase Change Memory (PCM) and Magnetic Random Access Memory using Spin Torque Transfer technology (STT-RAM) [12] are promising candidates for next generation memory hierarchy.

While PCM has higher density than STT-RAM, its long read/write latency (longer than DRAM [20]) and limited write endurance ($10^8 \sim 10^9$ writes [4, 18, 19]) make it more suitable for main memory instead of cache [20]. STT-RAM, on the other hand, has nearly the same read speed as SRAM [2] and virtually no write endurance problem ($10^{15}$ writes [15]). Although STT-RAM is not as dense as PCM, it is still approximately 4x denser than SRAM [2, 6]. These characteristics make STT-RAM a very attractive candidate for large and fast on-chip cache. With 3D stacking technology, large STT-RAM cache can be stacked on top of cores, addressing the cost and complexity problem in a hybrid magnetic-CMOS process [2, 17].

Despite of these advantages, one major challenge of STT-RAM is its high write energy. Recent studies have shown that STT-RAM cache consumes 6-14 times more energy per write access than SRAM [2, 14]. Furthermore, our evaluation shows that write energy contributes more than 70% of dynamic energy in a 16MB STT-RAM cache (Figure 1). Therefore, reducing write energy of STT-RAM is important to improve its energy efficiency.

Write  Read
100% 80% 60% 40% 20% 0%
ammp apsi art barnes bzip2 fmm facerec gcc gzip lucas mcf mgrid milc ocean raytrace sphinx3 swim Average

**Figure 1: Dynamic energy breakdown of a 16MB STT-RAM cache.**

In this paper, we propose Early Write Termination (EWT), a novel technique that can significantly reduce the write energy of an STT-RAM. Our design is based on the observation that many bits are written with the same value that is already stored in the cache. Those writes are therefore unnecessary, and should be removed to save energy. We designed a simple EWT circuit to terminate such redundant bit writes at their early stages. EWT does not introduce any overhead to access latency and can easily be integrated with existing write circuit. Our experiments show that EWT can reduce up to 80% of write energy and up to 67% of dynamic energy. As a result, the total energy consumption and $ED^2$ are reduced by 33% and 34%, respectively.

The remainder of this paper is organized as follows. Section 2 introduces the background of STT-RAM. Section 3 elaborates on the circuit design of our proposed EWT technique. Section 4 presents how we modeled the delay and energy for read and write accesses in an STT-RAM. Section 5 explains our experimental settings and presents our evaluation results. Section 6 discusses related work and finally, Section 7 concludes this paper.

## 2. STT-RAM BACKGROUND

STT-RAM, or SPRAM, is a new generation of Magnetic Random Access Memory (MRAM) [12, 21]. As in a conventional MRAM, a STT-RAM cell uses Magnetic Tunnel Junction (MTJ) to store binary data. An MTJ consists of two ferromagnetic layers — reference layer and free layer — and one tunnel barrier layer ($MgO$). The magnetic direction of reference layer is fixed, while the magnetic direction of free layer can be changed. The relative magnetization direction between the reference layer and free layer results in different resistance of MTJ, which is used to represent the binary data stored in the cell. When the magnetic field of the free layer and reference layer are parallel, the MTJ resistance is low representing a logical "0". When these two layers are in anti-parallel direction, the MTJ resistance is high which represents a logical "1" [2, 21]. Fig. 2 shows a conceptual view of the MTJ structure [2].

Reference Layer
MgO
Free Layer
Logical 1
(High resistance)
Logical 0
(Low resistance)

**Figure 2: Conceptual view of MTJ structure [2].**

Unlike previous MRAM designs, STT-RAM uses spin-polarized current flowing through the MTJ to change the direction of its magnetic direction in the free layer. Therefore, STT-RAM uses much

*This project is supported in part by NSF CAREER 0747242, NSF CAREER 0641177, CCF-0734339 and CNS-0720595.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
*ICCAD'09* November 2–5, 2009, San Jose, California, USA.
Copyright 2009 ACM 978-1-60558-800-1/09/11 ...$10.00.

lower switching current than conventional MRAMs (100-200uA v.s. 10mA [2,12,21]), which makes it feasible for cache or memory. Furthermore, STT-RAM's threshold current decreases as the size of MTJ becomes smaller, resulting in better scalability than previous MRAM designs [2]. However, STT-RAM still faces challenge of high write energy, compared to an SRAM cell write [2, 14].

A typical design of STT-RAM cell array is shown in Figure 3 [2, 12]. An NMOS connected to word line (WL) is used to select a row of cells, and voltage is applied between bit line (BL) and source line (SL) to perform read/write operations:

- When reading from the cell, a small negative voltage (usually -0.1V) is applied between BL and SL [2, 12]. The amount of current flowing through the cell is dependent on the resistance of MTJ, which is sensed by a sense amplifier to output the stored data.
- When writing a '0' to the cell, a positive voltage (typically 1.2V) is applied between SL and BL, creating a current flow from SL to BL. When writing a '1', a negative voltage (typically -1.0V) is applied between SL and BL, creating a current flow in the opposite direction. The write current is significantly larger than the read current and the duration is also much longer [2, 3].

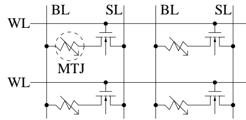


**Figure 3: Typical design of STT-RAM cell array [2, 12].**

# 3. EARLY WRITE TERMINATION (EWT)

## 3.1 The Opportunity

Several existing studies have shown that there is a high probability that a write into a cache or memory location does not change its content, and therefore can be removed. Such an observation has been used at the word level for L1 cache [8], multiprocessors [9], and off-chip memories [7, 20]. Our evaluation demonstrates that this phenomenon is more significant at the bit level, and can be used to throttle those bit-writes that do not change the stored value. Fig. 4 shows the results of our evaluation for a 16MB STT-RAM L2 cache (experimental settings will be discussed in Section 5.1). On average, we see that about 88% of bit-writes are redundant, which implies a significant amount of removable bit writes and a great potential of energy savings in a STT-RAM cache.

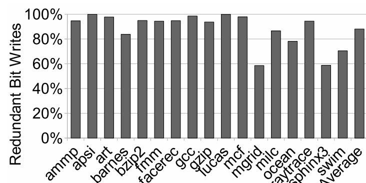


**Figure 4: Redundant bit writes in 16MB STT-RAM L2 cache.**

## 3.2 The Rationale

To remove the unnecessary writes, one needs to know if the new bit value is different from the old bit value. Since our STT-RAM is at the L2 level, one simple way is to utilize the "dirty" bit in L1 to pass or terminate the writes into the L2, analogous to the write removal technique at the memory level [7]. That is, each L1 word is tagged with a dirty bit, and only dirty words are written back to L2 upon eviction. However, this increases the number of dirty bit management in L1 (one bit per word vs. one bit per line by default). More importantly, it cannot capture all the opportunities in write removal. For example, an L1 word being dirty does not warrant that the value is different from what is already stored in L2. Also, even if the value is different, there still might be many bits that are the same. Hence, this technique cannot bring significant reduction in write energy.

To exploit the full opportunity in redundant bit-writes, we could first read out the cache content, compare it with the new values, and write back only the different bits. However, this method entails that every write is preceded with a read operation. Although reads consume much less energy and are much faster than writes in STT-RAM, we could do better than the above method based on the following unique features of STT-RAM cells:

1. When writing to an STT-RAM cell, the change of MTJ resistance is not a gradual procedure. Instead, resistance changes abruptly near the end of a write cycle [1]. This means that at the early stage of a write operation, STT-RAM cell still holds its valid old value.
2. A read operation is performed through applying a voltage between the bit-line and source line, followed by sensing the resulting current on the two lines to determine the MTJ's resistance. This can be performed during the write operation as the latter simply keeps the voltage for a sustained amount of time to change the MTJ state.
3. A write operation in STT-RAM is much longer than a read. A typical write pulse of a STT-RAM cell is 10ns, below which the switching current increases rapidly [2,3]. However, reading from a cell can complete in less than 1ns even in a large L2 [2]. This not only gives us adequate room to sense the old value during the early stage of a write, but also implies great opportunity in saving energy by terminating the write current as soon as redundancy is detected.

Based on those observations, we propose a novel write scheme with the capability of early termination in case of a redundant write. The main goal is to greatly reduce the write energy without impact on performance. The basic idea is to sample the resistance of the MTJ (old value) at early stage of a write operation, and throttle the write current if old value is the same as the new value. To achieve this goal, we need an additional sense amplifier $SA_w$ for each column to sense the resistance of the MTJ during a write operation, and some additional logic to generate the control signals. The reason we use a separate sense amplifier is because the original sense amplifier is used only for read operations which work with only small currents. Hence, it cannot be used for write operations which generate much larger current than reads. As we will show later, we use a normal voltage sense amplifier for $SA_w$ due to its simplicity and low power. The procedure of EWT is as follows.

- When a write operation begins, write voltage is applied between BL and SL to form a write current.
- When the signals are stabilized, sense amplifier $SA_w$ is enabled to sense the resistance of the MTJ (i.e. old value).
- After the old value is sensed, it is saved in a latch and $SA_w$ is turned off to save power.
- If the old value is the same as the new value, a control signal $WCUT$ is generated to shut off the write circuit and terminate the operation on this cell. Otherwise, write operation on this cell continues normally.

As we can see, the above process does not require an extra read to precede a write because sensing the old value is done together with the write operation. Comparing to a previous scheme used in Phase Change Memory that mandates a read before a write [20], EWT does not introduce any overhead in performance. In fact, our experiments show that EWT sometimes even improves performance a little because some write requests (an entire L1 cache line) can be completely throttled and terminated in their early stage.

## 3.3 The Circuit Design

Our EWT implementation does not require any change to existing read or write circuit. Instead, it is designed to work as an "add-on" to a working MRAM read and write circuit. Therefore, EWT can be easily integrated into existing designs.

The write operation starts with applying a positive voltage, for writing a '0', or negative voltage, for writing a '1', between the SL and BL. We add a pass gate on both BL and SL, as shown in Figure 5. These pass gates serve two purposes: 1) when it is detected that the write is redundant, the pass gates are turned off to cut the write current on BL and SL; 2) they are also small loads added to

the BL and SL to show the write path voltage distribution that is determined by the MTJ's resistance. This is used to detect the stored value in MTJ.

For example, when a '0' is written, a positive voltage is applied between SL and BL creating a current flow from SL to BL. Hence, there is a voltage drop on the pass gate on the SL. However, the magnitude of this drop is determined mainly by the resistance of the MTJ (other wire loads are relatively constant). If it is storing a '1', meaning that the resistance is high, the voltage drop on the pass gate is relatively small and $V_{in0}$ is relatively high. On the other hand, if the MTJ is storing a '0', $V_{in0}$ is relatively low. Such a voltage difference is used to determine the stored value in MTJ.

To avoid disturbance on write current and ensure that $V_{in0}$ can be correctly sensed, a conversion circuit is used to magnify $V_{in0}$ into $V_{sense}$, as illustrated in Figure 6. The circuit is similar to a basic differential amplifier [22]. In this conversion circuit, $P_1$, $P_2$ and $N_2$ form a current mirror to provide output current on $N_1$. They are properly sized so that output current is large enough for sensing. Signal $SE$ is used to turn on and off the conversion circuit. Input voltage $V_{in}$ is connected to $N_1$'s gate, which controls its equivalent resistance. For example, if $V_{in}$ is relatively higher, $N_1$'s equivalent resistance is smaller, and vice versa. This difference is reflected at $V_{sense}$ as output current flows through $N_1$. In this way, the difference on $V_{in}$ is magnified into the difference on $V_{sense}$, which is used as the input to the sense amplifier referred in Figure 5.

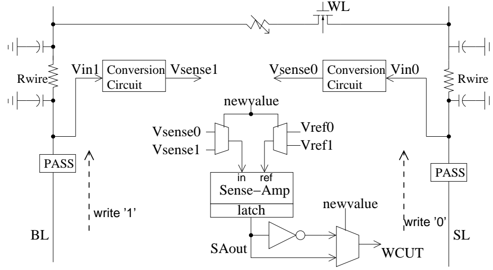


**Figure 5: EWT circuit design in a column.**

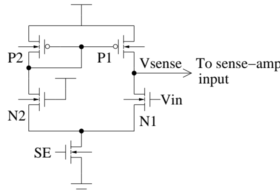


**Figure 6: The conversion circuit.**

Writing a '1' is carried by applying a negative voltage between SL and BL, and the process is opposite to writing a '0'. Therefore, we used a symmetric conversion circuit on the BL for writing a '1'. The two outputs, $V_{sense0}$ and $V_{sense1}$ are then sent to the sense amplifier $SA_w$ through a mux controlled by the new bit value, e.g. $V_{sense0}$ is selected if a '0' is written. At the same time, two reference voltages $V_{ref0}$ and $V_{ref1}$ are also sent to $SA_w$ for comparing with $V_{sense0}$ and $V_{sense1}$ respectively. We need different reference voltages because the circuit is not entirely symmetric [1, 12] and variation of $V_{sense}$ is also different on BL and SL.

The last step is to generate a control signal $WCUT$ to either turn off the write circuit or let it continue, based on the result of the sense amplifier $SA_{out}$. The $SA_{out}$ indicates whether the MTJ is storing a '0' or a '1'. We would need to compare it with the new value to derive $WCUT$. If they are equal, $WCUT$ is high, otherwise, $WCUT$ is low. Fortunately, we do not need to use a comparison circuit such as a XOR/XNOR gate to implement it. Table 1 summarizes the signals and actions during a write. As we can see that $WCUT = SA_{out}$ when writing a '0', and $WCUT = \overline{SA_{out}}$ when writing a '1'. Hence, either $SA_{out}$ or $\overline{SA_{out}}$ is used to turn off the pass gates and the write circuit. This can be implemented by an inverter and mux, which is smaller and simpler than XOR/XNOR gate. In our HSPICE simulation, $WCUT$ can be generated in 0.536ns in a 16MB L2 cache after the word-line is selected. Therefore, redundant bit writes can be detected and throttled at a very early stage.

**Table 1: Sensing Signals**

| Old | New | $V_{in0}$ | $V_{sense0}$ | $SA_{out}$ | Action |
|---|---|---|---|---|---|
| 0 | 0 | Lower | $> V_{ref0}$ | 1 | Cut |
| 1 | 0 | Higher | $< V_{ref0}$ | 0 | Continue |
| Old | New | $V_{in1}$ | $V_{sense1}$ | $SA_{out}$ | Action |
| 0 | 1 | Lower | $> V_{ref1}$ | 1 | Continue |
| 1 | 1 | Higher | $< V_{ref1}$ | 0 | Cut |

## 3.4 Overhead

We implemented the EWT circuit and simulated them in HSPICE using 45nm technology. We measured the additional energy introduced by EWT circuits, including pass gates, conversion circuit, muxes, sense amplifier, latch, and inverters. These components are added on per column basis. That is, all cells in one column share one set of the EWT circuit. We measured that the average energy overhead per cell per write is 89.29fJ. Comparing to the 2.767pJ cell write energy (discussed in Section 4.2), the energy overhead introduced by EWT is 3.23%.

The estimated area introduced by EWT circuit is about $13.44\mu m^2$ per column ($330\mu m$ long column). Comparing to the total area of a 16MB STT-RAM cache (calculated by CACTI), the estimated area overhead is 4.17%.

Since EWT is carried within a write operation, there is no performance overhead to the write latency. On the contrary, some write requests can even finish earlier if all the bits are the same as what are already stored in the cache. This leads to a slight performance gain, which we will see in Section 5.2.

# 4. MODELING STT-RAM AND EWT

To measure how much energy savings we can achieve through our EWT design, we first modeled a STT-RAM L2 cache in both performance and energy, and then compared it to a baseline STT-RAM without the EWT. As we have shown in Figure 3, STT-RAM cache uses similar peripheral logic as a regular SRAM cache. Hence, we use the existing cache modeling tool CACTI [16] to derive both the latency and the energy values for the peripheral logic such as the H-tree, decoder, word-line, bit-line, sense amplifiers etc., and combine them with the STT-RAM cell's latency and energy. We chose the 45nm technology library, and the low operation power (LOP) peripheral design due to the high dynamic power required by STT-RAM cells.

## 4.1 Latency

The breakdown of read and write component latencies are listed in Table 2. The latencies are rounded up to CPU cycles when used in our simulator. We refer to a recent work on STT-RAM cache [2] for STT-RAM's cell latency. For read operation, this is essentially the sense-amplifier delay, which is assumed to be 20% slower than SRAM's sense amplifier [2].

For write operations, we used a 10ns pulse width as mentioned earlier. However, if the entire write access (a cache line) is throttled, the write pulse width is equal to the time required for redundancy detection which is 0.536ns, as measured from our HSPICE simulation. Therefore, a write with EWT may take shorter time than in the baseline.

**Table 2: Per-Access Read/Write Latency**

| | Read | Write (Base) | Write (EWT) |
|---|---|---|---|
| H-tree in | 2.010ns | 2.010ns | 2.010ns |
| Word-line + Decoder | 0.544ns | 0.544ns | 0.544ns |
| Bit-line | 0.800ns | N/A | N/A |
| Sense-amp | 1.006ns | N/A | N/A |
| H-tree out | 1.872ns | N/A | N/A |
| Write Pulse | N/A | 10ns | 10ns / 0.536ns |
| Total | 6.232ns | 12.554ns | 12.554ns / 3.090ns |

## 4.2 Dynamic Energy

The breakdown of dynamic energy for reads and writes are shown in Table 3–4. For dynamic cell energies, we referred to the results from the recent work [2], and scale them to 45nm technology.

When EWT is enabled, the write energy is no longer a fixed value. Instead, it is the sum of three parts: peripheral energy $E_{peripheral}$, overhead $E_{overhead}$ and a varying cell energy $E_{cells}$ due to a value change:

$$E_{EWTwrite} = E_{peripheral} + E_{overhead} + E_{cells}$$

$E_{peripheral}$ is the energy consumed by the peripheral logic. This is 0.203nJ, same as in baseline. $E_{overhead}$ is the energy consumed by the EWT circuits. This part is 0.0457nJ per write access, calculated as per cell overhead multiplied by the number of cells in a cache line (512 in our case) since there is one set of EWT circuit per column. $E_{cells}$ is the energy required by those cells that are updated. This variable part depends on how many cells are actually changed in a write request. It can be expressed as:

$$E_{cells} = N_{changed} \times E_{cellchange} + N_{unchanged} \times E_{unchanged}$$

Where $E_{cellchange}$ is the energy used to change one cell, which is 2.767pJ in our model. This is obtained from scaling the results in [2] to 45nm technology. Write operations on unchanged cells are terminated at the end of 0.536ns, which amounts to 0.148pJ per cell for $E_{unchanged}$. In summary, per-access write energy with EWT can be expressed as:

$$E_{EWTwrite} = E_{peripheral} + E_{overhead} + N_{changed} \times 2.767pJ$$
$$+N_{unchanged} \times 0.148pJ$$

**Table 3: Per-Access Read/Write Energy**

| | Read | Write (Base) | Write (EWT) |
|---|---|---|---|
| Peripheral | 0.192nJ | 0.203nJ | 0.203nJ |
| Overhead | N/A | N/A | 0.0457nJ |
| Cells | 0.013nJ | 1.417nJ | Variable (Table 4) |
| Total | 0.205nJ | 1.620nJ | Variable (Table 4) |

**Table 4: EWT Write Energy (Per-Cell)**

| | |
|---|---|
| $E_{cellchange}$ | 2.767pJ per cell |
| $E_{unchanged}$ | 0.148pJ per cell |

## 4.3 Leakage Energy

Since STT-RAM has negligible leakage in the cell, peripheral leakage becomes the dominant part in the total leakage of the cache. We used CACTI to estimate the leakage power for the peripheral logic. This value is 0.265W for the 16MB STT-RAM cache used in our experiment. Due to the non-volatile nature of STT-RAMs, we can further reduce the leakage power by power gating the cache banks if they are idle. This significantly reduces the leakage power of a STT-RAM cache. CACTI results show that the leakage power of an idle bank in our model is 0.388mW, in a 16-bank 16MB cache. In addition, we assume that there is a 1ns delay to power on a bank. Such leakage saving scheme is applied to both the baseline and our design with EWT.

# 5. EVALUATIONS

## 5.1 Experimental Setup

We used a simulator based on Simics [11] to simulate a 3D architecture with a 16MB L2 STT-RAM stacked on top of a 4-core CMP, the most feasible way to integrate STT-RAMs with CMOS technology as studied previously [14]. Core frequencies are set to 1GHz due to thermal constraint in a 3D architecture. We enhanced the cache model in Simics to model the energy and delay of STT-RAM, as well as the contention on cache banks and wake-up delays of idle banks.

We use a variety of workloads from SPLASH2 [13], SPEC2K and SPEC2006. Both memory intensive and computational workloads are evaluated in our experiments. Results are collected through execution-driven simulations. For each workload, we skipped its initialization phase, warmed up for 50M instructions and ran for 100M instructions.

**Table 5: Simulation Parameters**

| | |
|---|---|
| Processor core | 4 cores, each core runs at 1GHz |
| L1 Cache | Private L1 cache (32K I-cache and 32K D-cache), 64-byte lines, 4-way set associative, 3 cycles access time |
| L2 Cache | Shared L2 cache (STT-RAM), 16MB, 64-byte lines, 16-way set associative, 16 banks |
| Main Memory | 4GB memory, 50 cycles access time |

## 5.2 Performance

As we discussed previously, EWT does not introduce any performance penalty to cache accesses. Instead, write requests may finish early if no bit change is needed. Therefore, EWT can reduce average write latency and the contention on cache banks, which results in slight improvement in performance. Figure 7 shows our results in Cycles Per Instruction (CPI), normalized to the baseline. We observe a 3%-7% of CPI improvement in memory intensive workloads such as `mcf, art, lucas`, and the average CPI improvement over all workloads is 1%.

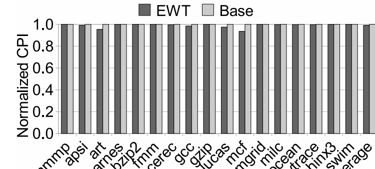


**Figure 7: Performance improvements.**

## 5.3 Energy Savings

As expected, the EWT design achieved significant energy savings in writes. Figure 8 shows the write energy in each workload, normalized to the baseline. With EWT, up to 80% of write energy reduction is observed. Among all 17 workloads, 14 of them get more than 60% of write energy reduction. Even for workloads with lower bit write redundancy such as `mgrid, sphinx3 and swim`, EWT still achieves 40%-60% savings. The average saving on write energy is 70%.

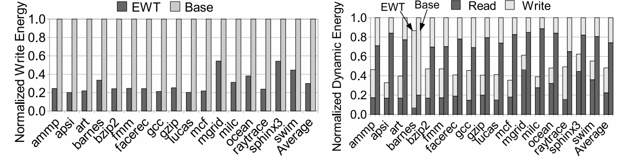


**Figure 8: Write energy savings.** **Figure 9: Dynamic energy savings.**

We combined write energy and read energy together to evaluate our savings in total dynamic energy. Figure 9 shows the measured results. As write energy contributes to more than 70% of total dynamic energy in baseline, applying EWT leads to significant reduction in total dynamic energy (52% on average). We then compared the total energy (dynamic plus leakage) between EWT and baseline in Figure 10. With EWT, total energy can be reduced by up to 53%, and the average reduction is 33%.

## 5.4 Energy-Delay Product

Last, combining our results in both energy and performance, we present results for $ED^2$ in Figure 11. Due to the significant savings in energy and slight improvement in execution time, we obtain up to 59% of $ED^2$ reduction, with an average reduction of 34%. These results show that EWT can effectively improve energy efficiency using a STT-RAM cache.

# 6. PRIOR ART

There have been active efforts recently on STT-RAM designs. Most device-level studies focus on improving the MTJ properties, cell array structures, and prototyping. To name a few, Hosomi *et al.* fabricated a 4K bit STT-RAM using their tailored MTJ design in 0.18$\mu m$ technology [3]. The goal was to demonstrate that STT-RAM is a prominent candidate for next generation memory due its high speed, low power and high scalability. Kawahara *et al.*

Figure 10: Total energy savings.

Figure 11: $ED^2$ savings.

prototyped a larger 2Mb STT-RAM in $0.18\mu m$ [5] technology with new features to ensure low read and write access time.

Miura *et al.* presented a SPRAM with synthetic ferrimagnetic (SyF) free layer, which has high immunity to read disturbance and sufficient margin between read and write currents [23]. This SyF free layer scheme in MTJ is further investigated in [24] to develop low critical current density without degrading the thermal-stability factor. Gogl *et al.* built a 16-Mb MRAM with a novel bootstrapped write driver circuit in $0.18\mu m$ technology [27]. In an earlier work, Durlam *et al.* prototyped a 1-Mbit MRAM based on 1T 1MTJ bit cell integrated with copper interconnect technology [25]. Su *et al.* presented a write disturbance fault (WDF) model for MRAM [26]. The fault affects the data stored in MRAM cells due to excessive magnetic field during a write operation. A 1-Mb prototype fabricated with $0.18\mu m$ technology is also demonstrated.

Chen *et al.* proposed a dynamic MTJ model which provides more accurate (transient) description for MTJ resistance switching [1]. As a result, more than 20% pessimism in write time can be reduced with TSMC $0.13\mu m$ technology. Li *et al.* considered the failure probability of STT-RAM cells due to parameter variations [10]. They developed a model to characterize STT-RAM cells so that one can predict memory yield and design optimizations to minimize memory failure.

At architecture level, there are several recent efforts in using STT-RAM as an on-chip last level cache. Dong *et al.* developed a delay and energy model for MRAM-based cache [2]. They used the model to compare MRAM with SRAM and DRAM in terms of area, performance and energy in the context of 3D stacking. They found that MRAM cache offers competitive IPC improvement with a large reduction in power. We have leveraged their model heavily in this paper. Sun *et al.* proposed techniques to improve the latency and to reduce the write energy for an L2 STT-RAM [14]. Their proposal is to use a buffer in front of the L2 to serve the writes. Also, a hybrid SRAM-MRAM cache architecture was developed to reduce the accesses to the MRAM cache banks. We do not require architecture modifications in our design. Instead, we took a circuit level approach to remove completely the write redundancy in the L2 cache itself without impacting the performance.

There are also some recent work on PCMs that are closely related to ours. Lee *et al.* proposed techniques to reduce the write accesses to PCM-based main memory to improve its endurance [7]. One of the techniques utilizes the dirty bits in L2 at the word granularity to check if a word has been updated since it was last fetched on-chip. We also proposed a technique along a similar line for PCM memory [20] at bit level. In this technique, a memory row is first read out, then compared with the new data, and finally written back for those changed values. As we have discussed before, checking redundancy at the word level looses significant opportunity in removing writes. Also, we do not require a pre-write read operation in this paper, which is a significant saving in both performance and energy.

# 7. CONCLUSION

We propose a novel scheme, Early Write Termination, to improve the energy efficiency of STT-RAM cache. Our scheme throttles redundant cell write at very early stage, which leads to significant savings on write energy without any latency overhead. We implement critical circuits of EWT, and perform detailed modeling and simulation. In our experiment, up to 80% of write energy reduction and 34% of average $ED^2$ reduction are observed. Our results show that EWT is an effective and practical scheme to improving the energy efficiency of a STT-RAM cache.

# 8. REFERENCES


[1] Y. Chen, X. Wang, H. Li, H. Liu, D. V. Dimitrov, "Design Margin Exploration of Spin-Torque Transfer RAM (SPRAM)," *International Symposium on Quality Electronic Design,* pp. 684-690, 2008.

[2] X. Dong, X. Wu, G. Sun, Y. Xie, H. Li, Y. Chen, "Circuit and Microarchitecture Evaluation of 3D Stacking Magnetic RAM (MRAM) as a Universal Memory Replacement," *Design Automation Conference,* pp. 554-559, 2008.

[3] M. Hosomi *et al.* "A Novel Nonvolatile Memory with Spin Torque Transfer Magnetization Switching:Spin-RAM" *IEEE International Electron Devices Meeting,* pp. 459-462, 2005.

[4] D. H. Kang, et al., "Two-bit Cell Operation in Diode-Switch Phase Change Memory Cells with 90nm Technology," *IEEE Symposium on VLSI Technology Digest of Technical Papers,* pp. 98-99, 2008.

[5] T. Kawahara *et al.* "2 Mb SPRAM (SPin-Transfer Torque RAM) with Bit-by-Bit Bi-Directional Current Write and Parallelizing-Direction Current Read," *IEEE Journal of Solid-State Circuits,* Vol. 43, No. 1, pp. 109-120, Jan. 2008.

[6] S. Lai, T. Lowrey, "OUM - A 180nm Nonvolatile Memory Cell Element Technology for Standalone and Embedded Applications," *International Electron Devices Meeting,* pp. 36.5.1-36.5.4, 2001.

[7] B. C. Lee, E. Ipek, D. Burger "Architecting Phase Change Memory as a Scalable DRAM Alternative," *to appear, International Symposium on Computer Architecture,* 2009.

[8] K. M. Lepak and M. H. Lipasti, "On the Value Locality of Store Instructions," *International Symposium on Computer Architecture,* pp. 182-191, 2000.

[9] K. M. Lepak and M. H. Lipasti, "Silent Stores for Free," *International Symposium on Microarchitecture,* pp. 22-31, 2000.

[10] J. Li, C. Augustine, S. Salahuddin, K. Roy, "Modeling of Failure Probability and Statistical Design of Spin-Torque Transfer Magnetic Random Access Memory (STTMRAM) Array for Yield Enhancement," *Design Automation Conference,* pp. 278-283, 2008.

[11] P. S. Magnusson, et al., "Simics: A full system simulation platform," *Computer,* 35(2):50–58, 2002.

[12] M. Hosomi, et al. "A Novel Nonvolatile Memory With Spin Torque Transfer Magnetization Switching: Spin-RAM," *International Electron Devices Meeting,* pp. 459-462, 2005.

[13] J. P. Singh, W. Weber, A. Gupta. "SPLASH: Stanford Parallel Applications for Shared-Memory". *In Computer Architecture News*, vol. 20, no. 1, pp 5–44.

[14] G. Sun, X. Dong, Y. Xie, J. Li, Y. Chen, "A Novel Architecture of the 3D Stacked MRAM L2 Cache for CMPs," *The 15th International Symposium on High-Performance Computer Architecture,* pp. 239-249, 2009.

[15] F. Tabrizi, "The Future of Scalable STT-RAM as a Universal Embedded Memory," *Embedded.com,* February 2007.

[16] S. Thoziyoor, N. Muralimanohar, J. H. Ahn, N. P. Jouppi, "CACTI 5.1", *http://www.hpl.hp.com/techreports/2008/HPL-2008-20.html*, 2008.

[17] Y. Xie, G. H. Loh, B. Black, K. Bernstein, "Design space exploration for 3D architectures," *ACM Journal on Emerging Technologies in Computing Systems (JETC),* pp. 65-103, 2006.

[18] F. Yeung, et al., "$Ge_2Sb_2Te_5$ Confined Structures and Integration of 64Mb Phase-Change Random Access Memory," *Japanese Journal of Applied Physics,* pp. 2691-2695, 2005.

[19] "The International Technology Roadmap for Semiconductors, Process Integration, Device and Structures," *http://www.itrs.net/links/2007itrs/2007_chapters/2007_PIDS.pdf* 2007.

[20] P. Zhou, B. Zhao, J. Yang, Y. Zhang, "A Durable and Energy Efficient Main Memory Using Phase Change Memory Technology," *The 36$^{th}$ International Symposium on Computer Architecture,* To Appear, 2009.

[21] J. G. Zhu, "Magnetoresistive Random AccessMemory: The Path to Competitiveness and Scalability," *Proceedings of the IEEE,* pp. 1786-1798, 2008.

[22] J. M. Rabaey, A. Chandrakasan, B. Nikolic, "Digital Integrated Circuits: A Design Perspective 2nd Edition," *Prentice-Hall Electronics And VLSI Series,* page 680-681, 2003.

[23] K. Miura, et al., "A novel SPRAM (SPin-transfer torque RAM) with a synthetic ferrimagnetic free layer for higher immunity to read disturbance and reducing write-current dispersion," *Symposium on VLSI Technology Digest of Technical Papers*, pp. 234-235, 2007.

[24] J. Hayakawa, et al., "Current-Induced Magnetization Switching in MgO Barrier Magnetic Tunnel Junctions With CoFeB-Based Synthetic Ferrimagnetic Free Layers," *IEEE Transactions on Magnetics, Vol. 44, No. 7*, pp. 1962-1967, 2008.

[25] M. Durlam, et al., "A 1-Mbit MRAM Based on 1T1MTJ Bit Cell Integrated With Copper Interconnects," *IEEE Journal of Solid-State Circuits, Vol. 38, No. 5,* pp. 769-773, 2003.

[26] C. L. Su, et al., "Write Disturbance Modeling and Testing for MRAM," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 16, No. 3,* pp. 277-288, 2008.

[27] D. Gogl, et al., "A 16-Mb MRAM Featuring Bootstrapped Write Drivers," *IEEE Journal of Solid-State Circuits, Vol. 40, No. 4,* pp. 902-908, 2005.