





“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.

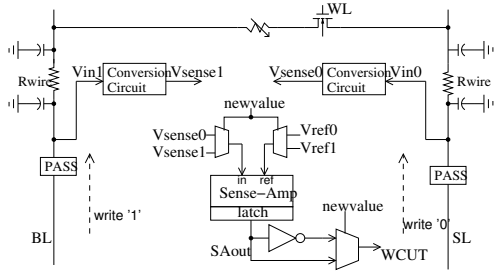


Figure 5: EWT circuit design in a column.

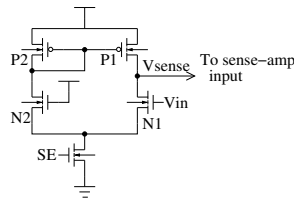


Figure 6: The conversion circuit.

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, other-

wise,  $WCUT$  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.

### 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.

**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

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.

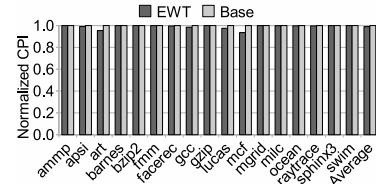
We use a variety of workloads from SPLASH2 [13], SPEC2K and SPEC2006. Both memory intensive and computational workloads are evaluated in our experiments. Results are collected through execution-driven simulations. For each workload, we skipped its initialization phase, warmed up for 50M instructions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

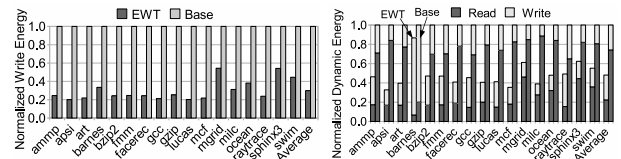
### 5.2 Performance

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,

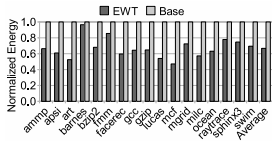


Figure 10: Total energy savings.

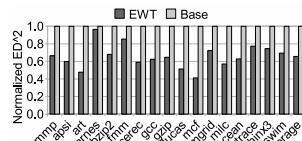


Figure 11:  $ED^2$  savings.

cell array structures, and prototyping. To name a few, Hosomi *et al.* fabricated a 4K bit STT-RAM using their tailored MTJ design in  $0.18\mu\text{m}$  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.* prototyped a larger 2Mb STT-RAM in  $0.18\mu\text{m}$  [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  $0.18\mu\text{m}$  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  $0.18\mu\text{m}$  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  $0.13\mu\text{m}$  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 loses 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](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 36th 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.