Impact of Process-Variations in STTRAM and Adaptive Boosting for Robustness

Seyedhamidreza Motaman, Swaroop Ghosh and Nitin Rathi
Computer Science and Engineering, University of South Florida.
motaman@mail.usf.edu, sghosh@cse.usf.edu, nitinr@mail.usf.edu

***Abstract*— Spin-Torque Transfer Random Access Memory (STTRAM) is a promising technology for high density on-chip cache due to low standby power. Additionally, it offers fast access time, good endurance and retention. However, it suffers from poor write latency and write power. Additionally we observe that process variation can result in large spread in write and read latency variations. The performance of conventionally designed STTRAM cache can degrade as much as 10% due to process variations. We propose a novel and adaptive write current boosting to address this issue. The bits experiencing worst-case write latency are fixed through write current boosting. Simulations show 80% power improvement compared to boosting all bit-cells and 13% performance improvement compared to worst case latency due to process variation over a wide range of PARSEC benchmarks.**

Keywords—STTRAM, write power, write current boosting, Process variation, Variation tolerant design

I. INTRODUCTION

Conventional CMOS memory i.e., Static Random Access Memory (SRAM) has been the popular choice for embedded memory application for last several decades. However, SRAM seems to be approaching a brick wall. On one hand process variability and leakage power is posing severe obstruction towards SRAM scaling to future nodes and on the other hand, emerging energy-constrained and bandwidth hungry electronic gadgets demand for larger as well as energy-efficient on-chip cache which cannot be satisfied with SRAM. To address the changing landscape of consumer market, there is a corresponding need of changing the design paradigm. Several emerging memory technologies are on the horizon such as Spin Transfer Torque RAM (STTRAM) [1], Domain Wall Memory (DWM), Resistive RAM (ReRAM) and memrister [2]. STTRAM is an energy-efficient variant of Magnetic RAM (MRAM) where the magnetization switching is through current induced spin-transfer torque. Due to fast switching, low-power, endurance and superb retention, STTRAM is widely studied for universal memory. Fig. 1 shows the schematic of STTRAM bitcell where Magnetic Tunnel Junction (MTJ) is used as the storage element. The resistance of MTJ is high (low) when fixed layer and free layer are in antiparallel (parallel) configuration with respect to each other.

One of the primary challenge of STTRAM is long write latency. Additionally, our analysis indicate that process variations in the STTRAM bitcell increases write latency further for large cache (Section 3). Similarly the read latency is also degraded due to process variations. The sources of process variations are summarized in Fig. 2(a). We note that the process variations result in long tail in write and read latency. This results in significant performance degradation and power overhead. In this work we model the tail for correct estimation of number of failing bits. We also find that write latency can be lowered by boosting the write current. We propose circuit level technique to implement adaptive write current boosting and exploit them at micro-architecture level to mitigate process variation induced performance and power degradation. The proposed approach is summarized in Fig. 2.

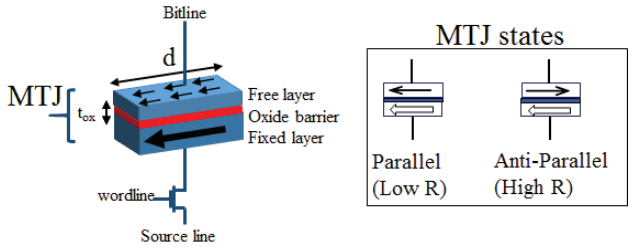


Fig. 1 Schematic of a Spin Transfer Torque Random Access Memory (STTRAM).

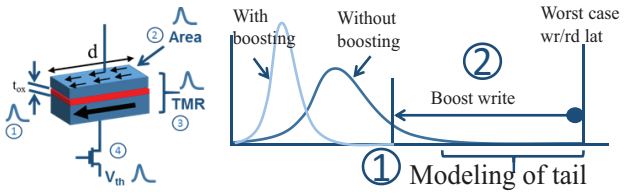


Fig. 2 (a) Various sources of variations in STTRAM bitcell and, (b) the proposed methodology that involves modeling of tail of the distribution and adaptive boosting to accelerate the tail.

Note that the proposed methodology can be employed dynamically. However in this work we have investigated the static (one time programming) column boosting for the sake of simplicity. The proposed technique can also be perceived as a repair mechanism to fix the slow columns. In summary, we make following contributions in this paper:

- We investigate the impact of process variations on the read and write latency of STTRAM.
- We propose write driver circuit to enable adaptive boosting.
- We propose a methodology to enable write current modulation adaptively to mitigate process variation induced write latency degradation.

The rest of the paper is organized as follows. In Section II, we describe the impact of process variations on read and write

This paper is based on work supported by Semiconductor Research Corporation (#2442.001).

TABLE 1: Parameters used for process variation study

Device	Parameter	Mean	Std. Dev.
Transistor	V_{TH}	0.5	60 mV
MTJ	MgO Thickness	1.1nm	5%
	Shape Area	Pi*25e^{-9}	15%

latency. The proposed write driver to enable adaptive modulation is introduced in Section III. The cache architecture and simulation results are presented in Section IV. Related work is described in Section V and conclusions are drawn in Section VI.

II. Process Variation Analysis

In this Section, we analyze the impact of process variations in the STTRAM bitcell during read and write operation. We also investigate the modeling of read/write latency distribution and impact of current boosting.

A. Process Variation in Write Operation

Process variation analysis is important due to the large cache size that is employed at the last level. The process variations in the MTJ is modeled by incorporating variations in MTJ as well as access transistor as shown in Fig. 2(a). For MTJ, we have assumed tunnel oxide barrier and surface area variations. The variations in access transistor is lumped in threshold voltage fluctuation. The mean and standard deviation of these parameters are provided in Table-1. The variations in the MTJ can increase the intrinsic thermal energy barrier and resistance of MTJ which in turn can increase the write time. The write latency is asymmetric in nature. Therefore, we have considered the worst case polarity (1→0 transition) for latency analysis.

Fig. 3(a) shows the Monte-Carlo analysis for 5000 simulation points at typical process corner using MTJ model from [3]. It can be noted that write latency has wide distribution. Therefore performance analysis with mean write latency assumption can result in significant overestimation. The write latency also shows a long tail and the worst case write bits could eventually limit the system performance. In order to gain detailed understanding we use curve fitting based functions in Matlab to model the write latency distribution (especially the tail). Fig. 3(a) depicts different models (empirical, Extreme Value Theory, lognormal, inverse Gaussian and loglog) used to fit the distribution in Matlab. Empirical model indicated better match for the tail. Therefore we used this model for the cache level analysis. Note that the cache size for our study is 8MB. The curve fitting model is used to extrapolate the distribution to 8MB bits. Accuracy of estimation can be improved by running larger Monte-Carlo simulation. At 70uA current the worst case write latency is found to be 23ns which is >5X larger than mean value underscoring the need of process variation-aware design (Fig. 3(b)).

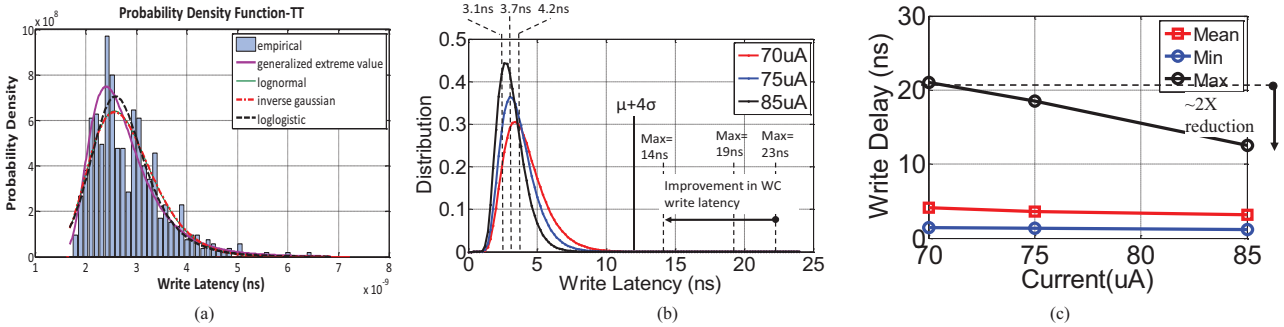
In order to improve the system performance it is crucial to fix the tail of the write latency. The distribution for boosted write currents are also shown in Fig. 3(b). It can be observed that write current boosting can be used to speed up tail bits and mitigate the impact of process variation on write latency. The distribution also indicates that the number of MTJs beyond μ+4σ point is reduced when write current is boosted. Fig. 3(c) plots the max, mean and min latency for different write currents. It can be noted that worst case points can gain significant benefit (as much as 2X) although the mean shows minor improvement from boosting.

B. Process Variation in Read Operation

The process variations in the STTRAM can reduce the TMR and read current which in turn can increase the sense time. We have analyzed the time needed to develop 100mV sense margin (to account for senseamp offset due to variations). The simulations are done using the settings described before. Fig. 4(a) plots the read latency distribution for 2000 runs of Monte Carlo. Different curve fitting models are also plotted. The read latency distribution for 8MB MTJs is shown in Fig. 4(b). It can be noted that process variation can degrade the read latency as much as 8X.

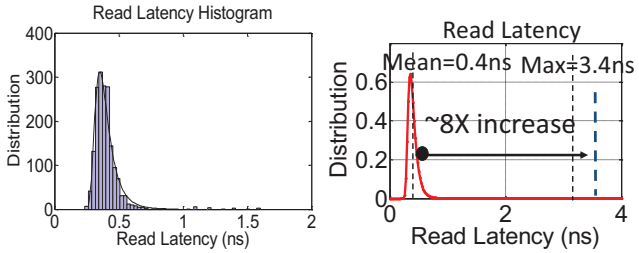
C. Process Variation Tolerant Design

From the above discussion, it is evident that write current boosting can be used as a knob to mitigate process variation. As depicted in Fig. 3(b), write current boosting reduces the number of MTJs beyond 4 sigma delay. Note that the current boosting for write is associated with power consumption. Therefore it should be used only for the tail bits to improve the performance with minimal impact of dynamic power. Mitigating read latency degradation is subject of further research.



(a) (b) (c)

Fig. 3 (a) Write latency distribution for 5000 Monte Carlo points. The curve fitting to model the tail is also shown; (b) write latency distribution using curve fitting model for three different write currents for 8MB cache. The worst case MTJ can be accelerated through high write current. The 4 sigma delay is also shown. By boosting the current the number of bits beyond 4 sigma delay can be reduced; and, (c) min, mean and max write latency with write current.



(a) (b)

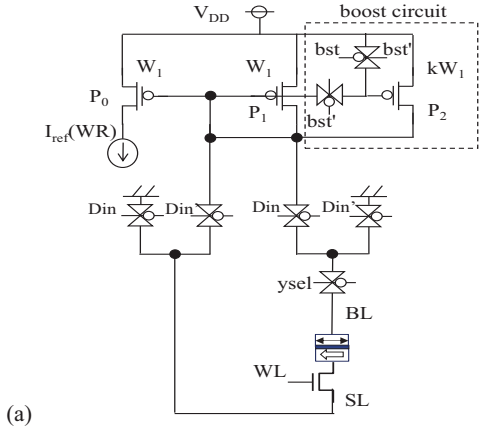
Fig. 4 (a) Read latency distribution for 2000 Monte Carlo points. The curve fitting to model the tail is also shown; (b) read latency distribution for 8MB cache.

III. SUBARRAY CIRCUIT DESIGN

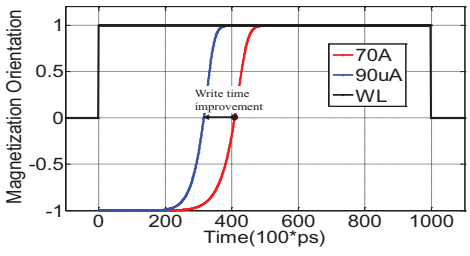
In the previous section we studied the impact of process variation and write current boosting as design time techniques to improve performance under variability. In this Section, we will present write driver design to enable boosting. The subarray architecture is also be presented.

A. Write Driver Design

We propose a novel current mirror based write driver to boost the write current of the column if needed (Fig. 5(a)). A reference write current I_{ref}(WR) is mirrored on the leg that is driving BL/SL. The direction of current flow is controlled by the polarity of data to be written (D_{in}). The BL (SL) is connected to current source (VSS) if the data to be written is 1 (0). The sizing of PMOS P_1 is ratioed wrt to reference leg to generate the required write current. We add an extra PMOS transistor P_2 with size *k* so that extra current needed for the boosting is generated when boost signal is asserted (i.e., bst=1). For nominal conditions P_2 is disabled by connecting the gate to V_{DD}.



(a)



(b)

Fig. 5 (a) Boost enabled write and sense circuit; and (b) simulation results showing write time improvement by enabling write boost.

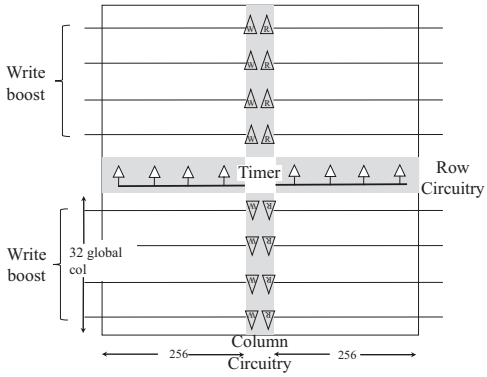


Fig. 6 Subarray architecture showing boost enabled write circuit.

The proposed driver needs 4 transistors for multiplexers and an extra PMOS to generate the boosted current. Considering the fact that gate leakage is negligible and *bst* is a DC signal the multiplexers can be designed using minimum sized transistors. Therefore the area overhead of the proposed boosting can be kept below 1%. Fig. 5(b) shows the hspice simulation waveform of magnetization switching during write process for nominal and boosted current.

B. Subarray Architecture

Fig. 6 shows the proposed sub-array design with integrated boost enabled write drivers. There are a total of 512 WLs (256 in each sector) and 512 local columns. Column muxing of 8:1 is used for one global column. A total of 64 global columns provide 64 bits of data in/out. The column area holds read/write circuitries. The write drivers are designed per global column basis. Therefore, boosting a write driver will boost the write current for the 8 local columns. *Note that it is possible to disable the boost for fast MTJs at the cost of decoding complexity*. Furthermore the power overhead of boosting small number of global columns is found to be minimal (Section 5).

IV. CACHE DESIGN FOR ADAPTIVE BOOSTING

In the previous section we explained the subarray circuit design techniques. This section is focused on methodology to identify the slow bits and implementation of current boosting. This is followed by cache organization and simulation results. The limitations and possible improvements are also discussed.

A. Methodology

The proposed boosting is employed after a test routine that screens the slow write bits. The test pattern can be any of the conventional March patterns (e.g., March C [24]) that is performed at different frequencies to determine the write time of the bits in absence of boosting. The columns containing slow write are marked individually. In this context it is worth mentioning that the entire global column is marked slow even if one of the local columns are found slow. This is due to the fact that write drivers are shared per global column basis. Next the same patterns are repeated with the boosted write currents to ensure that the bits pass. Since the amount of current boosting is determined statistically through simulations we expect that all bits will pass after this step. If not, the existing column or

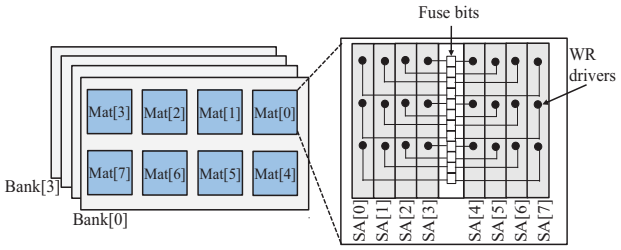


Fig. 7 Cache organization and fuse bits.

row redundancies can be used to replace the remaining slow bits. It is also possible to provide an extra setting in the drivers during design phase to boost the current further.

Fuses are used to program the individual columns for boost/no-boost. The fuse bits are decoded and loaded in the flip-flops to assert the DC signals controlling boost (Fig. 7). Note that fuse-based infrastructure is commonly used in micro-processors for redundancy programming, SRAM assist setting etc. Therefore the proposed technique can be easily incorporated in the system.

B. Cache Organization

We have considered a 8MB L2 cache for this study. The L2 cache is divided into following sections (Fig. 7): (a) Sub-array, (b) Mat that consists of a group of sub-arrays which share a common pre-decoder. Each mat contains multiple ways. A group of mats provides output cache-line (e.g., 8 mats provide 64 bits each totaling 512 bits) and, (c) Bank that operates independently.

Each subarray contains 512 rows and 512 columns. This amounts to 0.25Mb data. Each mat is composed of 8 subarrays (SA[7:0]). The write drivers of each subarray receives global column based boost signal. This will require 64 DC tracks (i.e., two tracks per global column) to be routed for each subarray i.e., 256 DC tracks per mat. Note that minimum pitch metals can be used for routing these signals. Each bank contains 8 mats (mat[7:0]) of total size 8MB. There are four independent banks (bank[3:0]) in the cache.

Each way in L2 is implemented in a different subarray in mat for parallelism. The column mux selects the desired BL and senseamp senses bit-cell states in either data or tag array. Each mat provides 64-bits of data by accessing a subarray. For example, way0 is accessed by enabling SA[0] of Mat[7:0] providing 512 bits of cache line. The L1 cache comprises of traditional SRAM.

C. Simulation Setup

We evaluate SRAM and several cases of STTRAM in terms of power and performance. The evaluations are performed on a 4-core Alpha processor in Gem5 [4]. The processor configuration is provided in Table 2. Gem5 is modified accordingly to implement variable read and write latencies for STTRAM cache. We simulate process variation for 5000 runs of Monte Carlo and find a model to fit the distribution in Matlab. Next the model is used to estimate the write and read latency

TABLE 2: Processor Configuration

Processor	Alpha,O3,4 cores, 2GHz, 8-way issue
SRAM L1-Cache	Private, Icache=16KB, Dcache=16KB, 64B Cache-line, 2 cycle Read/Write latency, Write back.
LLC Cache	Shared, 8MB, 4 banks, 8 ways, 64B cache-line, writeback, R/W latency based on memory tech.
Main Memory	4GB, DDR3, 200-cycle latency

distributions for 64 million MTJs. Next the steps described below are followed:

1. The number of MTJs with write latency greater than 4 sigma (N_{wr}) are determined from the latency distribution obtained from Matlab.

2. N_{wr} is randomly distributed among the 64 million MTJs. The slow global columns numbers are determined in Matlab and fed to Gem5.

3. Gem5 matches the global columns for each access with the list and finds the number of times the slow global columns are accessed. This information is used to estimate the dynamic power of boosted columns.

The simulations are performed over a wide range of Parsec Benchmarks [5]. For power simulation we used McPAT [6] multi-core power simulator with modified CACTI [7] integrated in Gem5 simulator. We have simulated following cases to evaluate STTRAM under process variations:

(a) STTRAM-no-PV: STTRAM without any process variation.

(b) STTRAM-WC-PV: STTRAM with worst-case write and read latency due to process variation.

(c) STTRAM-bWR: STTRAM with write boosting of slow columns.

(e) STTRAM-bAll: STTRAM with write boosting of all columns.

The cache latency and energy is obtained using CACTI and Hspice model of STTRAM. The parameters used for simulations are provided in Table 3. Mean write latency is considered for STTRAM-no-PV whereas worst case write latency is considered for STTRAM-WC-PV (Fig. 3(b)). We use write current of 70uA for STTRAM-no-PV and STTRAM-WC-PV and 85uA for boosted cases. For boosted cases, we assume 4 sigma write latencies for normal columns and boosted columns. The write and read energy with and without boosting is also shown in the Table 3.

D. Simulation Results

Fig. 8(a) shows the performance result represented by the normalized (normalized to SRAM) instruction per cycle (IPC). STTRAM-no-PV provides 4% performance improvement over SRAM. However STTRAM-WC-PV indicates that process variation can degrade the IPC by 10% on average compared to STTRAM-no-PV. Boosting the write current (STTRAM-bWR) can improve the IPC by 13% compared to STTRAM-WC-PV.

Table 3: Design parameters for different cache configurations (22nm PTM Technology)

Cache parameters	Cell Size	Total Area	Read Lat.	Write Latency boost/orig.	Read Energy	Write Energy	Write Pulse (boost/orig.)	Leakage Power (W)
SRAM	146F^2	17.3mm^2	7.1ns	5ns	1.1nJ	0.8nJ	------	10.2
STTRAM-no-PV	40 F^2	6.9 mm^2	2.9ns	5.2ns	0.9nJ	1.4nJ	3.9 ns	1.72
STTRAM-WC-PV	40 F^2	6.9 mm^2	5.5ns	22.2ns	0.6nJ	0.7nJ	21ns	1.72
STTRAM-PV	40 F^2	6.9 mm^2	5.5ns	13.4ns/22.2ns	0.6 nJ	0.7nJ/1.2nJ	12.5ns/21ns	1.72

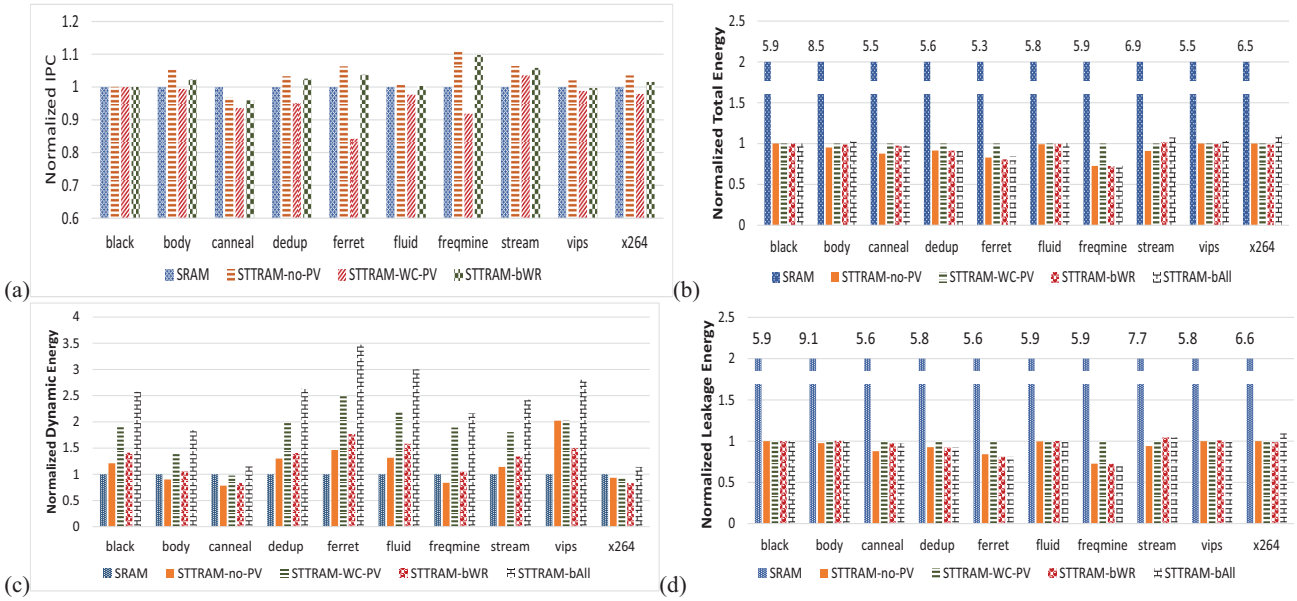


Fig. 8 (a) IPC; (b) L2 total energy comparison; (c) L2 dynamic energy; (d) L2 leakage energy.

The maximum benefit is observed for write intensive benchmarks such as dedup.

Fig. 8(b) shows the normalized total energy dissipation (normalized to STTRAM-WC-PV). The STTRAM architecture shows ~6.4X saving compared to SRAM. This is owing to elimination of bitcell leakage and reduction in peripheral leakage (due to less number of peripherals). STTRAM-bAll increases the power for benchmarks dedup and freqmine because they are write intensive. The other benchmarks observe power reduction due to lower peripheral leakage as the run-time is faster with boosted write.

Fig. 8(c)-(d) shows the breakdown of total energy into leakage and dynamic energy. The proposed STTRAM-bWR decreases the dynamic energy consumption by 30% compared to STTRAM-WC-PV due to write pulse time reduction. However, it reduces the dynamic energy by 80% relative to STTRAM-bAll. Therefore the proposed write boosting is effective in improving the IPC (13%) and the energy (30%) in compare to STTRAM-WC-PV.

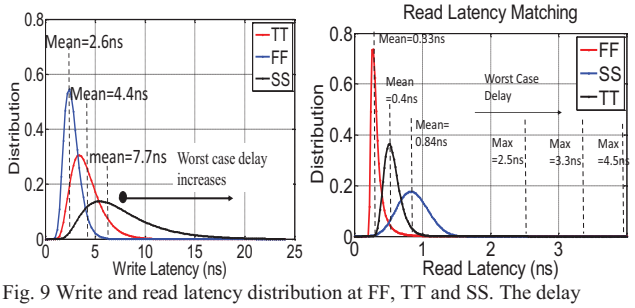


Fig. 9 Write and read latency distribution at FF, TT and SS. The delay spreads out at SS affecting the performance significantly.

E. Discussions on Improvement/Future Research

The scope of improvement for the proposed work is as follows:

Considerations for inter-die process variations: Although simulations are carried out at typical corner the proposed methodology is equally applicable for dies at other process corners. Simulation results indicate that read and write latency show similar spread in fast and slow corners (Fig. 9). The boost transistors can be designed taking inter- and intra-die process variations into account. Therefore the boost circuit should be able to provide the current needed for all process corners.

Fine grained boosting: In this work the entire global column is boosted even if only a single MTJ in a single local column is slow. This implementation is simple but it wastes power for fast MTJs. Fine grained boosting to consider data polarity (to exploit asymmetric write latency), MTJ and column selection can be explored to cut down unnecessary power.

Static vs. dynamic boosting: The proposed work implements static boosting. Therefore the slow columns are always boosted. It is possible to incorporate combination of static and dynamic boosting where static boosting will fix process variations and dynamic boosting can compensate for temporal degradations.

V. Related Work

The impact of inefficient writes is minimized in STTRAM by reducing the number of write operations using hybrid caches in which the frequently written blocks are stored in SRAM [8]. In [13], early write termination to prevent redundant write operations has been employed to reduce write energy of STTRAM. In [9], a hybrid design of SRAM L1 caches STTRAM L2 and L3 caches is proposed. The retention time of STTRAM is exploited to improve the write latency and write power [9]. Read-verify-rewrite scheme is proposed [10] that verifies the success of write operation and rewrites if needed. An improved idea that uses adaptive write period to improve performance while eliminating write errors in STTRAM [11]. A current source based two-step write scheme is proposed to improve the write energy and write latency [12]. Device-architecture space is explored to reduced write power by lowering the thermal energy to trade volatility [9,14-16]. Interesting circuit-architecture methods e.g., balanced write, flipped MTJ with sequential tag-data access and partial line update, 2T-1R with negative bitline, read optimized bitcell with stretched write cycle [17-20] have also been proposed. Sense margin related challenges in STTRAM and domain wall memory have been explored in [25-26] and a sizing, voltage biasing and shift current modulation technique has been proposed. For resistive memories such as Phase Change RAM (PCRAM), architectural techniques have been proposed to lower the write power through write termination [21] and improve performance by write pause [22]. Besides error correction techniques have also been suggested to mitigate failures in resistive memories [23]. *However, process variation induced write latency spread mitigation through write boosting is proposed for the first time*.

VI. Conclusions

STTRAM is a promising non-volatile memory technology for cache application due to high-density, low standby power, excellent retention, fast access time and good endurance. However, it can suffer from severe performance and power degradation due to process variation induced write and read latency variations. We propose a novel low-overhead write current boosting methodology that comprehends modeling, circuits and micro-architecture to address this issue. Simulations show 80% power improvement compared to boosting all bit-cells and 13% performance improvement compared to worst case latency due to process variation over a wide range of PARSEC benchmarks.

References

[1] Hosomi, M., et al. "A novel nonvolatile memory with spin torque transfer magnetization switching: Spin-RAM." *Electron Devices Meeting, 2005. IEDM Technical Digest. IEEE International*. IEEE, 2005.

[2] Kryder, Mark H., and Chang Soo Kim. "After hard drives—What comes next?." *Magnetics, IEEE Transactions on* 45.10 (2009): 3406-3413.

[3] Xuanyao, Fong, et al. "SPICE Models for Magnetic Tunnel Junctions Based on Monodomain Approximation," https://nanohub.org/resources/19048.

[4] Gem5, http://www.gem5.org.

[5] Parsec, http://parsec.cs.princeton.edu/index.htm.

[6] McPAT, http://www.hpl.hp.com/research/mcpat

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

[8] Zhou, Ping, et al. "Energy reduction for STT-RAM using early write termination." *Computer-Aided Design-Digest of Technical Papers, 2009. ICCAD 2009. IEEE/ACM International Conference on*. IEEE, 2009.

[9] Smullen, Clinton W., et al. "Relaxing non-volatility for fast and energy-efficient STT-RAM caches." *High Performance Computer Architecture (HPCA), 2011 IEEE 17th International Symposium on*. IEEE, 2011.

[10] Sun, Hongbin, et al. "Design techniques to improve the device write margin for MRAM-based cache memory." *Proceedings of the 21st edition of the great lakes symposium on Great lakes symposium on VLSI*. ACM, 2011.

[11] Bi, Xiuyuan, et al. "Probabilistic design methodology to improve run-time stability and performance of stt-ram caches." *Proceedings of the International Conference on Computer-Aided Design*. ACM, 2012.

[12] Lee, Dongsoo, and Kaushik Roy. "Energy-Delay Optimization of STT MRAM Write Operation Under Process Variations." (2014): 1-1.

[13] Rasquinha, Mitchelle, et al. "An energy efficient cache design using spin torque transfer (STT) RAM." *Proceedings of the 16th ACM/IEEE international symposium on Low power electronics and design*. ACM, 2010.

[14] Sun, Zhenyu, 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*. ACM, 2011.

[15] Swaminathan, Karthik, et al. "When to forget: A system-level perspective on STT-RAMs." *ASP-DAC*. 2012.

[16] Xu, Cong, et al. "Device-architecture co-optimization of STT-RAM based memory for low power embedded systems." *Proceedings of the International Conference on Computer-Aided Design*. IEEE Press, 2011.

[17] Li, Jing, et al. "Design paradigm for robust spin-torque transfer magnetic RAM (STT MRAM) from circuit/architecture perspective." *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on* 18.12 (2010): 1710-1723.

[18] Park, Sang Phill, et al. "Future cache design using STT MRAMs for improved energy efficiency: devices, circuits and architecture." *Proceedings of the 49th Annual Design Automation Conference*. ACM, 2012.

[19] Kim, Yusung, et al. "Write-optimized reliable design of STT MRAM." *Proceedings of the 2012 ACM/IEEE international symposium on Low power electronics and design*. ACM, 2012.

[20] Lee, Dongsoo, Sumeet Kumar Gupta, and Kaushik Roy. "High-performance low-energy STT MRAM based on balanced write scheme." *Proceedings of the 2012 ACM/IEEE international symposium on Low power electronics and design*. ACM, 2012.

[21] Zhou, Ping, et al. "A durable and energy efficient main memory using phase change memory technology." *ACM SIGARCH Computer Architecture News*. Vol. 37. No. 3. ACM, 2009.

[22] Qureshi, Moinuddin K., Michele M. Franceschini, and Luis Alfonso Lastras-Montaño. "Improving read performance of phase change memories via write cancellation and write pausing." *High Performance Computer Architecture (HPCA), 2010 IEEE 16th International Symposium on*. IEEE, 2010.

[23] Schechter, Stuart, et al. "Use ECP, not ECC, for hard failures in resistive memories." *ACM SIGARCH Computer Architecture News*. Vol. 38. No. 3. ACM, 2010.

[24] Bushnell, Michael, and Vishwani D. Agrawal. *Essentials of electronic testing for digital, memory and mixed-signal VLSI circuits*. Vol. 17. Springer, 2000.

[25] Motaman, Seyedhamidreza, and Swaroop Ghosh. "Simultaneous Sizing, Reference Voltage and Clamp Voltage Biasing for Robustness, Self-Calibration and Testability of STTRAM Arrays." *Proceedings of the The 51st Annual Design Automation Conference on Design Automation Conference*. ACM, 2014.

[26] Motaman, Seyedhamidreza, Anirudh Iyengar, and Swaroop Ghosh. "Synergistic circuit and system design for energy-efficient and robust domain wall caches." *Proceedings of the 2014 international symposium on Low power electronics and design*. ACM, 2014.