A Cross-layer Adaptive Approach for Performance and Power Optimization in STT-MRAM

Nour Sayed Rajendra Bishnoi Fabian Oboril Mehdi B. Tahoori
Karlsruhe Institute of Technology (KIT) Karlsruhe, Germany
E-mail: {nour.sayed, rajendra.bishnoi, fabian.oboril, mehdi.tahoori}@kit.edu

***Abstract*—Spin Transfer Torque Magnetic Random Access Memory (STT-MRAM) is a promising candidate as a universal on-chip memory technology due to non-volatility, high density and scalability. However, high write energy and latency are major challenges in this memory technology due to the asymmetry and stochastic nature of the write operation. Typically, the write current is set for the minimum energy point, which can further impact the write latency. To mitigate these issues, we propose an adaptive write current scaling technique that adjusts the write current, and hence the write latency and energy based on the performance needs at run-time. Using this technique, optimal energy and performance points for write current are obtained using detailed device and system level analysis. Furthermore, we use run-time adaptation of write current by predicting the write access rate for the next execution phase. We evaluate the efficiency of the proposed approach on SPEC2000 applications for STT-MRAM-based L1 and L2-cache levels. The results show that the effective write latency of L1 and L2 is reduced by 52.4% and 55.7% with 7.6% and 1.4% area overheads, respectively, corresponding to the overall system performance optimization of 15.5% while the total memory energy consumption is increasing by only 3.2%.**

I. Introduction

The conventional memory technologies such as Dynamic RAM (DRAM) and Static RAM (SRAM) suffer from high leakage power and scaling limitations in advanced technology nodes [1]. As the number of cores on a chip continues to increase with technology down-scaling, the demand for more on-chip memories grows significantly, which further aggravates the leakage and scaling issues. Therefore, the semiconductor industry is actively searching for alternative memory technologies, including non-volatile memories, which have close-to-zero leakage power for the bit-cell.

Spin Transfer Torque Magnetic Random Access Memory (STT-MRAM) is a promising candidate to be used as a universal memory for future multi-core and embedded systems. This is because of its higher density (as good as DRAM), non-volatility, scalability, fast read access (almost as fast as SRAM), CMOS-compatibility, and virtually unlimited endurance [2]. Despite all these benefits, the write operation is a main challenge in STT-MRAM due to the high energy and latency needed. The write operation in STT-MRAM is of stochastic nature, which means that the switching time of the bit-cell content is non-deterministic even under the same ambient and operational conditions, and hence, a large timing margin is required to satisfy a reasonable *Write Error Rate* (WER) [3]. This in turn further increases the write latency and energy penalty. On the other hand, the write probability depends mainly on a *Thermal Stability Factor* so-called Δ of the bit-cell, which indicates the energy barrier that must be overcome for state transition between '0' and '1' logic values. A lower Δ will make the cell switch faster and would require lower write current [4].

A direct replacement of SRAM with STT-MRAM for on-chip memories (caches) significantly reduces the leakage power at the cost of performance and dynamic energy due to the high write current and latency needed. Several techniques have been proposed to mitigate this issue. At the device-level, it is proposed to trade the retention ability of the cell for reduced write latency and energy by lowering Δ [4]. Consequently, the retention time of the memory, which is expected to be as large as possible, will be reduced exponentially and any slight increase in the operating temperature at run-time can affect it dramatically, resulting in an unacceptably high retention failure rate. Whereas, at circuit-level, several techniques [5–10] have been also proposed to improve the write characteristics of STT-MRAM. They are based on either i) detecting the actual switching time of each bit cell and stopping the unnecessary current flow immediately after the write completion [5–8], or ii) terminating the redundant bit writes at an early stage to remove the unnecessary writes [9, 10]. Such techniques do not improve the overall write latency for the complete word, since the write latency is fixed based on the cell with worst case switching characteristics.

At architecture-level, the common approach for reducing the write latency of STT-MRAM is to reduce the timing margin and exploit appropriate *Error Correction Codes* (ECCs) to correct latent write errors in the tail of the stochastic write distribution, as it is proposed in [11–14]. The amount of latency reduction is maximized with complex ECCs, however, encoding and decoding of complex ECCs impose significant area, energy, and performance overheads, which could erode the write latency improvement. All the previous techniques try to address the write challenges at design time regardless of the application behavior which have to be considered at run-time, and may affect the required write parameters significantly.

At system-level, two opposing forces of energy consumption are at play, as the system energy is the product of power and time. Any increase in the write current increases the power, but reduces the latency, hence there is a minimum energy point corresponding to the optimal write current. Based on the sensitivity of the application to the latency of the write operation, which correlates to write access rate, we can adaptively increase the write current for effective performance improvement. Thus, we propose a dynamic write approach for STT-MRAM to adjust the write current value on-the-fly according to the performance needs. In this work, we make the following contributions:

- We identify the opportunity of using an adaptive write current scaling approach for STT-MRAM by presenting a detailed energy-performance trade-off at device, architecture and system levels.
- Based on our cross-layer analysis, we obtain the optimal write current levels for different performance and energy needs of the system.
- We predict the sensitivity of the application performance to write latency, based on the write rate, to adjust the write current level at run-time.
- We allow to control the write current at run-time by modifying the write circuitry of our STT-MRAM design, which allows multiple levels of write currents.

We evaluate the proposed approach by analyzing the performance and energy consumption of a system for both L1 and L2 STT-MRAM caches using detailed simulations of SPEC2000 workloads. The results demonstrate that the effective write latency across all application phases is reduced by 52.4% and 55.7% for L1 and L2 caches, respectively. This corresponds to system performance improvement 15.5%, on average. The associated memory dynamic energy overhead is 1.7% because of the write current increase, and memory leakage energy overhead is 1.5% due to the proposed write circuit.

The rest of this paper is structured as follows. We start with an overview of the basics of STT-MRAM and further motivation of our work in Section II. Afterwards, in Section III, the proposed approach is discussed, followed by a comprehensive experimental study in Section IV. Finally, Section V summarize the main conclusions drawn from this work.

978-3-9819263-0-9/DATE18/©2018 EDAA

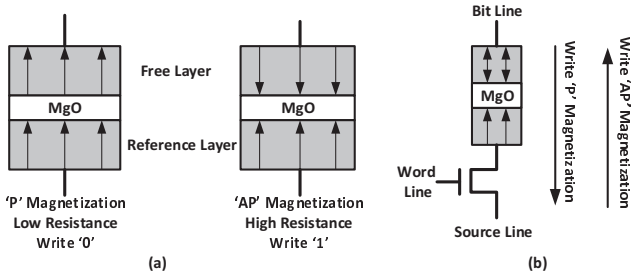


Fig. 1. Typical STT-MRAM bit-cell structure

II. Background

A. Basics of STT-MRAM

In STT-MRAM, data is stored in a *Magnetic Tunnel Junction* (MTJ), which consists of two ferromagnetic layers separated by a thin oxide layer. The magnetic orientation of one layer, namely free layer, can be freely rotated, while the magnetization of the other layer, the reference layer, is fixed. The magnetization of the free layer can be in parallel 'P' or anti-parallel 'AP' to the reference layer, which corresponds to low and high resistance states respectively. These two states are used to represent logic '0' and '1' (see Fig. 1(a)). A typical STT-MRAM bit-cell architecture is shown in Fig. 1(b). It consists of one MTJ cell and an NMOS access transistor, which is used to select the MTJ for memory operations.

In order to read the data stored in an STT-MRAM cell, a low current flows through the MTJ to sense the resistance state of the cell. On the other hand, to write a data into an STT-MRAM cell, a write current (I_w) much higher than the critical current (I_c), which is the minimum current required to switch the magnetization of the MTJ for a given write pulse, has to flow. The final magnetization state can be controlled by the write current direction (see Fig. 1). High write current results in a high dynamic energy. This issue is further exacerbated due to the stochastic nature of the writing (switching) process as well as the high sensitivity to process variation, leading to large timing margins [15].

B. Write Margin due to Stochastic Write Behavior

The MTJ cell has an inherent asymmetric switching behavior [16]. This phenomena is because of two main reasons: i) the spin-transfer efficiency factor of the free layer for switching its magnetic orientation to the same or the opposite spin direction of the reference layer, and ii) the impact of the voltage degradation of the access transistor, which reduces the current density. This increases the switching delay of the free layer to 'AP' state more significantly compared to 'P' state. As shown in Fig. 2(a), the 'AP' switching delay has a wider distribution with very long tail compared to the 'P' switching delay. On the other hand, the switching of the MTJ magnetic orientation is stochastic in nature due to the random thermal fluctuations. This means that the switching delay of MTJ-cell magnetization is not deterministic. However at system-level, the write period has to be fixed to a certain duration, which guarantees a target Write Error Rate (WER) per bit. The WER model can be expressed as in [17]:

$$WER_{bit}(t_w) = 1 - exp[\frac{-\pi^2 \cdot (I-1) \cdot \Delta}{4(I \cdot exp[C(I-1)t_w] - 1)}], \quad I = \frac{I_w}{I_c} \quad (1)$$

where t_w is the write period and C is a technology dependent parameter. I is the ratio of the write current (I_w) to the critical current (I_c) and Δ is the thermal stability factor. According to Eq. 1, any increase in the write current reduces the write period for a given error rate. However, the current increase leads to a higher power (P), as it is given as: $P = I_w^2.R$, R is the MTJ resistance value. Typically, the target WER for on-chip memories is of the order of 10^{-18} [3]. Fig. 2(b) illustrates the write error rate with the associated write latency for different write current values. As shown, for the same target write error rate, the write latency (i.e., the minimum time required to switch the cell content) reduces significantly with increasing the write current ratio.

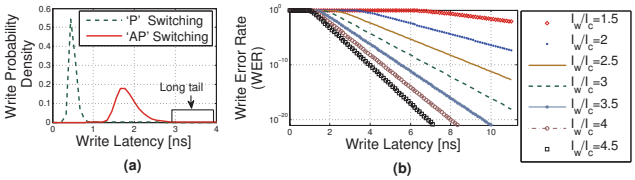


Fig. 2. (a) Write latency distribution for AP and P switching delays for a single bit-cell, (b) WER versus write latency for 512-bit data with different write current values, [$\Delta = 40$]

III. Adaptive Write Current Scaling

In this section, the proposed adaptive write current scaling approach for STT-MRAM is presented. In this regard, we first explain the main objective of the proposed approach. It is then followed by a detailed description based on analyzing the relation between the write latency and energy at device and system levels. The proposed application phase prediction methodology is presented next. Then, the modification of the write circuitry to allow dynamic current scaling is illustrated. Finally, the proposed cross-layer framework is described.

A. Objective

In this paper, the STT-MRAM design is improved based on the strong correlation between the write access rate and the write latency to maximize the performance along with minimal energy overhead. At run-time, the write current can be adjusted to different values (e.g., high, medium and low). In situations, where the write rate is high, the overall performance is very sensitive to the write latency. Thus, it is reduced by adjusting the write current. In contrast, when the performance is less sensitive to the write latency, the write current is reduced to save write energy. To enable such an adaptation, the overall execution time is divided into distinct phases, based on the sensitivity of the performance to the write access rate. For optimal performance improvements, we need to find an optimal write current value for each run-time phase. Therefore, this paper provides an analysis of write latency and its energy relation from device to system level, which accordingly is used for run-time phase identification and optimization.

B. Write Latency and Energy Relation

1) At device-level: The opportunity of improving the performance by increasing the write current at device-level is illustrated in Fig. 3. This figure shows the write energy and the write latency relations with the write current for both L1 and L2 caches. For L1, since the write operation is on the critical path, simple ECC-1 for one bit error correction is used to reduce the impact of stochastic write around 50%, at cost of only one cycle decoding/encoding penalty per each read/write access. Moreover, the adopted thermal stability factor (Δ) is low to guarantee fast write operations. Fig. 3(a) illustrates the energy and latency plots versus the write current for target WER of 10^{-9} with Δ = 30. On the other hand, the normal WER of 10^{-18} is considered for L2 with higher Δ equals to 40 to guarantee high retention time (see Fig. 3(b)). As write energy and latency plots in Fig. 3(a) and Fig. 3(b) present, the energy consumption is minimized for the write current values of 102 μA and 82 μA, which are corresponding to write latencies of 19.8 ns and 10.7 ns for L2 and L1-cache lines, respectively. However, the write latencies reduce modestly along with a significant increase in the write energy for higher write current. The figure shows that by increasing the write current two times, the write latency reduces by 66%, while the related write energy increases by 35% per access for L1 and L2 caches. It is important to note that such an increase in the write current does not impair the MTJ reliability due to time dependent dielectric breakdown (TDDB) of the oxide barrier.

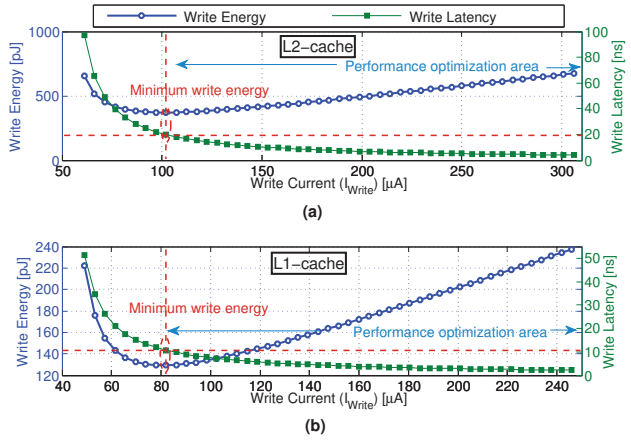


Fig. 3. Impact of increasing the write current on write energy consumption and write latency for a 512-bit cache line in case of AP switching at device-level [Δ = 30 for L1, Δ = 40 for L2, detailed setup in Table III]

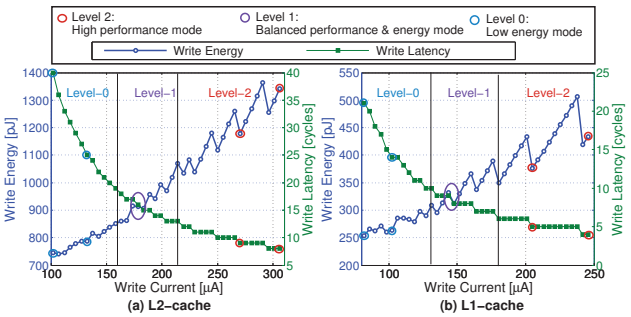


Fig. 4. Impact of increasing the write current on write energy consumption and write latency for a 512-bit cache line in case of AP switching at system-level [Δ = 30 for L1, Δ = 40 for L2, detailed setup in Table III]

2) At architecture-level: The continuous extracted device latency is from system-level perspective quantized into multiple clock cycle times with a fixed length, thus, the write latency and energy are discretized. Therefore, the curves of the write energy and latency against the write current are no longer smooth or monotonic (see Fig. 4). As shown in Fig. 4, different energy points have the same write latency, which in turn reduces the search space for the energy-latency optimal pareto points. Furthermore, Fig. 4 illustrates that the energy and latency curves are spread over three regions of the write current (low, medium and high). This helps us to determine the initial candid points for all regions. Identifying the optimal pareto points needs for each level (i.e., Level-0, Level-1 and Level-2) results simulations at system-level with detailed memory access patterns of applications. This enables to identify the impact of write latency and energy on the overall system performance and energy, respectively.

Table I summarizes the STT-MRAM parameters for both device and architecture levels, including two candid points (Point-1 and Point-2) for each scaling level. The selection of the actual point requires system level analysis which will be described next.

C. Proposed Write Phase Prediction Methodology

Our phase analysis method is split between an off-line write rate analysis and a run-time write rate prediction.

1) Off-line Write Access Analysis: The write access phase is defined as a period of execution time that exhibits a consistent write access rate which is distinguishable from the write rates in the other phases. From our off-line analysis, we generate a set of training data for run-time control using memory intensive workloads of the SPEC2000 benchmarks suite. From the write access counter measurements, we observed the following: i) Write rate can significantly vary over the time. ii) Write rate is highly periodic in repetitive and long-stable patterns for more than 90% of the entire run-time for all executed benchmarks. iii) While the periods and write rates are different for L1 and L2 caches, the repetitive and stable behavior is common for both. Fig. 5 plots write rate values for L1 and L2 caches over the entire execution time for the compression programs bzip2 and gzip. While we have done the analysis for all memory-intensive benchmarks and observed similar trends, we only show these two benchmarks for brevity.

Algorithm 1 Control algorithm for the $(i+1)^{th}$ iteration

Input 1- Off-line analysis:
{Window length L_W, tracing period TP}
2- Run-time history information:
{Run-time phase ($Phase_j$), write rate of the previous window WR_{Wi}}
3- STT-MRAM design configurations:
{Look-up table for available write current levels with correspond write rates }
Output 1- Write rate for the current window WR_{Wi+1}
2- Adopted write current (I_w) for the stable period SP_{Wi+1}
3- Run-time phase for W_{i+1}
1: **For** run-time **from** end of W_i **to** end of TP_{Wi+1} **do**
2: Calculate the number of write operations
3: **end for**
4: $WR_{Wi+1} \longleftarrow \frac{total\ write\ operations\ of\ TP_{i+1}}{TP}$
5: **if** WR_{Wi} != WR_{Wi+1} **then**
6: End the current run-time phase ($Phase_j$)
7: Set correspond write current I_w for WR_{Wi+1}
8: Start new run-time phase ($Phase_{j+1}$)
9: **end if**
10: **return** $Phase_{j+1}$, WR_{Wi+1} and I_w

It is important to note that the amount of the performance improvement offered by the adaptive write current scaling approach is derived from the amount of the Instruction Per Cycle (IPC) improvement. For on-chip memory, the IPC is closely linked with the number of memory write accesses. As the number of write accesses increases, a significant acceleration for the write operation is required to reduce the IPC degradation due to the long write latency. On the other hand, for low write rates, the IPC is not significantly affected by the adopted write latency. Therefore, in this case, the write parameters corresponding to the minimum energy point will be considered. This is the foundation for building our approach based on the off-line write phases analysis.

2) On-line Write Current Control Algorithm: The off-line analysis of stable write rate phases is the main guide to predict the write rate at run-time. However, the major challenge of the predictor is to accurately identify the correct time for changing the write current value (i.e., switching between modes). This relies on identifying the stable duration of write rate. The simplest way is to perform the prediction periodically. Therefore, we divide the execution time into small windows with a constant length. Each window consists of two parts: i) *Tracing Period* (TP), and ii) *Stable Period* (SP). SP is orders of magnitudes longer than TP. The tracing period information tells the predictor that the write rate, which was stable during the last tracing period, will be stable for the next stable period. Based on the off-line write rate analysis, we identify the possibility of dividing the execution time into windows. The optimal length of the run-time window is determined by greatest common divisor of all observed off-line stable phases.

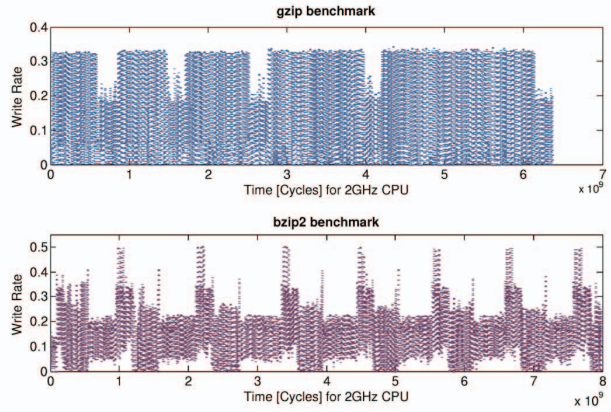
At run-time, the proposed predictor is implemented by a periodic control algorithm. The data learned during off-line analysis such as the window parameters (i.e., length, TP and SP) and the observed levels of the write rate with the required write current values are passed to the control algorithm. The pseudo-code of the iterative control algorithm is presented in Algorithm 1, which explains the diagram of run-time phases as shown in Fig. 6.

D. Modification of Write Circuitry

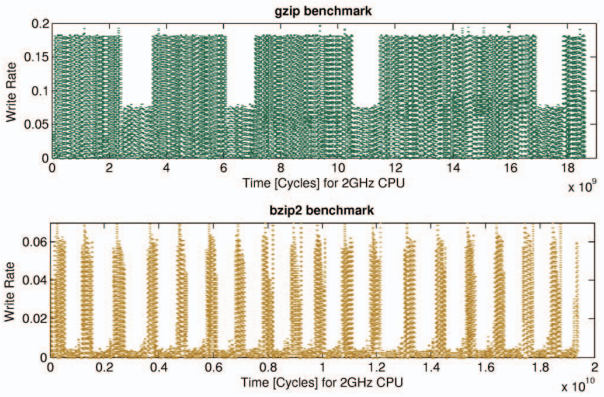
The circuit-level implementation of the proposed write circuitry to enable the selection of multiple levels of write currents is shown

TABLE I. STT-MRAM DESIGN CONFIGURATION FOR L1 AND L2 CACHES

				Write access rate					
				Low		Medium		High	
				Energy-efficient mode Level-0 configurations		Balanced-energy-performance mode Level-1 configurations		Fast-performance mode Level-2 configurations	
Cache level	Δ	WER	Candidate	Current [μA]	Latency [ns]	Current [μA]	Latency [ns]	Current [μA]	Latency [ns]
L1	30	10^{-9}	Point-1	82	10.7	147.6	4.1	200.9	2.8
			Point-2	102.2	7.1	164	3.6	246	2.1
L2	40	10^{-18}	Point-1	102	19.9	183.6	7.7	275.4	4.5
			Point-2	132.6	12.5	193.8	7.1	306	4



(a) L1-cache write access rate



(b) L2-cache write access rate

Fig. 5. Off-line write access phase analysis for L1 and L2 caches

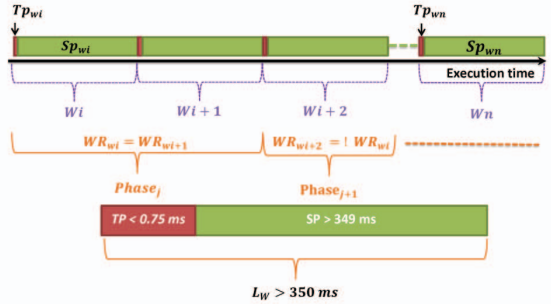


Fig. 6. Run-time write access rate prediction

in Fig. 7. Here, the write circuitry is the same as the one used for the standard write operation. The write circuitry is optimized to deliver the current that is adjusted based on the minimum energy point (i.e., energy-efficient mode) by default. In order to increase the write current and switch between levels, several transistors are employed that are activated using control circuits (C1, C2, .., Cn). Table II shows the truth table for switching between the three levels.

E. Proposed Cross-Layer Framework

In order to analyze the efficiency of the proposed idea, we have developed a cross-layer framework (as demonstrated in Fig. 8). We start from device-level analysis in SPICE all the way up to the system-level analysis in a performance simulator. For the bit-cell characterization, we run a device-level analysis for L1 and L2-caches in SPICE, based on TSMC 65 nm transistor models and the perpendicular STT-MRAM model presented in [19]. The device-level results are fed to NVSim [20] to obtain the read and write latencies of memory word for L1 and L2-caches to extract the architecture-level results. However, for the write latency, we model the stochastic behavior distribution according to the methodology described in Sec. II (B). At System-level we have two sub-components: off-line analysis and on-line control. In Off-line analysis, we simulated various applications of the SPEC2000 benchmark suite for the entire execution time. On the other hand, in on-line control, performance and energy analysis of the proposed approach are done for L1- and L2-caches.

TABLE II. TRUTH TABLE FOR CONTROL CIRCUIT

Level-1	Level-2	BL	SL	Operation	Mode
X	X	0	0	No write operation	—
0	0	0/1	1/0	Normal write operation	Energy-efficient
0	1	1	0	C1&C4 are ON	Balanced-
0	1	0	1	C2&C3 are ON	energy-performance
1	X	1	0	C1&C4&C5&C8 are ON	Fast-
1	X	0	1	C2&C3&C6&C7 are ON	performance

TABLE III. SIMULATION SETUP

Device model	Perpendicular STT-MRAM model with radius of 20 nm [19]
gem5 configuration	ISA ALPHA
Processor	Single-core, 2 GHz, Out-of-order, 4-issue
L1 cache	64 KB, 4-way set associative, 64B line size STT-MRAM, $\Delta = 30$, different write latencies
L2-cache	512 KB, 4-way set associative, 64B line size STT-MRAM, $\Delta = 40$, different write latencies
SPEC Applications	gzip, bzip2, mcf, twolf, vpr, sjeng, lbm

IV. SIMULATION RESULTS

This section first provides the details of our experimental setup. Then, the performance and energy analysis are discussed to demonstrate the total gain of the proposed approach for on-chip memories. Finally, the related area-overhead is evaluated.

A. Experimental Setup

At System-level, we extended gem5 simulator [21] to support different read and write latencies for each cache. The evaluations were performed using various applications of the SPEC2000 benchmark suite. For each application, the simulation was conducted over the entire execution time and over 0.5 ms for off-line analysis and on-line control estimations, respectively. Table III summarizes the set-up for our experiments.

B. SPEC2000 Behavior Analysis

From our off-line analysis, it can be noticed that the write rates of the SPEC2000 workloads for both L1- and L2-cache levels are repetitive and predictable over the run-time. The phases can be clustered into three levels of the write rate, high, medium and low, see Table IV. The greatest common divisor of all stable off-line phases is 350 ms. This identifies the length of the run-time window.

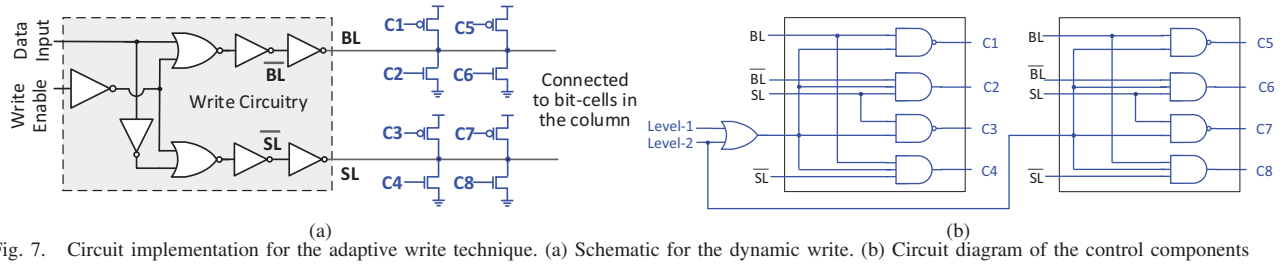


Fig. 7. Circuit implementation for the adaptive write technique. (a) Schematic for the dynamic write. (b) Circuit diagram of the control components

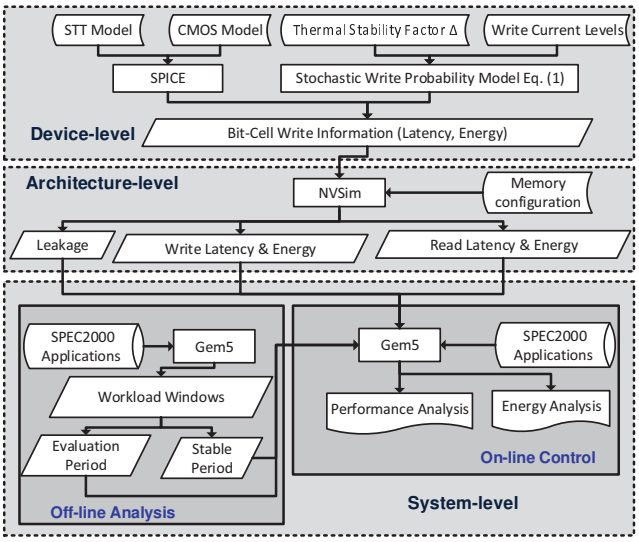


Fig. 8. Proposed cross-layer tool flow

TABLE IV. WRITE RATE PHASES OF SPEC2000

Write rate	L1-cache	L2-cache	Application
Low	WR $<$ 0.1	WR $<$ 0.05	bzip2,lbm,vpr
Medium	0.1 $<$ WR $<$ 0.2	0.05 $<$ WR $<$ 0.1	gzip,bzip2 twolf,sjeng
High	0.2 $<$ WR	0.1 $<$ WR	gzip,mcf

Whereas, the tracing period is determined by observing the longest time of the unstable write current value, which is less than 0.75 ms, as shown in Fig. 6.

C. Performance and Energy Analysis

For a direct comparison between a standard STT-MRAM design and our adaptive write current scaling approach, we normalized all performance and energy results to the extracted results of the minimum energy point (default current value), which refers to Point-1 of Level-0 (see Table I). In fact, as the L1-cache filters most accesses, the access rate for higher level cache (i.e., L2-cache) is low. Therefore, the leakage energy is responsible for 60% of the overall energy consumption for L2-cache. In contrast, for an L1-cache, the access energy is the major contributor. Fig. 9 illustrates the effects of the write current levels on the IPC and energy consumption for different SPEC2000 applications.

For the energy-efficient mode (i.e., Level-0), the total IPC of the low write rate phase is optimized with the second candidate (i.e., Point-2) by 14%, compared to the baseline (i.e., Point-1). This in turn results in reducing the leakage energy, which is the dominant portion of the total energy for L2-cache, by 15%. However, the dynamic energy is increased by 5%. Consequently, the total energy consumption is reduced by 8%.

For the balanced-energy-performance mode (Level-1), the performance (i.e., IPC) is improved by 10% and 11%, and the leakage is reduced by 10% and 12% for Point-1 and Point-2, respectively. However, the dynamic energy is increased by 17% and 23% with Point-1 and Point-2 compared to the standard design (i.e., Point-1 of Level-0). This leads to increase the total energy consumption by 1.5% and 3.5%.

In the high-performance mode (Level-2), the performance is optimized by 25% and 27% along with increasing the dynamic energy by 45% and 58% for Point-1 and Point-2, respectively. Whereas, leakage dissipation is reduced by 15% and 17%. Therefore, total energy dissipation increases by 15.3% and 24.3% for Point-1 and Point-2, respectively.

As illustrated in Fig. 9b (e), Point-2 of level-0, Point-1 of level-1 and level-2 are considered as optimal current points for L1 cache, since they offer maximum IPC with minimum dynamic energy consumption. Whereas, for L2 cache, where leakage energy is dominant, Point-2 of level-0, level-1 and level-2 are considered as optimal points. Table V summarizes the percentage of IPC and energy overhead improvements per phase of the proposed write approach over the standard one.

In order to estimate the overall improvements of the proposed adaptive approach, the energy overhead and performance improvements over each phase and the total breakdown of the execution over different phases should be considered. Our results show that the effective write latency for L1 is reduced by 52.4% (from 21 cycles to 10 cycles), while for L2, it is reduced by 55.7% (from 40 cycles to 18 cycles). Fig. 10 illustrates the overall performance improvement and energy overhead for different SPEC2000 applications for the proposed approach. It can be seen, that our approach improves the IPC compared to standard design, on average, by 15.5% at cost of increasing total memory energy dissipation by only 1.7%. It is worth mentioning that the proposed approach does not impair the MTJ reliability due to TDDB. Firstly, the increased current in the highest level is within the device tolerance. Secondly, this current increase happens only during write intensive phases of application execution, and the overall impact is limited.

It is important to note that the amount of IPC improvement and energy overhead due to the proposed approach are application dependent. For instance, for the applications with a low write access rate (e.g., lbm and vpr), there is no energy overhead compared to the standard design, as the energy-efficient mode has been considered. However, the total energy consumption is reduced by 8% because of improving the performance by 14%, which in turn reduces the leakage energy significantly, while the dynamic energy negligibly increases. However, for write intensive applications (e.g. mcf and gzip) with several high performance phases (level-2), the differences of the dynamic and the total energy dissipations between the proposed and the standard design become significant, up to 45% and 15.3%, respectively (see Fig. 10).

D. Area-overhead Evaluation

We need to organize the write circuit components in such a way that it is able to facilitate the highest required current in our proposed technique. This is handled by adding some additional drivers and its controlling circuitry as described in Fig 7. These additional circuitry

TABLE V. PERCENTAGE OF IPC IMPROVEMENT AND ENERGY INCREASE OF THE PROPOSED APPROACH OVER STANDARD STT-MRAM DESIGN

Power mode	Write Rate	Point	Increase of write current %		Reduction of write latency %		Increase of write energy %	Optimization of total performance %	Overhead of total energy %
			L1	L2	L1	L2			
Energy-efficient	Low	Point1	0%	0%	0%	0%	0%	0%	0%
		Point2	24%	29%	32.8%	27.8%	3%	14%	-8%
Balanced-energy-performance	Medium	Point1	70%	79%	58%	67%	38.8%	10%	1.5%
		Point2	100%	135%	62.8%	69.8%	45,7%	11%	3.5%
Fast-performance	High	Point1	140%	169%	76.6%	75%	64.6%	25%	15.3%
		Point2	200%	200%	81.3%	77.4%	76,6%	27%	24.3%

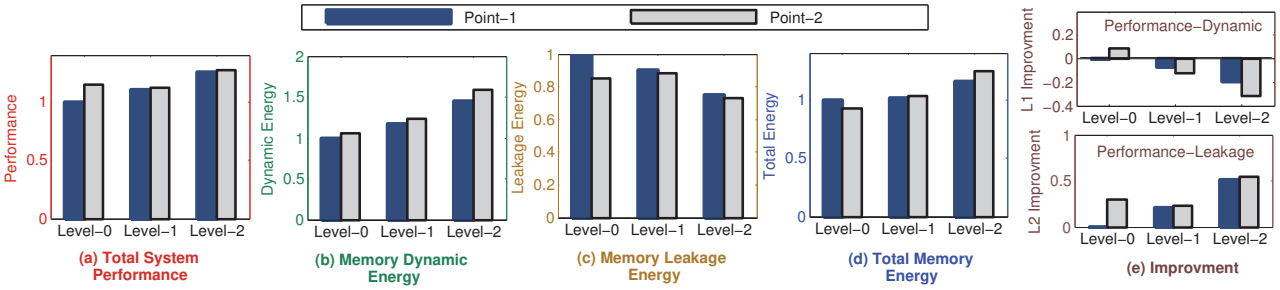


Fig. 9. Performance and total energy overhead (Dynamic+Leakage) for L1 ad L2 caches of the proposed approach normalized to standard STT-MRAM design without including (Dynamic+Leackage) overhead of the additional circuitry

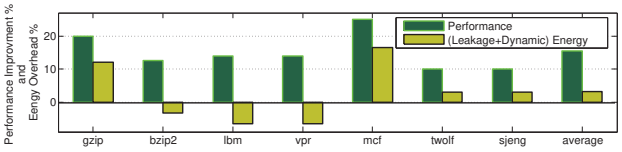


Fig. 10. Performance improvement and energy overhead (Dynamic+Leakage) of the proposed approach for various SPEC2000 workloads compared to standard approach including (Dynamic+Leackage) overhead of the additional circuitry

TABLE VI. AREA AND ENERGY OVERHEADS OF CIRCUIT-LEVEL IMPLEMENTATION OF PROPOSED APPROACH FOR L1 AND L2 CACHES

Cache	Area	Leakage	Dynamic energy
L1	7.6%	1.4%	1%
L2	1.4%	0.1%	0.5%

imposes area overhead of 7.6% and 1.4% for L1 and L2 caches, respectively. Those additional circuitry is employed column-wise and due to small size of bit-cell array in L1 cache, it incurs relatively high area overhead ratio compared to the L2 cache. Furthermore, the dynamic energy and leakage for our proposed circuit are also increased as mentioned in Table VI, due to addition of the extra components. On the other hand, the hardware cost associated with our control algorithm consists of a 16-bit counter and a simple look-up table for adjusting the write current based on the write rate of each level.

V. CONCLUSIONS

The write operation in STT-MRAM is the main bottleneck for the performance, energy and reliability of STT-MRAM. Traditionally, the write current is set to the minimum energy point. In this paper, by analyzing the behavior of the workloads, we observe that based on the write access rate of the application, the sensitivity of the overall performance to write latency changes. Based on this observation, we have developed a cross-layer adaptive write current level scaling scheme, in which a runtime predictor is used to predict the write access rate, and the write circuitry is modified to allow adjusting the write current levels at runtime. A detailed cross-layer analysis is performed to obtain the optimal write current level for the corresponding write access rate phases. Our results demonstrate that we can improve the overall system performance by 15.5% for a maximum energy overhead of 3.2% with 7.6% and 1.4% area overheads for L1 and L2- cache levels, respectively.

VI. ACKNOWLEDGEMENT

This work was partly supported by the European Commission under the Horizon-2020 Program as part of the GREAT project (http://www.great-research.eu/) and by ANR/DFG as part of the MASTA project.

REFERENCES

[1] W. Zhao et al., "New generation of predictive technology model for sub-45 nm early design exploration.", *IEEE*, vol. 53, no. 11 pp. 2816-2823, 2006.
[2] S. A. Wolf et al., "The promise of nanomagnetics and spintronics for future logic and universal memory." *IEEE*, vol. 98, no. 12, pp. 2155-2168, 2010.
[3] D. Apalkov et al., "Spin-transfer torque magnetic random access memory (STT-MRAM).", *JETC*, vol. 9, no. 2, pp. 13, 2013.
[4] C. W. Smullen et al., "Relaxing non-volatility for fast and energy-efficient STT-RAM caches.", *HPCA*, pp. 50-61, 2011.
[5] R. Bishnoi et al., "Self-timed read and write operations in STT-MRAM.", *VLSI*, vol. 24, no. 5, pp. 1783-1793, 2016.
[6] R. Bishnoi et al., "Improving write performance for STT-MRAM.", *IEEE Transactions on Magnetics*, vol. 52, no. 8, pp. 1-11, 2016.
[7] D. Suzuki et al., "Cost-efficient self-terminated write driver for spin-transfer-torque RAM and logic.", *IEEE*, vol. 50, no. 11, pp. 1-4, 2014.
[8] T. Zheng et al., "Variable-energy write STT-RAM architecture with bit-wise write-completion monitoring.", *ISLPED*, pp. 229-234, 2013.
[9] P. Zhou et al., "Energy reduction for STT-RAM using early write termination." *ICCAD*, pp. 264-268, 2009.
[10] R. Bishnoi et al., "Avoiding unnecessary write operations in STT-MRAM for low power implementation.", *ISQED*, pp. 548-553, 2014.
[11] B. D. Bel et al., Improving STT-MRAM density through multibit error correction., *DATE*, pp. 1-6, 2014.
[12] X. Bi et al., Probabilistic design methodology to improve run-time stability and performance of STT-RAM caches., *ICCAD*, pp. 88-94, 2012.
[13] N. Sayed et al., "Opportunistic write for fast and reliable STT-MRAM.", *DATE*, pp. 554-559, 2017.
[14] N. Sayed et al., "Leveraging Systematic Unidirectional Error-Detecting Codes for fast STT-MRAM cache.", *VTS*, pp. 1-6, 2017.
[15] M. Seyedhamidreza et al., Impact of process-variations in STTRAM and adaptive boosting for robustness. *DATE*, pp. 1431-1436, 2015.
[16] KW. Kwon et al., AWARE (Asymmetric Write Architecture With REdundant Blocks): A High Write Speed STT-MRAM Cache Architecture., *TVLSI*, vol. 22, no. 4, pp. 712-720, 2014.
[17] K. Munira et al., "A Quasi-analytical model for energy-delay-reliability tradeoff studies during write operations in STT-RAM.", *IEEE*, vol. 59, no. 8, pp. 2221–2226, 2012.
[18] J. Ahn et al., "DASCA: Dead write prediction assisted STT-RAM cache architecture." *HPCA*, pp. 25-36, 2014.
[19] A. Mejdoubi et al., "A compact model of precessional spin-transfer switching for MTJ with a perpendicular polarizer.", *MIEL*, pp. 225-228, 2012.
[20] X. Dong et al., "NVSim: A Circuit-Level Performance, Energy, and Area Model for Emerging Nonvolatile Memory.", *TCAD*, pp. 994-1007, 2012.
[21] N. Binkert et al., "The gem5 simulator.", *ACM SIGARCH*, vol. 39, no. 2, pp. 1-7, 2011.