Bulkyflip: A NAND-SPIN-Based Last-Level Cache With Bandwidth-Oriented Write Management Policy

Bi Wu, *Student Member, IEEE*, Pengcheng Dai, Zhaohao Wang, *Member, IEEE*, Chao Wang, Ying Wang, *Member, IEEE*, Jianlei Yang, *Student Member, IEEE*, Yuanqing Cheng, *Member, IEEE*, Dijun Liu, Youguang Zhang, *Member, IEEE*, Weisheng Zhao, *Fellow, IEEE*, and Xiaobo Sharon Hu, *Fellow, IEEE*

***Abstract*— High capacity last-level caches (LLCs) are being used to help alleviate the growing speed gap between the processor and main memory. However, traditional CMOS based memory technologies (SRAM, DRAM, et al.) for such LLCs consume high static power. Non-volatile memory such as STT-MRAM has been proposed as a low power solution for LLCs. Nevertheless, the high write current induces a so-called "supply current threshold" issue and limits the maximum number of bit-cells that can be written concurrently in one cycle in an STT-MRAM cache. This drawback significantly decreases the bandwidth of the STT-MRAM cache compared with SRAM. In this work, we present a hardware implementation of NAND-like spintronic memory (NAND-SPIN) LLC for the first time. By exploiting the unique *erase-then-program* operation for writing NAND-SPIN, we propose an adaptive buffer entry (ABE) write policy for each cache write access. Instead of writing a fixed number of bits sequentially, our method adaptively extends the write data length under a fixed maximum cache supply current. Compared to existing STT-MRAM caches, 'ABE' can achieve 70% performance improvements on average. Compared with the conventional early write terminate (EWT) policy, 'ABE' can save 33% write energy on average with negligible hardware overhead.**

***Index Terms*— NAND-SPIN, spin orbit torque (SOT) MRAM, last level cache, write throughput, high performance.**

I. Introduction

As THE technology node aggressively shrinks, static power consumption becomes a challenge of CMOS integrated circuits due to sub-threshold leakage [1]. Memory systems including on-chip caches and off-chip main memory suffer from even more severe leakage power consumption as their capacities grow up to meet the ever increasing memory bandwidth requirement [2], [3]. To tackle the notorious "Power Wall" problem, some non-volatile technologies, such as STT-MRAM [4], PCRAM [5] and ReRAM [6], have emerged to reduce leakage power consumption. Among them, STT-MRAM (whose bit-cell design is shown in Fig. 1 (a)) is one of the most promising candidates for on-chip cache design because of its fast access speed, near zero leakage and unlimited read/write endurance [7], [8].

Unfortunately, several shortcomings still hinder the commercial application of STT-MRAM such as the high program latency, inefficient power consumption, and asymmetrical switching process [9]–[11]. These issues are partly addressed by exploiting spin orbit torque (SOT) which is an emerging physical mechanism for ultrafast magnetization switching [12], [13]. However, as shown in Fig. 1(b), the standard SOT-MRAM bit-cell contains two access transistors associated with a three-terminal MTJ device, which induces large area overhead and array complexity especially when it is utilized as large capacity caches. From this perspective, SOT-MRAM still cannot fully meet the design requirements of large-capacity cache, despite its ultrafast speed and low energy. In addition, the access transistors in both STT-MRAM and SOT-MRAM suffer from severe source degeneration issue [14] due to the bi-directional write current, which also exacerbates asymmetry between two write directions.

To overcome the shortcomings of both STT-MRAM and SOT-MRAM, recently a novel spintronics memory with NAND-like cell structure (called NAND-SPIN) is proposed and is shown in Fig. 1 (c) [15]. In the NAND-SPIN, multiple MTJs are fabricated above the same heavy metal strip, and the

Manuscript received August 18, 2019; accepted October 1, 2019. Date of publication November 15, 2019; date of current version January 15, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61704005, in part by the International Collaboration Project under Grant B16001, in part by the National Key Technology Program of China under Grant 2017ZX01032101, in part by the Special Foundation of Beijing Municipal Science and Technology Commission under Grant Z161100000216149, and in part by the State Key Laboratory of Computer Architecture, Institute of Computing Technology, Chinese Academy of Sciences, under Grant CARCH201602. This article was recommended by Associate Editor M. Mozaffari Kermani. *(Bi Wu and Pengcheng Dai contributed equally to this work.) (Corresponding authors: Zhaohao Wang; Ying Wang; Weisheng Zhao.)*

B. Wu, P. Dai, and C. Wang are with the Beijing Advanced Innovation Center for Big Data and Brain Computing, School of Microelectronics, Fert Beijing Research Institute, and the School of Electronics and Information Engineering, Beihang University, Beijing 100191, China (e-mail: bi.wu@buaa.edu.cn).

Z. Wang, Y. Cheng, and W. Zhao are with the Beijing Advanced Innovation Center for Big Data and Brain Computing, School of Microelectronics, Fert Beijing Research Institute, Beihang University, Beijing 100191, China (e-mail: zhaohao.wang@buaa.edu.cn; weisheng.zhao@buaa.edu.cn).

Y. Wang is with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China (e-mail: wangying2009@ict.ac.cn).

J. Yang is with the Beijing Advanced Innovation Center for Big Data and Brain Computing, School of Computer Science and Engineering, Fert Beijing Research Institute, Beihang University, Beijing 100191, China (e-mail: jerryyangs@gmail.com).

D. Liu is with the China Academy of Information and Communications Technology (CAICT), Beijing 100191, China (e-mail: liudj@datanggroup.cn).

Y. Zhang is with the School of Electronics and Information Engineering, Fert Beijing Research Institute, Beihang University, Beijing 100191, China (e-mail: 05396@buaa.edu.cn).

X. S. Hu is with the Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN 46656 USA (e-mail: shu@nd.edu).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSI.2019.2947242

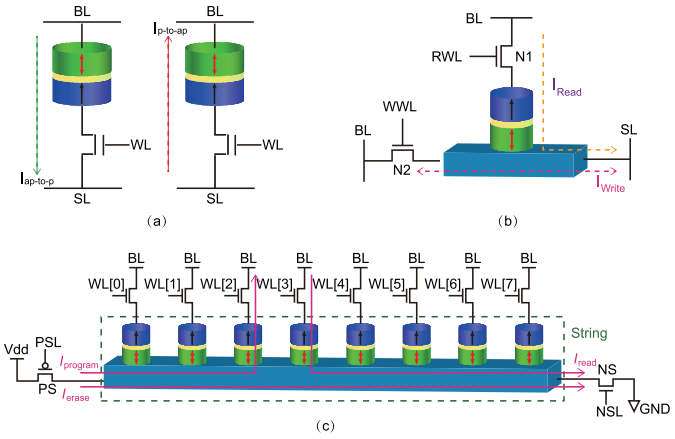


Fig. 1. Standard bit-cell of MRAMs: (a)1T1MTJ STT-MRAM cell structure; (b)2T1MTJ SOT-MRAM cell structure; (c)NAND-SPIN cell structure.

write operation is achieved by a SOT-induced erasing followed by a STT-induced programming. Compared with STT-MRAM and SOT-MRAM, NAND-SPIN could achieve larger density, faster operating speed and lower write power consumption especially when adopted as large capacity caches. In addition, owing to the unique write operation method, NAND-SPIN could completely overcome the source degeneration issue and the asymmetric switching drawback of STT-MRAM. However, most of the traditional architecture-level write policies cannot function well in the NAND-SPIN architecture because bit-cells in NAND-SPIN are no longer independent of each other [15], as can be seen in Fig. 1(c).

Another critical weakness of STT-MRAM is the limited number of bits that can be written concurrently. It is mainly caused by the imbalance between the high write current of STT-MRAM and the limited power supply current, and can severely restricts the cache write throughput. According to the recent tapeout work by TSMC [16], the write buffer width is only 16 bits, which is much shorter than the 512 bit wordline and incurs inefficient data exchange. Fortunately, this weakness can be overcome by using the NAND-SPIN combined with a well-designed write policy.

In this work, for the first time, we present a hardware implementation of novel NAND-SPIN LLC beyond the previous non-random-access NAND-SPIN memory [15], where the entire block has to be erased before writing any one bit-cell. For further optimization of the write bandwidth, we propose a novel write policy for NAND-SPIN based LLCs by utilizing the unique *erase-then-program* write operation. A pre-set flag bit and an inverting operation are introduced to the LLC design to control the number of actual data transfers between the cacheline and the write buffer to minimize write energy. To improve the write throughput, we introduce two control policies for the cache write buffer. One of the policies is fixed width invert (FWI) which doubles the write throughput by writing a fixed number of write buffer entries. The other policy is adaptive buffer entry (ABE) which adaptively adjusts the write data size depending on the total number of '0's. To validate the effectiveness and efficiency of our proposed policies, we construct both 'FWI' and 'ABE' cache systems on GEM5 simulation platform. Both single core and multi-core processors simulations are performed. As a result, the write throughput is increased significantly. In addition, the write power and write speed can be improved as well thanks to the high-efficiency switching mechanisms of NAND-SPIN.

The rest of the paper is organized as follows. Section II introduces the basics of STT-MRAM, SOT-MRAM and NAND-SPIN. Section III describes the motivation of our work by investigating the NAND-SPIN cache performance and write throughput bottleneck of conventional STT-MRAM cache designs. Section IV details the proposed 'FWI' and 'ABE' write policies, and investigates the design tradeoff between these two policies. Experimental results are given in Section V. Section VI presents the related work, and Section VII concludes the paper.

II. Preliminaries of MRAM

As one of the most promising candidates for the next generation memory technology, STT-MRAM has some unique benefits like nonvolatility, high integration density and high read speed. As shown in Fig. 1(a), the typical 1T-1MTJ cell structure consists of 1 NMOS transistor and 1 Magnetic Tunnel Junction (MTJ). The bi-directional current across the MTJ switches the magnetization of the free layer (upper layer) to be parallel or anti-parallel (P or AP) to that of the pinned layer (bottom layer). Depending on magnetizations of the two layers, the MTJ can be set to low or high resistance states, and a '0' or '1' can be stored.

However, STT effect incurs an incubation delay which limits the switching speed. In addition, read and write paths are coupled in a STT-MTJ, leading to difficulty in addressing the read disturb. In this case, SOT mechanism has been investigated to resolve these problems. As shown in Fig. 1(b), the spin accumulation in the bottom heavy metal generated by a charge current induces a SOT to switch the magnetization of the free layer [17]. SOT-driven magnetization dynamics can be described by the following Landau-Liftshitz-Gilbert equation as [18],

$$\frac{\partial \vec{m}}{\partial t} = -\gamma\,\mu_0 \vec{m} \times \vec{H_{eff}} + \alpha \vec{m} \times \frac{d\vec{m}}{dt} - \lambda_{DL}\xi J \vec{m} \times (\vec{m} \times \vec{\sigma}) - \lambda_{FL}\xi J \vec{m} \times \vec{\sigma} \quad (1)$$

where $\vec{m}$ is the unit vector of the free layer magnetization, $\vec{H_{eff}}$ is the effective field, $\vec{\sigma}$ is the unit vector of the SOT-induced spin polarization, γ is the gyromagnetic ratio, μ_0 is the vacuum permeability, α is the damping constant, J is the SOT current density, ξ is a device-dependent parameter, and λ_{DL} and λ_{FL} represent the strengths of the damping-like and field-like torques respectively. When J exceeds the critical current density J_0, a '0' or '1' is programed into the upper MTJ with ultralow energy consumption and ultra fast switching speed compared with STT-MRAM.

However, a standard SOT-MRAM bit-cell contains two access transistors as seen from Fig. 1(b), leading to lower integration density compared to STT-MRAM. Note that the two access transistors need to have the same width due to the layout limitation, even if the read current is much smaller than the write current [19]. As the on-chip cache capacity

is increasing rapidly, large area overhead would swallow the benefits obtained.

In addition, in a STT-MTJ the P-to-AP switching is intrinsically harder than AP-to-P switching. Such an asymmetry degrades the write performance of the STT-MRAM. Moreover, for 1T1MTJ bit-cell, the source degeneration issue of the transistor induced by the bi-directional current further exacerbates the write asymmetry.

To overcome the above challenges of SOT-MRAM, NAND-SPIN has been proposed. As shown in Fig. 1(c), similar to NAND-Flash memory [20], NAND-SPIN is organized physically in strings. A NAND-SPIN string contains 2^N MTJs which can be integrated into a single structure. The free layers of MTJs within the same string are contacted to the same heavy-metal strip. Therefore the actual number of terminals for each MTJ is reduced compared to the SOT-MRAM. As a result, higher integration density can be achieved. Each MTJ is connected to an access transistor whose gate and drain act as the wordline (WL) and bitline (BL), respectively. In addition, each string is equipped with two selection transistors (PS and NS in Fig. 1(c)), one of which can be shared by multiple strings within the same row.

Inherited from SOT-MRAM switching mechanism, NAND-SPIN write operation is divided into two steps: first, activate the selection transistors (PS and NS), a charge current (I_{erase}) passes the heavy metal strip to generate the SOT effect, which erases all the MTJs within the strip to AP states (data '1'). Second, for those bit-cells to be switched to P states (data '0'), their access transistors and PS transistors are activated. The BLs are grounded to produce a current ($I_{progrem}$ in Fig. 1(c)) flowing from the free layer to pinned layer to switch those MTJs to P states. Thus the harder P-to-AP STT switching is avoided in the NAND-SPIN. Instead, the AP state is written by the high-efficiency SOT erase operation. Moreover, both SOT and STT currents are unidirectional in the NAND-SPIN, therefore the source degeneration issue of access transistor is resolved. However, without any specific scheduling strategy, compared to SOT-MRAM, the improvement caused only by the intrinsic performance of NAND-SPIN is not significant. In this case, the unique two-step write operation of NAND-SPIN produces an opportunity to optimize the cache performance. To read a specific MTJ on a strip, the access transistors and NS transistors are activated. Then BLs are connected to the sensing amplifier to read out data.

TABLE I

NAND-SPIN DEVICE PARAMETERS USED IN SIMULATIONS [15]

Symbol	Value
MTJ free layer thickness	$1nm$
Heavy metal thickness	$4nm$
Damping constant	0.02
H_k	$1.44 \times e^5 A/m$
Resistance-area product	$5\Omega \cdot \mu m^2$
Saturation magnetization	$1150KA/m$
Spin Hall angle	0.3
Tunneling spin polarization	0.62
Heavy metal resistivity	$200\mu\Omega \cdot cm$

TABLE II

CIRCUIT LEVEL PARAMETERS USED IN SIMULATIONS [15]

Parameter	STT-MRAM	SOT-MRAM	NAND-SPIN
$S(F^2)$	40.33	9∗2	7.95
$T_{read}(ns)$	1.62	1.63	1.65
$T_{write}(ns)$	6 for '0' to '1' 4 for '1' to '0'	0.7 for '0' to '1' 1 for '1' to '0'	1 for erase 4 for program
$P_{Read}(fJ)$	15.336 for '0' 16.285 for '1'	15.987 for '0' 16.78 for '0'	19.173 for '0' 20.134 for '0'
$P_{Write}(fJ)$	627 for '0' to '1' 1387 for '1' to '0'	178.6 for '0' to '1' 127.2 for '1' to '0'	erase:30.91 /bit Program:369.7 /bit

∗Prerequisites: Write latency of STT-MRAM and SOT-MRAM are clamped at 6 ns and 1 ns respectively.

III. MOTIVATION

A. Performance of NAND-SPIN Cache

According to the above discussions, NAND-SPIN promises to achieve higher-density and more energy-efficient write operation compared with the STT-MRAM and SOT-MRAM. These advantages enable the NAND-SPIN to be applied in high-capacity scenarios like LLCs. To validate this point, we have conducted a detailed study to explore the NAND-SPIN based cache design space using NVSim [21]. The NAND-SPIN parameters used for the simulation are shown in Table I and the circuit level parameters are shown in Table II (gathered from [15]) which represents the state-of-the-art technologies.

Note that, this work adopts the delay clamp assumption for all the MRAM access transistors size setting. This means that the transistors can be adjusted to different sizes for meeting appropriate switching speed prerequisites. For example, as shown in Table II, instead of using the same size as the STT-MRAM (e.g. 40.33 F^2 each in this work) [22], the SOT-MRAM cell area could be reduced to 9 F^2. These settings lead to 6 *ns* and 1 *ns* switching delays of STT-MRAM and SOT-MRAM, respectively, which are consistent with the state-of-the-art technologies. The rest of this paper follows the same assumption.

As shown in Fig. 2, we compared cache area of NAND-SPIN with other two MRAM technologies under different capacities for the 40 nm technology node. NAND-SPIN achieves the best area efficiency among all the technologies. Compared to the SOT-MRAM, the NAND-SPIN cache still shows almost 50% area reduction. In Fig. 2, when the capacities are larger than 8MB, the NAND-SPIN read latency become mighty as the much smaller area leads to much lower interconnection latency/power. Considering the write operation, although the latency of NAND-SPIN is larger than that of SOT-MRAM due to the two-step operation, it is still lower than the other technologies. Note that, the write latencies of both STT-MRAM and SOT-MRAM vary little in the case of small capacity (e.g. $<$ 32 MB), since the inner cell latency is a dominating factor. However, as the capacity increases, the interconnection delay plays a more and more important role, and thus the write latency rises significantly (see the results of Fig. 2 while larger than 32 MB). In addition, the write energy for NAND-SPIN is always better when the capacity is larger than 16 MB due to the less interconnection overhead. In summary, under traditional cache without any optimizations, NAND-SPIN works best in most of the parameters when

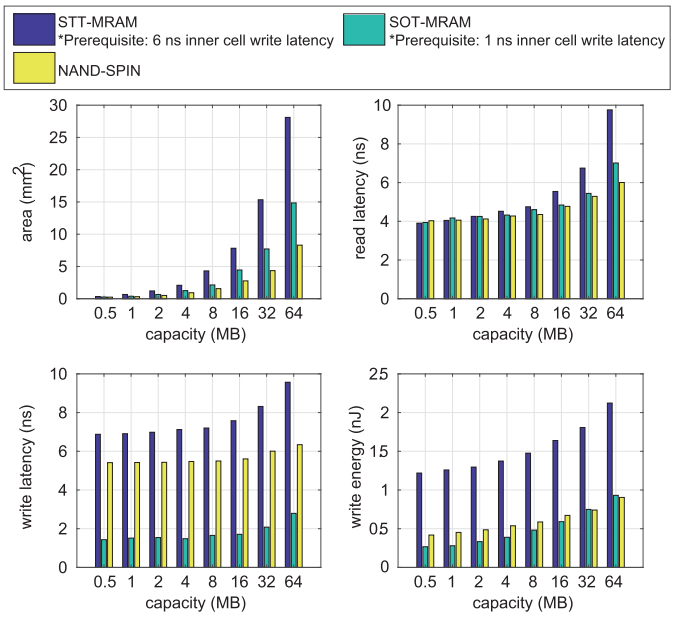


Fig. 2. Comparisons among STT-MRAM, SOT-MRAM and NAND-SPIN.

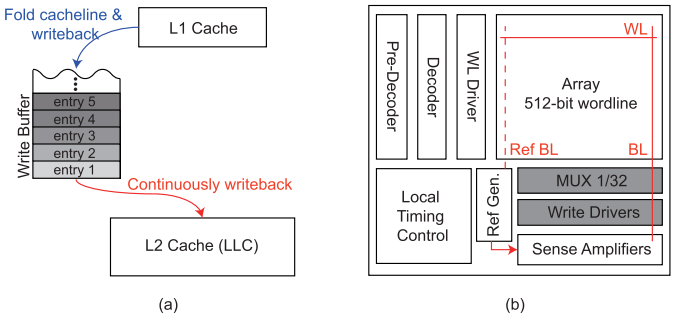


Fig. 3. (a) Write back workflow in typical memory architecture; (b) Typical cache memory hardware structure of STT-MRAM [16].

it is adopted into the large capacity application scenarios (such as LLC).

B. Write Throughput Bottleneck of STT-MRAM LLC

An important factor to be considered in cache design is the write policy. Two general write policies are writethrough and writeback [23]. In these two policies, a fixed size write buffer plays an important role in system performance. As shown in Fig. 3(a), the write buffer is a FIFO (First-In-First-Out) like cache queue consisting of several entries. Each entry carries the data that can be written in parallel within one write cycle. In this case, the entry width (also called write buffer width) determines the performance of the LLC.

However, for STT-MRAM cache, the high write current limits the number of bits that can be written concurrently, i.e., the write buffer width. As shown in Fig. 3(a), before pop down, L1 cache will firstly fit the data into the write buffer width. This operation produces a bottleneck for the write efficiency. For example, if the baseline L1 to L2 data width is 64 bytes, for a 8-bit width write buffer, the writeback data will be fit into 64 segments for the following continuous writing. This issue indeed exists in practical applications. As shown

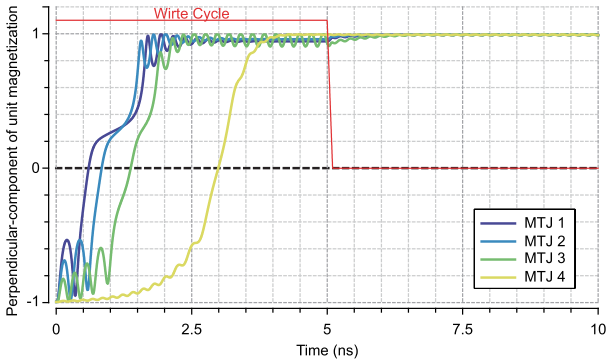


Fig. 4. An example of writing 4 MTJs on the same heavy metal strip in parallel.

in Fig. 3(b), in the most recently tapeout of STT-MRAM by TSMC [16], the concurrent write width is only 16 bits (1/32 MUX for 512 bits cacheline). Simply enlarging the write buffer width encounters the on-chip 'supply current threshold' problem. Furthermore, the low write efficiency induces performance degradation in two aspects: first, for large-capacity cache with a multi-bank structure, the write operation can block the interface leading to the decrease of read speed since the number of interconnection ports for each bank is very limited. Second, for the entire LLC, the blocked write buffer exacerbates the risk of the buffer overflow which blocks the following write accesses. Thus, it is desirable to find an approach to overcome this write buffer width challenge.

IV. The Invert Based Cache Write Policy of NAND-SPIN LLCs

A. Hardware Implementation of NAND-SPIN LLCs

It is easy to observe that the occurring probabilities of '0' and '1' in an entry are usually not equal, and they strongly depend on the application characteristics. By leveraging the data inversion methodology, the two-step write operation of NAND-SPIN provides an opportunity to optimize the cache throughput. Below, we first introduce the hardware implementation of NAND-SPIN LLCs which is used for benchmarking the practical effectiveness of the following write policy. Then two write buffer control policies are proposed to settle the throughput problem.

As shown in Fig. 3(b), the LLC in this work is similar to the recent tapeout STT-MRAM chip equipped with 64 single port banks [16]. The write buffer entry of L1 cache is 16 bits wide. In addition, the cache hardware structure is shown in Fig. 5. The PS and NS transistors are shared by MTJs within the same string, which minimizes the area occupation of the NAND-SPIN cache. For each strip, more MTJs require larger PS and NS transistors, in this work we place 4 MTJs on each strip which can be written concurrently by I_{erase} and $I_{write'0'}$. As shown in Fig. 4, we performed simulations with the Cadence Spectre simulator [24] to test the feasibility of writing 4 MTJs in parallel in the NAND-SPIN. The results show that within the same long enough write cycle, all 4 MTJs can be switched to '0' concurrently despite their different

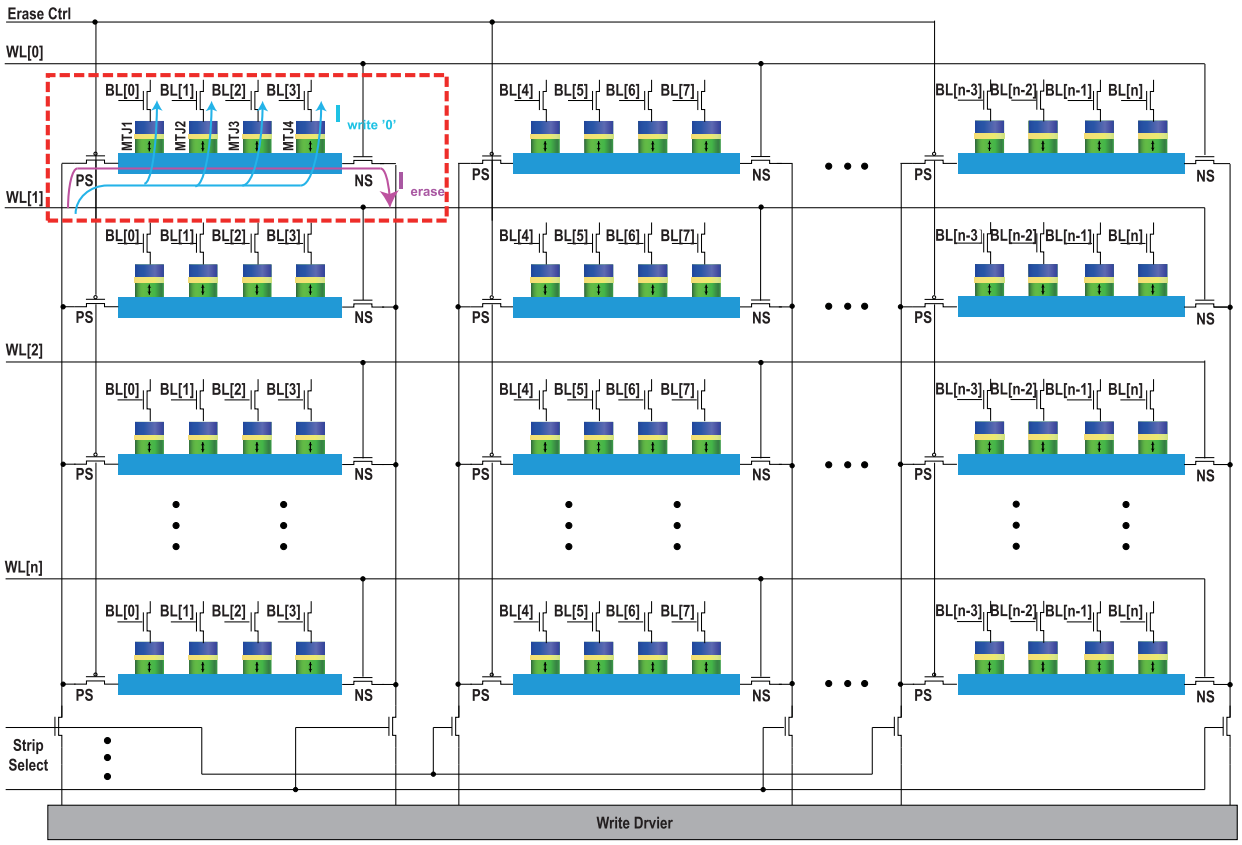


Fig. 5. Cache array circuit schematic diagram of NAND-SPIN LLCs.

switching delays. The feasibility and reliability of this parallel writing will be discussed in Section V.

To validate the efficiency of our write policies, we also construct another comparison architecture with the 'EWT' read before write policy [25]. Before data is written into the target cacheline in the LLC, a read operation should be performed to determine the original data. After that, the new data is compared with the read out data bit-by-bit. If the new data bit is already stored in the cache line, the write process for this bit will be terminated. With this policy, the redundancy can be completely avoided to achieve a dramatic energy reduction. However, as discussed in [25], the read out process incurs a large latency and energy overhead, which is the main drawback of this policy.

B. Fixed Width Invert Write Policy for NAND-SPIN LLCs

Although the currents of writing '1' and '0' in STT-MRAM are different, the circuits have to be designed to tolerate the worst case. However, this design incurs excessively high write energy and additional write driver complexity for the better case. Even worse, without the 'EWT' policy, all the data including the redundant data should be written together. In contrast, NAND-SPIN can achieve more energy-efficient write operation. We use an example shown in Fig. 6 to illustrate further. During step 1, all the bit-cells are uniformly erased to '1'. As mentioned above, this erase operation is implemented by SOT effect. Thus ultrafast erase speed can

Fig. 6. An example of NAND-SPIN two-step data writing process.

be obtained. During step 2, only '0's are written into the specific bit-cells by the STT. During the entire write operation, the STT is only responsible for low-cost writing '0', as writing '1' has been implemented with high-efficiency SOT effect and shared by multiple bit-cells. Therefore, the waste of write energy induced by the STT asymmetry is avoided.

Although the step 2 achieves lower energy and smaller latency than those of conventional STT operation, it still accounts for a large portion of the total energy and latency compared with step 1. Considering the ideal case that only step 1 is performed (i.e. the data to be written are all '1's), the write efficiency could be maximized. However, such an ideal case is highly unlikely to happen in the real system. Our goal is to reduce the number of '0' to be written by designing a write policy and a supporting architecture. In this way, the energy consumption can be lowered.

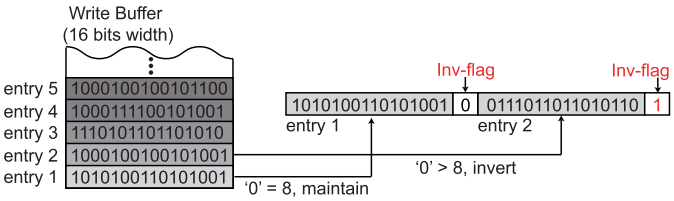


Fig. 7. An example of 'FWI' write policy.

Refer to the write buffer design in Section III, the write buffer is a multi-entry memory with a fixed width. Within an entry, we proposed a Fixed Width Invert (FWI) write policy to keep the total number of writing '0' less than or equal to half of the entry width. To support 'FWI', we add an additional flag bit, inv-bit, to each write buffer entry (see Fig. 7). Besides judiciously deciding whether to write the original or the inverted data, 'FWI' also bundles two write buffer entries into one which doubles the write throughput while still ensure that the total write current is below the supply current threshold.

Specifically, 'FWI' works as follows (referring to Fig. 7). Before writing the data from the write buffer to the specific cache line, the cache controller first checks the head two entries. After that, instead of fetching data one entry by one entry, the first two entries are obtained together. If the number of data '0's in any one of these two entries is larger than half of the entry width (8 bits in this work), all the data in this entry are inverted for the following cacheline writing. Additionally, the invert flag, inv-flag appended to this 16-bit data indicates the inversion state of this data segment. In the read process, the readout data segment only needs to be re-inverted if the inv-flag is set (i.e., storing '1'). By applying the 'FWI' to NAND-SPIN LLCs, all the data of these two entries are erased to write all 32 bits data '1's into the target cacheline. Then less than (or equal to) 16 bits data '0's will be written concurrently. Actually, in this process, 32 bits valid data are written into the target cacheline in parallel. As we discussed previously, the write throughput is mainly dependent on the practical width of the write buffer. Thus, 'FWI' policy almost doubles the buffer width with an acceptable hardware overhead. As a result, the write throughput can be significantly improved.

In terms of overhead, as shown in Fig. 7, within memory array each 16 bits data need 1 bit flag to label the inverted state of this data segment, which incurs a 6.25% capacity overhead. Even we adjust the hardware structure of write buffer to 32 bits width, the overhead is still as large as 3.125%. For a cache chip, this magnitude of capacity loss results in performance degradation. For the buffer part, we construct RTL simulations for three types of write buffers, the conventional 16 bits width buffer, 17 bits width buffer for 'FWI' and 16 bits width buffer in 'ABE' with 1-bit additional tag for every 512 bits (1 wordline). Each buffer depth is 1024 entries. As shown in Table III, the overhead of 'FWI' policy is 5.26% larger than the orginal write buffer. In addition, for the last level cache memory, the area of memory cell array occupies most of the area of the entire chip. Thus, this part of overhead is ignorable. In addition, in each buffer entry, the maximum number of

TABLE III

HARDWARE OVERHEAD COMPARISONS IN THREE TYPES OF WRITE BUFFERS

	Original (16 bits)	FWI (17 bits)	ABE (16 bits with 1 bit register for each 512 bits)
Logic utilization	570	600	587
Hardware overhead	–	5.26%	2.98%

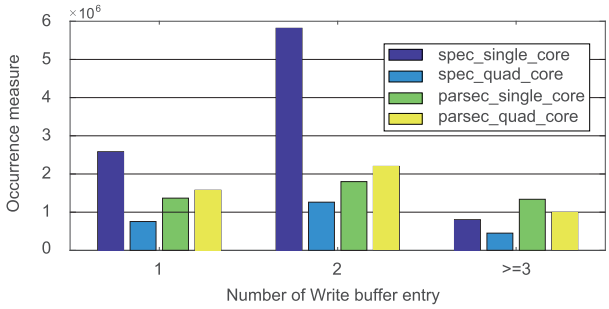


Fig. 8. The average number of write buffer entries that contains no more than 16 '0's in total, assuming a fixed write buffer width of 16.

writing '0' is 8. Considering the fixed 16 bits buffer bandwidth, only two entries are written during one write cycle. So in these aspects, although the 'FWI' write policy could achieve nearly double speedup, it is still has room for improvements.

C. Adaptive Buffer Entry Write Policy of NAND-SPIN LLCs

As discussed previously, in NAND-SPIN LLCs, the data '1' could be regarded as redundant data, which can be handled together in the erase process. To explore the redundant data distribution in cache access process, we conducted simulations for a 16MB 8-way associative L2 cache in a single-core and quad-core systems by using modified gem5 [26]. (Details about the experimental setup are presented in Section V). We gathered the number of write buffer entries that contains no more than eight '0's in total assuming a fixed write buffer width of 16. The average results for both PARSEC and SPEC suites are shown in Fig. 8. In this experiment we fix the bit-count of writing '0's to 16 per cycle to preliminarily capture the accessing features of the cache system. From the experimental data, one can see that during each write cycle, the 16 data '0's can be distributed into more than 2 buffer entries. Assume that the supply current threshold sets the maximum number of concurrently writable '0's to 16, if the number of data '0's of the first 3 entries is less than 16 or just 16, these three entries could be written into a cacheline simultaneously. Based the data shown in Fig. 8, it is easy to see that just serially writing data entry like in STT-MRAM cache is the lowest-efficiency policy. Even adopting the 'FWI' to bundle two entries into one write, there are still quite some $\geq$3 entries cases that are not taken advantage of.

Based on the above observation, we introduce a new write policy, Adaptive Buffer Entry (ABE). 'ABE' works as follows (shown in Fig. 9). When a cache miss occurs in L1 cache, a new data is loaded into L1 cache. Consequently, a stale data should be evicted. Before this cache line be pushed

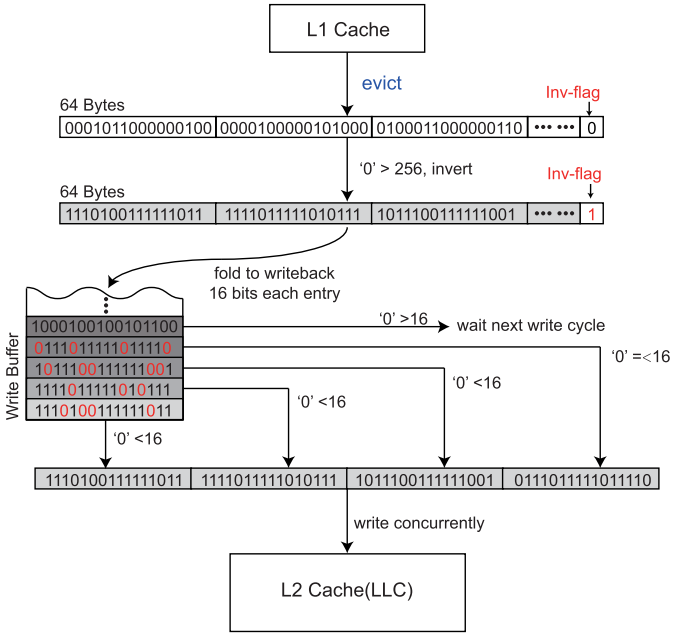


Fig. 9. An example of 'ABE' write policy.

into L2 cache write buffer, check the number of '0' in the entire cacheline (64 Bytes). If it is larger than half of the cacheline bits, we invert this cacheline bit-by-bit and append 1 single bit invert flag to it similar to the use of inv-flag in 'FWI'. Then during the following write process from the write buffer to L2 cache, the system checks each buffer entry and counts the number of '0's. Once it reaches the maximum number of concurrently writable bits (16 bits in this work according to the tapeout case [16]), all the checked entries are stored into LLCs. As shown in Fig. 9, the first 4 entries located at the head of the buffer contain 16 '0's. In this case, all the data in these 4 entries are written concurrently to the LLC. Note that the number of '0's to be concurrently written cannot be larger than 16 due to the limitation of supply current threshold. Meanwhile, all the writable data in a write cycle will belong to the same cacheline in this case.

Refer to the results in Fig. 8, 'ABE' can handle varied '0' distribution patterns within neighboring buffer entries in an adaptive manner and ensure that the total write current is always below the supply current threshold. Meanwhile, the capacity overhead is controlled within 0.2 % (1 bit flag for 512 bits data). For the buffer part, only 1 bit tag is needed for each wordline which incurs just 2.98% buffer resource overhead (shown in Table III). Furthermore, compared with the LLC array, this overhead is negligible. On the other hand, compared with 'FWI' policy, although the number of writing '0's cannot be strictly controlled under 50 % in each buffer entry, the enhanced parallelism at the entry level will offset this point. Note that for 'FWI' and 'ABE', considering the read operation scenario, non-block read/write buffers are used, which could handle the read/write buffering operation separately. Meanwhile, in this work, the inversion process depends on the output control of the sense amplifier [10]. In the sensing process, a pair of opposite binary states could be sensed. And the invert operation could be realized by just controlling the corresponding output which depends on the invert flags. With this methodology, the inversion performance overhead could be hidden into the normal read process.

TABLE IV
THE CMP ARCHITECTURE CONFIGURATIONS

Processor	8-core @ 3.3 GHz, out-of-order, alpha
L1-Cache	I-cache 128KByte 8-way set associative D-cache 128KByte 8-way set associative
L2-Cache	16 MByte, 8-way set associative, shared 16 bits width, 8 cacheline entries write buffer MOESI cache coherence protocol
Main Memory	4 GByte DDR3 DRAM
Benchmark	Spec 2000&2006, Parsec

V. EXPERIMENTAL RESULTS

To validate the effectiveness of the proposed 'ABE' design in terms of write energy and performance, we perform extensive simulations as described below.

A. Experiment Setup

The CMP architecture used in our simulation consists of 8 Alpha 21264 cores. Each core has 128 KB private instruction and data cache respectively. L2 cache is shared by all cores. L2 cache read/write separated non-block buffer width is fixed to 16 bits which can hold 8 cacheline entries. The writeback operations from L1 to L2 are all handled by write buffer which will not interferes with the buffering process of reading. The detailed architecture setup for simulations is tabulated in Table IV. We extended the GEM5 simulator [26] to model the 'ABE' policy. The conventional STT-MRAM without any optimization is used as the baseline in our simulations. The 'EWT' policy-optimized STT-MRAM, standard SOT-MRAM, 'EWT' policy-optimized SOT-MRAM and 'FWI' policy-optimized NAND-SPIN are adopted as comparison groups in our experiments.

The cell read/write energy and latency of NAND-SPIN were obtained from the HSPICE simulations on the 40 nm NAND-SPIN technology using the model developed in [15]. The MTJ parameters used for the simulation are shown in Table I. Then, circuit level simulation results, listed in Table. II, were fed into NVSim [21] to obtain the write energy and latency of corresponding L2 cache. We assumed that L2 cache and the bank capacity are respectively 16MB and 256KB. Note that, the calculation of the write latency (T_{all}) in 'EWT' STT-MRAM (or SOT-MRAM) cache architecture is different from that in standard STT-MRAM (or SOT-MRAM) and is given below.

$$T_{all} = T_{write-access} + T_{inner-read} + T_{comparison} \quad (2)$$

where $T_{write-access}$ is the normal write access latency of STT-MRAM. $T_{inner-read}$ is the read process latency consisting of the bit-line active latency and the sense amplifier latency. $T_{comparison}$ is the latency of comparison process between the coming data and the read-out original data. The latter two portions in Eq. 2 were obtained from [25] where the same policy has been simulated under the same technology node.

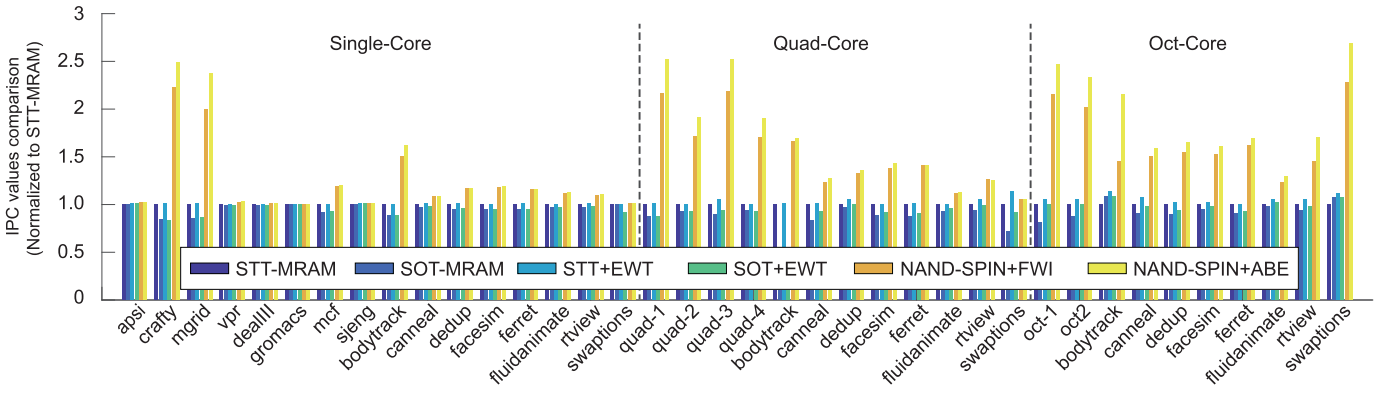


Fig. 10. IPC comparisons of six write policies.

TABLE V

THE BENCHMARK QUAD-TASK AND OCT-TASK GROUPS CONSTRUCTED FROM THE SPEC2000 & SPEC2006 BENCHMARK SUITE

Group	Benchmark
Quad-1	art110, gzip, vpr, bzip2
Quad-2	gap, swim, twolf, crafty
Quad-3	mgrid, mesa, ammp, vortex1
Quad-4	hmmer, cactusADM, sjeng, gromacs
Quad-5	mcf, hmmer, gromacs, dealII
Quad-6	sjeng, mcf, cactusADM, gromacs
Oct-1	vortex, gzip, apsi, art, mgrid, vpr, swim, gcc
Oct-2	hmmer, libquantum, cactusADM, gromacs dealII, sjeng-mcf, bzip2

The experiments were performed on the single-core/quad-core/eight-core architecture respectively. The benchmark suites used in simulations include SPEC bencmark [27] as multi-program applications and PARSEC [28] as multi-thread applications. We constructed 10 single task cases for single core simulations, 6 combinations of SPEC benchmarks for quad-core multi-program simulations and 2 combinations of SPEC benchmarks eight-core multi-program simulations, which are tabulated in Table V. We ran one instance of these workloads per core to simulate the multi-programing case. The PARSEC benchmarks were used for single-core, quad-core and eight-core multi-thread simulations. In our simulations, 200 million instructions were fast-forwarded to warm up the cache and then 30 million instructions were executed to generate the simulation statistics. L2-cache access statistics obtained from gem5 were used to estimate the overall write energy consumption induced by normal write accesses.

B. Performance Analysis

IPC performance comparisons for the single core case is plotted in Fig. 10(a), and those for quad-core and eight-core cases are plotted in Fig. 10(b). The results are normalized to the baseline, i.e. STT-MRAM cache without any optimization. Single task denotes the case that we only run one benchmark on the multi-core processor. As shown in Fig. 10(a), left-most bars represent the baseline IPC values of single-core simulation results. The STT-MRAM with 'EWT' results are also shown in these figures. Among the three write policy designs, 'ABE' works the best, and can improve performance by 15% on average. For "bodytrack" group, 'FWI' scheme can improve 58% performance compared to the baseline while

TABLE VI

THE NUMBERS OF CONCURRENTLY WRITING ENTRIES IN 'ABE' POLICY

Group	Numbers of entries could be written in parallel					
	1	2	4	8	16	32
Quad-2	310428	661608	217397	87626	78909	248208
Quad-3	400694	892765	420024	38822	13740	267547
Qct-1	227292	500372	36029	22363	8921	52283
Swaptions	222716	342730	20548	12846	5285	15146

'ABE' can obtain 69% improvement at most. Considering the multi-core simulations, IPC comparisons when running multi-program and multi-thread applications are shown in Fig. 10(b), the IPC improvement of 'FWI' is 55% on average, and 'ABE' can further improve performance by 70.2% on average compared to the baseline. These results are attributed to the fact that, for single-core simulations, 16MB LLC cannot be fully utilized as a result of the small dataset size of a single application. Note that, the buffer searching overhead has already been taken into consideration by adding a read cycle period for each additional entry which can be written parallelly. However, compared with the long term write delay which can be reduced by the parallelism of our policy, the additional read cycle can only slightly affect system performance, even with this overhead. This can explain why 'FWI' shows better performance than 'ABE' in some results (e.g. 'retrview'). Otherwise, the capacity reduction of 'FWI' is also taken into account by setting the inversion flag bits in cache matrix as invalid bits which are treated as cache miss accesses in Gem5 simulation framework. In this term, the performance overhead of this capacity reduction could be taken into consideration in our evaluations.

The reason for the additional improvement of 'ABE' over 'FWI' is the enhanced parallelism. As shown in Table VI, groups with significant performance improvements in multi-core simulations are picked out. In the table, the number of entries which could be written in parallel is stretched across 2 to 64. Compared to the 'FWI' policy where only 2 entries are written, in 'ABE' policy most of entries can be written in higher parallelism, which further improves the performance of NAND-SPIN LLCs.

C. Energy Analysis

Write energy comparisons of three write policies are presented in Fig. 11. Fig. 11(a) illustrates write energy

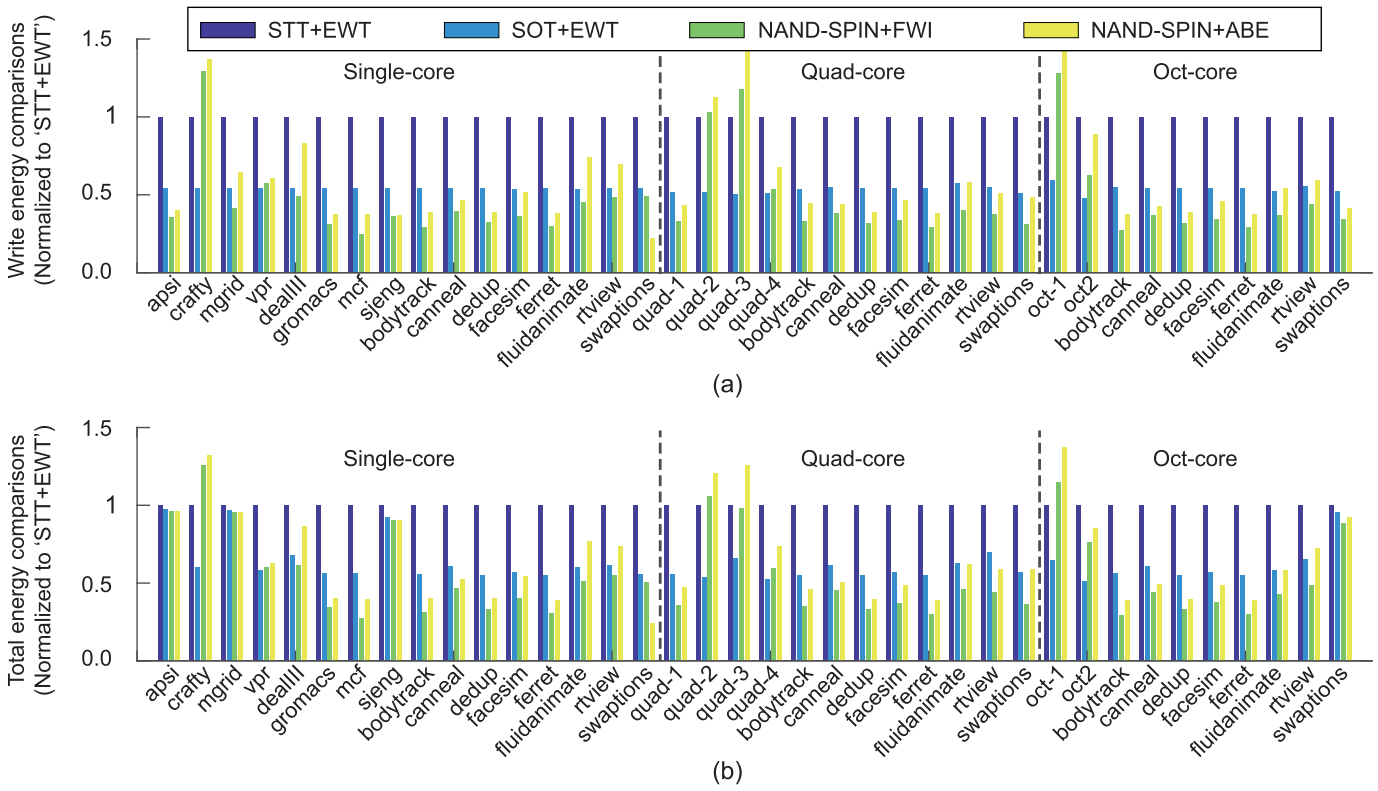


Fig. 11. Energy comparisons of four write policies: (a) The write energy comparisons; (b) The overall energy comparisons.

comparisons, including single-core, quad-core and eight-core cases. The total energy consumption comparisons, considering both read and write energy, are shown in Fig. 11(b). All results are normalized to the 'STT+EWT' policy which has already been considered as an optimal policy for STT-MRAM cache energy saving. As shown in the figures, 'FWI' works the best in terms of write energy and overall energy consumptions. As for write energy, 'ABE' can save 33% on average, and 58% at most ("sjeng" case) compared with the 'EWT'. Meanwhile, 'FWI' achieves about 48% write energy reduction on average. Even compared with SOT-MRAM, our policies optimized NAND-SPIN cache also can achieve a certain level of energy reduction. Considering the overall energy consumption as shown in Fig. 11(b), 'ABE' also works well. For most of the energy results, the energy consumptions of 'FWI' is lower than those of 'ABE'. Note that, in some cases the 'FWI' could show better energy efficiency. The reason is that the inversion method in 'FWI' could reduce the total write bits more precisely with smaller granularity. Meanwhile, in some results like quad-2 and 3, both 'FWI' and 'ABE' exceeds other 'EWT' groups. This is because for a small number of high repetition rate data patterns, the 'EWT' strategy could minimize the number of redundant writings. Nevertheless, the IPC results of 'ABE' always show a considerable improvement compared to 'FWI' policy.

D. Concurrently Write Reliability Analysis of NAND-SPIN

Now we discuss the write operation of NAND-SPIN at the device level. Generally the write operation is affected by two

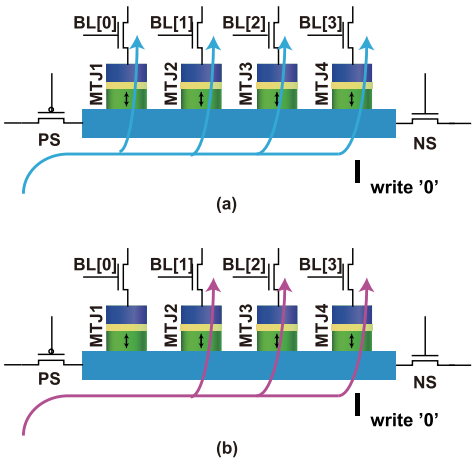


Fig. 12. Examples of concurrent writing operation in NAND-SPIN:(a) all-switched write operation; (b)partly-switched write operation.

issues: first, as shown in Fig. 12(a), while writing all the MTJs with STT effect, the supply current must switch all the MTJs with a reliable yield [29], [30]. Second, as shown in Fig. 12(b), while writing the MTJs at the tails of the strip with STT effect, the current flowing through the strip must not disturb [31] the other idle MTJs (called write disturbance).

To investigate the write disturbance and read reliability issues, we perform circuit simulations with a commercial CMOS 40-nm design kit (with process variations 3σ [32]). Monte Carlo statistical simulations (1000 runs) were performed to evaluate the robustness of the write operation.

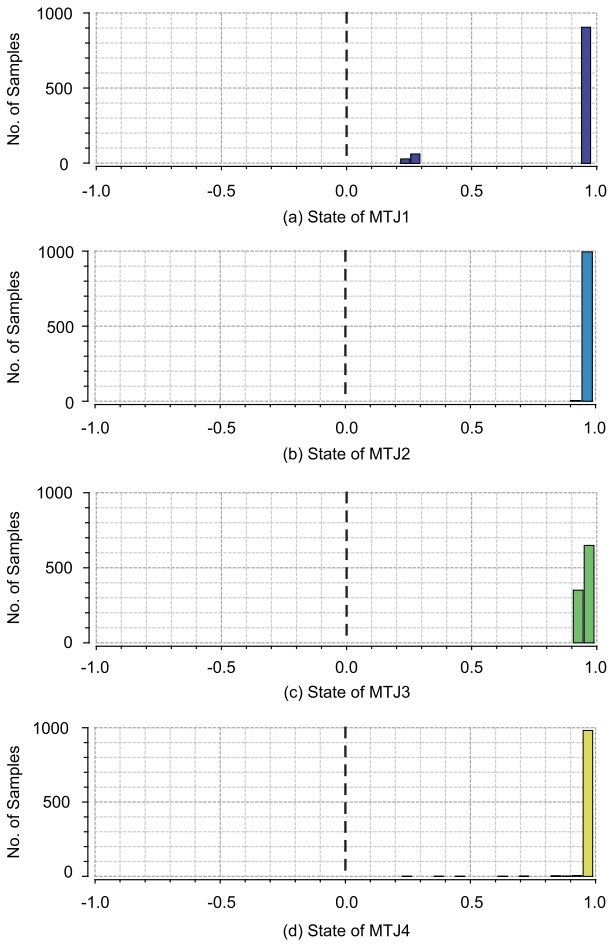


Fig. 13. Monte Carlo simulation results of writing 4 MTJs on the same heavy metal strip in parallel.

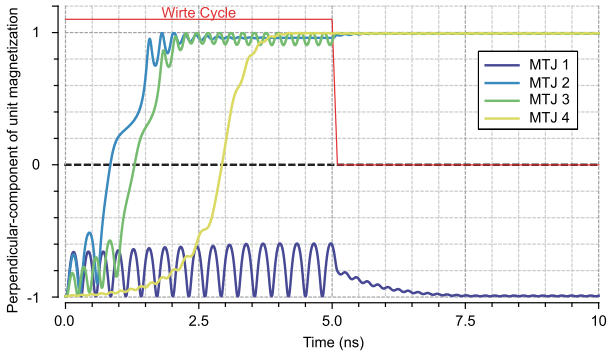


Fig. 14. An example of writing latter 3 MTJs on the same heavy metal strip in parallel. We set different heavy-metal widths for different bit-cells in order to enhance the robustness against the write disturbance. For instance, for the left-most MTJ, the heavy-metal width is intentionally increased to weaken the disturbance current density.

First, we consider the case shown in Fig. 12(a) where all 4 MTJs are concurrently written by the STT. The results of 1000 runs Monte-Carlo simulations are shown in Fig. 13, where the duration of applied STT current is 3 ns. In the figure, the x-axis represents the final perpendicular-component magnetization of the free layer in the MTJ (positive and negative values mean data '0' and '1', respectively). Y-axis indicates the number of samples in 1000 runs. As can be

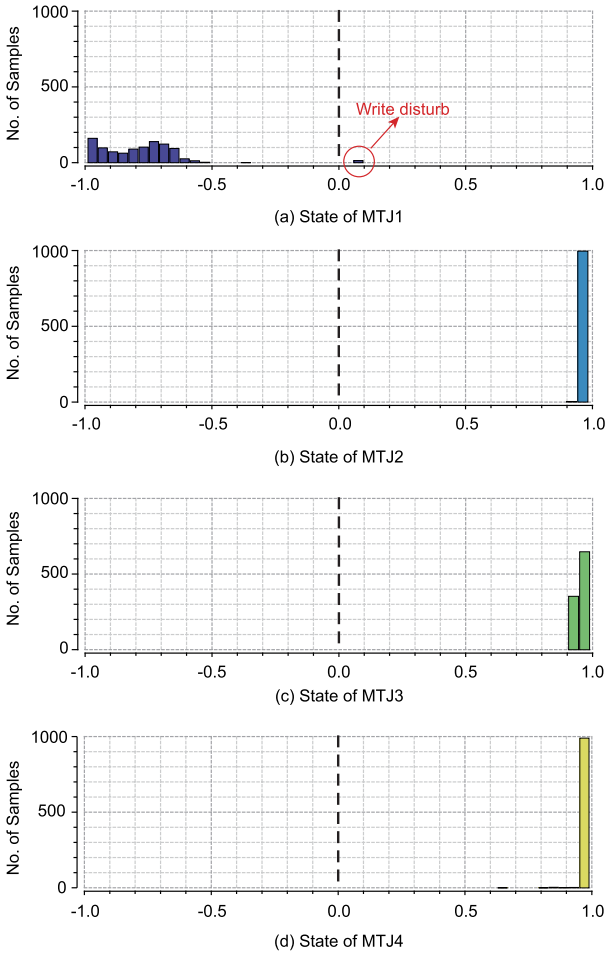


Fig. 15. Monte Carlo simulation results of write disturbance on the left-most MTJ.

seen, even with the process variation, all the 4 MTJs could be written successfully.

Second, we consider the case shown in Fig. 12(b), where the right three MTJs are concurrently written by the STT. In this case, the STT currents passing the strip will disturb the state of the left-most MTJ through the SOT effect. Note that this figure shows the worst case of write disturbance. The transient simulation results shown in Fig. 14 indicates the time-dependent magnetization switching in this worst-case scenario. Obviously, the magnetization precession occurs on the left-most MTJ during the write operation, which reflects the write disturbance. Nevertheless, the state of the left-most MTJ remains unchanged. Moreover, the Monte-Carlo simulation results shown in Fig. 15 indicate that, in most of cases the influence of write disturbance is negligible. Only 1.4% of samples are unintentionally switched. With the help of Error Collecting Code (ECC) [33], [34], the reliability of entire cache system is acceptable.

Additionally, if the three MTJs on the right side need to maintain their original states while writing the leftmost MTJ to '0', there will be no any interference existing in the NAND-SPIN. Because in this case, only the leftmost MTJ needs to be written '0' with the STT mechanism. Thus the current only flows through the leftmost MTJ and underneath heavy metal.

No any current passes the heavy metal below the three MTJs on the right side, hence no any interference.

VI. Related Work

With technology node continuously shrinking, leakage power occupies a large portion of chip power consumption. This problem is much more severe for large capacity on-chip cache. To settle these standby power overheads, non-volatile technologies start to get on the stage, such as PCRAM, ReRAM and MRAM. Among them, MRAM has experienced a series of evolutions. From in-plane STT-MRAM [35] to perpendicular magnetic anisotropy STT-MRAM [36], [37] and the following emerging SOT-MRAM [38], [39], VCMA STT-MRAM [40], Skyrmion-based MRAM [41], MRAM devices always go straightly towards lower write energy and faster access speed [42].

Beyond the device level optimizations, many efforts on architecture level also target on the high write cost issue of STT-MRAM. Zhou et al. proposed a write early termination policy to reduce write energy of STT-MRAM [25] which mainly aimed at reducing redundant data writing. The experimental results indicated that the total write energy can be reduced by more than 33%. Sun et al. proposed a multi-retention level STT-MRAM cache design and associated adaptive data refresh policy to reduce the energy overhead and improve cache access performance [43]. Chi *et al.* explored the data encoding scheme for STT-MRAM [44]. By mapping frequent occurring data patterns to the energy efficient resistive states, the write energy of STT-MRAM can be reduced dramatically [45]. Bi *et al.* proposed a compression based multi-level cache [46]. By enabling the hard-bit region to store another compressed cacheline, the system performance for memory intensive workloads can be improved significantly. Min *et al.* also STT-MRAM/DRAM hybrid buffer to extend the lifetime of object-based NAND flash device [47]. Wu *et al.* exploited the data redundancy within the write back data from upper level cache to STT-MRAM LLCs and proposed a compare-and-write technique to eliminate the redundant write back data for write energy optimization [48].

Different from the above related works, this work exploits a device-architecture co-design write policy for the state-of-the-art erasable NAND-SPIN large capacity cache which enlarging the bandwidth of LLCs.

VII. Conclusion

As the modern processor enters into the multi-core and many-core era, cache capacity increases rapidly for the cache performance improvement. To mitigate the leakage power, spintronic-based LLC is promising to replace the conventional SRAM cache. In this work, we take advantage of the emerging NAND-SPIN device to reduce the write energy consumption, and improve cache access performance of LLCs. To improve cache write back throughput, we propose an 'ABE' write policy which could adaptively extend the write data length to be written back to the fixed-width cache write buffer. Compared to the STT-MRAM cache design and a non-adaptive 'FWI' write policy, our proposed 'ABE' method can achieve 70 % performance improvements over the baseline. Meanwhile, in comparison with the conventional early write terminate (EWT) policy, our 'ABE' could save 33% write energy with negligible hardware overhead. Our fulture work will focus on ameliorating the software level design to make it aware of the characteristics of underlying NAND-SPIN, as well as some OS level optimizations like instruction reordering and more.

Acknowledgment

This author acknowledges the support of the Asian Research Grant by the University of Notre Dame.

References

[1] N. S. Kim *et al.*, "Leakage current: Moore's law meets static power," *Computer*, vol. 36, no. 12, pp. 68–75, Dec. 2003.
[2] M. Jagtap, "Era of multi-core processors," *Power*, vol. 2, no. 2, p. 2, 2009.
[3] M. B. Taylor, "A landscape of the new dark silicon design regime," *IEEE Micro*, vol. 33, no. 5, pp. 8–19, Sep./Oct. 2013.
[4] C. Chappert, A. Fert, and F. N. Van Dau, "The emergence of spin electronics in data storage," *Nature Mater.*, vol. 6, no. 11, pp. 813–823, Nov. 2007.
[5] S. Raoux *et al.*, "Phase-change random access memory: A scalable technology," *IBM J. Res. Develop.*, vol. 52, nos. 4–5, pp. 465–479, 2008.
[6] S. Gaba, P. Knag, Z. Zhang, and W. Lu, "Memristive devices for stochastic computing," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Melbourne VIC, Australia, Jun. 2014, pp. 2592–2595.
[7] M. Mao, H. Li, A. Jones, and Y. Chen, "Coordinating prefetching and STT-RAM based last-level cache management for multicore systems," in *Proc. ACM Int. Conf. Great Lakes Symp. VLSI*, New York, NY, USA, May 2013, pp. 55–60.
[8] A. Qoutb and E. Friedman, "MTJ magnetization switching mechanisms for IoT applications," in *Proc. ACM Great Lakes Symp. VLSI (GLSVLSI)*, New York, NY, USA, May 2018, pp. 347–352.
[9] W. Kang *et al.*, "An overview of spin-based integrated circuits," in *Proc. 19th Asia South Pacific Design Autom. Conf. (ASP-DAC)*, Singapore, Jan. 2014, pp. 676–683.
[10] B. Wu, Y. Cheng, J. Yang, A. Todri-Sanial, and W. Zhao, "Temperature impact analysis and access reliability enhancement for 1T1MTJ STT-RAM," *IEEE Trans. Rel.*, vol. 65, no. 4, pp. 1755–1768, Dec. 2016.
[11] L. Xue *et al.*, "An adaptive 3T-3MTJ memory cell design for STT-MRAM-based LLCs," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 26, no. 3, pp. 484–495, Mar. 2018.
[12] Z. Wang, W. Zhao, E. Deng, J.-O. Klein, and C. Chappert, "Perpendicular-anisotropy magnetic tunnel junction switched by spin-Hall-assisted spin-transfer torque," *J. Phys. D, Appl. Phys.*, vol. 48, no. 6, pp. 065001-1–065001-7, Jan. 2015.
[13] Z. Wang *et al.*, "Progresses and challenges of spin orbit torque driven magnetization switching and application (Invited)," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Florence, Italy, May 2018, pp. 1–5.
[14] A. Sheikholeslami, "Source degeneration [circuit intuitions]," *IEEE Solid-State Circuits Mag.*, vol. 6, no. 3, pp. 5–6, Aug. 2014.
[15] Z. Wang *et al.*, "High-density NAND-like spin transfer torque memory with spin orbit torque erase operation," *IEEE Electron Device Lett.*, vol. 39, no. 3, pp. 343–346, Mar. 2018.
[16] Q. Dong *et al.*, "A 1 Mb 28 nm STT-MRAM with 2.8 ns read access time at 1.2 V VDD using single-cap offset-cancelled sense amplifier and *in-situ* self-write-termination," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, San Francisco, CA, USA, Feb. 2018, pp. 480–482.
[17] A. Brataas and K. Hals, "Spin-orbit torques in action," *Nature Nanotechnol.*, vol. 9, no. 2, p. 86, 2014.
[18] M. Cubukcu *et al.*, "Spin-orbit torque magnetization switching of a three-terminal perpendicular magnetic tunnel junction," *Apply Phys. Lett.*, vol. 48, no. 6, p. 065001, 2015.
[19] Y. Seo, K.-W. Kwon, and K. Roy, "Area-efficient SOT-MRAM with a Schottky diode," *IEEE Electron Device Lett.*, vol. 37, no. 8, pp. 982–985, Aug. 2016.
[20] M. d'Abreu, "NAND Flash memory: The driving technology in digital storage—Overview and challenges," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI (ISVLSI)*, Natal, Brazil, Aug. 2013, p. 1.

[21] X. Dong, C. Xu, Y. Xie, and N. P. Jouppi, "NVSim: A circuit-level performance, energy, and area model for emerging nonvolatile memory," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 31, no. 7, pp. 994–1007, Jul. 2012.

[22] F. Oboril, R. Bishnoi, M. Ebrahimi, and M. Tahoori, "Evaluation of hybrid memory technologies using SOT-MRAM for on-chip cache hierarchy," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 34, no. 3, pp. 367–380, Mar. 2015.

[23] P. P. Chu and R. Gottipati, "Write buffer design for on-chip cache," in *Proc. IEEE Int. Conf. Comput. Design, VLSI Comput. Process.*, Cambridge, MA, USA, Oct. 1994, pp. 311–316.

[24] Cadence. *Spectre Circuit Simulator*. Accessed: Jan. 2004. [Online]. Available: http://www.cadence.com

[25] P. Zhou, B. Zhao, J. Yang, and Y. Zhang, "Energy reduction for STT-RAM using early write termination," in *IEEE/ACM Int. Conf. Comput.-Aided Design-Dig. Tech. Papers*, San Jose, CA, USA, Nov. 2009, pp. 264–268.

[26] N. Binkert *et al.*, "The gem5 simulator," *ACM SIGARCH Comput. Archit. News*, vol. 39, no. 2, pp. 1–7, 2011.

[27] Spec Benchmark. *Standard Performance Evaluation Corporation*. Accessed: Feb. 2007. [Online]. Available: https://www.spec.org/

[28] C. Bienia and K. Li, *Benchmarking Modern Multiprocessors*. Princeton, NJ, USA: Princeton Univ. Princeton, 2011.

[29] E. Cheshmikhani, H. Farbeh, S. Miremadi, and H. Asadi, "TA-LRW: A replacement policy for error rate reduction in STT-MRAM caches," *IEEE Trans. Comput.*, vol. 68, no. 3, pp. 455–470, Mar. 2019.

[30] Y. Zhang, X. Wang, Y. Li, A. K. Jones, and Y. Chen, "Asymmetry of MTJ switching and its implication to STT-RAM designs," in *Proc. IEEE Autom. Test Eur. Conf. Exhib. (DATE)*, Dresden, Germany, Mar. 2012, pp. 1313–1318.

[31] R. Wang, L. Jiang, Y. Zhang, L. Wang, and J. Yang, "Selective restore: An energy efficient read disturbance mitigation scheme for future STT-MRAM," in *Proc. ACM/EDAC/IEEE Design Automat. Conf. (DAC)*, San Francisco, CA, USA, Jun. 2015, pp. 1–6.

[32] *Manual Design Kit for CMOS 40 nm*, STMicroelectron., Geneva, Switzerland, 2012.

[33] R. W. Hamming, "Error detecting and error correcting codes," *Bell Syst. Tech. J.*, vol. 29, no. 2, pp. 147–160, 1950.

[34] Z. Pajouhi, X. Fong, A. Raghunathan, and K. Roy, "Yield, area, and energy optimization in STT-MRAMs using failure-aware ECC," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 13, no. 2, p. 20, 2017.

[35] S. H. Kang and K. Lee, "Emerging materials and devices in spintronic integrated circuits for energy-smart mobile computing and connectivity," *Acta Mater.*, vol. 61, no. 3, pp. 952–973, 2013.

[36] S. Ikeda *et al.*, "A perpendicular-anisotropy CoFeB–MgO magnetic tunnel junction," *Nature Mater.*, vol. 9, pp. 721–724, Jul. 2010.

[37] M. Wang *et al.*, "Current-induced magnetization switching in atom-thick tungsten engineered perpendicular magnetic tunnel junctions with large tunnel magnetoresistance," *Nature Commun.*, vol. 9, no. 671, pp. 1–7, 2018.

[38] L. Liu, C.-F. Pai, Y. Li, H. W. Tseng, D. C. Ralph, and R. A. Buhrman, "Spin-torque switching with the giant spin Hall effect of tantalum," *Science*, vol. 336, no. 6081, pp. 555–558, May 2012.

[39] I. M. Miron *et al.*, "Perpendicular switching of a single ferromagnetic layer induced by in-plane current injection," *Nature*, vol. 476, no. 7359, pp. 189–193, Aug. 2011.

[40] W. Kang, L. Chang, Y. Zhang, and W. Zhao, "Voltage-controlled MRAM for working memory: Perspectives and challenges," in *Proc. IEEE Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Lausanne, Switzerland, Mar. 2017, pp. 542–547.

[41] W. Kang, Y. Huang, X. Zhang, Y. Zhou, and W. Zhao, "Skyrmion-electronics: An overview and outlook," *Proc. IEEE*, vol. 104, no. 10, pp. 2040–2061, Oct. 2016.

[42] R. Patel, E. Ipek, and E. Friedman, "2T–1R STT-MRAM memory cells for enhanced on/off current ratio," *Microelectron. J.*, vol. 45, no. 2, pp. 133–143, 2014.

[43] Z. Sun *et al.*, "Multi retention level STT-RAM cache designs with a dynamic refresh scheme," in *Proc. IEEE/ACM Int. Symp. Microarchitecture (MICRO)*, Porto Alegre, Brazil, Dec. 2017, pp. 329–338.

[44] P. Chi, C. Xu, X. Zhu, and Y. Xie, "Building energy-efficient multi-level cell STT-MRAM based cache through dynamic data-resistance encoding," in *Proc. IEEE Int. Symp. Qual. Electron. Design*, Santa Clara, CA, USA, Mar. 2014, pp. 639–644.

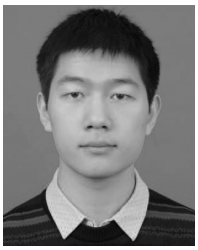
[45] M. Rasquinha, D. Choudhary, S. Chatterjee, S. Mukhopadhyay, and S. Yalamanchili, "An energy efficient cache design using spin torque transfer (STT) RAM," in *Proc. ACM/IEEE Int. Symp. Low Power Electron. Design (ISLPES)*, Lausanne, Switzerland, Jul. 2010, pp. 389–394.

[46] X. Bi, M. Mao, D. Wang, and H. H. Li, "Cross-layer optimization for multilevel cell STT-RAM caches," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 6, pp. 1807–1820, Jun. 2017.

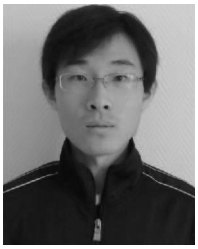
[47] C. Min, J. Guo, H. H. Li, and Y. Chen, "Extending the lifetime of object-based NAND flash device with STT-RAM/DRAM hybrid buffer," in *Proc. IEEE Asia South Pacific Design Autom. Conf. (ASP-DAC)*, Jan. 2017, pp. 764–769.

[48] B. Wu *et al.*, "Write energy optimization for STT-MRAM cache with data pattern characterization," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI (ISVLSI)*, Jul. 2018, pp. 333–338.

Bi Wu (S'15) received the B.S. degree from the China University of Mining and Technology, Xuzhou, China, and the M.S. degree from Beihang University, Beijing, China, where he is currently pursuing the Ph.D. degree in electrical engineering. His research interests include circuit level and architecture level design and optimization of STT-MRAM, SOT-MRAM, and the corresponding reliability analysis and improvement, and so on. In 2017, he received the China National Scholarship for Ph.D. by the Ministry of Education of China.

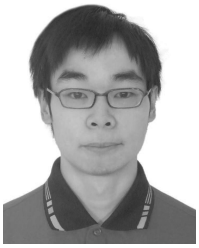


Pengcheng Dai received the B.S. degree from Beihang University, Beijing, China, where he is currently pursuing the M.S. degree in electrical engineering. His research interests include the usage of MRAM in computer architecture, architecture of embedded devices, and so on.



Zhaohao Wang (S'12–M'16) received the B.S. degree in microelectronics from Tianjin University in 2009, the M.S. degree from Beihang University, China, in 2012, and the Ph.D. degree in physics from the University of Paris-Saclay, France, in 2015. His current research interests include the modeling of non-volatile nano-devices and design of new non-volatile memories and logic circuits.

Chao Wang received the B.S. degree from Beihang University, Beijing, China, where he is currently pursuing the Ph.D. degree in electrical engineering. His research interests include the modeling of non-volatile nano-devices and design of new non-volatile memories and logic circuits.



Ying Wang (M'14) received the B.S. and M.S. degrees in electrical engineering from the Harbin Institute of Technology in 2007 and 2009, respectively, and the Ph.D. degree in computer science from the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), Beijing, in 2014. He is currently an Assistant Professor with ICT, CAS. His research interests include computer architecture and VLSI design, specifically memory systems, on-chip interconnects, resilient and energy-efficient architecture, and machine learning accelerators.

Jianlei Yang (S'12) received the B.S. degree in microelectronics from Xidian University, Xi'an, China, in 2009, and the Ph.D. degree in computer science and technology from Tsinghua University, Beijing, China, in 2014. From 2013 to 2014, he was a Research Intern with Intel Labs China, Intel Corporation. He is currently an Associate Professor with the School Computer Science and Engineering, Beihang University, Beijing. His current research interests include numerical algorithms for VLSI power grid analysis and verification, spintronics, and neuromorphic computing. He was a recipient of the first place on TAU Power Grid Simulation Contest in 2011, the second place on TAU Power Grid Transient Simulation Contest in 2012, the IEEE ICCD Best Paper Award in 2013, and the ACM GLSVLSI Best Paper Nomination in 2015.

Yuanqing Cheng (S'11–M'13) received the Ph.D. degree from the Key Laboratory of Computer System and Architecture, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. After spending one year post-doctoral study at LIRMM, CNRS, France, he joined Beihang University, China, as an Assistant Professor. His research interests include reliable design for 3D integrated circuits, as well as computing architecture design for emerging technologies, including spintronic and carbon nanotube devices. He is an ACM member.

Dijun Liu received the B.S. and M.S. degrees in electrical engineering from the China University of Petroleum, Huadong, in 1993 and 1998, respectively, and the Ph.D. degree in geophysics from the China University of Petroleum, Beijing, China, in 2001.

He is currently a Chief Scientist with the China Academy of Telecommunications Technology (CATT), Beijing, and a Professor with the School of Electronic and Information Engineering, Beihang University. He is also the Director of the China Institute of Communications (CIC) and the Chairman of the China Communications Integrated Circuit Committee (CCIC). His main research interests include high-performance SoC design and software-defined radio communication.

Youguang Zhang (M'13) was born in Jinhua, China, in 1963. He received the M.S. degree in mathematics from Peking University, Beijing, China, in 1987, and the Ph.D. degree in communication and electronic systems from Beihang University, Beijing, in 1990. He is currently a Professor with the School of Electronic and Information Engineering, Beihang University. His research interests include microelectronics and wireless communication. In particular, he recently focuses on the circuit and system co-design for the emerging memory and computing systems.

Weisheng Zhao (M'06–SM'14–F'19) received the Ph.D. degree in physics from the University of Paris-Sud, France, in 2007. From 2004 to 2008, he investigated Spintronic devices-based logic circuits and designed a prototype for hybrid Spintronic/CMOS (90 nm) chip in cooperation with STMicroelectronics. Since 2009, he has been a Tenured Research Scientist with CNRS. He has authored or coauthored more than 150 scientific articles, including *Advanced Material*, *Nature Communications*, and IEEE TRANSACTIONS. His interests includes the hybrid integration of nano-devices with CMOS circuit and new non-volatile memory (40 nm technology node and below) like MRAM circuit and architecture design. He is also the principal inventor of four international patents. Since 2014, he becomes a youth 1000 Plan Distinguished Professor with Beihang University, Beijing, China. He is also the Associated Editor of the IEEE TRANSACTIONS ON NANOTECHNOLOGY.

Xiaobo Sharon Hu (S'85–M'89–SM'02–F'16) received the B.S. degree from Tianjin University, China, the M.S. degree from the Polytechnic Institute of New York, and the Ph.D. degree from Purdue University, West Lafayette, IN, USA. She is currently a Professor with the Department of Computer Science and Engineering, University of Notre Dame. Her research interests include computing with beyond-CMOS technologies, low-power system design and cyber-physical systems. She has published more than 300 peer-reviewed articles in these areas. She received the NSF CAREER Award in 1997 and the Best Paper Award from the Design Automation Conference in 2001 and ACM/IEEE International Symposium on Low Power Electronics and Design in 2018. She was the General Chair of the 2018 Design Automation Conference (DAC), and the Program Chair and the TPC Chair of 2016 and 2015 DAC. She served as an Associate Editor for the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION, *ACM Transactions on Design Automation of Electronic Systems*, and *ACM Transactions on Embedded Computing*. She is also an Associate Editor of *ACM Transactions on Cyber-Physical Systems*.