A Low-Power Hybrid Magnetic Cache Architecture Exploiting Narrow-Width Values

Mohsen Imani†, Abbas Rahimi‡, Yeseong Kim†, Tajana Rosing†

†Computer Science and Engineering, UC San Diego, La Jolla, CA 92093, USA

‡Electrical Engineering and Computer Sciences, UC Berkeley, Berkeley, CA 94720, USA

{moimani, yek048, tajana}@ucsd.edu, abbas@eecs.berkeley.edu

***Abstract*— Modern microprocessors have increased the word width to 64-bits to support larger main memory sizes. It has been observed that data can often be represented by relatively few bits, so-called *narrow-width values*. To leverage narrow-width data, we propose a hybrid cache architecture composed of magnetic RAM (MRAM) and SRAM to save the upper and lower 32-bits of each word in MRAM and SRAM respectively. To address write performance issue of MRAM, we propose an optimal dynamic write buffer (DWB) allocation mechanism. To enhance efficacy of our hybrid cache in the absence of narrow-width values, we propose a double row write (DRW) technique that adaptively partitions non-narrow data to two 32-bit pieces for consecutive row writes in the SRAM part. DWB and DRW jointly guarantee the performance of the proposed hybrid cache and balance a tradeoff between the buffer size and the number of double row writes. Our evaluation on SPEC CPU2000, SPEC CPU2006 and Mibench benchmarks shows that our hybrid cache can achieve up to 46% power and 24% area savings at the same performance as the conventional SRAM cache.**

Keywords—Hybrid cache, Non-volatile memory, Spin-transfer torque memory, Narrow-width values

I. Introduction

Over last few decades, scaling of conventional CMOS technology has been motivated by the need of higher integration density and performance. Static power consumption is a major concern in designing nano-scaled integrated circuits, due to the exponential dependence of subthreshold current on the threshold voltage. The embedded SRAM comprises a dominant portion of chip area as well as the power consumption in modern microprocessors [1]. Emerging non-volatile memories (NVMs) are promising candidates to replace conventional SRAMs due to their low leakage power, high density, and comparable read latency/energy [2].

In modern microprocessors, data can often be represented by a fewer number of bits than the full word width. Such data composition is called narrow-width values [3]. For example, in 64-bit processors, the majority of data uses only the first 32 bits of data for data representation. This narrow-width feature has been widely observed in several CPU benchmarks such as *SPEC CPU2006, SPEC CPU2000*, *Mibench* [4], as well as GPU benchmarks including *Rodinia* [5], and *Parboil* [6]. Low data activity on the upper 32-bit provides an opportunity to replace high leakage SRAMs with more efficient memory cells to improve the power efficiency of the cache. Low leakage power and high density of NVMs make these devices an appropriate candidate for this replacement. However, these memories suffer from high write energy and long write latency. Hence, careful design and placement of such NVM modules are needed to avoid energy and performance penalties in data-intensive caches and register files. Spin-transfer torque RAMs (STT-RAM) and spin-orbit torque RAM (SOT-RAM) are two types of high endurance and fast NVMs based on the magnetic tunneling junction devices (MTJ). Various optimization techniques have been proposed to mitigate the cost of write operations on NVMs. These techniques are mainly focused on either reducing write latency, or improving dynamic energy [7-9].

In this paper, we exploit the narrow-width values to design an efficient hybrid cache architecture using NVM parts to reduce the cache energy consumption and area without a performance penalty. The key contributions of this paper are as follows:

- We first design a hybrid cache architecture that symmetrically locates the narrow-width data in MRAM and SRAM parts without any data migration and performance overhead. Our evaluation on several CPU benchmarks shows there is a significant proportion of the narrow-width data on L2 cache and last level cache (LLC). Hence, we use the NVM as an alternative memory to decrease the leakage power of the upper 32-bits of L2 cache. Our hybrid cache architecture saves the upper and lower 32-bits of every word lines in MRAM and SRAM respectively.
- The proposed hybrid cache is able tolerating the non-narrow data as well as the narrow one with no performance and energy overheads with respect to SRAM cache. To hide the write latency penalty of the MRAM part, we enhance our hybrid cache structure with two mechanisms; i) a double row write (DRW) technique to guarantee the performance of the proposed hybrid cache due to limited write buffer size. ii) A dynamic write buffer (DWB) allocation that balances a tradeoff between buffer size and the number of double row write on the cache.
- We evaluate efficacy of our hybrid cache using two types of NVMs for MRAM part: STT-RAM and SOT-RAM. The simulation results of the hybrid caches in the form of STT-RAM/SRAM and SOT-RAM/SRAM on SPEC CPU2006 benchmarks show up to 43% and 46% (on average 36% and 40%) power saving when using STT-RAM and SOT-RAM for the NVM part. None of these hybrid L2 caches incurs performance overhead.

II. Related Work

Caches are the dominant part of chips and their power optimization can significantly improve total power and thermal properties of microprocessors [10]. NVMs are

known as a promising technology to reduce the static power consumption of on-chip memories. However, NVMs suffer from high write latency, energy consumption and endurance. Several techniques are introduced to address NVM dynamic energy, write latency and endurance[11, 12]. From other side, there are several optimization techniques to improve the power and reliability characteristics of microprocessors for narrow-width values [13-17]. The target of these techniques is soft-error protection or power/performance efficiency of the narrow-width values in register files and caches. In this section we briefly review the related work in area of narrow-width and hybrid cache architecture.

In narrow-width domain, Brooks *et al.* [13] proposed an aggressive clock gating method to turn off partitions of integer arithmetic units that have non-necessary data. They merged the narrow integers and allowed them to share a single function. An asymmetrically ported register file (RF) with different sizes is used to reduce RF power consumption [14]. The thermal behavior of a low power and value-aware register file (VARF) in multicore processor with narrow-width values is studied in [15]. This work uses three thermal-aware control methods: access counter, register-ID and thermal sensor to dynamically control the different partitions in VARF. Another effort [17] evaluates the effect of negative bias temperature instability (NBTI) on narrow-width values on the RF. They showed that in narrow-width values the NBTI degradation on the upper 30-bit entry of cache is destructive. They used power gating on the upper 30-bit entry and implemented a bit flipping technique on the lower 34-bit to balance the percentage of zero and one on register files. For caches, Wang et al. [16] implemented multiple dirty bits and read-before-write optimization techniques on phase change random access memory (PCRAM) to improve the endurance of PCRAM as LLC. Their technique leverages the characteristics of narrow-width data and partially decreases the number of writes on LLC from upper level caches. However, PCRAM as the LLC suffers from the slow write and limited endurance.

Using a hybrid NVM/SRAM cache is an effective technique to decrease the power consumption of SRAM-based caches. A hybrid cache is able to put most of the writes stress on SRAM and uses low leaky NVMs for non-write intensive workloads [18] [19]. To monitor and control the number of write and read operations on NVM and SRAM caches, Wu. *et al*, [20] added two flag bits to tag store and a technique to migrate the data between memories. Smullen, *et al*, [8] proposed a SRAM/STT-RAM hybrid cache which uses relaxed MTJ to decrease data retention of STT-RAM in order to make STT-RAM write operation as fast as SRAM. This cache decreases the non-volatility of STT-RAM with the expense of lower retention time. Using similar idea, Sun, *et al.* [21] split the STT-RAM to two low retention and high retention regions and utilized refreshed scheme (as DRAM) to update the data on low retention region. In this structure the high energy consumption of refresh schemes can be high such that it reduces the advantage of using NVM.

In contrast, our proposed hybrid cache architecture exploits the feature of narrow-width values symmetrically locates the narrow-width data in MRAM and SRAM parts with no data migration and performance overhead. Our design automatically hides the cache write energy over a half of cache wordline. High endurance hybrid cache also improves the cache energy consumption by balancing a tradeoff between buffer size and the number of double row write.

III. MRAM BACKGROUND

Magnetic random access memory (MRAM) is a memory device that is designed based on nano-scale characteristics of electrons. Different types of spin-based NVMs have been proposed such STT-RAM and SOT-RAM. MTJ is the main element of MRAMs that stores data in the form of spin orientation. An MTJ consists of two ferromagnetic layers separated by a barrier oxide layer (e.g., MgO). In the case of parallel (antiparallel) magnetization between the ferromagnetic layers, the resistance of MTJ will be low (high) which represents logic zero (one) [22].

STT-RAM consists of an MTJ and an access transistor (see Figure 1a). This cell has the same path for read and write operations. For both read and write operations, the current passes through source to bitline or vice versa. STT-RAM cell consumes zero leakage power, and it is approximately 4X denser than SRAM. The endurance of STT-RAM is very high due to high MTJ robustness. Several articles report $>10^{15}$ cycles endurance for STT-RAM, making them suitable for replacing SRAM [8]. However, their write energy higher and write latency longer compared to SRAM cells. The leakage power advantage of these memories significantly reduces their total power consumption compared to SRAMs in non-memory intensive applications. However, direct usage of STT-RAM does not provide efficiency for data intensive and write latency critical structures.

SOT-RAMs use an extra terminal to separate read and write operation paths. SOT-RAMs have a separated *word bitline* and *read bitline* (see Figure 1b) for read and write operations respectively. The read operation is similar to STT-RAM devices. Based on MTJ resistance, different currents (high or low) will pass through MTJ and *read bitline*. While in the write mode the current from *write bitline* and access transistor sets/resets the magnetization direction of the perpendicular MTJ [23]. This enables separate optimizations for read and write operations, while the write optimization in STTRAM affects all other cell parameters (e.g., write energy, read latency, etc.).

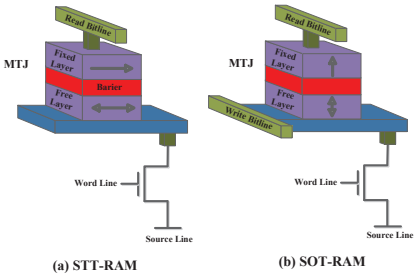

Figure 1. MRAM Standard cells (a) STT-RAM (b) SOT-RAM

IV. PROPOSED CACHE

In narrow-width values the upper 32-bit of data are mostly inactive with very low number of writes. Our evaluations on SPEC CPU2006 benchmarks show that more than 85% of data in L2 cache can be represented by only using first 32-bit and the rest upper half of values contain more than 93% zeros. Figure 2 shows the average percentage of zeros on different index bits for SPEC CPU2006 benchmarks. In this paper, based on this observation we propose a L2 hybrid cache architecture to decrease the cache power consumption and area for system with narrow-width values. To detect narrow-width values, we use existing efficient zero detection logic in execution unit [24]. Our proposed hybrid cache architecture saves the

upper half 32-bit of data on a non-volatile memory module, while the other lower half is stored in a SRAM module. The proposed cache is shown in Figure 3. The non-volatile partition could be implemented with different NVM technologies, depending on the endurance, performance and power requirements. We focus on STT-RAM and SOT-RAM for our NVM part, due to their high endurance, low power and comparable read performance with SRAMs. However, the high write energy and latency of these memories limit their direct usage on caches, especially L1 caches. Indeed, in data intensive workloads with high write request, the write power and write latency of NVMs can be dominant. Hence, our proposed cache locates NVMs in an appropriate L2 cache with a new cache protocol to compensate the high write energy and latency of such NVMs. The advantages of using NVMs in upper half of cache entry are:

- The upper half of data mostly contains inactive values without frequent updates. This idleness leads to a small number of writes in the NVM segment of the cache without incurring high dynamic energy.
- Due to this limited number of writes on the NVM part of the cache, a small write buffer can reduce the write latency of NVMs in cache level. This results in significant area and power improvement of the entire cache.

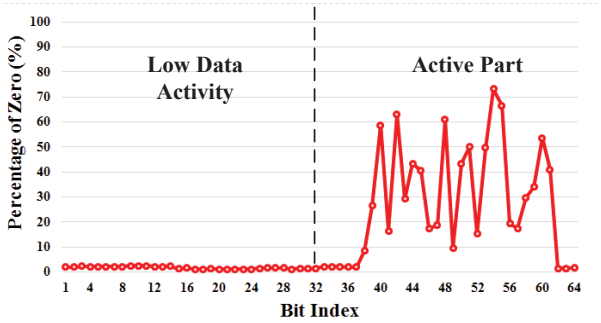

Figure 2. Average data distribution in different bit index for L2 cache running SPEC CPU 2006 benchmarks

We used NV*sim* tool [25] to estimate the power, performance and area of the SRAM and the non-volatile caches. This tool estimates the write/read access time, read/write access energy, leakage power and area of SRAM and STT-RAM. However, NV*sim* does not support SOT-RAM cell. We modified the NV*sim* source code to support SOT-RAM features and verified our results accordingly [26]. We used a circuit level HSPICE simulation to extract the cell features of STT-RAM and SOT-RAM based on the experimental setup in [27] using 45nm predictive technology modeling (PTM).

Table 1 shows the performance and power consumption of 64-bit wordline, 512KB SRAM, STT-RAM and SOT-RAM cache. The result shows that STT-RAM and SOT-RAM consume ~90% lower leakage power than the SRAM cache. At iso-capacity points, both STT-RAM and SOT-RAM are 52% and 46% smaller than the SRAM cache. In cell level SOT-RAM and STT-RAM have slow latency characteristics than SRAM. However, in large caches such as L2/LLC, NVMs have more comparable latency as SRAM due to higher NVM density (~3-4X denser cell) which results lower bitline/interconnect delay. This fact makes the hit latency of NVM-based caches much lower than the SRAMs. Our observation shows that in cache larger than 256KB, SOT-RAM cache has comparable read/write energy with respect to the SRAM cache. This alongside non-volatility and low static power make it suitable to replace SRAM in L2 cache without any performance penalty. The write energy of this memory is still higher than SRAM that is addressed by its proper usage to only store upper 32-bit. In contrast, STT-RAM has 1.7X slower write operation respect to the SRAM L2 cache. In the following, we exploit a write buffer and two new cache protocols to compensate the long write delay when using STT-RAM as the NVM part.

A. Write Buffer

To solve the performance problem of the proposed STT-RAM/SRAM cache, we added the SRAM write buffer in the cache structure. Utilizing an appropriate buffer size hides the long write latency of STT-RAM such that proposed cache can work in same performance as the SRAM cache. However, it is not required to write all coming data on buffer, because the narrow-width data can be written directly on the cache without any performance overhead. This technique increases the buffer availability for critical input data (non-narrow data). In addition, processor can directly access to write buffer for reading the data. For write intensive workloads we require large write buffer that can handle all non-narrow data of the cache. But such a large SRAM buffer has two disadvantages: high leakage power, and high dynamic energy of write back operations from buffer to cache.

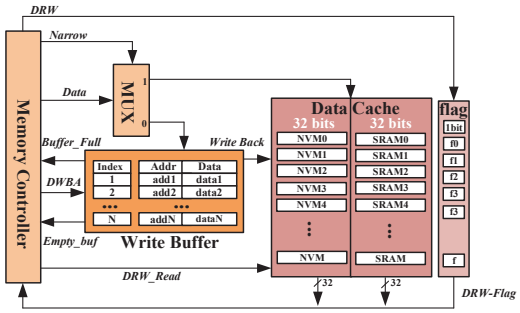

Figure 3. Proposed hybrid cache structure

Table 1. L2 cache parameters in different memory technologies (512KB cache - 64bit)

	*STT-RAM**	*SOT-RAM*†	*SRAM*‡
Non-Volatile	Yes	Yes	No
Data Storage	Magnetization	Magnetization	Latch
Hit Latency	1.83ns	2.05ns	4.26ns
Miss Latency	1.12ns	1.35ns	2.03ns
Write Latency	5.89ns	2.93ns	3.44ns
Hit/Miss Energy	0.175nJ	0.123nJ	0.088nJ
Write Energy	0.044nJ	0.036nJ	0.029nJ
Leakage Power	103.38mW	11.42mW	905.02mW
Area	0.76mm^2	0.93mm^2	1.74mm^2

*STT_RAM optimized based on write EDP
†SOT-RAM optimized base on latency
‡SRAM optimized based on leakage power

B. Partial Double Row Writing Technique

To solve the aforementioned buffer size problem, we introduce a new cache protocol for our hybrid STT-RAM/SRAM cache structure. The technique is called double row write (DRW) protocol. The proposed protocol works as follow (see Figure 4):

(i) When the input data is narrow, it is directly written on cache without checking or using the write buffer. This

can be done as fast as SRAM based write, because the NVM part does not need to be updated.

(ii) If the input data is not narrow-width and the SRAM write buffer has enough space (*Buffer_full=0*), the input data is written on the buffer. The buffer data can be transferred to the cache on write-back mode when the cache is not in critical write mode. The utilization of cache buffer is directly depends on the number of consecutive non-narrow write request on the input file.

(iii) When the input data is not narrow and the write buffer is full (*Full_Buffer=1)*, writing the non-narrow data on the cache will be slowed by the NVM latency. Our solution is using DRW techniques where the non-narrow data is partitioned to two 32-bit pieces, and each one is written on first 32-bit entry of two consecutive rows on the SRAM part (*DRW=1*). This write method bypasses the NVM latency in the critical cases by partially leaving the 32-bit upper part of the hybrid cache *blank*.

In the worst-case for the critical cases, this method decreases the effective cache capacity by a factor of two. However since the NVM part consumes very low leakage power and occupies low area, the power overhead of the cache with double cache write will be small. Based on the NV*sim* tool, in 45nm technology 10K write buffer occupies 0.079mm^2 and an extra flag bit increases the area of SRAM based cache by (1/64)≈1.6% of the area of conventional SRAM caches. Thus, the proposed cache has still 24% area efficiency respect to conventional SRAM based cache. Further, this double row write is only activated for a short period of time until the write buffer becomes empty again and the flag bit becomes zero (*Buffer_full=0*). To save data as a double row write, we use an extra flag bit on the cache structure. During the read operation if the *DRW_Flag* signal is one, the cache reads the first 32 bits of two consecutive rows and merges them to 64-bit data. In the case of zero flag bit (*DRW_Flag=0*) the data can be read from cell as a conventional cache.

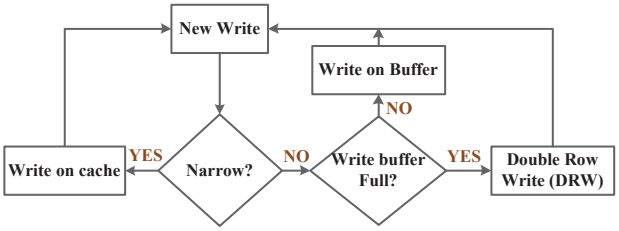

Figure 4. Proposed cache protocol to write data

C. Dynamic Write Buffer Allocation

The goal of dynamic write buffer (DWB) allocation is to decrease write buffer energy overhead and adjust the optimum buffer size for different applications. In this technique, at first system activates 2KB rows in write buffer and power gate the rest of the buffer. Based on the application request (which is related to the consecutive non-narrow write operation), the buffer size can increase up to the maximum level with 2KB steps. If the maximum buffer could not satisfy our performance (the *buffer_full* flag activates when system uses the maximum buffer size), the cache switches to DRW mode till the buffer becomes empty again. This technique balances a tradeoff between buffer size and the number of double row write on the cache (effective cache capacity). Large buffer consumes high leakage power but it reduces the average time that memory is in DRW mode. So, the DWB chooses an optimum buffer size requirement such that it can balance between buffer and DRW energy consumption. This technique dynamically changes the buffer size using power gating technique. More details about the functionality and impact of this technique are presented in Section 5.

V. USING THE TEMPLATE

A. Experimental Setup

We used GEM5 simulator [28] to extract L2 cache trace for benchmarks. We first fast-forward 500 million instructions and then simulated the next one billion instructions in detail. The processor configurations are listed in Table 2. To set L2 cache latency in GEM5 simulator, we used the worst-case latency (number of cycles) from NV*sim* tool. For example, in hybrid STT-RAM/SRAM structure, we used hit latency of SRAM and write latency of STT-RAM as the worst case latencies in GEM5. Accordingly, we used extensive simulations on GEM5 for a wide range of benchmark: SPEC CPU2006, SPEC CPU2000 and Mibench. The latency configuration and benchmarks are listed on Table 3.

Table 2. Baseline Processor Configuration

Frequency	Single-core @ 2 GHz, Out-of order execution
L1 Cache	32KB, 2-way associative, 64B cache line
L2 Cache	512KB, 8-way associative, 64B cache line
Memory	DRAM, DDR3, 4GB

Table 3. Configuration details of different cache delay and used benchmarks

Extracted delay @L2	
Cache Type	***Worst Case Read/Write Latency***
SRAM	Write:3.4ns Read:4.3ns
STT-RAM/SRAM	Write:5.9ns Read:4.3ns
SOT-RAM/SRAM	Write:3.4ns Read:4.3ns
STT-RAM	Write:5.9ns Read:1.9ns
SOT-RAM	Write:2.6ns Read:2.1ns
Benchmarks	
SPEC CPU2006	mcf, bzip2, gcc, hmmer, sjeng, libquantum, gobmk, milc, perlbench, bwaves, leslied3d, games, h246ref
SPEC CPU2000	bzip2, gcc, vortex, twolf, swim, lucas, fma3d, apsi, ammp
Mibench	basicmath, bitcount, qsort, susan, dijkstra, patricia, sha

B. Proposed Cache without Write Buffer

This section compares the impact of the proposed cache architecture on the performance and power of system without using write buffer. As we explained before, the STT-RAM has faster read and slower write operations than SRAM in L2 cache. Therefore, the write operation on STT-RAM decreases the performance of pure STT-RAM or the hybrid SRAM/STT-RAM caches. In SOT-RAMs both read and write operations (in L2 cache level) are faster than SRAM and the system will not have any performance overhead compare to the SRAM based cache. However, higher write energy of this cache may degrade the energy efficiency of the SOT-RAM and SRAM/SOT-RAM caches. Figure 5 and Figure 6 depict the energy consumption and performance of the hybrid STT-RAM/SRAM, SOT-RAM/SRAM and pure STT-RAM and SOT-RAM caches normalized to conventional the SRAM based cache considering the overhead of of the DRW and DWB technique. The results indicate that the proposed

SRAM/STT-RAM and SRAM/SOT-RAM caches consume up to 43% and 46% (on average 36% and 40%) lower energy with less than 13% performance overhead on STT-RAM and no performance overhead on SRAM/SOT-RAM. Therefore, SOT-RAMs are appropriate candidates to replace the first 32-bit entry of last level cache. In this region, they have low data activity and consume low dynamic energy. Pure SOT-RAM has 9% lower latency than SRAM but it suffers from high dynamic energy of 32-bit of cache entry where the data are very active. The results show that the pure SOT-RAM has 13% higher energy consumption than SRAM.

In write intensive benchmarks with low percentage of narrow-width data (e.g. *mcf* and *hmmer* in SPEC CPU2006 benchmarks) the performance of STT-RAM/SRAM cache reduces up to 21%. In these benchmarks, the high number of non-narrow writes increase the NVM dynamic energy. We will show how the introduced write technique solves the performance overhead of STT-RAM/SRAM cache. However in pure STT-RAM cache the low read latency of STT-RAM can be used to compensate the long write latency of STT-RAM. For SPEC 2006 benchmarks pure STT-RAM cache has 4% performance overhead on average and the energy consumption increases up to 19%. Note that the power overhead of pure NVMs cache will increase severely for write intensive workloads.

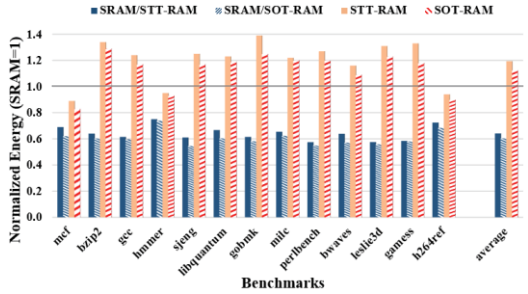

Figure 5. Normalized energy of STT-RAM/SRAM and SOT-RAM/SRAM hybrid cache without buffer.

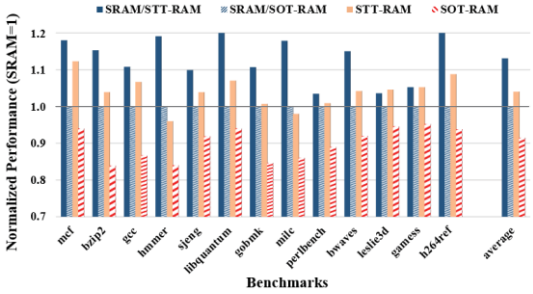

Figure 6. Normalized performance of STT-RAM/SRAM and SOT-RAM/SRAM hybrid cache whiteout buffer

C. Buffer Utilization

The optimum buffer size requirements for different SPEC 2006 benchmarks are listed in Table 4. Based on this table, benchmarks need at least 5.9KB SRAM buffer on average to have a zero performance overhead compare to the SRAM based cache. Although the large buffer size guarantees the performance of our proposed cache for more write intensive workloads, the leakage power of buffer can be dominant and decrease the power efficiency of the cache structure. Figure 7 shows the normalized performance and energy consumption of the proposed cache using different buffer sizes. Cache structure without write buffer improves energy consumption 40% on average but it has up to 21% performance overhead (13.1% on average). This worst-case performance overhead decreases to 5.1% with 4KB write buffer. Utilizing buffer size equal or larger than 6KB, guarantees the same performance as the conventional cache for all benchmarks. For other write intensive workloads, the system may still have performance overhead for 6KB and larger buffer size. To have a general solution which can guarantee system performance in all buffer size, we introduced double row write and dynamic write buffer allocation techniques.

Table 4. Minimum buffer requirement for different SPEC CPU2006 benchmarks

Benchmarks						
Mcf	*bzip2*	*gcc*	*hmmer*	*sjeng*	*Libquant*	*Gobmk*
5.9KB	5.3KB	4.9KB	5.5KB	3.6KB	3.7KB	0KB
Benchmarks						
Milc	*perlbench*	*bwaves*	*leslie3d*	*gamess*	*h264ref*	*Average*
0.5KB	0.1KB	3.1KB	3.5KB	4.3KB	6.0KB	3.6KB

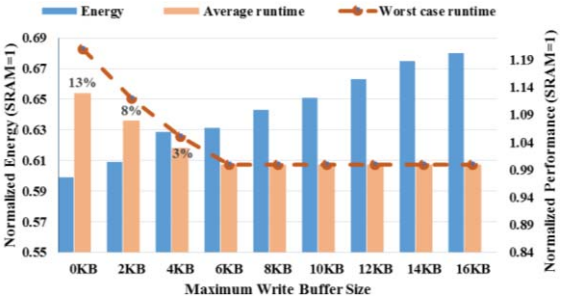

Figure 7. Average energy consumption of proposed STT-RAM/SRAM cache with different buffer size

D. Proposed Cache Protocol Implementation

To avoid using very large write buffer, the double row write technique is used to control high write latency of STT-RAM. In this technique utilizing any size of write buffer guarantees the performance of cache architecture.**Error! Reference source not found.** shows the advantage of the proposed method on different buffer sizes. Note that the buffer size values on table are the maximum available buffer size. It shows the obvious tradeoff between the maximum buffer size and the number of double row write on the cache. Large buffer size consumes high leakage power while it decreases the average time of the system in DRW mode. On the other hand using DRW mode, memory consumes the power of two half-write and two half-read operation. So, this power can be dominant when write buffer is too small. For SPEC CPU2006 benchmarks, using 4KB and 6KB write buffer provides 42% and 40% energy improvement without any performance overhead. Increasing the maximum buffer size larger than 8K, does not have positive impact on power saving, because for several SPEC applications the small buffer can be enough to write non-narrow data. To gain the maximum energy saving, 4KB buffer is the optimum write buffer requirement.

Table 5. Proposed cache efficiency in different buffer size

Maximum Buffer Size	2K	4K	6K	8K	10K	12K	14K	16K
Normalized Runtime	1	1	1	1	1	1	1	1
# of DRW (KB)	6.8	2.7	0	0	0	0	0	0
Normalized Energy	0.61	0.58	0.60	0.61	0.63	0.64	0.65	0.67

* Power gating considered on buffer with granularity of 2KB

To show the effect of proposed architecture on the other benchmarks with different proportions of narrow-width data, we test the proposed architecture with Mibench and SPEC CPU2000 benchmarks. Figure 8 shows the average double row writes on the proposed memory when the system runs different benchmarks. This value depends on the buffer size. Large buffer decreases the percentage of time that the cache is in DRW mode. Therefore, the minimum energy point balances the amount of DRW energy and buffer size energy consumption. Figure 9 shows the average normalized energy on SPEC CPU2000 and Mibench benchmarks in different maximum buffer sizes. In Mibench and SPEC 2000 benchmarks using 8KB and 4KB buffer are optimum energy point such that the cache achieved 36% and 39% energy saving respect to the SRAM based cache. Note that in the optimum energy point, number of DRW write is not zero because the proposed technique balances the number of double row writes and buffer energy consumption.

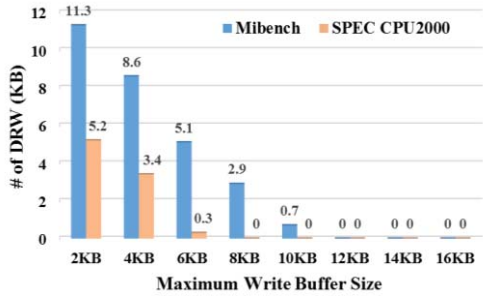

Figure 8. Number of DRW write on STT-RAM/SRAM cache in different maximum buffer size

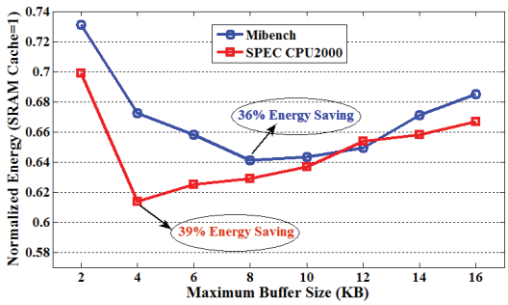

Figure 9. Normalized energy of STT-RAM/SRAM cache in different maximum buffer size

VI. Conclusion

This paper proposes a new hybrid MRAM/SRAM cache architecture that leverages narrow-width values to reduce the cache power. Based on the cache data distribution on modern processors, much data can be represented just by the first 32-bit of the word line and the rest of the bits are mostly zero and idle. The proposed architecture is designed such that the upper 32-bit of wordline (non-active data) is saved on NVM which consumes very low leakage power. To overcome the high write latency of NVMs in non-narrow data, we propose a partial double row write (DRW) technique and dynamic write buffer (DWB) allocation to decide the trade-off between effective cache availability and the maximum buffer size. Our evaluation on several benchmarks shows that proposed hybrid cache can achieve up to 46% power improvement and 24% area efficiency in the same performance to conventional SRAM cache.

Acknowledgment

This work was supported by NSF grant #1527034 and UCSD Powell Fellowship.

References

[1] N. S. Kim, et al., "Leakage current: Moore's law meets static power," IEEE computer, vol. 36, pp. 68-75, 2003.

[2] C. J. Xue, et al., "Emerging non-volatile memories: opportunities and challenges," IEEE/ACM CODES+ISSS, pp. 325-334, 2011.

[3] M. Imani, et al., "DCC: Double Capacity Cache Architecture for Narrow-Width Values," ACM GLSVLSI, pp. 113-116, 2016.

[4] M. R. Guthaus, et al., "MiBench: A free, commercially representative embedded benchmark suite," IEEE Workshop WWC, pp. 3-14, 2001.

[5] S. Che, et al., "Rodinia: A benchmark suite for heterogeneous computing," IEEE IISWC, pp. 44-54, 2009.

[6] J. A. Stratton, et al., "Parboil: A revised benchmark suite for scientific and commercial throughput computing," Center for Reliable and High-Performance Computing, 2012.

[7] A. Jog, et al., "Cache revive: architecting volatile STT-RAM caches for enhanced performance in CMPs," IEEE DAC, pp. 243-252, 2012.

[8] C. W. Smullen, et al., "Relaxing non-volatility for fast and energy-efficient STT-RAM caches," IEEE HPCA, pp. 50-61, 2011.

[9] S. Li, et al., "Leveraging emerging nonvolatile memory in high-level synthesis with loop transformations," IEEE/ACM ISLPED, pp. 61-66, 2015.

[10] L. Benini, et al., "System-level power optimization: techniques and tools," ACM TODAES, vol. 5, pp. 115-192, 2000.

[11] M. Imani, et al., "Low power data-aware STT-RAM based hybrid cache architecture," IEEE ISQED, pp. 88-94, 2016.

[12] S. Li, et al., "Leveraging nonvolatility for architecture design with emerging NVM," IEEE NVMSA, pp. 1-5, 2015.

[13] D. Brooks, et al., "Dynamically exploiting narrow width operands to improve processor power and performance," IEEE HPCA, pp. 13-22, 1999.

[14] A. Aggarwal, et al., "Energy efficient asymmetrically ported register files," IEEE ICCD, pp. 2-7, 2003.

[15] S. Wang, et al., "Exploiting narrow-width values for thermal-aware register file designs," IEEE DATE, pp. 1422-1427, 2003.

[16] S. Wang, et al., "Low power aging-aware register file design by duty cycle balancing," IEEE DATE, pp. 546-549, 2012.

[17] G. Duan, et al., "Exploiting narrow-width values for improving non-volatile cache lifetime," IEEE DATE, 2014.

[18] C. J. Xue, *et al.*, "Emerging non-volatile memories: opportunities and challenges," *IEEE/ACM Codess*, pp. 325-334, 2011.

[19] Q. Li, et al., "MAC: migration-aware compilation for STT-RAM based hybrid cache in embedded systems," ACM/IEEE ISLPED, pp. 351-356, 2012.

[20] X. Wu, et al., "Power and performance of read-write aware hybrid caches with non-volatile memories," IEEE DATE, pp. 737-742, 2009.

[21] Z. Sun, et al., "Multi retention level STT-RAM cache designs with a dynamic refresh scheme," IEEE/ACM microarchitecture, pp. 329-338, 2011.

[22] Y. Xie, "Modeling, architecture, and applications for emerging memory technologies," IEEE Design & Test of Computers, pp. 44-51, 2011.

[23] I. M. Miron, et al., "Current-driven spin torque induced by the Rashba effect in a ferromagnetic metal layer," Nature materials, vol. 9, pp. 230-234, 2010.

[24] J. Tan, et al., "Soft-error reliability and power co-optimization for GPGPUS register file using resistive memory," IEEE DATE, pp. 369-374, 2015.

[25] X. Dong, et al., "Nvsim: A circuit-level performance, energy, and area model for emerging nonvolatile memory," IEEE TCAD, vol. 31, pp. 994-1007, 2012.

[26] R. Bishnoi, et al., "Architectural aspects in design and analysis of SOT-based memories," IEEE ASP-DAC, pp. 700-707, 2014.

[27] X. Fong, et al., "SPICE Models for Magnetic Tunnel Junctions Based on Monodomain Approximation," 2013.

[28] N. Binkert, et al., "The gem5 simulator," ACM SIGARCH Computer Architecture News, vol. 39, pp. 1-7, 2011.