D–MRAM Cache: Enhancing Energy Efficiency with 3T–1MTJ DRAM / MRAM Hybrid Memory

Hiroki Noguchi, Kumiko Nomura,
Keiko Abe, Shinobu Fujita
Toshiba Corporate R&D Center, Kawasaki, Japan
{hiroki.noguchi, kumiko.nomura, keiko2.abe,
shinobu.fujita}@toshiba.co.jp

Eishi Arima, Kyundong Kim, Takashi Nakada,
Shinobu Miwa, Hiroshi Nakamura
The University of Tokyo, Tokyo, Japan
{arima, kim, nakada, miwa, nakamura}
@hal.ipc.i.u-tokyo.ac.jp

***Abstract*—This paper describes a proposal of non-volatile cache architecture utilizing novel DRAM / MRAM cell-level hybrid structured memory (D-MRAM) that enables effective power reduction for high performance mobile SoCs without area overhead. Here, the key point to reduce active power is intermittent refresh process for the DRAM-mode. D-MRAM has advantage to reduce static power consumptions compared to the conventional SRAM, because there are no static leakage paths in the D-MRAM cell and it is not needed to supply voltage to its cells when used as the MRAM-mode. Besides, with advanced perpendicular magnetic tunnel junctions (p-MTJ), which decreases the write energy and latency without shortening its retention time, D-MRAM is capable of power reduction by replacing the traditional SRAM caches. Considering the 65-nm CMOS technology, the access latencies of 1MB memory macro are 2.2 ns / 1.5 ns for read / write in DRAM mode, and 2.2 ns / 4.5 ns in MRAM mode, while those of SRAM are 1.17 ns. The SPEC CPU2006 benchmarks have revealed that the energy per instruction (EPI) of the total cache memory can be dramatically reduced by 71 % on average, and the instruction per cycle (IPC) performance of the D-MRAM cache architecture degraded only by approximately 4 % on average in spite of its latency overhead.**

I. Introduction

In recent years, mobile processors have been remarkably developed, and multi-core processors are becoming a mainstream even in mobile applications. This trend causes increasing the total chip area and, of course, its power consumption. For mobile processors, it is clearly required to maximize the performance and to minimize the power consumption simultaneously.

To meet the chip power budget, however, the CPU cores cannot operate at the maximum operation frequency without considering total power consumption and the chip temperature, which is called the dark silicon problem [1–2]. In the dark silicon era, managing cache static power has become one of the most critical design constraints to meet the chip power budget. This is because caches dissipate the most part of total power consumption of SoCs [3]. To decrease the leakage power of caches, power gating (PG) is currently used for long-term stand-by states when no applications are running on the CPU. However, this PG cannot be used while application is running due to power / performance overhead and operation instability after PG, even though there are frequent short stand-by states of cache memory [4–7]. Therefore, reduction in leakage power for the short stand-by states without PG is the key target to reduce the power of mobile SoC further.

To reduce static power consumptions effectively, non-volatile memory technologies, which consume zero-stand-by power because of their non-volatility, have been explored, such as magnetic RAM (MRAM) [8], phase-change RAM (PCRAM) [9–10], resistive RAM (ReRAM) [11], and legacy NAND Flash. Among these non-volatile memories, STT-MRAM is the most promising candidate [12–13], because of its several advantages; high-speed accesses (1 to 50 ns), high endurance ($>10^{15}$), and notable characteristics, low-power CMOS logic compatibility [14].

In this paper, to save the leakage power consumption, without PG during application running, we introduce a cell-level hybrid DRAM / MRAM memory (D-MRAM) [26] and propose a power saving architecture by adapting DRAM mode or MRAM mode with this D-MRAM as a cache memory.

The rest of this paper is organized as follows. Section II introduces the key issues in applying STT-MRAM to caches and related works. Section III describes the D-MRAM design concepts and its evaluated performance. Section IV proposes our cache architecture utilizing D-MRAM and Section V discusses experimental results before summarizing this paper.

II. Background and Related Work

A. Write Latency and Energy of STT-MRAM

Delay and power overhead is fatal for cache memory, if slow-operation MTJ-based memory such as the previous generation toggle MRAM [15] or the other MRAMs which spends more than 25 ns to write in [12, 16] is used for the cache. Until now, delay and energy overheads have been acknowledged as critical issues on MRAM-based caches. Although STT-MRAM can reduce the power overhead [8], it is not sufficient for decreasing power compared with SRAM. Very recently, advanced STT-MRAM [26] has been developed with its power overhead is about 1/10 of that for previous STT-MRAM. On the other hand, architecture approaches to conceal these overheads in write operations have been proposed [17–18]. Therefore, much emphasis has been placed on the design of write buffer, and asymmetrical-write methods [19] have

978-3-9815370-0-0/DATE13/©2013 EDAA

TABLE I. ARCHITECTURE COMPARISON. STATIC ENERGY IS INCLUDING CONTINUOUS REFRESHING ENERGY AND BUS ENERGY FOR REPLACING.

Architecture	Latency	Dynamic	Static
SRAM	Fast	Low	High
Write optimize [17, 19]	Slow	Medium	Low
3D stacked [18]	Slow	High	Low
Relaxed retention [20]	Medium	Medium	Medium
Multi retention [22]	Fast	Medium	Low
Short retention [23]	Fast	High	Medium
Hybrid cache [24–25]	Fast	Medium	Medium
D-MRAM cache (This work)	Fast	Low	Low

been proposed so far. Delay time and energy consumption still have been bottlenecks in using MRAM in cache memory.

B. Short Retention STT-MRAM

Short retention STT-MRAM (SR-STT-MRAM) which is able to write with short-pulse width by improving MTJ devices themselves has been proposed [20–21], and some researches aimed at applying it to cache memory have been conducted [22–23]. However, they need to be refreshed regularly for continuous data retention due to its short retention time. Zhenyu Sun et al. presented the cache architecture utilizing several retention time domains [22]; it requires moving data among domains and refreshing. Thus, when physical replace occurs between domains, extra energy is consumed due to not only read and write dynamic energies in memory array but also transfer dynamic energies in on-chip data bus. Because this data transfer in bus is issued in units of a cache line (e.g., 64 B), this bus energy cannot be neglected. The most severe issue of the SR-STT-MRAM is instability of fundamental memory function, i.e., increasing risk of both write error and read disturbance. SR-STT-MRAM cannot therefore be adopted for real applications.

C. Previous STT-MRAM Cache Architectures and Proposal

Table I summarizes architecture comparisons among recently proposed STT-MRAM caches. Early write termination reduces full-time write opportunity [17] and write-asymmetric memory reduces write per thousand instructions (WPKI) [19]; the write energy is surely reduced depending on benchmarks but write latency deteriorates CPU performance. 3D stacked enables to implement larger capacity because of stacking implementation [18], but it cannot conceal the latency overhead of STT-MRAM. Relaxed retention succeeds to average merits and weak points of STT-MRAM [20], but the relaxed STT-MRAM needs to refresh constantly and it leads to inadequacy. This refreshing overhead is also large for multi retention and short retention proposals [22–23] and it deteriorates the performance per energy. Multi retention [22] and hybrid [24] caches require data replacement via swap buffers, and it leads to dissipate transfer energy in bus.

It is reasonable that short retention memory is used for cache memory, since long retention is not needed generally for data in cache. Embedded DRAM (e-DRAM) is, therefore, considered to be candidate for cache, as read/write power and refresh power can be smaller than that of short retention STT-MRAM. It is furthermore expected that hybridization of e-DRAM and STT-MRAM is more effective for power reduction of cache. In addition, we propose cell-level hybrid DRAM/MRAM where DRAM is composed of only NMOS transistors to avoid area overhead and process cost overhead for hybridization [26].

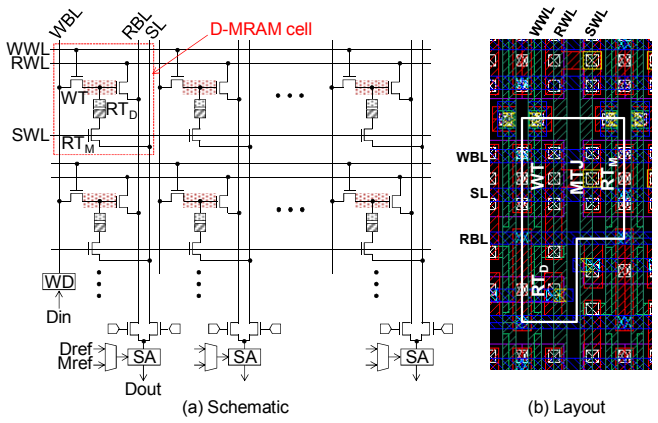

Fig. 1. D-MRAM (a) schematic and (b) cell layout. The cell size is 1.002 μm^2 at 65 nm and can be reduced because of its small transistor counts.

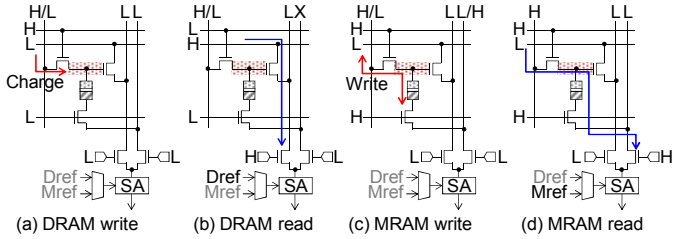

Fig. 2. D-MRAM operation mechanisms.

TABLE II. SIMULATED MEMORY PARAMETERS OF D-MRAM BASED CACHE AT THE 65-NM CMOS (USING TYPICAL CC-PROCESS CORNER, 25°C), CONSIDERING THE ADVANCED P-MTJ [27].

Memory type		Read / write latency (ns)	Read / write / internal-write dynamic energy (nJ)	Leakage (mW)
L1 32 KB	SRAM	0.68	0.367 / 0.357	7.96
	DRAM	1.4 / 0.7	0.240 / 0.510 / 0.289	3.53
	MRAM	1.4 / 3.7	0.392 / 0.763 / 0.543	3.53
L2 1 MB	SRAM	1.17	1.26 / 1.24	144
	DRAM	2.2 / 1.5	1.04 / 1.58 / 0.578	34.3
	MRAM	2.2 / 4.5	1.63 / 2.10 / 1.09	34.3

III. PROPOSAL OF HYBRID DRAM / MRAM (D-MRAM)

A. Circuit Design of D-MRAM

In D-MRAM depicted in Fig. 1, three nMOS transistors are connected to a MTJ device; WT for a wordline (WWL), RT_M for a source wordline (SWL) and RT_D for a read wordline (RWL). The capacitance of RT_D's gate and RT_M's source-drain capacitance (off-state) is equivalent to a capacitor for DRAM, as shown in Figs. 1 and 2. The operation mechanism of D-MRAM is explained in Fig. 2. The retention time is more than 10 μs for 1 MB at the typical process, estimated by Monte

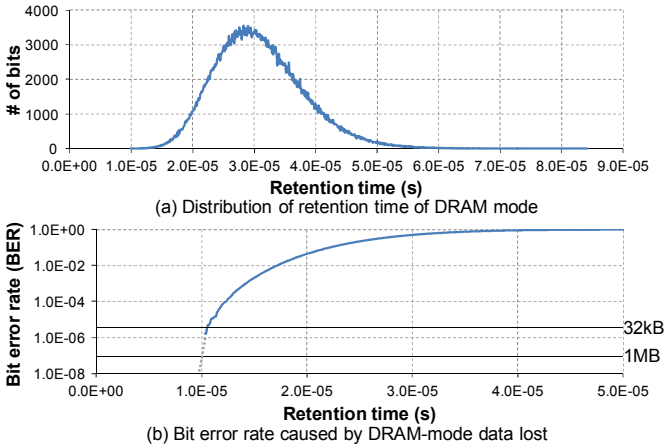

Fig. 3. Monte Carlo simulation with 65-nm CMOS process technology shows about 10-µs retention time for typical CC corner (25°C); (a) shows the raw distributed retention times of DRAM mode and (b) shows accumulated bit error rate (BER) curves. The 32 KB crossbar is for L1 cache, and 1 MB crossbar is for L2 cache.

Carlo simulation considering the device variation, shown in Fig. 3, while the retention time for MRAM mode is over 10 years. For MRAM mode, write operation is based on current injection into the MTJ through two transistors by the spin torque transfer, as shown in Fig. 2 (c), and read operation uses current sensing scheme, as shown in Fig. 2 (d).

B. Memory Parameters for Cache Design

We estimated the macro-level performance of this D-MRAM with advanced p-MTJ, which decreases the write energy and latency without shortening its retention time [27]. The simulated memory performance is listed in Table II. The parameters in Table II are calculated based on SPICE circuit level simulation. The access latencies are not included bus latency and tag access. The dynamic and static powers are included for H-Tree logic circuits and routing to the memory banks. Although D-MRAM is non-volatile memory, it consumes static power for the peripheral circuits including H-Tree drivers implemented with traditional CMOS circuits and they should not be powered off to avoid large latency overheads due to power-up [28]. The number of I / O bits is set to 512 b for every set. Sub-banks are consists of 512 b × 512 b (= 32 KB) for L1 cache, and 1024 b × 1024 b (= 128 KB) for L2 cache. The comparison of accumulated dynamic energy considering DRAM refreshing is shown in Fig. 4. Although the DRAM dynamic energies are less than those of MRAM, but DRAM operates as a volatile memory to keep the stored data, refreshing is required. Since the crawling refresh consumes power, our approach is that DRAM refreshing is not conducted for all data, and spot refreshing is done when data are readout. When read operation energy is also included, total refreshing energy can be reduced by the spot refreshing. Furthermore, it should be noted that the refreshing energy is less than the write energy; because for the write operation transfer energy is needed to bring in the data from memory controller unit (MMU) via the H-Tree bus. However, the refreshing operation can be locally performed, and such transfer energy is not needed.

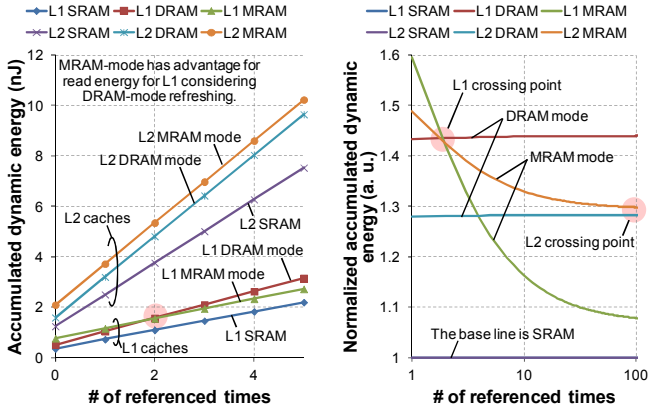

Fig. 4. Accumulated dynamic energy comparison among SRAM, DRAM (DRAM-mode) and MRAM (MRAM-mode) of hybrid D-MRAM. In this case, DRAM refreshing is performed at every access.

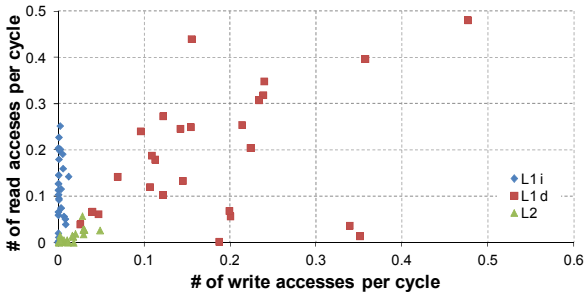

Fig. 5. Cache activities with SPEC CPU2006 benchmarks (INT: 12, FP: 14).

IV. D-MRAM Cache Architecture

A. Grand Design for Cache Architectures

STT-MRAM generally has larger dynamic power and slower write operation than SRAM, while the static leakage is relatively small. It is presumed that D-MRAM has these weak points; long write pulse is needed in storing data to MRAM, and refreshing energy consumed for DRAM. Our architecture described in the next subsection, however, can defeat these weak points and involves new cache design concept for effective power reduction. Adopting novel data migration policy, without long write pulses data can be stored to DRAM and once the data are stored in MRAM, the data can be kept without any follow-on refreshing.

B. Characteristics of Each Cache Hierarchies

To clear up the access pattern of caches, we analyze the cache activities using SPEC CPU2006 benchmarks [29], as shown in Fig. 5. The level 1 (L1) instruction cache and data cache have high-rate access repetition, and there is an asymmetrical characteristic between read access and write access. Of course this tendency changes depending on the application running on the CPU; the L1 instruction cache has much more frequent read accesses than write accesses and particularly all write requests are certainly issued from memory side, because CPU does not modify the running program code. Because most of accesses are read requests for L1 instruction cache, the increasing read latency sensitively deteriorates the

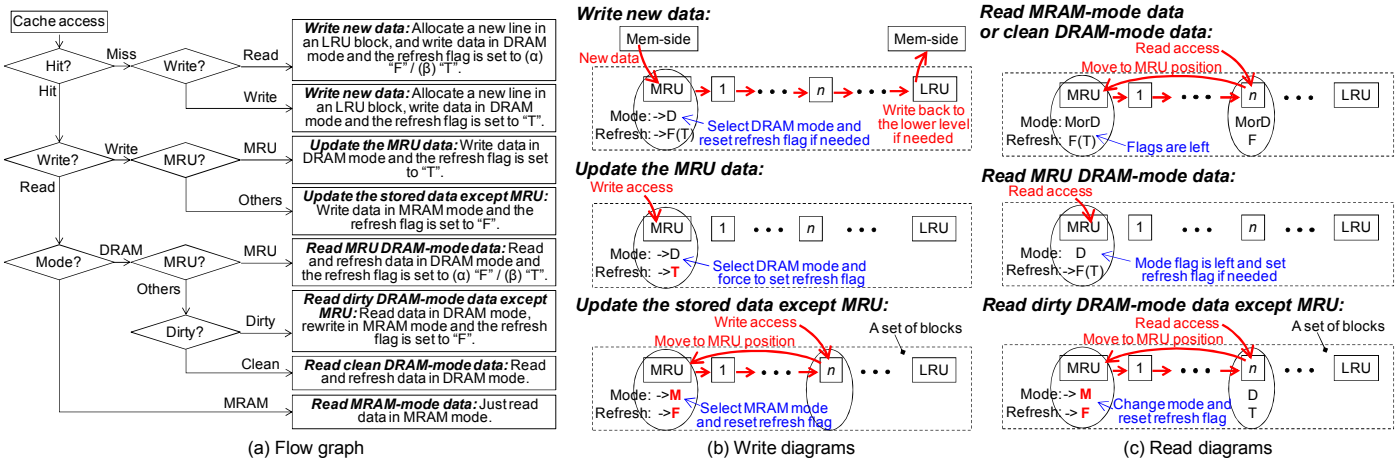

Fig. 6. Migration policy of D-MRAM cache: (a) flow graph (α) w/ (β) w/o low-power flag, (b) write diagrams and (c) read diagrams: "T", "F", "D", and "M" are represent "True", "False", "DRAM-mode" and "MRAM-mode" respectively.

instruction per cycle (IPC) performance due to the longer loading time of program code. On the contrary, for L1 data cache, as the treated data size in recent data-intensive mobile applications is increasing, the memory footprint is enlarged; the big-size web contents, high-definition multimedia, and even 3D games are implemented in handheld devices [30]. These applications have much more frequent accesses to L1 data cache than that to L1 instruction cache, yet the footprint of the running program is also enlarged. Consider to replace the L1 caches with non-volatile memory, write requests cause pipeline stalls, because the crucial read requests distracted by its longer write operations. To conceal these stalls, write buffers are very effective, but for L1 caches, the number of entries of write buffers is becoming blowup. The enough number of write buffers to prevent such stalls is over 100 for L1 data cache which is considering 4.5 ns write pulse. This number of write buffers becomes much bigger, if utilizing with the longer-write pulse MRAMs, such as [16].

For L1 caches, the access overhead derived from additional control circuits should be avoided, because L1 caches are sensitive to the CPU performance. On the other hand, the access requests to L2 cache is extremely fewer compared to L1 caches. Therefore, several proposals replace the traditional SRAM L2 cache with STT-MRAM or hybrid SRAM / STT-MRAM [18, 24–25]. To reduce the write accesses to L2 cache, even if only slightly, the L1 caches should make up not with write-through but with write-back protocol, because the write-through protocol is focusing on keeping coherency among several levels in memory hierarchy including caches in multi-core CPUs, rather than reducing energy consumption. Write-back protocol saves the required number of write buffer entries. These deep analyses have been used for creating D-MRAM cache architecture described in the next.

C. D-MRAM Cache Architecture

How to select operation modes, DRAM mode or MRAM mode, for the D-MRAM is the key point for increasing energy efficiency of cache memory. Basic policy how to select the mode is that when the fast operation is needed, DRAM-mode is continuously selected, when the long retention is needed, MRAM-mode is selected. At first, we can select the operation mode in terms of the number of refreshing times. Considering the L1 cache, the average-access intervals are relatively short enough for DRAM to hold the stored data. On the other hand, the access intervals of each L2 cache datum are much longer than that of L1. Since it leads to increase the miss rate of L2 cache, refreshing is required? The MRU position is a bell ringer to judge when the operating mode is translated from DRAM-mode into MRAM-mode and when to return DRAM mode. Since the MRU data have frequently short intervals until the next access, this policy is suitable to store data in the DRAM-mode memory cell. On the contrary, new data or LRU-sided data tend to have long intervals before the next access, and the next access is not issued while the data exists. For such data block, MRAM mode is preferred.

Next, we must select DRAM-mode when the access latency is critical. D-MRAM is not able to readout the data while the write process is operated. Therefore, the write latency in MRAM-mode cannot be neglected, when the large number of cache accesses is issued continuously. This situation cannot be largely changed, although the write-back protocol saves the number of write accesses and enough number of write-buffers enables temporal postponements of write processes. That is the reason that DRAM-mode should be selected as default operation mode to minimize the access-latency overheads.

Lastly, consider the number of referenced times, some memory cells do not need to be refreshed, because of their little referenced times. For such data, refreshing energy is waste. The LRU-sided data sometimes have long interval to next access or never be accessed forward. DRAM-mode has an advantage for these data, because after retention time the stored data is to be flushed natively without refreshing. To guarantee the robust function, however, unexpected data loss must be avoided. The important data, which are touched by CPU and have not been to be written-back to the lower level of memory, are required to keep surely by refreshing or rewriting to MRAM.

D. D-MRAM Management Policy

To solve the trade-off between DRAM and MRAM modes, the following D-MRAM management policies are proposed:

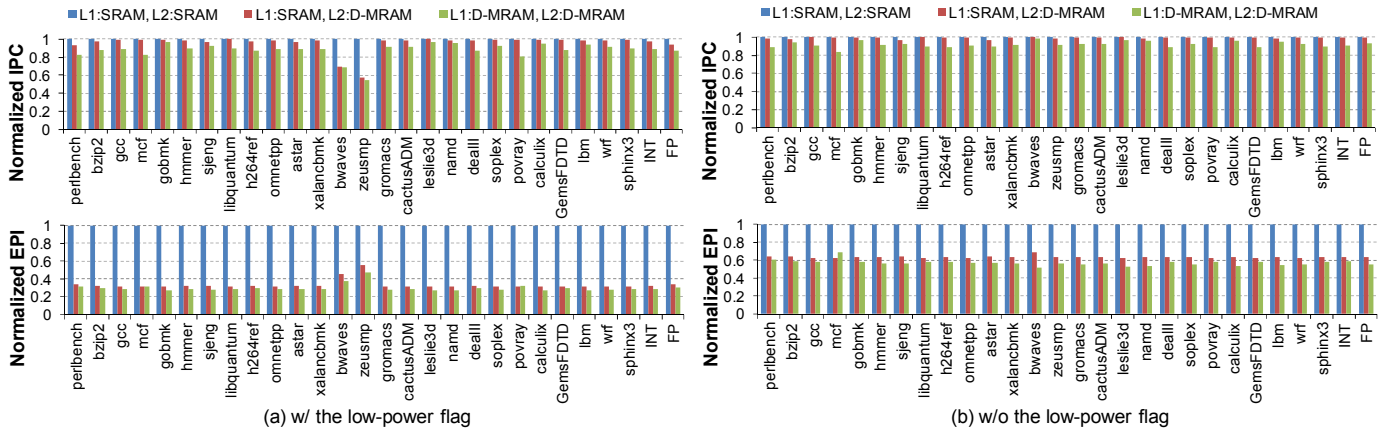

Fig. 7. Normalized instruction per cycle (IPC) and cache energy per instruction (EPI) comparisons. The low-power flag of the D-MRAM based caches is (a) "ON" and (b) "OFF". The baseline is SRAM caches.

***Policy 1*:** The write operation is always conducted in DRAM mode as a default. To reduce the number of refreshing in DRAM, when the write operation is issued, the cache controller checks the LRU-stack position of the accessing block. Only when the LRU-stack position is available (write hit) and it does not imply MRU position, the refresh flag is set. That is for the MRU data where the refresh flag is not set and refreshing will be skipped. This is because the MRU data have a tendency to have another access within short intervals.

***Policy 2*:** To avoid fatal error caused by DRAM data loss, the dirty data required to write back to lower-level memories are stored to DRAM-mode memory cells with refreshing.

***Policy 3*:** When the DRAM read operation is performed, the stored data are refreshed every time. In this operation, if the readout data are dirty and is not in MRU position, the data are rewritten in MRAM mode. This is because such data tend to waste refreshing energies. If the write-buffer is filled or next tag access is issued, the cache controller stalls the cache access until the operation of rewriting the data is finished. Because the number of read accesses to MRU positions is much more than that of the other positions, this stall is rarely occurred.

The above policies are applied all together. The flow graph and diagrams of above management policies are shown in Fig. 6. In Fig. 6 (a), the optional policy is shown at read miss operation. The low-power flag is added and in the case that the low-power flag is "OFF", the newcomer data are stored in DRAM-mode cells with refreshing flag to reduce the DRAM data loss and accelerate performance. On the other hand in the case that the low-power flag is "ON", the newcomer data are stored in DRAM-mode cells without refreshing flag to reduce refreshing energies.

V. Experimental Results

A. Simulation Setup

We modified a CPU simulator, gem5 [31] for non-volatile cache memories. Processor configurations used for the simulation are listed in Table III. The benchmarks used in this study are SPEC CPU2006 [29] benchmark suites. After 1G instructions bypassing and 100-M instruction warm-up for initializing cache, 200M instructions we evaluated with detailed CPU model. In order to estimate the access latency and energy consumption, the memory parameters shown in Table II were utilized. We assume that the tag memories both for SRAM-based cache and for D-MRAM-based cache are composed of typical SRAM memory, and there are no differences in latency and energy for those.

TABLE III. System configurations.

Processor	ARMv7-a, single core, 1 GHz, single thread, 3-way fetch, 3-way decode, 8-way issue out-of-order
SRAM L1 cache	32KB I/D, 2-way, 64B line, 1-cycle, write-back, 8-entry write buffers
D-MRAM L1 cache	32KB I/D, 2-way, 64B line, 2-cycle (MRAM write expends additional 2 cycles), write-back, 8-entry write buffers
SRAM L2 cache	1MB, 8-way, 64B line, 12-cycle, write-back, 16-entry write buffers
D-MRAM L2 cache	1MB, 8-way, 64B line, 13-cycle (MRAM write expends additional 2 cycles), write-back, 16-entry write buffers
Main memory	2GB, 100-cycle latency

B. Simulation Results and Discussion

Figure 7 show CPU performances, instruction per cycle (IPC) and cache energy per instruction (EPI) calculated by gem5 simulator. These results indicate that the performance degradation for D-MRAM based L1 cache is more dominated than for D-MRAM based L2 cache. The results on SPEC CPU2006 benchmarks reveal that the energy per instruction (EPI) of the total cache memory can be dramatically reduced by 71 % on average for the proposed D-MRAM cache architecture with limited degradation within 4 % on average.

For the benchmarks, "bwaves" and "zeusmp", the D-MRAM based caches decrease the IPC performances severely, in Fig. 7 (a). This is because memory-sided write operations are much performed in these benchmarks and DRAM-mode memory cell loses the stored data without refreshing, even for the data in MRU position, caused by the long-interval access due to their large-cycle latencies for floating-point computations. Although "bzip2" has also the large number of memory-side write operations, the access intervals are relatively shorter than that of "bwaves" and "zeusmp".

Therefore, D-MRAM based cache can keep the high IPC performance for "bzip2". To avoid this IPC performance degradation, the optional policy, the low-power flag, works effectively at the expense of refreshing energy in Fig. 7 (b).

VI. Conclusion

To consider the effective cache architecture suitable for innovative devices, such as non-volatile STT-MRAM, the traditional hierarchy makes no sense. In this work, we propose non-volatile cache architecture utilizing novel DRAM / MRAM cell-level hybrid structured memory (D-MRAM) that enables effective power reduction for high performance mobile SoCs. D-MRAM has advantage to reduce static power consumptions compared to the conventional SRAM, because there is no need to supply voltage for its cells. Besides with advanced perpendicular magnetic tunnel junctions (p-MTJ), which decreases the write energy and latency without shorten its retention time, D-MRAM has capable of replacing the traditional SRAM caches.

The presented cache architecture can save energies consumed in memory. Thus, the total energy budget which can be allocated to the CPU cores becomes relatively luxury. In near future, even when the large number of cores is implemented in a small-budget mobile system, the dark area can be reduced with adopting this non-volatile cache architecture. Furthermore, the non-volatile memory can hold the stored data at the extremely low-voltage operation even at zero volts, the dynamic voltage and frequency scaling (DVFS) techniques can be utilized with much wider voltage swings aggressively than ever used.

Acknowledgment

We thank the anonymous reviewers for their constructive comments. This study was partly supported by Normally-Off Computing Project of NEDO in Japan.

References

[1] O. Azizi, A. Mahesri, B. C. Lee, S. J. Patel and M. Horowitz, "Energy-performance tradeoffs in processor architecture and circuit design: a marginal cost analysis," Proc. of *the 37th ISCA*, pp. 26-36, Jun. 2010.

[2] H. Esmaeilzadeh, E. Blem, R. St. Amant, K. Sankaralingam, and D. Burger, "Dark silicon and the end of multicore scaling," Proc. of *the 38th ISCA*, pp. 365-376, Jun. 2011.

[3] Semiconductor Industry Association, The International Technology Roadmap for Semiconductors (ITRS): http://www.itrs.net/reports.html.

[4] S. Kaxiras, H. Zhigang, and M. Martonosi, "Cache decay: exploiting generational behavior to reduce cache leakage power," Proc. of *the 28th ISCA*, pp. 240-251, Jun. 2001.

[5] K. Flautner, N. S. Kim, S. Martin, D. Blaauw, and T. Mudge, "Drowsy caches: simple techniques for reducing leakage power," Proc. of *the 29th ISCA*, pp. 148-157, May 2002.

[6] H. Lakdawala et al., "32nm x86 OS-compliant PC on-chip with dual-core Atom® processor and RF WiFi transceiver," *ISSCC Dig. Tech. Papers*, pp. 62-63, Feb. 2012.

[7] G. Gerosa et al., "A Sub-1W to 2W low-power IA processor for mobile internet devices and ultra-mobile PCs in 45nm Hi-K metal gate CMOS," *ISSCC Dig. Tech. Papers*, pp. 256-257, Feb. 2008.

[8] T. Kishi et al., "Lower-current and fast switching of a perpendicular TMR for high speed and high density spin-transfer-torque MRAM," *IEDM Tech. Dig.*, pp. 1-4, Dec. 2008.

[9] S. J. Ahn et al., "Highly manufacturable high density phase change memory of 64 Mb and beyond," *IEDM Tech. Dig.*, pp. 907-910, Dec. 2004.

[10] S. Kang et al., "A 0.1 μm 1.8 V 256 Mb phase-change random access memory (PRAM) with 66 MHz synchronous burst-read operation," *IEEE J. Solid-State Circuits*, vol. 42, no. 1, pp. 210-218, Jan. 2007.

[11] S. S. Sheu et al., "A 4 Mb embedded SLC resistive-RAM macro with 7.2 ns read-write random-access time and 160 ns MLC-access capability," *ISSCC Dig. Tech. Papers*, pp. 200-202, Feb. 2011.

[12] K. Tsuchida et al., "A 64Mb MRAM with clamped-reference and adequate-reference schemes," *ISSCC Dig. Tech. Papers*, pp. 258-260, Feb. 2010.

[13] J. P. Kim et al., "A 45nm 1Mb embedded STT-MRAM with design techniques to minimize read-disturbance," *Symp. VLSI Circuits Dig. Tech. Papers*, pp. 296-297, Jun. 2011.

[14] X. Guo, E. Ipek, and T. Soyata, "Resistive computation: avoiding the power wall with low-leakage, STT-MRAM based computing," Proc. of *the 37th ISCA*, pp. 371-382, Jun. 2010.

[15] T. Sugibayashi et al., "A 16-Mb toggle MRAM with burst modes," *IEEE J. Solid-State Circuits*, vol. 42, no. 11, pp. 2378-2385, Nov. 2007.

[16] T. Ohsawa et al., "1Mb 4T-2MTJ nonvolatile STT-RAM for embedded memories using 32b fine-grained power gating technique with 1.0ns/200ps wake-up/power-off times," *Symp. VLSI Tech. Dig. Tech. Papers*, pp. 46-47, Jun. 2012.

[17] P. Zhou, B. Zhao, J. Yang, and Y. Zhang, "Energy reduction for STT-RAM using early write termination," *ICCAD Dig. Tech. Papers*, pp. 2-5, Dec. 2009.

[18] G. Sun, X. Dong, Y. Xie, J. Li, and Y. Chen, "A novel architecture of the 3D stacked MRAM L2 cache for CMPs," Proc. of *the 15th HPCA*, pp. 239-249, Feb. 2009.

[19] G. Sun, Y. Zhang, Y. Wang, and Y. Chen, "Improving energy efficiency of write-asymmetric memories by log style write," Proc. of *the ISLPED*, pp. 173-178, Jul. 2012.

[20] C. W. Smullen, V. Mohan, A. Nigam, S. Gurumurthi, and M. R Stan, "Relaxing non-volatility for fast and energy-efficient STT-RAM caches," Proc. of *the 17th HPCA*, pp. 12-16, Feb. 2011.

[21] H. Li et al., "Performance, power, and reliability tradeoffs of STT-RAM cell subject to architecture-level requirement," *IEEE Trans. on Magnetics*, vol. 47, no. 10, pp. 2356-2359, Oct. 2011.

[22] Z. Sun et al., "Multi retention level STT-RAM cache designs with a dynamic refresh scheme," Proc. of *the 44th MICRO*, pp. 329-338, Dec. 2011.

[23] A. Jog et al., "Cache revive: architecting volatile STT-RAM caches for enhanced performance in CMPs," Proc. of *the DAC*, pp. 243-252, Jun. 2012.

[24] X. Wu, J. Li, L. Zhang, E. Speight, R. Rajamony, and Y. Xie, "Hybrid cache architecture with disparate memory technologies," Proc. of *the 36th ISCA*, pp. 34-45, Jun. 2009.

[25] A. Jadidi, M. Arjomand, and H. Sarbazi-Azad, "High-endurance and performance-efficient design of hybrid cache architectures through adaptive line replacement," Proc. of *the ISLPED*, pp. 79-84, Aug. 2011.

[26] K. Abe et al., "Novel hybrid DRAM/MRAM design for reducing power of high performance mobile CPU," *IEDM Tech. Dig.*, pp. 243-246, Dec. 2012.

[27] E. Kitagawa et al., "Impact of ultra low power and fast write operation of advance perpendicular MTJ on power reduction for high-performance mobile CPU," *IEDM Tech. Dig.*, pp. 677-680, Dec. 2012.

[28] K. Kawasaki, T. Shiota, K. Nakayama, and A. Inoue, "A Sub-μs wake-up time power gating technique with bypass power line for rush current support," *IEEE J. Solid-State Circuits*, vol. 44, no. 4, pp. 1178-1183, Apr. 2009.

[29] SPEC CPU2006 benchmark suite: http://www.spec.org/cpu2006.

[30] A. Gutierrez et al., "Full-system analysis and characterization of interactive smartphone applications," Proc. of *the IEEE Int'l Symp. Workload Characterization (IISWC)*, pp. 81-90, Nov. 2011.

[31] N. Binkert et al., "The gem5 simulator," *ACM SIGARCH Computer Architecture News (CAN)*, vol. 39, no. 2, pp.1-7, May 2011.