Improving Energy Efficiency of Write-asymmetric Memories by Log Style Write

Guangyu Sun[1], Yaojun Zhang[2], Yu Wang[3], Yiran Chen[2]
[1]Center for Energy-efficient Computing and Applications, Peking University
[2]Department of Electrical and Computer Engineering, University of Pittsburgh
[3]Department of Electronic Engineering, Tsinghua University
gsun@pku.edu.cn, yu-wang@mail.tsinghua.edu.cn, {yaz24, yic52}@pitt.edu

ABSTRACT

The significant scaling challenges of conventional memories, i.e., SRAM and DRAM, motivated the research on emerging memory technologies. Many promising memory technology candidates, however, suffer from a common issue in their write operations: the switching processes at different write operations (i.e., $0 \rightarrow 1$ and $1 \rightarrow 0$) are asymmetric. Using a pessimistic design corner to cover the worst case of a write operation incurs large power and performance cost in the existing emerging memory technology designs. In this work, we propose a universal log style write methodology to mitigate this asymmetry issue by operating two switching processes in separate stages. The dedicated design optimizations are allowed on either switching process. The simulation results on the spin-transfer-torque random access memory based last-level cache show that our technique can improve the system performance by 4% while receiving 35% power reduction on average [1].

Categories and Subject Descriptors

B.3.0 [**MEMORY STRUCTURES**]: General

Keywords

Write-asymmetric memory, log write, energy efficiency

1. INTRODUCTION

In order to overcome the memory wall, extensive research has been done to find alternatives of the traditional SRAM and DRAM technologies. Various emerging memory technologies, such as STT-RAM (Spin Transfer Torque RAM), PCM (Phase Change Memory), FBDRAM (Floating Body DRAM), have been proposed to replace SRAM/DRAM in different levels of the memory hierarchy [8, 5, 13, 7, 3]. STT-RAM is normally employed as a competitive replacement of SRAM as on-chip memories because of its advantages of fast read speed, low leakage power, and high density [8, 5]. PCM is widely studied as a potential candidate of the main memory because it has higher density and lower standby power compared to DRAM [13, 7, 3]. As an emerging memory compatible to CMOS technology, FBDRAM is also attracting more attention, recently [4, 1]. With these emerging memory technologies, prior research has shown improvements in performance, power consumption, and reliability.

[1]This work was partially supported by National Natural Science Foundation of China (No.60870001, 61028006).

Table 1: Comparisons of data switching (45nm) .

	STT-RAM		PCM		FBDRAM	
Cell Area	$40F^2$		$16F^2$		$8F^2$	
Switching	$0 \rightarrow 1$	$1 \rightarrow 0$	$0 \rightarrow 1$	$1 \rightarrow 0$	$0 \rightarrow 1$	$1 \rightarrow 0$
Latency	$30ns$	$10ns$	$150ns$	$40ns$	$10ns$	$2ns$
Current	$250\mu A$	$200\mu A$	$300\mu A$	$600\mu A$	$30\mu A$	$30\mu A$

These emerging memories, however, have some common limitations such as long write latency, and high write energy. Although the memory capacity is increased after using these high density memories, the long write latency may offset the benefits and degrade the performance for workloads with intensive write operations. Similarly, although the standby power is reduced by using these emerging memories, the dynamic power can be significantly increased due to high write energy. Some research has been done to hide the long write latency by using hybrid memory [9] or stalling the write and preempting the blocked read [9, 6]. The replacement policy can also be tailored to reduce the write intensity [14]. Recently, Smullen and Li *et al* proposed to scarify the non-volatility to improve the write speed [8, 10].

In most of these techniques, the write operations of these memories are designed for the worst case of the asymmetric data switching processes. The asymmetry means that the switching processes of $0 \rightarrow 1$ (SET) and $1 \rightarrow 0$ (RESET) are quite different from each other, in respect of timing, energy consumption, and even reliability. Table 1 compares the differences between SET and RESET for some typical values of various memory technologies. It is easy to find that a RESET is much faster and consumes much less energy, compared to a SET. In the worst case design, however, the write latency and driving current are designed to satisfy the constraints of the SET. Thus, the RESET is normally overdriven, and much overhead is induced. As the technology scales down, this problem is becoming more severe because the increasing effect of process variations further aggravate the difference between a SET and a RESET. Smullen addressed this issue in a STT-RAM cache by flipping the data to have more RESET in a write operation. This scheme, however, only reduce write energy. Since RESET and SET are still operated simultaneously, the write latency cannot be reduced and the design constraints are not changed.

In this work, we propose a log style write methodology, in which the SET and RESET are processed separately in two stages. This universal methodology is feasible for most write-asymmetric memories designed for various levels of the memory hierarchy. With this methodology, the memory to be written is prepared by a recycle stage, in which only SETs are processed. Then, there are only RESETs when new data are written into the memory. Since SETs and RESETs are separated, they are processed with their own

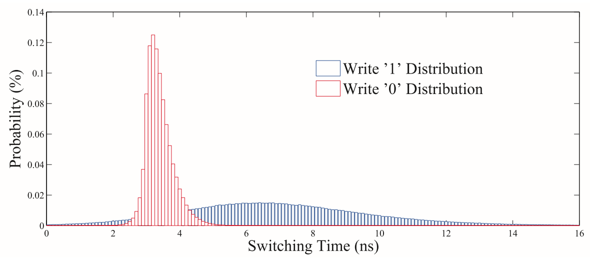

Figure 1: MTJ switching time distributions at the transistor width of 720nm.

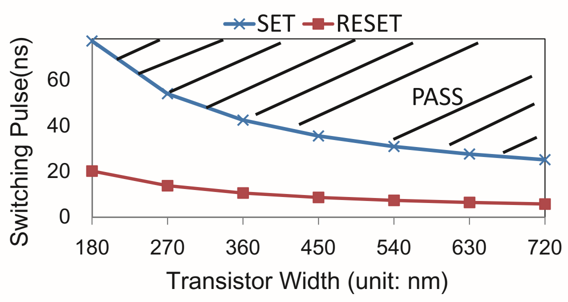

Figure 2: Design space based on transistor width.

constraints. Moreover, more design options and optimization opportunities are provided. We use STT-RAM based cache as a case study of applying this log style write. After using this methodology, we will show that energy efficiency of STT-RAM cache is greatly improved by reducing power consumption and improving performance at the same time. In addition, we also discuss how to extend the methodology for other write-asymmetric memories and compare it with read-before-write scheme.

2. PRELIMINARIES

In this work, we use STT-RAM as an example to study the write asymmetry and its impact. The STT-RAM is chosen because of significant differences between its SET and RESET. In addition, STT-RAM is normally proposed as on-chip memory, which is normally sensitive to long write latency and high write energy.

2.1 Basic of STT-RAM and Its Asymmetric Write

STT-RAM uses magnetic tunneling junction (MTJ) devices to store the information. A MTJ has two ferromagnetic layers (FL) and one oxide barrier layer (BL). The resistance of MTJ depends on the relative magnetization directions (MDs) of the two FLs. When their MDs are parallel or anti-parallel, the MTJ is in its low (bit '0') or high resistance state (bit '1'). R_h and R_l are usually used to denote the high and the low MTJ resistance, respectively. In a MTJ, the MD of one FL (reference layer) is pinned while the one of the other FL (free layer) can be flipped by applying a polarized write current though the MTJ. In the popular "1T1J" (one-transistor-one-MTJ) cell structure, the MTJ write current is supplied by a NMOS transistor. A larger write current can shorten the MTJ switching time by paying the additional memory cell area and increasing the breakdown possibility of the MTJ device.

The asymmetry of STT-RAM's write operations mainly comes from two sources: the asymmetric MTJ switching property and the NMOS transistor driving ability for two switching directions. The asymmetry of MTJ switching property in SET and RESET is mainly due to the different spin-transfer efficiency at the both sides of the oxide barrier. The driving ability of NMOS transistor is also asymmetric due to the different bias conditions at two STT-RAM cell switching directions. In addition, the different resistance in SET and RESET also affect the switching processes[12].

When process variations are considered, the asymmetry becomes more significant between SET and RESET operations because the current through the MTJ is affected by device variations of the MTJ and the NMOS transistor. For instance, the driving ability of the NMOS transistor is affected by the variations of transistor channel length (L)/width (W) and the threshold voltage (V_{th}). The MTJ resistance R_{MTJ}, which determines the voltage drop across the MTJ device, is reversely proportional to the MTJ surface area and exponential to the oxide layer thickness. The ïňĆuctuations of the above device parameters in the chip fabrication introduce bit-to-bit variations of STT-RAM cells [11]. Due to different switching properties and driving abilities, MTJ switching time has a larger mean and a wider distribution in SET than those in RESET under the same switching current[12], as illustrated in Figure 1. Thus, more overhead is induced in performance and power for the worst case design with process variations.

2.2 Impact of Asymmetry in Worst Case Design

In order to study the impact of asymmetry in write operations, we present the design space of SET and RESET operations with process variations and thermal fluctuations in Figure 2. The detailed device parameters of STT-RAM and corresponding process variations are listed in Table 2 and Table 3, respectively. A corner design methodology is usually used to find the boundary of design space with the consideration of device variations. The design corner can be setup as the combinations of device parameters.

Figure 2 shows that, for the same size of access transistor, RESET can be finished much faster than SET. At the same time, the energy consumption of RESET is much less than that of SET. With the worst case design methodology, however, the switching of RESET bits are over-driven for the same latency as the SET bits. Obviously, if there are only RESET in write operations, the performance can be improved with lower latency; the energy consumption can be reduced with lower driving current and less switching time; even the reliability can be improved because the device breakdown possibility is reduced with lower switching current.

3. LOG STYLE WRITE METHODOLOGY

Operating SET and RESET processes separately can improve energy efficiency of write asymmetric memories. Thus, we propose a log style write methodology to separate SET and RESET in two stages. In this section, we introduce this methodology by applying it to a STT-RAM last level cache (LCC). The discussion of applying this method to other memory level and technologies is presented in Section 4.

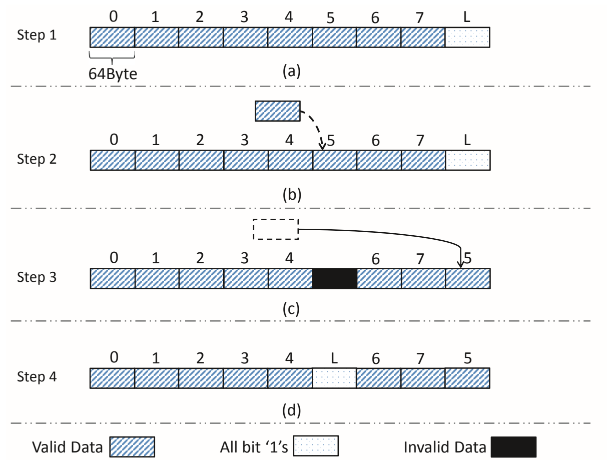

Figure 3: Illustration of the log style write operations.

3.1 Methodology Overview

Figure 3 illustrates the structure view of cache lines in an eight-associativity STT-RAM LLC, which supports log style write operations. The cache line size is set to 64Byte, which is commonly adopted in modern processors. In order to simplify the illustration, we assume that LLC cache line size is equal to that of its upper level cache. With this assumption, the data written back from the upper level cache and the data loaded from the main memory have the same size (64Byte). Note that the log style write also supports variable data lengths. The related issues are also discussed later in Section 4.

Compared to a traditional cache line size, the key difference is that an extra cache line (cache way) is added as the "log way". Thus, the length of one index of cache lines is increased to 576Byte, and there are nine cache lines in total. The eight cache ways storing valid data are labeled with number $0 \sim 7$ in the figure. The log cache way is labeled with L to be differentiated from the others. The cache lines are operated as follows to achieve the log style write:

1. Figure 3 (a) shows initial states of the cache lines. All bits in the log cache way are initialized as bit '1's.
2. In Figure 3 (b), there is an incoming write operation, which will update the $5th$ cache way.
3. In Figure 3 (c), the update data are stored in the log cache way instead of the original position ($5th$ cache way). Since there are only bit '1's in the log cache way, only bit '0's of the incoming data are updated. Thus, there are only RESETs in the write operation. At the same time, the old data in the original $5th$ sector are invalidated.
4. In Figure 3 (d), after the write operation, all bits in the invalidated cache way are programmed to bit '1's in a recycle stage so that the new log cache way is ready for the next write operation.

With the modified cache lines, there are only RESETs in the write operation so that both write latency and energy consumption can be greatly reduced. We call this methodology log style write because the data in each write operation is updated *out of its original place.* This methodology is similar to the log based file system used in NAND flash based SSD. The fundamental difference is that the sector holding invalid data in a SSD cannot be recycled immediately. For the STT-RAM LLC with log style write, the log cache way is recycled as soon as possible after each write operation.

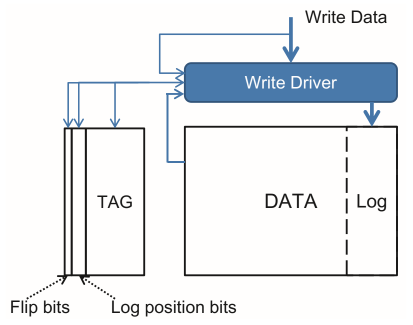

Figure 4: Modified control logic and data path.

A recycle stage is different from a normal write operation. There are no real data written so that the data bus is not occupied and only the local write driver in the sub-array is used. Thus, the access to other sub-arrays will not be blocked by the recycle operation. Theoretically, a read request to other cache lines of the same sub-array can also be operated at the same time in parallel by modifying the control logic. The experiment results (Figure 7) show that the latency of a recycle stage can be well hidden during idle time for most cases. Therefore, the circuitry modification in this work will not support simultaneous read and write requests in the same sub-array. The other design details and corresponding optimization techniques are introduced in the next subsection.

3.2 Design Details and Optimization Techniques

First, we need to decide how to modify cache lines to support the log cache way. As illustrated in Figure 3, we add one extra cache line per set. Thus, the "effective cache associativity", which means the maximum number of cache lines holding valid data in each set, is kept the same as the original cache design. This scheme, however, induces area overhead by adding extra cache lines and increases design complexity of locating cache lines. Alternatively, we can keep the original cache design but sacrifice one cache line per set as the log cache way. It means that the "effective cache associativity" is reduced by one to enable the log style write. In this work, we focus on the former scheme because using log write style can help reduce the cell size so that the area overhead can be offset. This issue will be discussed later in subsection 3.4.

Second, the tag structure of LLC needs to be modified for the log style write. Figure 4 shows corresponding control logic and data path. It should be mentioned that the tag part of the STT-RAM is normally designed with CMOS technology. It is because the STT-RAM based tag has the similar density as the CMOS tag. More important, some status bits of the cache tag (e.g. LRU bits) are updated in read operations. The overhead of long updating latency is too large for read operations. We need to add extra bits for each index of cache to represent the position of the log cache way. For the example of STT-RAM LLC, four bits are needed for each cache set to trace the position of log cache way among nine cache ways shown in Figure 3.

For each write operation, the data are always written to the log cache way, but the cache tag updating and the cache way selected for the following recycle stage are different. For the write hit from the upper cache level, the address bits in the tag of the original cache line is copied to the tag of the log cache way. At the same time, the status bits of the tag are updated in the log cache way, according to the cache management policy. For example, if LRU is used in the LLC of Figure 3, the tag of the last cache line is set to the most recently used one in Figure 3(c). In addition, the fifth cache line is invalidated. For the data write miss from the upper cache level, the tag of the log cache way is updated as in a traditional cache design. At the same time, a cache way (e.g. LRU cache way) is evicted and selected as the log cache way after a recycle stage. Note that this case happens in exclusive and non-inclusive caches. The process is similar for loading data from the main memory. In such a case, a cache line is evicted and erased as the log cache way while LLC is waiting for the data from main memory.

Third, extra control logic is required to operate SET and RESET in two stages efficiently. In order to avoid redundant SETs in the recycle stage, only the bit '0's in the cache line are switched. Thus, the original data need to be first read out and a transistor is needed to control the bit-line driver. Such control logic used for read-before-write has already been proposed in prior research [13]. The drawback of this technique is that the latency of the recycle stage is increased. Fortunately, the recycle stage is not on the critical timing path (shown in Figure 7), and the extra latency for data sensing (read) is trivial compared to the SET latency. For the similar reason, only bit '0's in the incoming data are written to the log sector with RESETs in the write operation. The control logic added in recycle stage can also be used in the write operation to avoid redundant RESETs. Thus, an extra multiplexer is needed to select control signal in different steps.

Forth, optimization technique is needed to reduce energy consumption. With the log style write operation, we can find that the total number of bits switched in two stages are different from that in a traditional write operation. The total write energy consumption in two cases are calculated and compared as follows. Let N_0 and N_1 represent the number of bit '0's and bit '1's in write data, respectively. Let E_0, E_1 and E'_0, E'_1 represent the RESET and SET energy consumption in a traditional write operation and a log style write operation. The total energy consumption of a traditional write operation is calculated in Equation 1. The total energy consumption in a log style write operation is listed in Equation 2.

$$Energy_{trad} = N_0 \times E_0 + N_1 \times E_1 \tag{1}$$

$$Energy_{log} = N_0 \times E'_0 + N_0 \times E'_1 \tag{2}$$

The Equation 2 is obtained from the fact that any sector will always be recycled in the future. Although we have $E'_1 < E_1$, we cannot promise that $Energy_{log}$ is always less than $Energy_{trad}$. In fact, N_0 in the write data can be much higher than N_1 for many workloads.

The "flip-N-write" [5] technique is employed to solve this problem efficiently. Before the RESET operated in a write operation, the numbers of bit '0's and bit '1's in write data are compared. If there are more bit '0's than bit '1's, all bits in data are flipped before being written into the log sector. An extra "flip-bit" is required for each sector to record whether the data has been flipped before being stored. In fact, we can gain more benefits with this technique in the log style write than in a traditional write operation. With this technique, the total energy consumption of the log style write operation is calculated as in Equation 3. We can find that the total number of switched bits can be much less than the total data length. The energy consumption is always reduced, compared to the traditional write.

$$Energy_{log} = min(N_0, N_1) \times (E'_0 + E'_1) + 1 \tag{3}$$

3.3 Overhead Analysis

First, there is no extra timing overhead. For each cache access, the position bits for the log cache way are always accessed together with the cache tag. For a write "miss" or a replacement, the extra bits added are updated in parallel with the original bits in a traditional tag . For a write hit, the cache tag is updated simultaneously with the data part. Thus, there is no extra timing overhead for the tag access. For the data access in a read operation, the data may be flipped before being read out. The latency of data flipping is negligible, compared to the total read latency of LLC [5]. Obviously, the write latency is always reduced.

Second, extra bits are accessed in read/write operation, but the total energy consumption is still reduced. In a read operation, an extra flip bit is accessed and the extra tag for the log cache way need to be read/compared. In a write operation, the position bits of the log cache way are updated and the extra flip bit may be updated. Since both the number of total switching bits and the RESET energy consumption are reduced in the write operation, the total energy consumption is reduced. This is proved by experimental results in Section 5.

Third, extra overhead is induced in the area. Most overhead comes from the extra storage space for the log cache way. The extra log cache way induces 12.5% storage overhead for the example of STT-RAM LLC. Besides, extra control logic and bits in the tag also induce space overhead, which is trivial compared to that of the log cache way. Thus, the area overhead is mainly decided by the capacity used for the log. Fortunately, the associativity of modern LLC is normally large (e.g. 16 or 32). It means that the area overhead is moderate. In next subsection, we will show that such overhead can be further compensated.

3.4 Design Trade-off with Log Style Write

With the extra log way, area overhead is induced to reduce write latency and energy consumption. In fact, when SET processes are moved into the recycle stage, we have more opportunities in the new design space for trade-off between capacity and performance/energy. As shown in Figure 2, RESET is much less sensitive to the size of the driving transistor, compared to SET. For example, the RESET latency is only increased a little when the transistor width is decreased from 720nm to 540nm. For memory like STT-RAM, the design area is closely related to the size of the driving transistor. As shown in Figure 5, the area of a 16M STT-RAM LLC can be reduced by about 25%, when the width of the driving transistor is reduced from 720nm to 450nm. However, the SET latency is increased by about 10ns with the smaller transistor. Since the latency of SET is not an critical issue, we can reduce the size of the driving transistor to compensate the log capacity without inducing much degradation in performance.

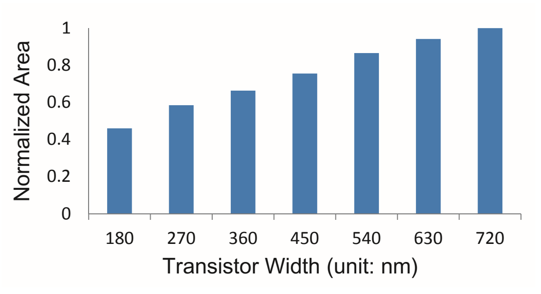

Figure 5: LLC area for different transistor widths ($16MB$ $45nm$).

Another opportunity comes from the error correction mechanisms. The total number of switching bits in each stage is equal to $min(E_0', E_1')$, which is less than half of that in the traditional caches. Thus, we can use weaker error correction mechanisms for the same error rates. with a smaller driving current in the RESET operation. The study about this issue is not included due to page limitation.

4. LOG WRITE FOR OTHER MEMORIES

The write style is not just limited to the LLC. For other levels of the cache hierarchy, one important issue is that the write data length may not be fixed. For example, if we want to apply the method to an L2 cache, which works with a write-through L1 cache. The data written through from L1 cache can be less than the size of a cache line, but a whole cache line is loaded from the next memory level. In such scenarios, the log record need to be set to the minimum granularity of the write operation, and the individual log record may be kept for each cache way. Thus, more storage space is needed to trace the log. For a L2 cache used in a 64-bit machine, the cache tag can be increased by about 40%. Fortunately, the cache tag only consumes a small part of the total area (e.g. less than 10%). It is still worth using the methodology because the L2 cache is normally more sensitive to the long write latency of SET.

The methodology is also feasible for other write-asymmetric emerging memories such as FBDRAM. For the capacity based memory such as FBDRAM, the different write voltage is applied the cell in different stages. It should be mentioned that such log write style has an impact on lifetime of the memory like PCM. On one side, the log style help spread the write intensity to different memory space. On the other side, the two-stage write may switch more bits, compared to the read-before-write scheme [13].

Although both log write style and read-before-write schemes can help reduce energy consumption, they cannot work with each other at the same time. However, we believe that these two schemes are used in different scenarios. For PCM, which is normally employed as off-chip memory, a read-before-write scheme is more attractive because the lifetime limitation of PCM makes it more important to reduce write intensity. For STT-RAM and FBDRAM, which are used as fast on-chip memory and have no lifetime issues, it is reasonable to leverage our log write style because read-before-write cannot reduce write latency but increases it instead.

5. EVALUATION

In this section, we evaluate a STT-RAM LLC with log style write in a chip-multiprocessors (CMP) system and compare the results to those of using a traditional one.

Table 2: Device Parameters

Device	Parameters	Mean	Std. Dev
NMOS	Channel Length L	$45nm$	$2.25nm$
	Channel Width W	design dependent	$2.25nm$
	Threshold Voltage V_{th}	$0.466V$	$30mV$
MTJ	MgO Thickness τ	$2.2nm$	2% of mean
	Cross Section A	$45 \times 90nm^2$	5% of mean

Table 3: $3sigma$ corner design parameters.

W	L	V_{th}	τ	A
-15%	$+15\%$	$+15\%$	$+6\%$	$\pm15\%$

5.1 Configuration Setup

For system level simulation, we use SIMICS simulator to evaluate performance. It is configured to model an eight-core processor. Each core is Ultra-SPARC-like with a 2GHz frequency. There are three levels of caches. The IL1/DL1 and L2 caches are SRAM based and the capacities are set to 64KB and 2MB, respectively. The baseline LLC (L3) is a 16-way 16MB STT-RAM cache. The simulator captures data addresses from all loads, stores, and prefetch operations. We use the information to calculate the memory access intensity, and use that to compute the energy consumption of the cache hierarchy. Our workloads are sets of multi-threaded benchmarks from SpecOMP and PARSEC. We choose benchmarks from the full set and mix them together to help us create a diverse set of intensive cache access patterns. The WPKIs (write per thousand instructions) of these benchmarks range from 1.8 to 9.8.

In order to estimate the access latency and energy consumption of different caches, we extend the widely used tool CACTI [2] to support STT-RAM technology with process variations considered. The parameters used in this work are listed in Table 2 and Table 3. We choose the latency as the value at 3σ of the distribution for both read and write (SET/RESET in STT-RAM) operations. The read/write latency is set to 1ns and 5ns for L1 and L2, respectively. For the traditional STT-RAM LLC, the read latency is 10ns, and the write latency is set to 35ns. As we discussed Section 3.4, we have different design options after using the log style write. In our experiments, we reduce the cell width to compensate the capacity overhead of an extra log cache way, which is about 6.5%. The SECDEC is employed as the error correction mechanisms for both STT-RAM caches, since the ECC optimization is not the focus of this work. Thus, for the STT-RAM with log write style, the read latency is still 10ns and the RESET and SET latency is 15ns and 38ns.

5.2 Evaluation Results

Figure 6 compares the normalized execution time for three different STT-RAM LLCs. The first set of baseline result is for the traditional STT-RAM cache design without using any write optimization techniques. The second result is for the STT-RAM cache using the write-halt technique [6], in order to provide a comprehensive comparison. The third set of result is for the STT-RAM cache using log style write operation. We can find that, after using the log style write, the performance is improved. In addition, using log style achieve even better performance than that of using write-halt technique in prior research. Compared to the baseline case, the performance is improved by about 4% on average after using the log style write. The performance improvement is not significant because the STT-RAM LLC is the

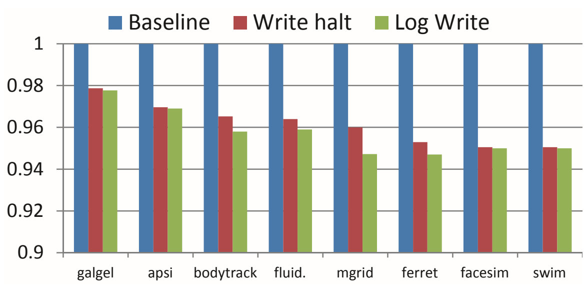

Figure 6: Normalized Execution Time.

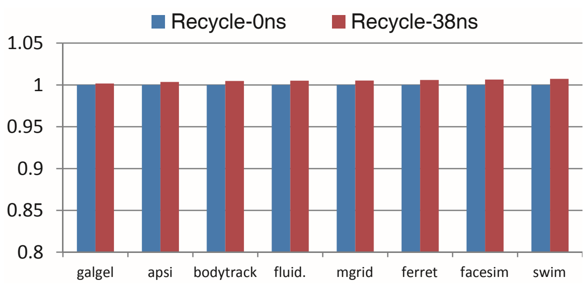

Figure 7: Normalized execution time with different recycle latency.

L3 cache, which is not that sensitive to write latency. We can expect more benefits when applying the log style write methodology in other memory levels with higher write intensity.

In order to analyze the impact of recycle stage on performance, we compare the results of two specific cases shown in Figure 7. The first set of results show normalized IPCs when we set the latency of recycle as zero. It represents a non-real case when recycle causes no overhead. The second set of results show IPCs of the real case in this work with the latency of recycle set to 38ns. The comparison shows that the performance is reduced by less than 1% with the real latency of recycle. It means that the latency of recycle stage can be hidden well and has little impact on performance.

The power consumption results are shown in Figure 8. After using write-halt technique, the total power consumption is increased. It is because the write operation can be canceled and by read operation in such a cache and will be re-written later. Due to the limitation of experimental environment, it is impossible to accurately estimate the energy consumption of an incomplete write. Thus, we assume that an incomplete write operation consumes $0\% \sim 75\%$ energy consumption of a complete write, and we randomly choose a value in this range for each write operation. Figure 8 also shows that the power consumption is greatly reduced after using the log style write operation. First, it is because the RESET energy consumption is greatly reduced. Second, the optimization using flip bit make sure that the total number of write bits are reduced. Thus, the ratios of bit '0' and bit '1' in write data also has an impact on the total energy consumption. For example, the power reduction for workload "facesim" is higher than the others because there are much more bit '0's in the write data. On average, the power consumption can be reduced by about 35% with the log style write.

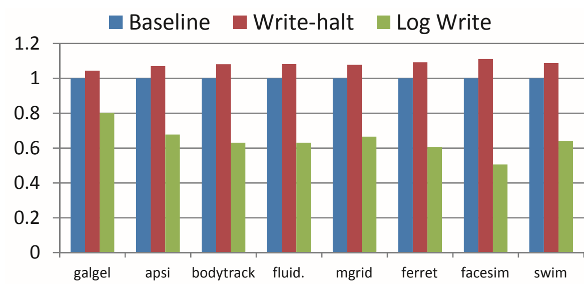

Figure 8: Normalized Power Consumption

6. CONCLUSION

Asymmetric write is a common character for most emerging memory technologies. When both SET and RESET exist in a write operation, the worst case design has to be used. Such a design scheme induces performance, power, and reliability overhead, which is further aggravated by the affect of process variations. In this work, we separate SET from RESET by using a log style write operation. With this methodology, the SET and RESET are optimized and processed individually. Moreover, separating SET and RESET provides more design options and optimization opportunities. After using the log style write, performance is improved and power consumption is reduced for the write-asymmetric memory design.

7. REFERENCES

[1] J. Kim, S. Chung, T. Jang, and *et al.* Vertical double gate Z-RAM technology with remarkable low voltage operation for DRAM application. pages 163–164, 2010.

[2] H. Labs. In *http://www.hpl.hp.com/research/cacti/*, .

[3] B. C. Lee, E. Ipek, O. Mutlu, and D. Burger. Architecting phase change memory as a scalable DRAM alternative. In *Proceedings of ISCA*, pages 2–13, 2009.

[4] Z. Lu, N. Collaert, M. Aoulaiche, B. De Wachter, A. De Keersgieter, W. Schwarzenbach, and *et al.* A novel low-voltage biasing scheme for double gate fbc achieving 5s retention and 10_{16} endurance at 85c. In *IEDM*, pages 12.3.1 –12.3.4, dec. 2010.

[5] A. Nigam, C. Smullen, V. Mohan, E. Chen, S. Gurumurthi, and M. Stan. Delivering on the promise of universal memory for spin-transfer torque ram (stt-ram). In *ISLPED 2011*, pages 121 –126, aug. 2011.

[6] M. Qureshi, M. Franceschini, and L. Lastras-Montano. Improving read performance of phase change memories via write cancellation and write pausing. In *HPCA*, pages 1 –11, jan. 2010.

[7] M. K. Qureshi, V. Srinivasan, and J. A. Rivers. Scalable high performance main memory system using phase-change memory technology. In *Proceedings of ISCA*, pages 24–33, 2009.

[8] C. Smullen, V. Mohan, A. Nigam, S. Gurumurthi, and M. Stan. Relaxing non-volatility for fast and energy-efficient stt-ram caches. In *High Performance Computer Architecture (HPCA), 2011 IEEE 17th International Symposium on*, pages 50 –61, feb. 2011.

[9] G. Sun, X. Dong, Y. Xie, J. Li, and Y. Chen. A novel architecture of the 3d stacked mram l2 cache for cmps. pages 239 –249, feb. 2009.

[10] Z. Sun, X. Bi, H. H. Li, W.-F. Wong, Z.-L. Ong, X. Zhu, and W. Wu. Multi retention level stt-ram cache designs with a dynamic refresh scheme. In *Proceedings of Micro*, pages 329–338, 2011.

[11] X. Wang, Y. Zheng, H. Xi, and D. Dimitrov. Thermal fluctuation effects on spin torque induced switching: Mean and variations. *Journal of Applied Physics*, 103(3):034507 –034507–4, feb 2008.

[12] Y. Zhang, Y. Li, A. Jones, and Y. Chen. Asymmetry of MTJ Switching and Its Implication to STT-RAM Designs. In *Proceedings of the DATE*, 2012.

[13] P. Zhou, B. Zhao, J. Yang, and Y. Zhang. A durable and energy efficient main memory using phase change memory technology. In *Proceedings of ISCA*, pages 14–23, 2009.

[14] P. Zhou, B. Zhao, J. Yang, and Y. Zhang. Energy reduction for stt-ram using early write termination. In *ICCAD*, pages 264 –268, nov. 2009.