Optimizing Energy in a DRAM based Hybrid Cache

Jiacong He
The University of Texas at Dallas
Email: jiacong.he@utdallas.edu

Joseph Callenes-Sloan
California Polytechnic State University
Email: jcallene@calpoly.edu

***Abstract*—The die-stacking DRAM cache can be used to increase bandwidth and reduce latency compared with conventional DRAM memory. However, energy becomes an inevitable challenge with the increasing size of DRAM cache. STT-RAM with near-zero leakage can be integrated with DRAM cache as a hybrid cache to reduce static energy, but the high write energy of STT-RAM brings another energy challenge. In this paper, we propose a tri-regional hybrid cache that can exploit the advantage of both DRAM and STT-RAM technologies. The asymmetric data access policy is introduced based on the non-uniform read/write property of the different hybrid cache regions. We also propose a prediction table that can reduce the searching energy of the hybrid cache. The results show that our hybrid cache reduces energy by 26% and improves performance by 11% on average compared with previous work.**

***Index Terms*—Hybrid cache, STT-RAM, DRAM cache, Energy, Performance.**

I. Introduction

To handle the memory wall problem and satisfy the high processing speed of the multicore processors, there is significant demand for a large cache capacity in future. However, the low density of SRAM hinders the improvement of cache capacity in the modern computing system, which can occupy up to half of total die area. Recently, emerging 3D-stacked DRAM cache has been used a large last-level cache, which can provide gigabytes of storage to break the memory wall.

DRAM cache consists of multiple layers of DRAM stacked on processors using Through-Silicon Via (TSV). Previous researchers proposed many techniques to optimize DRAM cache design, such as bandwidth usage improvements, large metadata management, latency and performance optimization. However, due to the process scaling and the incremental capacity of on-die DRAM cache, static energy contributes a large proportion of the total cache energy. Also, 3D IC design has more challenge in power and thermal management than its 2D counterpart because multiple stacking layers result in higher power densities. And previous work has not shown significant progress in handling this problem.

Emerging STT-RAM with minimal leakage and high density is a potential candidate to substitute for the DRAM cache. However, its high write energy and latency hinder its complete replacement of a large cache. To fully utilize the advantage of both emerging and traditional memory technology, STT-RAM is commonly used in a hybrid cache. For example, recent works in [3], [14], [15] leveraged non-volatile STT-RAM to build a hybrid cache with SRAM. There are challenges for fabricating a hybrid cache in the conventional 2D design, while the 3D stacking can overcome the fabrication difficulty by integrating different wafers with different memory technologies.

Although a hybrid cache with non-volatile STT-RAM can effectively reduce the static energy, extra write energy consumed by the STT-RAM usually needs further optimization, which often causes area overhead and design complexity. Recent research [11] has shown that the data retention time of STT-RAM could be relaxed to reduce its high write energy by shrinking the planar area of the MTJ or decreasing the thickness of the free layer. To solve the energy dilemma problem of conventional hybrid cache, we observe that volatile STT-RAM can be utilized in the hybrid cache as a buffer to balance the high static energy of DRAM and the high dynamic energy of non-volatile STT-RAM.

In this paper, we propose a last-level hybrid cache, which consists of three regions including a DRAM region, a Non-volatile STT-RAM (NSR) region and a Volatile STT-RAM (VSR) region. The DRAM region with high endurance is used as the main component of the hybrid cache. The NSR region with little leakage and VSR region with small write energy are used to reduce the static and dynamic energy of the hybrid cache respectively. Also, we observe the non-uniform read/write property of the tri-regional hybrid cache. Thus, we optimize the read/write orders of the different regions in the hybrid cache to achieve energy-efficient data accesses. Further, we observe that the data request needs to traverse the whole hybrid cache to find the required data. To reduce the searching energy and tag comparison latency in the different regions of the large cache, we propose a prediction table that can find out the data in the specific region directly, or bypass the whole hybrid cache by detecting a miss in advance.

Overall, our contributions are as follows:

- We reduce both the static and dynamic energy of a large hybrid cache by using volatile STT-RAM as a buffer region between non-volatile STT-RAM and DRAM regions.
- We avoid high static energy of DRAM region and high dynamic energy of STT-RAM region by optimizing the read and write data accessing orders.
- We reduce searching energy for every data access in the hybrid cache using a prediction table.
- We evaluate our design with SPEC workloads, and the results show that energy is reduced by 26% and performance is improved by 11% on average.

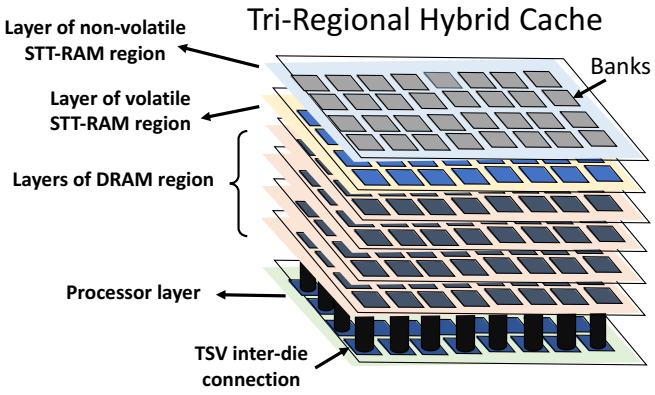

Fig. 1: The die-stacked tri-regional hybrid cache with the processor layer.

II. Tri-regional Hybrid Cache

A. Architecture Overview

We propose an energy-efficient hybrid cache consisting of three regions including DRAM region, volatile STT-RAM region and non-volatile STT-RAM region. The CPU cores with the proposed hybrid cache consist of four different regions stacked upon each other as illustrated in the Fig. 1. The lowest region is the region of processor die, which contains 8 cores and each core has private L1 SRAM cache and shared L2 SRAM cache. The second region is composed of 128MB DRAM cache region, which is divided into four layers with 32MB per layer and every layer is divided into banks. Because the density of STT-RAM is 1x-2x of the DRAM with the same die area. Thus, the volatile STT-RAM region and non-volatile STT-RAM region are divided into one 64MB layer. The layers from the second to the fourth region constitute the tri-regional hybrid cache, and every layer of the hybrid cache is stacked upon the processor layer. The data is transmitted between on-die hybrid cache and processor cores through a TSV inter-die connection.

Every layer of the hybrid cache is stacked upon the processor layer. There are 32 banks in every hybrid cache layer, and each bank is connected through a mesh-based Network-on-Chip (NoC). The communication among each layer is through the TSV vertical interconnection, which effectively reduces the 2D planar wire length and data transmission latency compared with conventional 2D cache structure. Whats more, there is one TSV connection to every layer for each core based on the proposed hybrid cache architecture. The die with cores is estimated to have 60mm^2 area, and the dies of tri-regional hybrid cache have an average area of 30mm^2 each based on the estimates from DESTINY [9] at 32nm technology.

B. Volatile STT-RAM

Based on the non-uniform property of the tri-regional hybrid cache, the refresh rate of the volatile STT-RAM region should be the same or closer to that of DRAM region, so as to reduce the complexity of hybrid cache refresh circuit. We observed that the DRAM cache uses fast logic transistors which have higher leakage and higher refresh requirement than the off-chip DRAM. Also, prior work [2] has observed that the retention time of a DRAM cell strongly depends on the threshold voltage V_{th} of its access transistor, which means there is variation in the retention time of DRAM cells. Thus, refresh mechanisms in DRAM cache designs typically use a single worst-case refresh period dictated by the cell with the lowest retention time. For example, previous work [3] used aggressive $40\mu s$ retention period of for eDRAM compared with the $64ms$ retention time of a commodity DRAM. However, this low refresh rate will increase the idle refresh energy and degrades the performance of the DRAM cache.

To achieve maximum energy-efficiency of the large hybrid cache, we need to decide the Pareto-optimal retention time of volatile STT-RAM by studying the relationship between retention time, write energy and refresh energy. The retention time T_{store} of STT-RAM can be modeled as [13]

$$T_{store} = \frac{1}{f_0} e^{\Delta} \tag{1}$$

where f_0 is the thermal attempt frequency and Δ is the thermal stability of MTJ [11]. Δ can be further characterized as

$$\Delta = \frac{M_s H_k V \cos(\theta)}{k_B T} \tag{2}$$

where M_s is the saturation magnetization, H_k is the anisotropy field, V is the volume, k_B is the Boltzmann constant, and T is the absolute temperature.

For successful switching within a target write time, the write current I_w must exceed a threshold known as the critical switching current I_c. The required switching current density J_C for the thermal activation is defined as [10]

$$J_C^{THM}(T_{sw}) = J_{C0}(1 - \frac{1}{\Delta} ln(\frac{T_{sw}}{\tau_0})) \qquad (T_{sw} > 10ns) \tag{3}$$

where J_{C0} is the switching threshold current density, T_{sw} is the switching pulse width, τ_0 is the relaxation time. Thus, we get the write current and its inherent relationship with retention time based on (1)–(3)

$$I_w = A \times J_C^{THM}(T_{sw}) \tag{4}$$

where A is the cross-sectional area of the free layer of MTJ.

Based on the power evaluation methodology in [1], the refresh power P_{ref} of volatile STT-RAM is modeled as

$$P_{ref} = P_0 \times \frac{t_{RFC}}{t_{REFI}} = P_0 \times \frac{T_{store}}{t_{REFI}} \tag{5}$$

where P_0 is the unit refresh power determined by the burst refresh current and active standby current, t_{REFI} is the refresh interval.

We have the write current and the refresh power relationship with the retention time respectively based on the analysis above. Given that the energy is proportional to the current and power, we can then describe the numerical relationship among the write energy, the refresh energy and the data retention time as shown in the Fig. 2. Since the write energy is proportional and refresh energy is inversely proportional to the retention

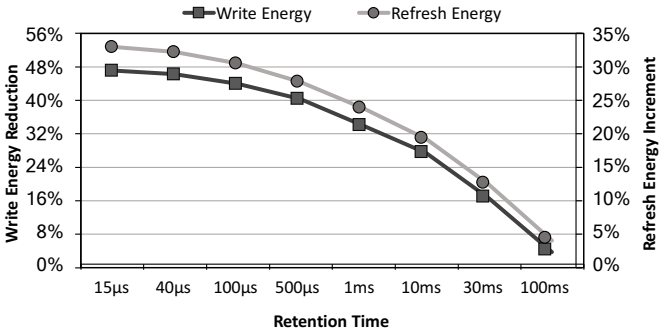

Fig. 2: Percentage of write energy reduction and refresh energy increment against retention time for the volatile STT-RAM region.

time respectively, the retention time should be selected to maximize the write energy reduction and minimize the refresh energy increment.

$$f(x) = E_{red}(x) - E_{inc}(x) \tag{6}$$

where x is the retention time, E_{red} is the write energy reduction changed with retention time, E_{inc} is the refresh energy increment changed with retention time. And we maximize $f(x)$ to get the optimal retention time, restricting retention time to be greater than aggressive eDRAM refresh rate and less than off-chip DRAM refresh period.

$$\text{maximize} \quad f(x) \quad (40\mu s < x < 64ms)$$

Thus, the retention time can be chosen in the range of $100\mu s$ to $1ms$ by the analysis above. Also, we consider the fact that the retention time of VSR should be closer to the refresh rate of DRAM cache to obtain a tradeoff of all retention time selections. And we also know the retention period of die-stacking DRAM cache is less than $100\mu s$, and the leakage of VSR is several times smaller than the DRAM cache even at the same refresh rate. Thus, to balance the refresh requirement of DRAM region and the energy saving from volatile STT-RAM region, we advocate our volatile STT-RAM region with a $0.5ms$ retention time by reducing the planar area of MTJ from $32F^2$ to $12F^2$.

C. 3D Asymmetric Data Access

Non-Uniform Cache Access (NUCA) is used to minimize wire delay of the cache by moving data near to the cores. However, the data movement inside the cache usually leads to extra energy consumption. Conventional 2D hybrid cache [3], [14], [15] only consider the difference of wire latency inside the specific region based on the location of the cache banks to the cores. Also, conventional 3D hybrid cache [8], [12] only consider the difference of wire latency of the vertical interconnection bus based on the location of the layers to the core layer. However, all these works do not consider the internal difference of latency and energy among different memory technologies, and they do not differentiate the order of read/write operation in the hybrid cache regions.

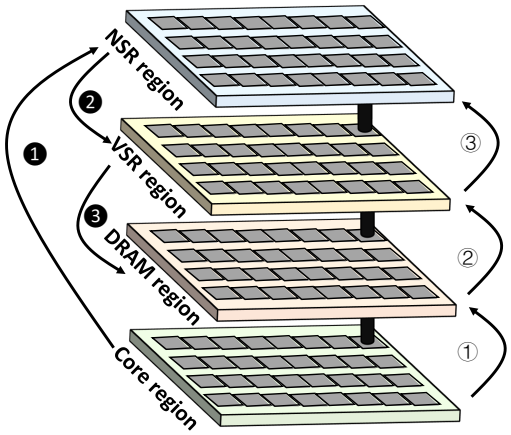

Fig. 3: Asymmetric data access in the tri-regional hybrid cache.

We observe that the latency and energy of different memory technologies inside different cache regions are more non-uniform than that of the 2D/3D cache wire interconnection. Instead of moving data near to the cores like the conventional NUCA design, our proposed approach is to change the accessing order to the different regions by fully utilizing the advantage of different memory techniques. Thus, based on the non-uniform read/write characteristics of the die-stacking hybrid cache, we propose an asymmetric data access policy for both read and write operations.

For the read request, it follows the accessing order from NSR region, VSR region and finally to the DRAM cache region (❶ → ❷ → ❸) as shown in the Fig. 3. The energy benefit of this order is that the DRAM region can be turned into a low power mode [4] to reduce high refresh and static energy when accessing NSR and VSR cache regions. The NSR and VSR regions, with little leakage and low read energy, can also achieve considerable energy savings. For the write request, it follows the accessing order from DRAM region, VSR region and finally to the NSR region (① → ② → ③). The energy benefit of this order is achieved by avoiding the high write energy of NSR region, and leveraging the write advantage of DRAM cache and VSR regions. Because every cache region is connected by TSV for data communication, it can easily bypass the lower cache region and access the upper cache region. Further, the wire latency and energy for bypassing inter-layer connection, $12ps$ to traverse 20-layer stack [12], is much smaller than that of we save.

D. Prediction Table

In the previous section, we show that data requests need to search all the hybrid cache regions until the tag is hit even if the requested data is not in that region, which wastes searching energy and time in the cache regions without the required data. To handle this problem, we propose a data prediction table to (1) skip regions in the hybrid cache by accessing the correct cache region directly, (2) reduce tag comparisons and reduce the tag lookup energy and latency, (3) go directly to off-chip DRAM on hybrid cache misses.

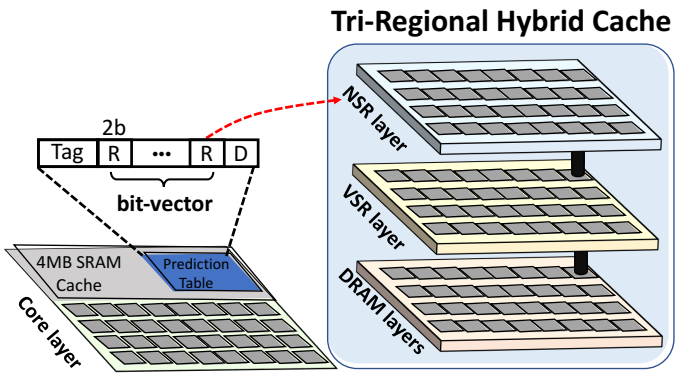

Fig. 4: Block diagram of the prediction table.

We maintain a prediction table stored in the L2 SRAM cache as shown in the Fig. 4. Every entry of the prediction table corresponds to every cache line of the hybrid cache. It consists of a tag for the address comparison of data accesses, and a bit vector for tracking the region of every data block. Because there are three separate regions in the hybrid cache, we use a 2-bit counter stored in the bit-vector to track the region location for every data block.

When there is a new data request for a data block in the hybrid cache, every bit-vector entry is first searched to fetch the region location information of that data block. We define the 2-bit (00, 01, 10, 11) information to represent (Inside DRAM cache, Inside NSR, Inside VSR, Non-existent). If the counter value is 00, 01 or 10, which means the data block is inside DRAM region, inside NSR region or inside VSR region respectively. Then the request is sent to the hybrid cache controller to access the specific region directly. Also, if it is 11, then the request can bypass the whole hybrid cache and access the off-chip memory as soon as possible.

Each time a data block is inserted into the hybrid cache, its corresponding bit in the prediction table will be set. The bit will be cleared after the data block is invalidated. We employ selective writeback to maintain consistency between the hybrid cache and the prediction table. When a prediction table entry is evicted, all hybrid cache blocks tracked by that entry are evicted from the hybrid cache to keep the prediction table precise.

III. Experimental Methodology

We extend gem5 simulator [2] to simulate an 8-core system with 2-level SRAM cache and a last-level hybrid cache. The STT-RAM regions and DRAM region are modeled similar to [5] and [7] respectively. The major system parameters are listed in the Table I. All the experimental parameters of 3D volatile/non-volatile STT-RAM and DRAM cache are obtained from the modified version of DESTINY [9]. McPAT [6] is used to get the power values of the hybrid cache. The 3D TSV model parameter is based on [12].

We use SPEC CPU2006 benchmarks to evaluate energy and performance of the hybrid cache. For each of the workloads, we warmed up the simulation for one billion cycles and collected results for one billion cycles. For the evaluation metrics, we use energy savings and IPC speedups to show how much energy and performance efficiency can be achieved.

TABLE I: SYSTEM CONFIGURATIONS

CPU core	8-Core OoO, 2.8GHz
SRAM Cache	L1: 32KB, 8-way, 4 cycles, 64B linesize L2: 4MB, 16-way, 10 cycles, 64B linesize LRU, write-back, write-allocate policy
Hybrid Last-level Cache	NSR (Non-Volatile STT-RAM): 64MB, 1.13/21.35 pJ/bit for R/W energy VSR (Volatile STT-RAM): 64MB, 1.16/1.28 pJ/bit for R/W energy DRAM: 128MB, 1.25/1.31 pJ/bit for R/W energy
Network Parameter	4 layers, 8 TSVs, 2-cycle router latency

IV. Results and Analysis

A. Energy Analysis

Static Energy: Fig. 5 illustrates the static energy, dynamic energy and the total energy savings of our tri-regional hybrid cache compared with the baseline. We observe that there is 36% reduction of static energy on average. This energy saving mainly comes from three sources: 1) the cell leakage reduction due to the non-volatility of STT-RAM region; 2) the DRAM cache region are turned into low power mode when applying the asymmetric data access policy; 3) the volatile STT-RAM is relaxed to have over 4× fewer leakage than a DRAM cache.

Dynamic Energy: We also observed that the dynamic energy is reduced by 16% compared with the similar size baseline. This energy saving can be attributed to the following reasons: 1) the volatile STT-RAM effectively reduces the high write energy over the conventional STT-RAM; 2) we use less aggressive refresh rates to minimize the refresh energy overhead of the volatile STT-RAM, and the refresh energy of the non-volatile STT-RAM is completely removed; 3) we apply an energy-efficient data access policy to reduce writing to the non-volatile STT-RAM; 4) unnecessary searching energy is largely eliminated by checking the prediction table beforehand.

Total Energy: The total energy is significantly reduced with 26% saving. Among all 12 workloads, 7 of them have more than 26% energy reduction. The primary reasons for these reductions are: 1) the proportion of static energy in the total energy is high for a large cache, hence the usage of the STT-RAM effectively reduces the energy; 2) the volatile STT-RAM helps to achieve a balance of static and dynamic energy reduction. Although there is a small increment in the refresh energy of the volatile STT-RAM, this is offset by the significant reduction in the write energy. Even for the write-intensive workloads such as $mgrid$ and mcf, our hybrid cache still achieves 25% to 27% energy saving.

B. Performance Analysis

Speedup: Fig. 6 shows the speedup of the proposed hybrid cache over the baseline. It improves the performance by 11% on average due to the following reasons. 1) The volatile STT-RAM is relaxed to have a write latency comparable to the

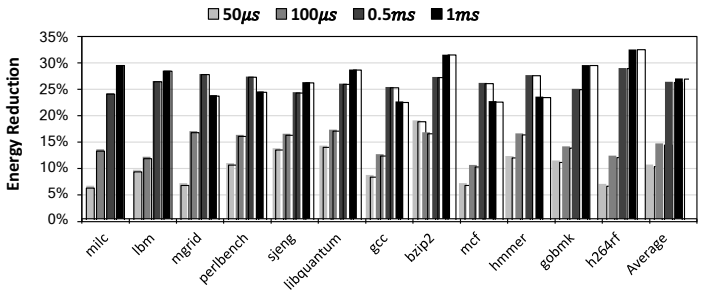

Fig. 5: Static and dynamic energy savings over baseline.

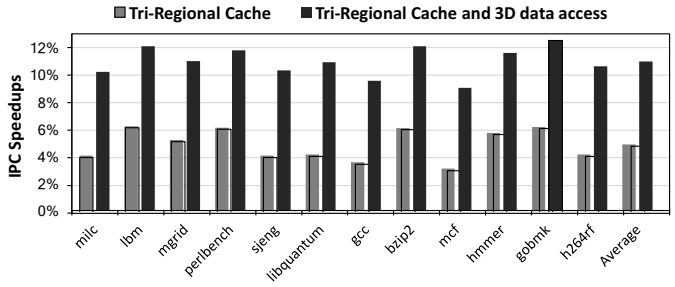

Fig. 6: Performance speedups compared to baseline.

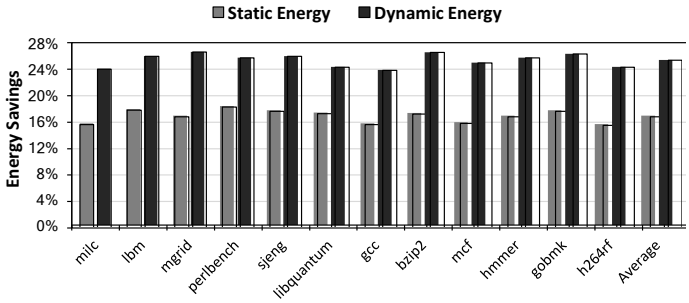

Fig. 7: Energy savings over traditional 2D hybrid cache.

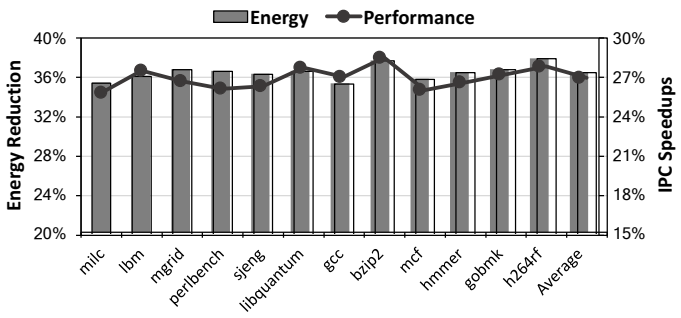

Fig. 8: Evaluation of asymmetric data access policy.

DRAM region, and it is used as a buffer to reduce the write operation in the conventional STT-RAM. 2) The tri-regional cache design with asymmetric access policy can restrict the data access in the latency-efficient regions. 3) The prediction table in the SRAM cache effectively reduces the tag lookup latency in the large cache. Our hybrid cache achieves 20% area reduction compared to baseline because STT-RAM is about 1x-2x denser than on-die DRAM. Thus, the performance can be further improved due to the larger cache capacity for the same chip area.

Benchmark Analysis: For the benchmarks which are not memory intensive and require modest cache sizes (e.g. $libquantum$, lbm and $gobmk$), the hybrid cache performs better by reducing access to the non-volatile STT-RAM region based on our asymmetric access policy. Also, for the write-intensive benchmarks ($hmmer$, $perlbench$ and $mgrid$), the performance improves above the average because the volatile STT-RAM effectively buffers a large number of writes to the other regions. However, the benchmarks that require large cache sizes (e.g. mcf and gcc), have relatively high miss rates and low performance in the proposed hybrid cache.

C. More Evaluations

2D Hybrid Cache Comparison: Compared with the conventional hybrid cache (DRAM and STT-RAM region), on average our hybrid cache achieved a 16.7% reduction in leakage and a 26.2% reduction in dynamic energy consumption as shown in Fig. 7. The static energy reduction comes the reason that the leakage of volatile STT-RAM is 4× smaller than die-stacking DRAM. And the reduction in dynamic energy can be attributed to 1) the low energy consumed during DRAM and volatile STT-RAM reads/writes; 2) the usage of volatile STT-RAM as buffer region to avoid accessing unoptimized STT-RAM with high write energy. Because the MTJ planar area of volatile STT-RAM is reduced from $32F^2$ to $12F^2$, so our design saves 8.25% total dies' area for the same cache capacity.

Asymmetric Access: We evaluated the asymmetric hybrid cache access policy and Fig. 8 shows that performance is increased by 27.5% and energy is reduced by 36.3% on average compared with the design without the policy. The IPC and energy benefits are due to the reasons that: 1) Traditionally, both read and write requests traverse the die-stacking hybrid cache structure die-by-die based on the order of location to the core die from near to far, but it is energy-inefficient to use the same order for read and write due to the non-uniform read/write energy in different dies. 2) The difference in wire delay and energy is much less than that of the different dies with different memory technologies.

Prediction Table Overhead: The area overhead of the prediction table is 1MB due to the 2 bits storage for every 64B data block in the 256MB hybrid cache. The extra storage occupies part of the precious L2 SRAM capacity, which may hurt the performance of L2 sensitive applications. We also need extra latency to access the prediction table compared with conventional cache access, it is 10 cycles for the L2 cache (4c tags + 6c data). However, this latency overhead is small compared with the average 50% reduction of the hybrid cache searching latency. Actually, the prediction table improves the performance of applications that are large cache sensitive by providing direct region access and reducing the tag comparison latency.

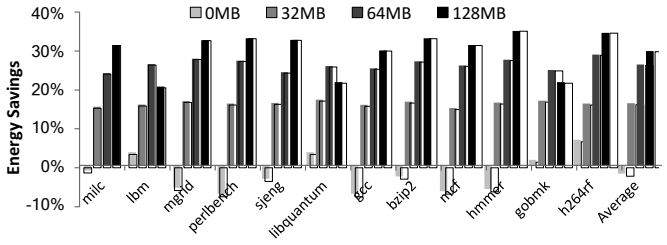

Fig. 9: Sensitivity analysis of different capacity of volatile STT-RAM region.

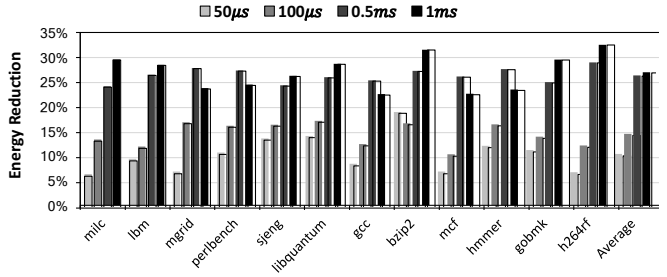

Fig. 10: Sensitivity analysis of various retention time.

D. Sensitivity Analysis

Size of Volatile STT-RAM Region: The key difference of our design compared with conventional hybrid cache is the usage of Volatile STT-RAM (VSR) as a buffer region. To better evaluate the impact of this design on the energy, we change the VSR with capacity 0MB (hybrid cache without VSR), 32MB, 64MB (default) and 128MB. The Fig. 9 shows that the VSR with the small capacity (32MB and 64MB) has more energy savings in the benchmarks with the smallest datasets, the VSR with large capacity (128MB) is more energy-efficient in the benchmarks with large cache footprints and write-intensities. For the hybrid cache without VSR (0MB), there is more variation in the energy savings. Because it is hard to balance the static and dynamic energy of a conventional hybrid cache without any optimizations, hence many benchmarks have negative energy reductions.

Retention Time: To better understand the effect of retention time on the hybrid cache energy consumption, we change the retention time to $50\mu s$, $100\mu s$ and $1ms$ in our experiments. As the energy results show in Fig. 10, for the benchmarks that are write-intensive, it is observed the $0.5ms$ retention time is the optimal compared with the other three in terms of energy. But for the benchmarks that are not write-intensive, the energy increases with the decreasing retention time. The results suggest that refresh energy is less dominant than write energy in the write-intensive benchmarks for our hybrid cache. However, although high retention time saves more energy than low retention in some conditions, it is not a proper choice when considering the refresh rate of die-stacking DRAM.

V. Conclusion

Modern computing systems have the increasing demand for the large cache capacity to reduce latency and increase bandwidth. However, the energy consumption of large cache becomes an inevitable challenge for the energy-constrained systems. Thus, in this paper, we have presented a novel hybrid cache architecture that can effectively reduce the energy of large die-stacking cache. First, we propose to use volatile STT-RAM as a buffer to balance the energy consumption of DRAM region and non-volatile STT-RAM region of the hybrid cache. Second, we analyze the optimal retention time to maximize the energy-efficiency of volatile STT-RAM. Also, we propose an asymmetric data access policy to fully utilize the energy benefit of different memory technologies of the tri-regional hybrid cache. Finally, a prediction table is proposed to reduce the searching energy of the hybrid cache. Compared with the baseline DRAM cache, the results show that our hybrid cache can achieve 26% energy reduction and 11% performance improvement on average.

References

[1] Micron technology, calculating memory system power for ddr3. 2007.
[2] N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, J. Saidi, D. R. Hower, T. Krishna, S. Sardashti, et al. The gem5 simulator. *ACM SIGARCH Computer Architecture News*, 2011.
[3] J. Cong, G. Gururaj, and Y. Zou. An energy-efficient adaptive hybrid cache. In *Proceedings of ISLPED*, 2011.
[4] H. David, C. Fallin, E. Gorbatov, U. R. Hanebutte, and O. Mutlu. Memory power management via dynamic voltage/frequency scaling. In *Proceedings of ICAC*, 2011.
[5] E. Kultursay, M. Kandemir, A. Sivasubramaniam, and O. Mutlu. Evaluating stt-ram as an energy-efficient main memory alternative. In *Proceedings of ISPASS*, 2013.
[6] S. Li, J. H. Ahn, R. D. Strong, J. B. Brockman, D. M. Tullsen, and N. P. Jouppi. Mcpat: an integrated power, area, and timing modeling framework for multicore and manycore architectures. In *Proceedings of MICRO*, 2009.
[7] G. H. Loh and M. D. Hill. Efficiently enabling conventional block sizes for very large die-stacked dram caches. In *MICRO*, 2011.
[8] N. Madan, L. Zhao, N. Muralimanohar, A. Udipi, R. Balasubramonian, R. Iyer, S. Makineni, and D. Newell. Optimizing communication and capacity in a 3d stacked reconfigurable cache hierarchy. In *Proceedings of HPCA*, 2009.
[9] M. Poremba, S. Mittal, D. Li, J. S. Vetter, and Y. Xie. Destiny: A tool for modeling emerging 3d nvm and edram caches. In *Proceedings of DATE*, 2015.
[10] A. Raychowdhury, D. Somasekhar, T. Karnik, and V. De. Design space and scalability exploration of 1t-1stt mtj memory arrays in the presence of variability and disturbances. In *Proceedings of IEDM*, pages 1–4, 2009.
[11] C. W. Smullen, V. Mohan, A. Nigam, S. Gurumurthi, and M. R. Stan. Relaxing non-volatility for fast and energy-efficient stt-ram caches. In *Proceedings of HPCA*, pages 50–61, 2011.
[12] G. Sun, X. Dong, Y. Xie, J. Li, and Y. Chen. A novel architecture of the 3d stacked mram l2 cache for cmps. In *Proceedings of HPCA*, 2009.
[13] 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.
[14] J. Wang, Y. Tim, W.-F. Wong, Z.-L. Ong, Z. Sun, and H. Li. A coherent hybrid sram and stt-ram l1 cache architecture for shared memory multicores. In *Proceedings of ASP-DAC*, pages 610–615, 2014.
[15] Z. Wang, D. A. Jiménez, C. Xu, G. Sun, and Y. Xie. Adaptive placement and migration policy for an stt-ram-based hybrid cache. In *Proceedings of HPCA*, pages 13–24, 2014.