Couture: Tailoring STT-MRAM for Persistent Main Memory

Mustafa M Shihab[1], *Jie Zhang*[3], *Shuwen Gao*[2], *Joseph Callenes-Sloan*[1], *and Myoungsoo Jung*[3]
[1]The University of Texas at Dallas, [2] Intel Corporation
[3] School of Integrated Technology, Yonsei Institute Convergence Technology, Yonsei University
Computer Architecture Memory Systems Laboratory
mustafa@camelab.org, jie@yonsei.ac.kr, shuwen@camelab.org, jcallenes.sloan@utdallas.edu, m.jung@yonsei.ac.kr

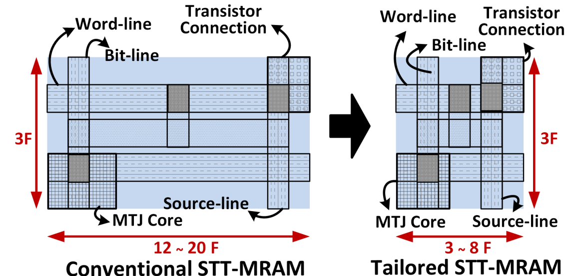
Abstract

Modern computer systems rely extensively on dynamic random-access memory (DRAM) to bridge the performance gap between on-chip cache and secondary storage. However, continuous process scaling has exposed DRAM to high off-state leakage and excessive power consumption from frequent refresh operations. Spin-transfer torque magnetoresistive RAM (STT-MRAM) is a plausible replacement for DRAM, given its high endurance and near-zero leakage. However, conventional STT-MRAM cannot directly substitute DRAM due to its large cell space area and the high latency and energy costs for writes. In this work, we present *Couture* – a main memory design using tailored STT-MRAM that can offer a storage density comparable to DRAM and high performance with low-power consumption. In addition, we propose an intelligent data scrubbing method (*iScrub*) to ensure data integrity with minimum overhead. Our evaluation results show that, equipped with the iScrub policy, our proposed Couture can achieve up to 23% performance improvement, while consuming 18% less energy, on average, compared to a contemporary DRAM.

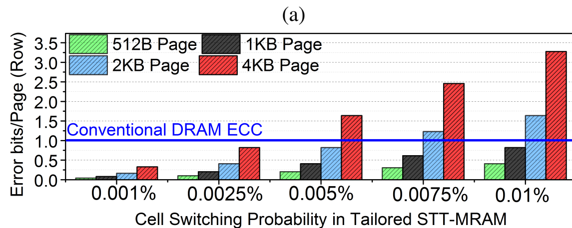
1 Introduction

Dynamic random-access memory (DRAM) has been instrumental in propelling the progress of computer systems, serving as the exclusive main memory technology. However, many recent data-intensive applications are demanding terabytes of working memory [1], forcing DRAM to scale-down to smaller process technologies that can adversely affect its performance and reliability. Specifically, scaling down DRAM cells increases off-state leakage, and reduces data-retention time. This results in frequent refresh operations that burden DRAM with extra latency and power overhead. The increased leakage also renders DRAM less reliable, as leakage from the "off" cells into the bitlines can generate errors.

While prior studies addressed these challenges using techniques such as data partitioning [2] and retention-aware refresh [3], it is expected that replacing DRAM with a non-volatile memory (NVM) can alleviate the power and reliability issues in the long term [4]. Spin-transfer torque magnetoresistive random-access memory (STT-MRAM) has received considerable attention as one such replacement candidate, owing to its excellent scalability, endurance, and near-zero leakage [4]. However, *the primary obstruction in replacing DRAM with STT-MRAM is their difference in cell area*. A typical STT-MRAM cell ($\sim 40F^2$) is roughly six times larger than a DRAM cell ($\sim 6F^2$), which renders it difficult for STT-MRAM to attain the density required for main memory. The other critical challenge with STT-MRAM is that, its write process involves "physically" switching the magnetic configuration of the magnetic tunnel junction (MTJ) with a large write current – rendering the energy cost unsuitable for write-intensive applications.



(a)



(b)

Figure 1: The reduction in STT-MRAM cell area by the proposed optimizations in our Couture design (a), and an analysis of retention errors in tailored STT-MRAM (b).

In this work, we present *Couture*, a persistent main memory that tailors STT-MRAM to offer a DRAM-comparable storage density, along with low-power operation, and excellent read/write performance. Specifically, we propose to tune the MTJ's thermal stability factor to lower the write current, *which allows STT-MRAM to use smaller access transistors and significantly reduce its cell area*. Figure 1a demonstrates this STT-MRAM cell area reduction achieved by our Couture design. By lowering the write current, we also reduce STT-MRAM's write energy. Furthermore, we present an optimized DDR3 [5] based dual in-line memory module (DIMM) design for our tailored STT-MRAM.

Unfortunately, optimizing the thermal stability factor partially truncates the conventional data retention time of the STT-MRAM ($\sim$10 years). In addition, at the end of the retention period, the STT-MRAM cells experience an increased probability of unintentional switching of the MTJs and can generate errors, which in turn can render

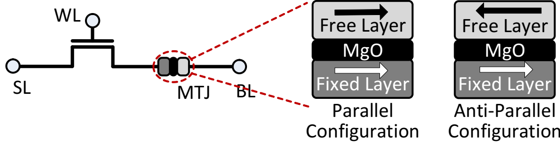


Figure 2: Internal structure of the STT-MRAM cell.

STT-MRAM difficult to be a reliable persistent working memory. Figure 1b demonstrates that, a small rise in the cell-level switching probability can escalate the error rate in a 4KB memory page, overwhelming the typical single-error correction and double-error detection (SECDED) DRAM ECC [6]. To address this, we propose *iScrub*, a novel data-scrubbing scheme using *reinforcement learning* to ensure data persistency with minimum overhead. The main **contributions** of this work are as follows:

• *Tailored STT-MRAM.* We tune the MTJ design parameters to render the STT-MRAM cell area comparable to that of a modern DRAM. Specifically, we adjust the thermal stability factor to lower the critical current for switching the MTJ's orientation, which in turn enables us to reduce STT-MRAM's cell area and its write energy.

• *Revamped memory module.* We propose a modified DIMM that incorporates unique requirements of STT-MRAM, while maintaining main memory's fundamental organization and DDR interface.

• *Reinforcement learning for intelligent scrubbing.* We employ a customized reinforcement learning algorithm to minimize the latency and power overhead caused by scrub operation in tailored STT-MRAM without incurring additional retention errors.

In this work, we compare our Couture memory and a DDR3 DRAM at system-level, and the evaluation results show that our Couture with iScrub can deliver up to a 23% performance gain, while consuming 18% less energy, on average.

2 Preliminary

2.1 STT-MRAM Fundamentals

Cell structure. Figure 2 shows the cell structure of STT-MRAM, which consists of an access transistor and a magnetic tunnel junction (MTJ). The access transistor is used to activate and control the cell, while MTJ works as the storage element. The MTJ demonstrates two different resistance levels by switching the polarity of its ferromagnetic plates to either parallel or anti-parallel configuration. These two resistance levels are used to represent the stored data bit (0 or 1) [7].

Characteristics of basic operations. The read operation of STT-MRAM is fast and power-efficient, because sense amplifier only needs to inject small current to measure the resistance of target cells. However, a large current is required to switch the the target MTJ's free layer orientation during a write operation, which results in high write energy and large STT-MRAM cell size. Specifically, to avoid write failures, transistors with a large W/L ratio are used to drive the write current, which in turn increases the cell size [8].

2.2 Exploring Critical Design Parameters

Thermal stability factor (Δ). In order to ensure data storage reliability, the MTJ in a STT-MRAM cell must maintain the parallel and the anti-parallel configurations in a discrete and stable manner. Thermal stability factor refers to the stability of an MTJ's magnetic orientation, and is modeled as [9]:

$$\Delta \propto E_b \approx \frac{H_K M_S V}{2 k_B T} \tag{1}$$

where E_b is energy barrier, T is temperature, H_K is anisotropic field, M_S is saturation magnetization, k_B is Boltzmann constant, and V is MTJ's volume.

One can note that, while the other parameters are fixed, we can set the thermal stability factor to fit our optimization goal by modifying the MTJ's volume.

Critical current (I_C). Critical current refers to minimum current that switches the polarity of MTJ's free layer [9]. The model for critical current can be expressed as:

$$I_C = \gamma[\Delta + \delta V T] \tag{2}$$

where γ and δ are fitting constants that represent the operational environment, V is MTJ's volume, and T is temperature.

The analytic model reveals that, critical current (I_C) can be reduced by reducing thermal stability factor (Δ).

Retention time. Retention time is the expected time before a random bit-flip occurs. It depends on the thermal stability factor, and can be expressed as follows [9]:

$$T_{Retention} = \frac{1}{f_0} exp(\Delta) \tag{3}$$

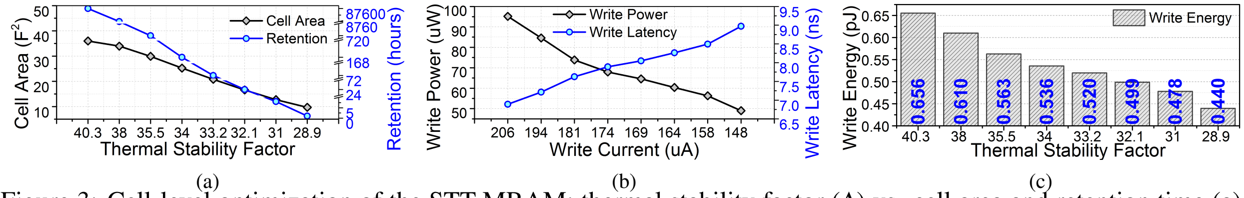
where f_0 is the operating frequency.

One can notice that, while a low thermal stability factor reduces the MTJ's critical current requirement, unfortunately, it also shortens the retention time.

3 Persistent Main Memory

3.1 Tailoring STT-MRAM

Access transistor optimization. A STT-MRAM cell is comprised of one access transistor and one MTJ. Since MTJ has comparatively small size, access transistor size becomes the main factor of cell area. As access transistor is used to control current to go through MTJ, the transistor size is mainly decided by the maximum current. Thus, in order to minimize the STT-MRAM cell area, the critical current should be reduced. Equations $1 \sim 3$ indicate the relationship of thermal stability factor and critical current. Derived from Equations $1 \sim 3$, we evaluate shows the optimized cell areas under the reduced thermal stability factors, and the results are shown in Figure 3a. One can observe from this figure that as the thermal stability factor decreases, the cell area keeps decreasing. However, it also shows that, the lowering of thermal stability factor (Δ) also shortens the retention time of our tailored STT-MRAM optimization. By con-



(a) (b) (c)

Figure 3: Cell-level optimization of the STT-MRAM: thermal stability factor (Δ) vs. cell area and retention time (a), write current vs. write power and write latency (b), and thermal stability factor (Δ) vs. write energy (c).

sidering the trade-off between the cell size and retention time, we shorten the thermal stability factor from 40.29 to 28.91 by reducing MTJ volume and the cell area is reduced from 36 F^2 to 10 F^2. While the cell area of our tailored STT-MRAM is not equal to that of DRAM, it attains a practically comparable size.

Energy and latency optimization. We also derived the relationship of write current and write power as well as write current and write latency, which is shown in Figure 3b. It is crucial to note that lowering the write current in order to reduce the write power only marginally increases the write latency. In fact, reducing the write current can significantly improve the overall write energy. As shown in Figure 3c, when we lower the thermal stability factor from its default value of 40.3 to 28.9, the STT-MRAM's write energy actually decreases from 0.66 pJ to 0.44 pJ.

3.2 Memory Module Design

Memory module organization. To ensure a smooth transition from DRAM, we designed our proposed Couture such that it can fit in existing main memory infrastructure, with only minimal changes to the host system. Therefore, we used a modified DIMM with the DDR3 PC3-12800 interface to implement our proposed Couture design. Our Couture configuration follows the consumer DRAM architecture and has two ranks per module and eight x8 memory chips (+1 ECC chip) per rank. Figure 4 illustrates the organization of our Couture design. Similar to DRAM, each Couture rank is logically divided into eight banks, and each bank is further divided into subarrays. Whereas global address decoders and row buffers are assigned to each bank, subarrays are internally accessed using local address decoders and row buffers. Individual memory cells, consisting of an access transistor and an MTJ, are connected via a bitline (BL), a sourceline (SL), and a wordline (WL). Because STT-MRAM requires different current paths for writing "0" and "1", our Couture cells are connected to two different write-input drivers, $W0_En$ and $W1_En$.

On-chip controller and sensing circuit. The on-chip controller for our Couture connects the DDR3 interface's 64-bit external data bus (EDB) running at 1600 MHz and the 16-bit external address bus (EAB) running at 800 MHz to the internal address (IAB) and data bus (IDB), respectively. As shown in Figure 5a, the on-chip controller utilizes a decoder and a multiplexer-demultiplexer pair to manage the data access and transfers to and from the different ranks of the DIMM.

Figure 5b shows the sensing circuit used in our Couture to detect the content of a cell. This design utilizes a

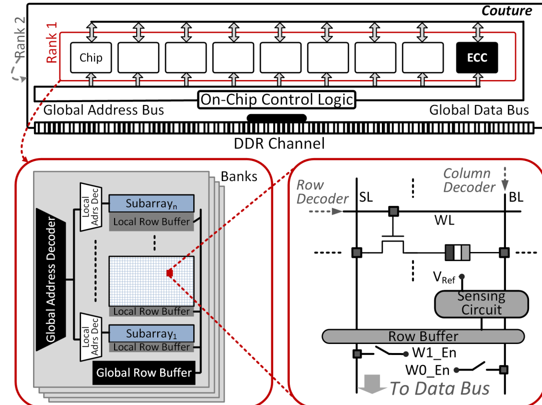
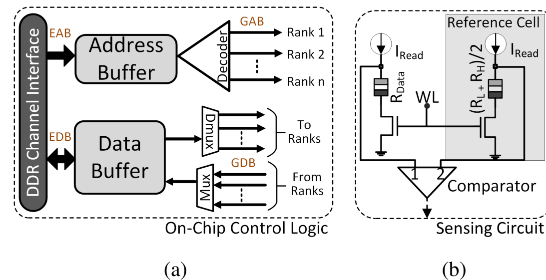


Figure 4: Proposed Couture main memory.



(a) (b)

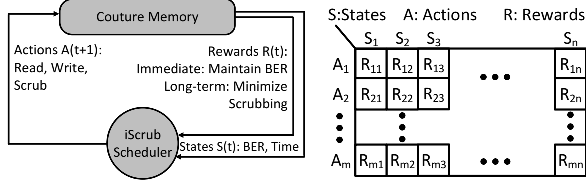
Figure 5: The on-chip controller (a), and sensing circuit (b) for Couture memory.

reference cell with the MTJ resistance level set exactly at the middle of the resistance spectrum (i.e., $\frac{R_H+R_L}{2}$).

Couture memory operations. The Couture memory follows the DDR standard for read/write operation, including decoding the row address to activate target wordline, decoding the column address to activate specific bitline, and leveraging sense amplifier and write driver to read/write data. The only difference in Couture memory is scrub operation. Data scrubbing in Couture is essentially a read followed by a write, and the naive scrub scheduling would be similar to DRAM refresh, albeit with a much longer interval (e.g., 64 ms vs. 1 hr). However, we propose *iScrub*, an intelligent scrub scheduler that can further minimize the scrub frequency, and does not rely on a conservative fixed period.

3.3 Reliability Management

Data retention in STT-MRAM depends on the switching of the MTJ's free layer, which is probabilistic in nature. Even with a fixed retention time, some cells can retain data for a longer time, while others can lose it before the



(a) Interaction between environment and controlling agent. (b) State-Action-Reward table.

Figure 6: iScrub based on reinforcement learning.

set period. Prior work on similar scheduling problems [10] [11] motivated us to consider an reinforcement-learning (RL) based self-optimizing scrub scheduler for Couture, that can exploit such probabilistic behavior and achieve a performance level otherwise unattainable with static scrubbing schemes.

As illustrated Figure 6a, in our RL model for the iScrub scheduler, the state functions (S) consist of the current scrub frequency, the time elapsed since the last scrub, and the current bit error rate (BER). The action function (A) includes two options: assign either a scrub or the scheduled I/O command. The immediate reward (R) goal for iScrub is to maintain the BER permitted by the ECC scheme, while the long-term reward is to minimize the scrub frequency and maximize I/O operations.

iScrub's scheduling policy. The working principle of our iScrub scheduler is shown in Algorithm 1. The iScrub scheduler operates on a table that records Q-values for all possible state-action pairs. Initially, all table entries are initialized with the highest possible Q-value (line 2). The scheduler then randomly issues either a scrub command or a command from the scheduled I/O operation in the transaction queue (line 3) and calculates the Q_P (line 4). At each predefined *test* cycle, the scheduler issues the command selected during the previous cycle (line 6) and collects the immediate reward (line 7). This command is selected by comparing it with the exploration parameter ε (line 8-11), which ensures that the RL algorithm is periodically and dynamically tuned, and it is assigned a very small value so as to ensure that commands with the highest Q-value are mostly selected. For each candidate command, the scheduler estimates the corresponding Q_{Sel} value from its Q-value table (line 12) and updates it (line 13). Finally, the scheduler sets this value as the Q_P for the next cycle.

Executing scrub operations. iScrub tracks each page write (and scrub) operation on the Couture memory. Whenever a page is written (or scrubbed), a *retention counter* is assigned to that page, which counts down with every clock cycle. When the counter hits zero, the memory controller is informed that it is a probable time for scrubbing that page. Using the aforementioned algorithm the controller then decides whether an immediate scrub operation is required in order to maintain reliable data retention. If a scrub operation is scheduled, then a *scrub-required* flag is set, and using a *bank ID* and a *scrub-sequence* number, the page is appended to the list of pages to be scrubbed in the corresponding bank. The scrub sequence number is assigned based

Algorithm 1: iScrub scheduling algorithm

Data: A: Action (i.e., Command), S: State, R: Reward
Input: γ: Discount parameter, ε: Exploration parameter
1 **Initialization**
2 All **Q-values** $\leftarrow \frac{1}{1-\gamma}$
3 **A**$\leftarrow$ select randomly: command from transaction queue or, scrub
4 $Q_P \leftarrow$ get **Q-value** for current **S** and **A**
5 **for** *Every "test" signal* **do**
6 Issue **A**, selected during the previous cycle
7 Collect immediate **R** for the issued command
8 **if** *rand()* $< \varepsilon$ **then**
9 Next **A** $\leftarrow$ random command (exploration)
10 **else**
11 Next **A** $\leftarrow$ command with the highest **Q-value** (exploitation)
12 $Q_{Sel} \leftarrow$ **Q-value** for the current **S** and **A**
13 $Update_Q \leftarrow$ SARSA update based on Q_P, **R**, Q_{Sel}
14 $Q_P \leftarrow Q_{Sel}$ `// Set Q-value for next cycle`

on a FIFO model, where a lower sequence number indicates a higher priority for scrubbing, and is updated every time a page is scrubbed (or written) in the bank. On the other hand, if scrubbing is not required immediately, the counter is then set again but to a smaller value (e.g., in our evaluation, to one-third of the previous value). The decreasing counter value allows the memory controller to check on a page more closely. This process is repeated until the page is overwritten or scrubbed, and then the counter is reset to its original value.

4 System Level Evaluation

4.1 Experimental Setup

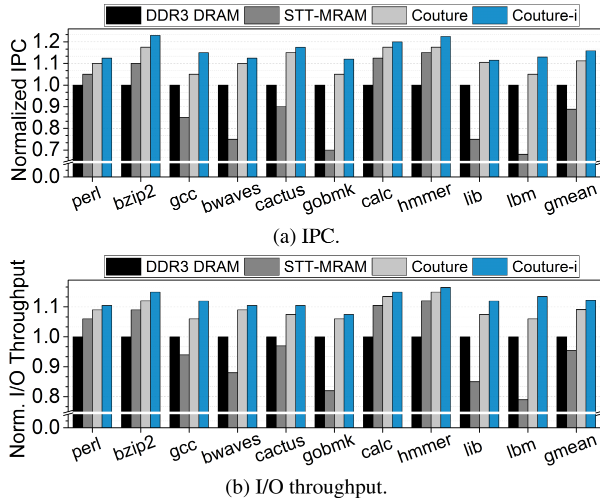
Simulated configurations. In order to discover the system-level benefits of our Couture scheme, we conducted the evaluation for four memory configurations:

- `DDR3 DRAM`: This configuration represents conventional DRAM memory with periodic refresh operations.
- `STT-MRAM`: This is a main memory design with STT-MRAM of ten year retention time.
- `Couture`: This configuration represents our Couture design without iScrub scheduler.
- `Couture-i`: This is the optimal configuration for our Couture design which utilizes iScrub scheme.

Processor	2.8GHz, OoO execution, SE mode
L1 Cache	Private 64KB Instruction and 64KB Data Cache
L2 Cache	Shared 8MB Unified Cache
Working Memory (Refresh freq.)	DRAM (64 ms), STT-MRAM (non-volatile), Couture (1 hour), Couture-i (varying)
Row Buffer Strategy	FR-FCFS and Open adaptive
Workloads	perl, bzip2, gcc, bwaves, cactus, gobmk, calc, hmmer, lib, and lbm

Table 1: Summary of the evaluation setup.

Evaluation method. We build our Couture latency model by considering both cell-level and peripheral circuit latency. Specifically, we calibrated CACTI [12] to get reasonable peripheral latency of main memory. For STT-MRAM's subarray access latency, we collect the data by modifying the non-volatile memory simulator, NVSim [13]. For system-level evaluation, we integrate our Couture latency model in the gem5 simulator [14] –



(a) IPC.

(b) I/O throughput.

Figure 7: Performance analysis.

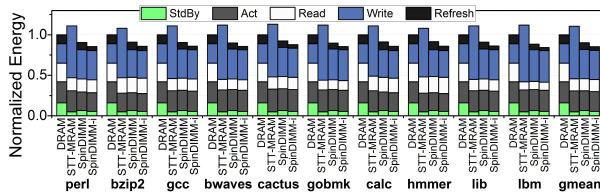


Figure 8: Energy analysis with detailed breakdown.

one of the recognized system simulators. We verify the performance of our Couture with ten workload applications from the SPEC2006 [15] benchmark suite. Table 1 details the simulation parameters used in our evaluation.

4.2 Evaluation Results

Performance analysis. Figure 7a shows a performance comparison of the four memory configurations in terms of the instructions per cycle (IPC), normalized to that of the `DDR3 DRAM` baseline. The results show that, although `STT-MRAM` entirely eliminated refreshes, its performance fell below that of `DDR3 DRAM` by 17%, on average. We believe that this is because of the long write latency. Moreover, by lowering the write latency, our `Couture` improved the average IPC by approximately 8%, compared to `DDR3 DRAM`. Finally, with iScrub in place, our `Couture-i` improved the average IPC by 16% over `DDR3 DRAM`, peaking at almost 23% for bzip2 and hmmer.

Figure 7b provides a second performance comparison, in terms of I/O throughput, that is the number of read/write operations served by each memory. Again, the results are normalized to that of `DDR3 DRAM`, and the unoptimized `STT-MRAM` fell short of it, by almost 5%, on average. As expected, `Couture` exceeded `DDR3 DRAM`'s performance by 8%, on average. However, `Couture-i` performed best, with an average improvement of nearly 13% over the `DDR3 DRAM` baseline.

Energy analysis. Figure 8 portrays the energy consumption trend for the four evaluated memory configurations, breaking each of them into five components: standby, activation, read, write, and refresh. All the values are normalized to the `DDR3 DRAM` baseline. Unsurprisingly, `DDR3 DRAM` consumed a significant amount of energy when in standby and refresh, reinforcing the primary concern of this work. Second, although `STT-MRAM` eliminated refreshes and reduced standby energy, the overall energy consumption nevertheless increased by 9.5%, on average, owing to the high write current. Our proposed `Couture`, improved the average energy consumption over `DDR3 DRAM` by around 14%, as a consequence of its optimized write current. Finally, by employing our iScrub technique, `Couture-i` reduced the scrub energy and further reduced the energy consumption. In particular, our `Couture-i` reduced the energy consumption by 20% for bzip2, and reduced it by 18%, on average.

To summarize, our Couture memory, equipped with our proposed iScrub mechanism, not only can reach a DRAM-like density, but also can deliver improved performance and consume significantly less energy.

5 Related Work

Adopting STT-MRAM. [16], [17], [4], [18] and [19] proposed last-level cache (LLC) designs based on STT-MRAM or combined with SRAM. Though these designs can reduce the write overheads of STT-MRAM, they address the challenges of SRAM-based cache – not the DRAM-based memory. While [20] proposed a STT-MRAM memory with partial-writes and write-bypasses for better performance, unfortunately, it overlooked the cell area drawback of STT-MRAM – one of the most critical challenges in deploying it as a main memory.

Optimizing STT-MRAM. [4] relaxed the retention time of their STT-MRAM cache for reducing write energy/latency. Whereas, [17] lowered the thermal stability factor for a scalable STT-MRAM cache. However, those optimization again targeted fitting the STT-MRAM in the cache architecture, and not that of the main memory.

6 Acknowledge

The authors would like to thank MemRay Corporation for technical support. This research is supported in part by MSIP "ICT Consilience Creative Program" IITP-R0346-16-1008, NRF-2015M3C4A7065645, NRF-2016R1C1B2015312 DOE grant DE-AC02-05CH1123 and MemRay grant (2015-11-1731). M. Jung has an interest in being supported for any engineering or customer sample product on emerging NVM technologies (e.g., PRAM, X-Point, ReRAM, STT-MRAM etc.). The corresponding author is M. Jung.

7 Conclusion

In this work, we proposed *Couture* – an area and energy optimized STT-MRAM designed to address the challenges faced by modern memory. Couture enables a smooth transition from DRAM, and ensures data integrity in the tailored STT-MRAM. Our system-level evaluation showed that our *Couture* with *iScrub* can achieve up to 23% performance improvement, while consuming 18% less power, compared to DRAM.

References

[1] NERSC, "NERSC computational systems," http://www.nersc.gov/systems/computational-systems-table/.

[2] S. Liu *et al.*, "Flikker: saving DRAM refresh-power through critical data partitioning," *ACM SIGPLAN Notices*, vol. 47, 2012.

[3] J. Liu *et al.*, "RAIDR: Retention-aware intelligent dram refresh," in *ISCA*, 2012.

[4] C. W. Smullen *et al.*, "Relaxing non-volatility for fast and energy-efficient STT-RAM caches," in *HPCA*, 2011.

[5] Micron, "Micron DDR3 SDRAM," https://goo.gl/memsWd.

[6] M.-Y. Hsiao, "A class of optimal minimum odd-weight-column SEC-DED codes," *IBM Journal of Research and Development*, vol. 14, 1970.

[7] H. Li and Y. Chen, *Nonvolatile Memory Design: Magnetic, Resistive, and Phase Change*. CRC Press, 2011.

[8] J. Li, *et al.*, "Design paradigm for robust Spin-Torque Transfer Magnetic RAM from circuit/architecture perspective," *IEEE VLSI*, vol. 18, 2010.

[9] A. Khvalkovskiy *et al.*, "Basic principles of STT-MRAM cell operation in memory arrays," *Journal of Physics D: Applied Physics*, vol. 46, 2013.

[10] J. Mukundan and J. F. Martinez, "MORSE: Multi-objective reconfigurable self-optimizing memory scheduler," in *HPCA*, 2012.

[11] E. Ipek *et al.*, "Self-optimizing memory controllers: A reinforcement learning approach," in *ISCA*, 2008.

[12] N. Muralimanohar *et al.*, "CACTI 6.0: A tool to model large caches," *HP Laboratories*, pp. 22–31, 2009.

[13] X. Dong *et al.*, "NVSim: a circuit-level performance, energy, and area model for emerging non-volatile memory," *IEEE TCAD*, vol. 31, 2012.

[14] N. Binkert *et al.*, "The gem5 simulator," *ACM SIGARCH Computer Architecture News*, vol. 39, 2011.

[15] J. L. Henning, "SPEC CPU2006 benchmark descriptions," *ACM SIGARCH Computer Architecture News*, vol. 34, 2006.

[16] Z. Sun *et al.*, "Multi retention level STT-RAM cache designs with a dynamic refresh scheme," in *MICRO*, 2011.

[17] H. Naeimi *et al.*, "STTRAM scaling and retention failure," *Intel Technology Journal*, vol. 17, 2013.

[18] J. Li *et al.*, "STT-RAM based energy-efficiency hybrid cache for CMPs," in *VLSI-SoC*, 2011.

[19] Y.-T. Chen *et al.*, "Dynamically reconfigurable hybrid cache: An energy-efficient last-level cache design," in *DATE*, 2012.

[20] E. Kultursay *et al.*, "Evaluating STT-RAM as an energy-efficient main memory alternative," in *ISPASS*, 2013.