Couture: Tailoring STT-MRAM for Persistent Main Memory

Mustafa M Shihab
Jie Zhang
Shuwen Gao
Joseph Callenes-Sloan
Myoungsoo Jung

CAMELab.org
Computer Architecture and Memory Systems Laboratory

Executive Summary

Motivation:

DRAM plays an instrumental role in modern computers, serving as the exclusive main memory technology. But process scaling has exposed it to high ***leakage*** and ***refresh power*** consumption.

STT-MRAM can be an excellent replacement for DRAM, given its high ***endurance*** and ***near-zero leakage***.

Executive Summary

Motivation:

DRAM plays an instrumental role in modern computers, serving as the exclusive main memory technology. But process scaling has exposed it to high ***leakage*** and ***refresh power*** consumption.

STT-MRAM can be an excellent replacement for DRAM, given its high ***endurance*** and ***near-zero leakage***.

Challenge:

Conventional STT-MRAM *cannot directly substitute* DRAM, because of

- *Large cell area*
- *High write energy*

Executive Summary

Motivation:

DRAM plays an instrumental role in modern computers, serving as the exclusive main memory technology. But process scaling has exposed it to high ***leakage*** and ***refresh power*** consumption.

STT-MRAM can be an excellent replacement for DRAM, given its high ***endurance*** and ***near-zero leakage***.

Challenge:

Conventional STT-MRAM *cannot directly substitute* DRAM, because of

- *Large cell area*
- *High write energy*

Solution:

We propose, ***Couture,*** a tailored STT-MRAM based memory that offers

- *DRAM-comparable storage density*
- *High performance with low write energy*
- *Intelligent data scrubbing (iScrub) to ensure data integrity with minimum overhead*

Executive Summary

Motivation:

DRAM plays an instrumental role in modern computers, serving as the exclusive main memory technology. But process scaling has exposed it to high ***leakage*** and ***refresh power*** consumption.

STT-MRAM can be an excellent replacement for DRAM, given its high ***endurance*** and ***near-zero leakage***.

Challenge:

Conventional STT-MRAM *cannot directly substitute* DRAM, because of

- *Large cell area*
- *High write energy*

Solution:

We propose, ***Couture,*** a tailored STT-MRAM based memory that offers

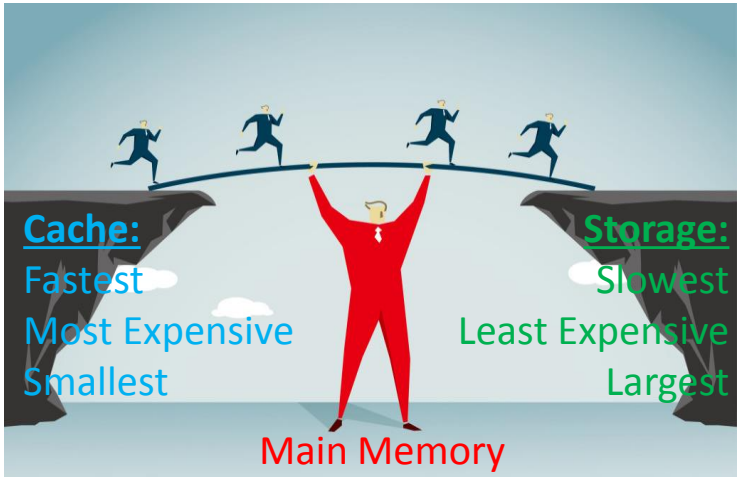
- *DRAM-comparable storage density*
- *High performance with low write energy*
- *Intelligent data scrubbing method (iScrub) to ensure data integrity with minimum overhead*

Results:

Compared to a contemporary DRAM, our proposed Couture can

- *Achieve up to 23% performance improvement*
- *Consume 18% less energy, on average*

Main Memory ≈ DRAM



Main memory ensures optimal performance-cost balance by bridging the gap between on-chip cache and storage

DRAM has been the ubiquitous choice for main memory - most often incarnated as DIMMs

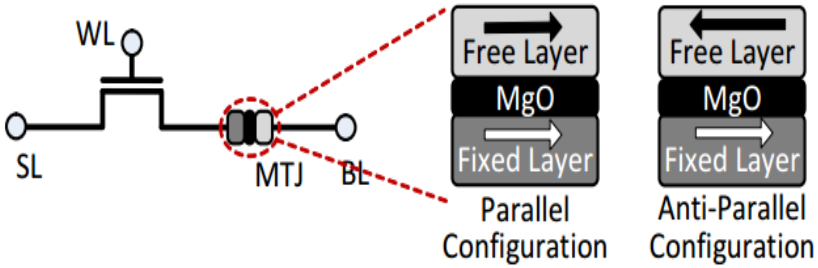
Applications are getting increasingly data-intensive, demanding larger memory. So, DRAM has to scale-down to process technologies that can hurt its performance

Scaling down DRAM cells increases off-state leakage

Leakage results in frequent refresh operations that burden DRAM with extra latency and power overhead. Particularly critical for HPCs and data-centers with TBs of memory

STT-MRAM: Opportunity and Challenges

Spin Transfer Torque Magnetoresistive RAM (STT-MRAM) stores data in a magnetic tunnel junction (MTJ) core



Physical configuration of the MTJ determines data type i.e., 0 or 1 .
Data can only be written by driving current to change MTJ configuration.
This is both good and bad!

✓ In absence of write current there is no switching.
This means there is no scope for off-state leakage

X Physical switching of the MTJ requires a large current.
This makes the write energy high

X Large write current requires a large access transistor to drive it, increasing the cell area.
A typical STTMRAM cell (~40F²) is roughly 6X larger than a DRAM cell (~6F²)

Tailored STT-MRAM: Critical Design Parameters

Couture proposes to reduce the STT-MRAM cell area and write energy by exploiting the design parameters for the MTJ core

Thermal Stability Factor (Δ): Refers to the stability of an MTJ's magnetic orientations

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

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

Critical current (I_C): Minimum current for switching the polarity of MTJ's free layer

$$I_C = \gamma[\Delta + \delta V T]$$

γ and δ = fitting constants that represent the operational environment

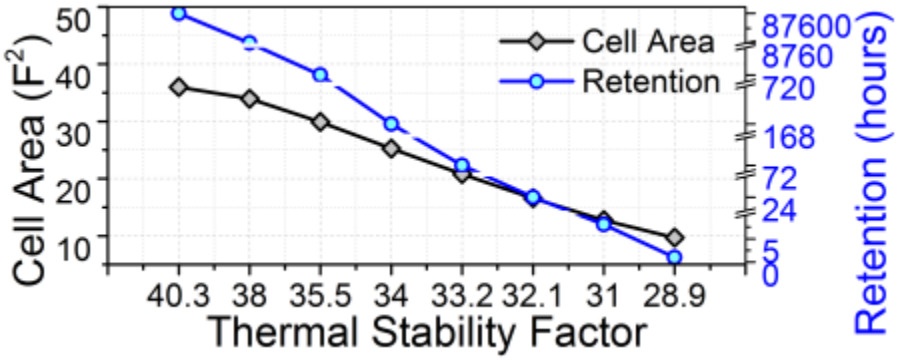
Retention time ($T_{Retention}$): The expected time before a random bit-flip occurs

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

f_0 is the operating frequency

Access Transistor Optimization

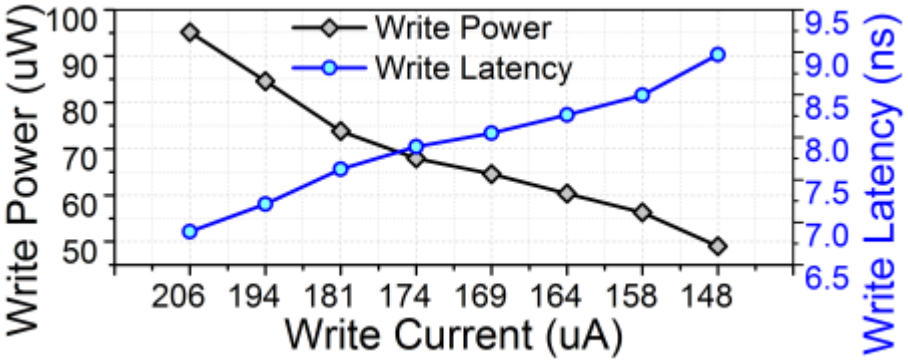
- STT-MRAM cell area is dominated by the access transistor, as MTJ is comparatively smaller in size
- The transistor size is mainly determined by the current driven by it. So, to make the transistor smaller, we need to reduce the critical current
- We reduce the MTJ thickness to lower the thermal stability factor, which in turn lowers the critical current

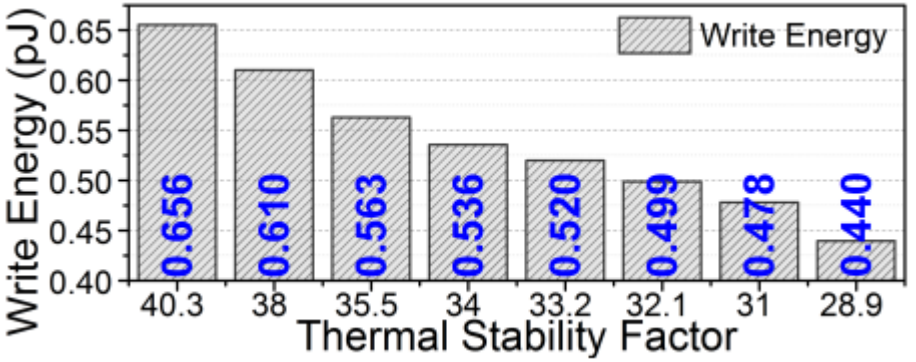


As we lowered the thermal stability factor from 40.29 to 28.91 by reducing MTJ volume, the cell area is reduced from ***36 F^2*** *to* ***10 F^2***

Unfortunately, lowering the thermal stability factor also shortens the retention time of our tailored STT-MRAM

Write Energy Optimization





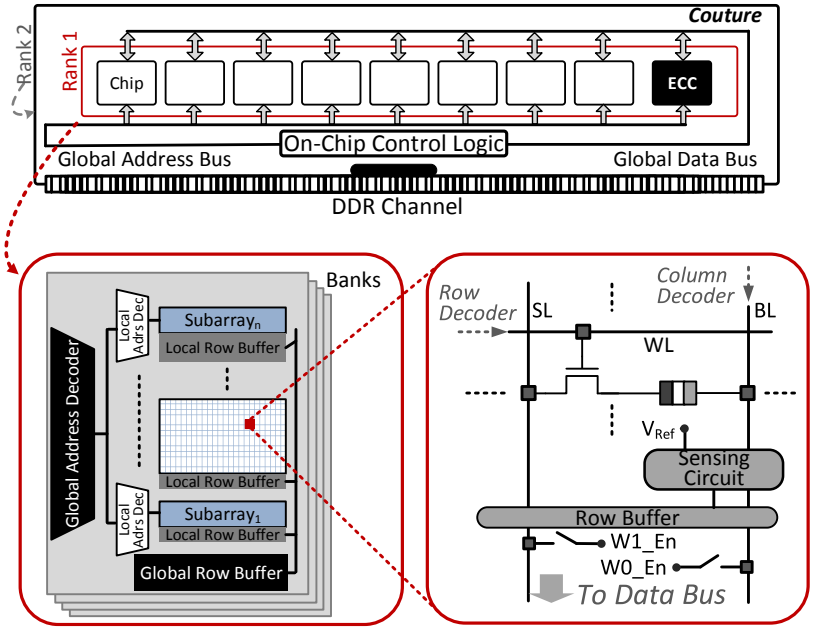
- Lowering the critical current allows us to lower the STT-MRAM write current.

  This in turn can reduce the write power by **~50%**
- The lowered write current results in a marginally increased the write latency
- But, reducing the write current can improve overall write energy by **~33%**

*Lowering the thermal stability factor from its default value of 40.3 to 28.9, STT-MRAM's write energy decreases from **0.66 pJ** to **0.44 pJ***

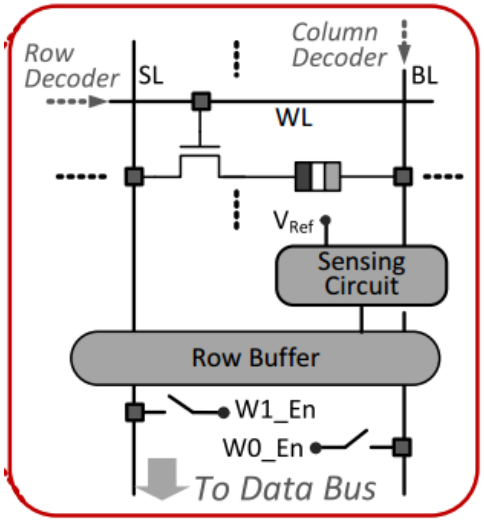
Couture Memory Module

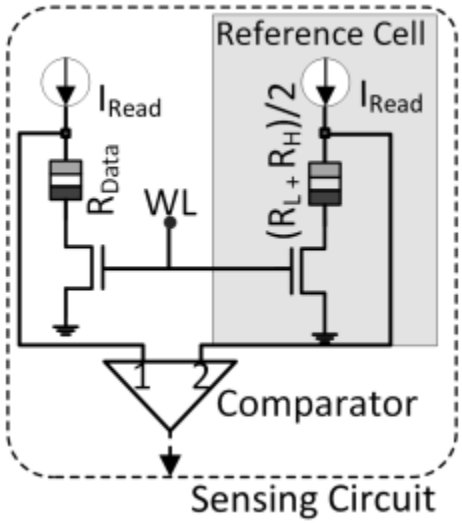
For a smooth transition from DRAM, our proposed Couture fits in the existing main memory DIMM packaging



- **2** Ranks in the Module >> **8** memory Chips (+1 ECC Chip) per Rank
- **8** logical Banks per Rank >> Each Bank is divided into subarrays of Cells
- Each Cell contains an Access Transistor and an MTJ, connected via a bit-line (BL), a source-line (SL), and a word-line (WL)

Dual Write Driver and Sensing Circuit





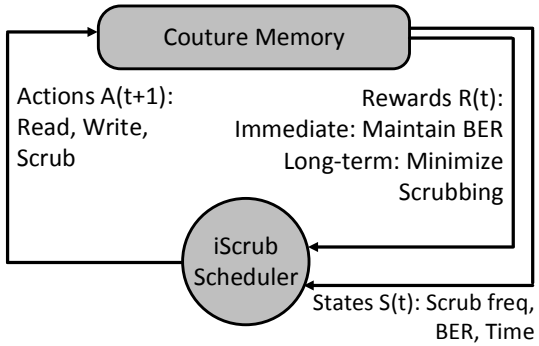
- STT-MRAM requires opposing current paths for writing "0" and "1", as the MTJ needs to be switched from anti-parallel to parallel configuration
  Couture connects the cells to two write-input drivers - ***W0_En*** and ***W1_En***

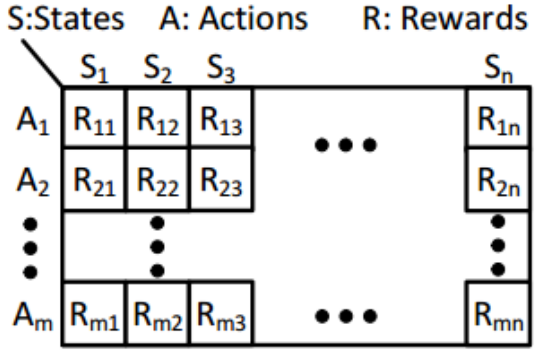
- Couture's sensing circuit detects the content of a cell by comparing it with a reference cell that has an MTJ that is set at a resistance level exactly in the middle of the resistance spectrum: $\frac{R_L + R_H}{2}$

Intelligent Data Scrub (iScrub)

- Cell-level optimization reduces the data retention time in tailored STT-MRAM
- But, data retention in STT-MRAM is actually probabilistic in nature
  - Even with a fixed retention time, some of the cells can retain data for a longer time, while others can lose it before the set period

iScrub – a reinforcement learning based scrub scheduler that can exploit such probabilistic behavior and ensure data integrity with minimum overhead

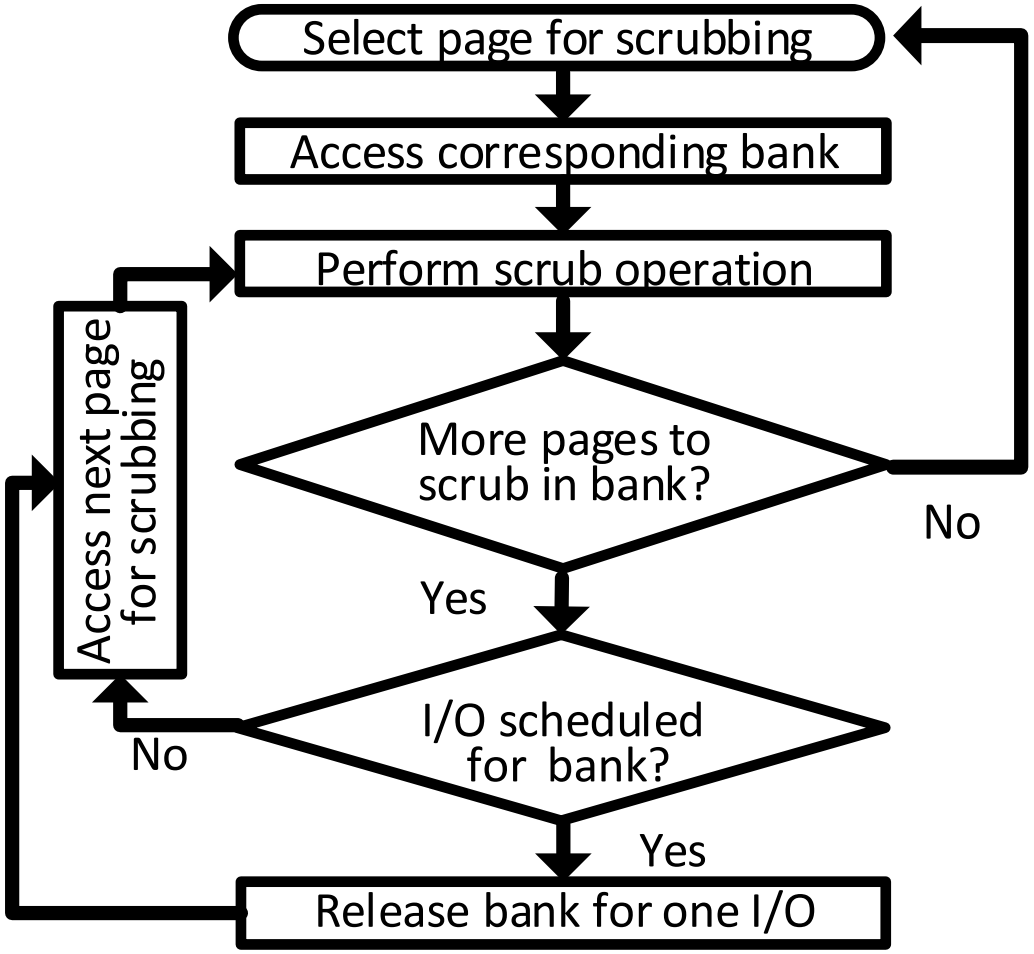




Reinforcement learning: ***learns and improve with experience***

- Interacts with its environment over time and senses the current state (S) of its environment and executes an action (A) that produces a reward (R)
- Goal is to maximize the cumulative reward by learning an optimal policy for mapping states to actions and gradually update a state-action-reward table

iScrub: Scheduling Algorithm



Evaluation Setup

Simulated configurations:

DDR3 DRAM: DRAM memory with periodic refresh operations

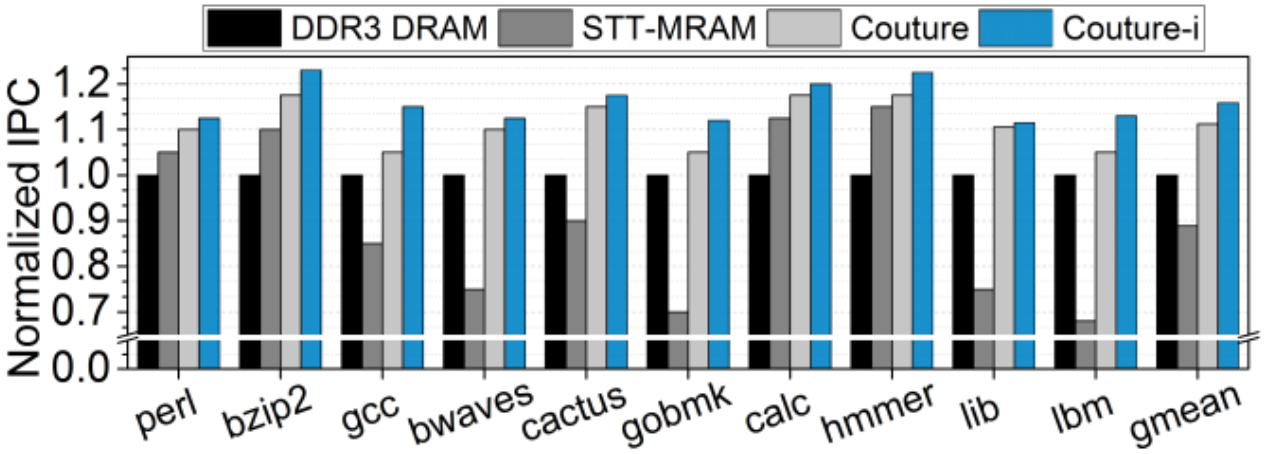
STT-MRAM: Main memory design with conventional STTMRAM (10yr retention)

Couture: Proposed Couture design without iScrub scheduler

Couture-i: Optimal configuration for our Couture design with iScrub scheme

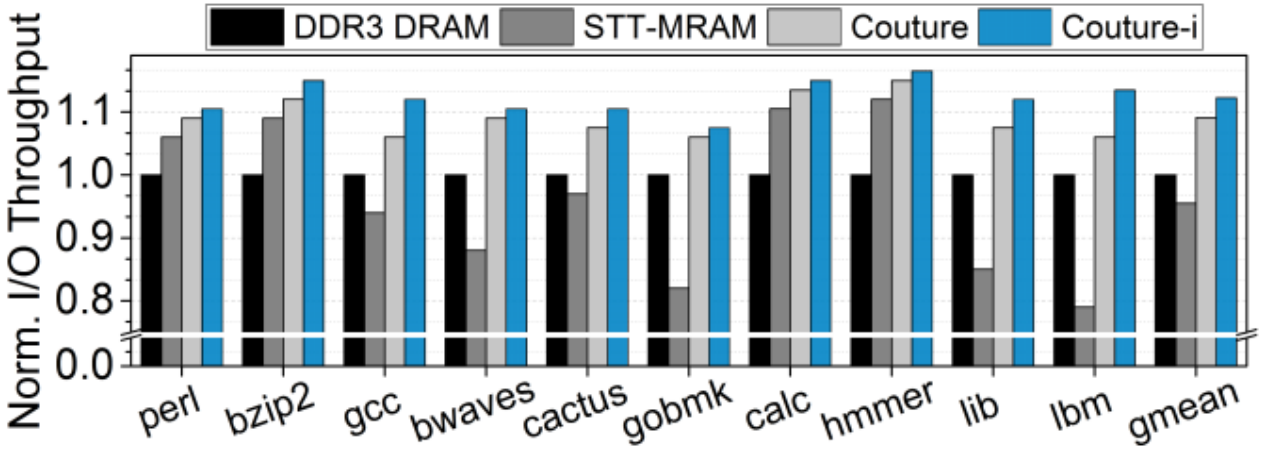
Processor	2.8GHz, OoO execution, SE mode
L1 Cache	Private 64KB Instruction and 64KB Data Cache
L2 Cache	Shared 8MB Unified Cache
Working Memory (Refresh freq.)	DRAM (64 ms), STT-MRAM (non-volatile), Couture (1 hour), Couture-i (varying)
Row Buffer Strategy	FR-FCFS and Open adaptive
Workloads	perl, bzip2,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

Performance Analysis - IPC



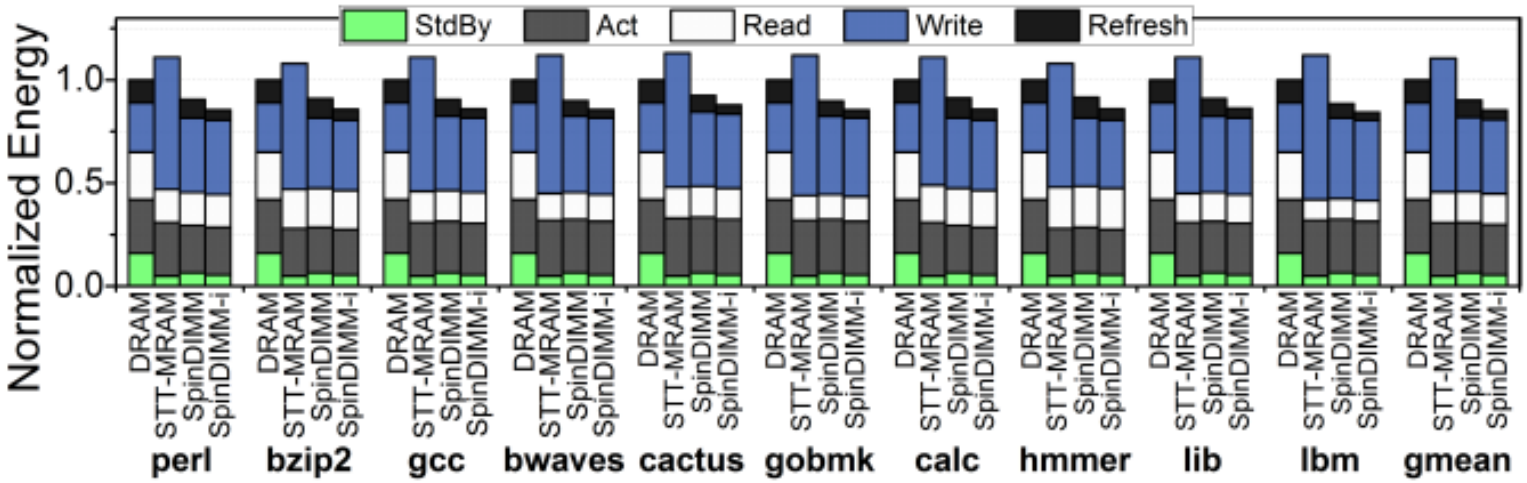
- Instructions per cycle (IPC) for the 4 memory configurations, normalized to that of the **DDR3 DRAM** baseline
- **STT-MRAM** eliminated refresh but average IPC fell below the baseline by **17%**
  - ➢ Because of the long write latency
- Lowering the write latency, **Couture** improved IPC by **8%**, on average
- With *iScrub*, **Couture-i** improved the average IPC by **16%** over **DDR3 DRAM**, peaking at **23%** for the bzip2 and hmmer benchmarks

Performance Analysis - Throughput



- Second performance comparison in terms of I/O throughput
- **STT-MRAM** fell short of the baseline **DDR3 DRAM** by **5%**, on average
- **Couture** exceeded the baseline's throughput by **8%**, on average
- **Couture-i** performed best with an average improvement of **13%** over the **DDR3 DRAM** baseline

Energy Improvement



- Energy consumption results:
  - Normalized to the **DDR3 DRAM** baseline
  - 5 components: *standby, activation, read, write,* and *refresh*
- DRAM consumed a significant energy for standby (leakage) and refresh
- **STT-MRAM** eliminated refresh and reduced standby energy, but the high write energy increases the overall consumption by **9.5%**
- **Couture**, with optimized write current, shows an average improvement of **14%**
- **Couture-i** reduced scrub energy with *iScrub* and further lowered total energy
  - Peak reduction: **20%** (for bzip2), Average reduction: **18%**

Thank You

CAMELab.org
Computer Architecture and
Memory Systems
Laboratory

Backup Slides

CAMELab.org
Computer Architecture and
Memory Systems
Laboratory

iScrub Algorithm

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`

Evaluation Process

- We build Couture latency model by considering cell-level and peripheral circuit latency
  - ➢ We calibrated CACTI to get reasonable peripheral latency of main memory.
- For STT-MRAM's subarray access latency, we collect the data by modifying NVSim
- For system-level evaluation, we integrate Couture latency model in the gem5
- We verify the performance of our Couture with ten workload applications from the SPEC2006 benchmark suite.