Architecting Volatile STT-RAM Caches for Enhanced Performance in CMPs

Adwait Jog† Asit Mishra‡ Cong Xu† Yuan Xie† N. Vijaykrishnan† Ravi Iyer‡ Chita Das†

†The Pennsylvania State University ‡ Intel Corporation

DAC 2012
San Francisco, CA

STT-RAM as Emerging Memory technology

• Spin-Torque Transfer RAM (STT-RAM) combines the speed of SRAM, density of DRAM, and non-volatility of Flash memory, making it attractive for on chip cache hierarchies.

• STT-RAM uses magneto-resistance to encode information.

• Read operation: Applying a small bias voltage across Select Line and Word Line, and sensing the current through the Magnetic Tunnel Junction (MTJ).

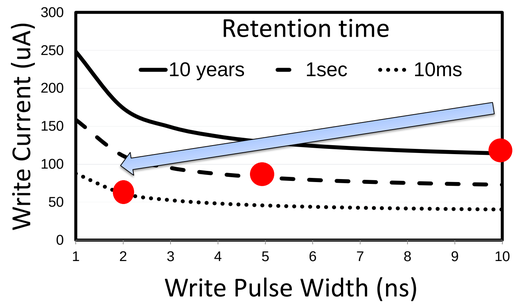
• Write operation: Activating the NMOS and applying a strong voltage to change the spin of the electrons in the free layer.

• This leads to high write latency and high write energy compared to read operation.

How to mitigate STT-RAM problems? (Answer: Trade off Non-Volatility for reduced write latency)

Device Characteristics ⟵ Optimal Retention Time? ⟶ Application Needs

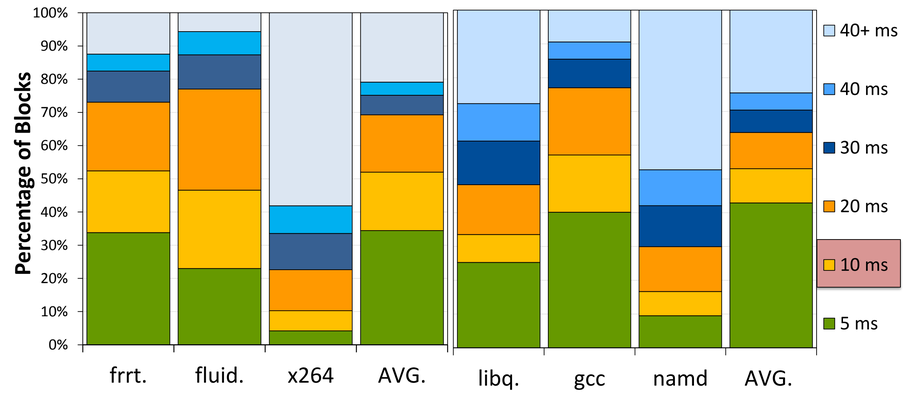
How can Non-Volatility be traded off?



Retention Time of STT-RAM	Write Latency @ 2 GHz
10 Years	22 cycles
1 second	12 cycles
10 milliseconds	6 cycles

How much Non-Volatility can be traded off?

Inter-Write Time Distributions of Multi-threaded and Multi-Programmed Benchmarks



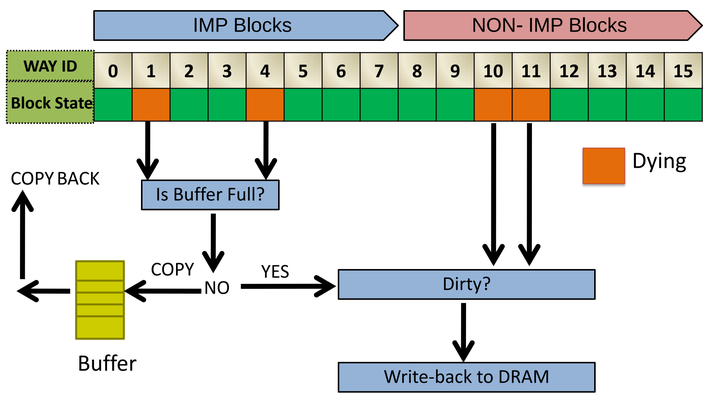
Majority (> 50%) of L2 Cache Blocks get refreshed within 10ms

Cache Refreshing Scheme

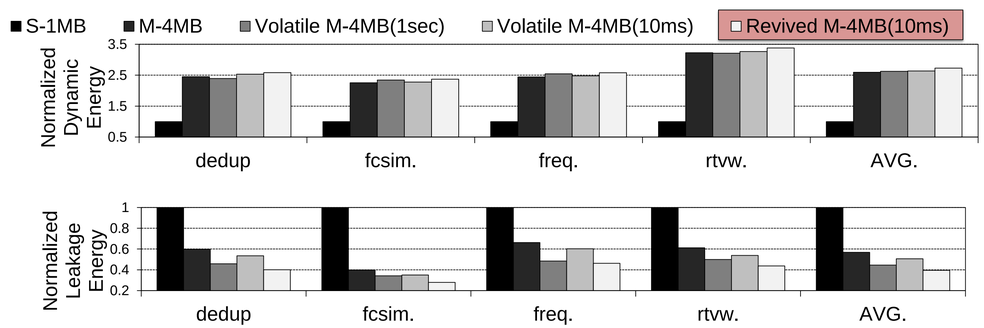
How to save rest 50% of the dying blocks because of volatile STT-RAM architecture?

Answer: Use Selective Refresh Policy.

Only refresh cache blocks which are in MRU Slots.

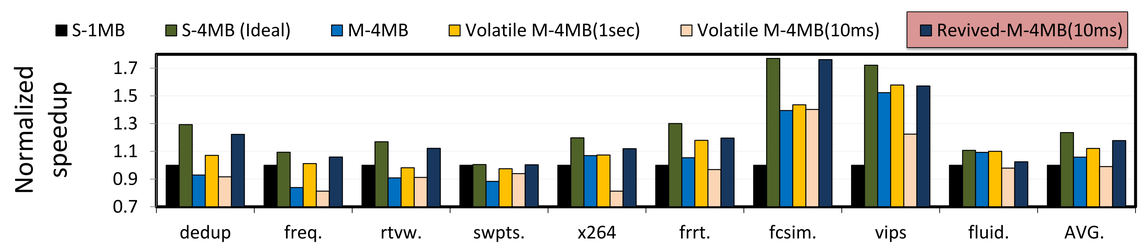


Energy and Performance Benefits



18% Improvement in IPC with our Cache Revive architecture

Nominal Increase in Dynamic Energy (4%), Substantial Reduction in Leakage Energy (60%) for multithreaded applications



Summary

• STT-RAM is a promising technology, which has high density, low leakage and competitive read latencies compared to SRAM.

• High Write Latency and Energy is impeding its widespread adoption.

• We propose trading off retention time for reduced write latency. Also, a simple buffering scheme is presented to store the important diminishing cache blocks.