# A Design Guideline for Volatile STT-RAM with ECC and Scrubbing

Namhyung Kim and Kiyoung Choi
Seoul National University
nhkim@dal.snu.ac.kr, kchoi@snu.ac.kr

***Abstract*—STT-RAM is considered as a promising alternative to SRAM due to its low static power (non-volatility) and high density. However, write operation of STT-RAM is inefficient in terms of energy and speed compared to SRAM and thus STT-RAM with low retention time (volatile STT-RAM) has been proposed at the cost of scrubbing and error correcting code (ECC). The more frequent scrubbing and stronger ECC are used, the shorter retention time is allowed. Based on extensive experiment and analytic STT-RAM model, this paper provides a guideline for designing a volatile STT-RAM with ECC and scrubbing.**

## I. Introduction

For a large cache in modern processors, Spin-Transfer Torque RAM (STT-RAM) is a promising candidate as an alternative to area and energy consuming SRAM. However, STT-RAM has poor write characteristics such as high write energy and long write latency. To overcome those poor write characteristics of STT-RAM, relaxing non-volatility of STT-RAM (thus making STT-RAM volatile) has been proposed [1].

By reducing STT-RAM cell area, thermal stability ($\Delta$) which represents height of thermal barrier for bit flipping is decreased to make STT-RAM volatile. As thermal stability decreases, write energy and latency is also decreased. However, lowering thermal stability makes bit flipping by thermal noise (retention failure) easier and thus decreases retention time.

In some previous work [1]–[3], simple DRAM-style read-and-write refresh techniques were employed to mitigate retention failure. However, the retention failure of volatile STT-RAM cannot be prevented by the refresh techniques because of the stochastic nature of STT-RAM retention failure. Instead, it was suggested to scrub STT-RAM periodically with error correction using error correcting code (ECC) [4], [5].

In this paper, from the analytic model of STT-RAM retention failure, we analyze and evaluate relationship among thermal stability, error correcting capability, and scrubbing overhead, and then provide a guideline of designing volatile STT-RAM cell and memory array.

## II. STT-RAM

### A. Overview

Unlike conventional charge based memory technologies, STT-RAM stores data in a special structure called Magnetic Tunnel Junction (MTJ). The stored data in the MTJ is determined by its magnetic direction. Static power consumption of STT-RAM is much lower than SRAM since it is unnecessary to consume extra power for maintaining the stored data because of its non-volatility. On the contrary, write characteristics such as write energy and latency are the main drawbacks of STT-RAM.

### B. Thermal Stability and Retention Failure

Thermal stability ($\Delta$) is one of STT-RAM cell design parameters. It is proportional to energy barrier between the two MTJ states (parallel and anti-parallel states) [6] as follows:

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

where $E_b$ is energy barrier, $k_B$ is the Boltzmann constant, and $T$ is temperature. $H_K$ is anisotropy field, $M_S$ is saturation magnetization, and $V$ is volume of the MTJ.

Retention failure is caused by an unwanted bit flip due to thermal noise. STT-RAM retention failure is a stochastic process and the probability of retention failure of a cell is modeled as a function of thermal stability ($\Delta$) and time ($t$) elapsed since the bit is written [4] as follows:

$$P_{ret_fail} = 1 - exp\left(\frac{-t}{exp(\Delta)}\right) \tag{2}$$

From (1) and (2), we can see that the retention failure rate can be controlled by cell design.

### C. ECC and Scrubbing

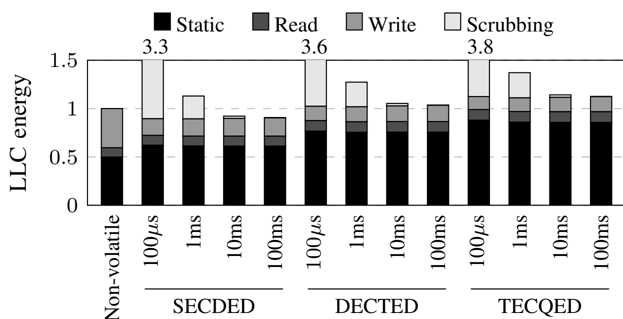
To tolerate increased error probability for volatile STT-RAM, ECC and periodic scrubbing is required. Every scrubbing operation reads whole data in memory array one by one and checks the data. If error is detected then it is corrected and written back in the same place. The scrubbing period should be short enough and the ECC should be strong enough so that errors are not accumulated. On the other hand, too frequent scrubbing may increase the performance overhead, energy consumption, and even wearing of STT-RAM.

### D. Memory Array of Volatile STT-RAM

As explained in the previous section, ECC and periodic scrubbing are mandatory for the reliability of volatile STT-RAM. Designing a memory array with ECC and scrubbing is complex since ECC and scrubbing depend on each other; ECC allows errors per its granularity up to its correcting capacity and scrubbing should be repeated before there occur too many errors to be handled by the ECC. Therefore, the STT-RAM cell (thermal stability) should be designed carefully considering memory array size, ECC strength, scrubbing period, and target reliability. The following equation shows how these parameters are related for proper operation [5].

$$\Delta \geq \frac{1}{1+k}\left[\sum_{i=0}^{k} \ln\left(\frac{m-i}{1+i}\right) + \ln\left(\frac{N}{\tau_0 F}\right) + k \ln\left(\frac{t_{ref}}{\tau_0}\right)\right] \tag{3}$$

Minimum thermal stability ($\Delta$) can be calculated from (3) when memory array is designed as an $N \times m$-bit array with $k$-bit ECC, target failure rate $F$, and scrubbing period $t_{ref}$.



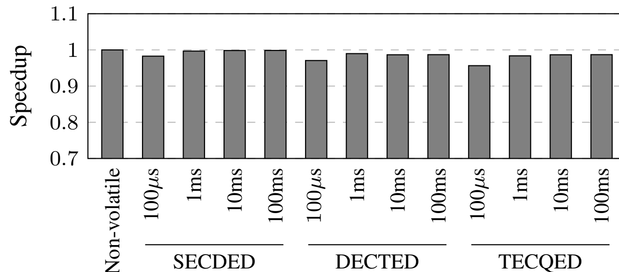




Fig. 1. Last-level cache energy consumption and performance of various configurations.

## III. Experimental Results

### A. Methodology

For evaluation, we use MacSim [7] simulator. The system runs at 4GHz with out-of-order core and has 2 MB 16-way set-associative last-level cache (LLC) with 64-byte blocks which is configured with STT-RAM of various thermal stability. STT-RAM caches are modeled by NVSim [8].

Extra overhead for ECC encoding/decoding and storing extra bits is also considered. The characteristics of the ECC encoders/decoders are extracted from the synthesized results obtained by using the Synopsys Design Compiler.

For the baseline, non-volatile STT-RAM without ECC and scrubbing is used. We evaluated volatile STT-RAM with four different scrubbing periods and three different ECC strengths.

### B. Last-Level Cache Energy

Fig. 1 shows the energy consumption of the LLC on different scrubbing period and ECC normalized to the baseline. Each bar shows the average energy consumption of all SPEC2006 benchmarks for each configuration.

As shown in Fig. 1, scrubbing overhead decreases as scrubbing period increases. The scrubbing overhead is negligibly small if the period is over 10ms (less than 2.4% when scrubbing period is 10ms). Thermal stability should be increased when the scrubbing period increases, which in turn increases the LLC energy. However, the effect of increased thermal stability by long scrubbing period is very small compared to the significant scrubbing overhead decrement.

For the same scrubbing period, LLC energy is increased as ECC strength is increased. Although the thermal stability of STT-RAM can be reduced with stronger ECC, the overhead of the encoder, decoder, and extra bits for the ECC is increased and the overhead grows faster than the energy saving from low thermal stability. Thus, using stronger ECC is not beneficial in terms of energy efficiency.

Compared to non-volatile STT-RAM baseline, among various evaluated configurations, using SECDED with long scrubbing period like 10ms or 100ms saves LLC energy (7.4% and 9.4% less energy than baseline, respectively). Others consume more LLC energy than the baseline because the overhead of scrubbing or ECC is larger than the energy saving from low thermal stability. In particular, 100$\mu$s scrubbing period is very inefficient because of the frequent scrubbing, which consumes more than 3x LLC energy compared to the baseline.

### C. Performance

Fig. 1 also compares performance of various configurations and that of the baseline. Using volatile STT-RAM as the LLC does not improve system performance at all and it even degrades performance. There are two main factors of performance degradation. First one is scrubbing overhead. As shown in Fig. 1, performance is degraded more with shorter scrubbing period. Since every scrubbing operation reads data/ECC and check whether the data is correct or not, it may incur more bank contentions. Second, extra cycles for decoding ECC may cause performance degradation. These extra cycles make read latency longer and may cause more bank contentions.

## IV. Conclusion

In this paper, we analyze and evaluate cache architectures using volatile STT-RAM, which has been proposed to address poor write characteristics of non-volatile STT-RAM. However, to utilize volatile STT-RAM, ECC with periodic scrubbing is mandatory due to its high retention failure rate. We evaluate volatile STT-RAM LLC with various ECC strength levels and scrubbing period configurations. Based on the evaluation, we reveal that ECC and scrubbing overhead hides the benefits of volatile STT-RAM over non-volatile STT-RAM in most configurations. We also reveal as our conclusion that weak ECC combined with long scrubbing period improves energy efficiency with negligible performance degradation.

## Acknowledgment

This work was supported by Samsung Advanced Institute of Technology, Samsung Electronics Co., Ltd.

## References

[1] C. W. Smullen IV *et al.*, “Relaxing non-volatility for fast and energy-efficient STT-RAM caches,” in *Proc. HPCA*, 2011.
[2] A. Jog *et al.*, “Cache revive: Architecting volatile STT-RAM caches for enhanced performance in CMPs,” in *Proc. DAC*, 2012.
[3] Z. Sun *et al.*, “Multi retention level STT-RAM cache designs with a dynamic refresh scheme,” in *Proc. MICRO*, 2011.
[4] H. Naeimi *et al.*, “STTRAM scaling and retention failure,” *Intel® Technology Journal*, vol. 17, pp. 54–75, 2013.
[5] C. W. Smullen IV, “Designing giga-scale memory systems with STT-RAM,” Ph.D. dissertation, University of Virginia, 2011.
[6] A. Khvalkovskiy *et al.*, “Basic principles of STT-MRAM cell operation in memory arrays,” *J. Phys. D: Appl. Phys.*, vol. 46, pp. 74 001–74 020, 2013.
[7] *MacSim Simulator.* https://github.com/macsimgt/macsim-public
[8] X. Dong *et al.*, “NVSim: A circuit-level performance, energy, and area model for emerging nonvolatile memory,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 31, pp. 994–1007, Jul. 2012.