InterA-ECC: Interconnect-Aware Error Correction in STT-MRAM

Surendra Hemaram*, Mahta Mayahinia*, Mehdi B Tahoori*†, Francky Catthoor†, Siddharth Rao†,
Sebastien Couet†, Tommaso Marinelli†, Anita Farokhnejad†, Gouri Sankar Kar†
* Department of Computer Science, Karlsruhe Institute of Technology (KIT), Germany, †Imec, Leuven, Belgium
Email: {surendra.hemaram, mahta.mayahinia}@kit.edu

***Abstract*—Spin-transfer torque magnetic random access memory (STT-MRAM) is a promising alternative to existing memory technologies. However, STT-MRAM faces reliability challenges, primarily due to stochastic switching, process variation, and manufacturing defects. These reliability challenges become even worse due to interconnect parasitic resistive-capacitive effects, potentially compromising the reliability of memory cells located far from the write driver. This can severely impair the manufacturing yield and large-scale industrial adoption. To address this, we propose an interconnect-aware error correction coding (InterA-ECC), which provides non-uniform error correction to a different zone of the memory subarray. The proposed InterA-ECC strategy selectively applies robust error-correction code (ECC) to specific rows within the subarray rather than uniformly across all rows, reducing ECC parity bits while enhancing bit error rate resiliency in the most vulnerable memory zone.**

I. Introduction

Spin-transfer torque magnetoresistive RAM (STT-MRAM) offers a promising alternative, specifically for system-on-chip level memories [1], [2]. However, STT-MRAM faces reliability issues, such as soft and hard errors, primarily due to the stochastic and asymmetric nature of magnetization and manufacturing defects [3]. The reliability concern is further increasing as interconnects become more critical with technology scaling due to the growing impact of parasitic resistive-capacitive (RC) delays [4]. This is problematic for cells farther from the write driver, where the voltage drop caused by parasitic resistance is more pronounced [5].

In the literature, several error correction strategies have been reported to improve STT-MRAM reliability [3], [6]. Additionally, some foundry efforts have also investigated device and circuit-level strategies to enhance STT-MRAM reliability [5], [7], [8], such as reducing parasitic resistance, using a common source line [7] or improving the read window with a hybrid resistance reference [8]. Besides, work in [5] employed design-process-test co-optimization, also depicted that most failed cells were located farther from the driver, underscoring the need for interconnect-aware error correction strategy.

In this work, we propose an interconnect-aware error correction coding (InterA-ECC) to mitigate the impacts of interconnect on STT-MRAM failure. The strategy divides the memory subarray into two zones based on the proximity of the rows to the write driver. The standard Single Error Correction-Double Error Detection (SEC-DED) Hamming code is applied for rows not far from the driver. A more robust error-correction code (ECC) is employed for rows farther from the driver, which is more susceptible to errors due to increased parasitic resistance. By strategically applying stronger ECC only to the most vulnerable rows, the InterA-ECC eliminates the need for uniformly strong ECC across the entire memory.

II. Proposed Strategy: InterA-ECC

Error-correction codes (ECCs) are widely used to address memory errors and are denoted as (n, k), where n is a codeword, and k is data bits [9]. The proposed InterA-ECC provides non-uniform error correction by dividing the subarray into hot and cold zones. The hot zone comprises the rows situated farther from the write driver and is protected by a stronger error correction than the cold zone. This paper aims to target both random and location-dependent failures by using simpler ECC in the cold zone and stronger ECC in the hot zone to address both random errors and interconnect effects.

II.A. InterA-ECC with distinct data block size for ECC

The cache block mapped to the hot zone is protected with SEC-DED Hamming code with a 32-bit block, while the cold zone operates with a 64-bit block, as shown in Fig. 1(a). The cold zone is protected with standard (72,64) code and requires a 64-bit (p_1=8×8) parity for a block of 512-bit, whereas the hot zone is protected with the (39,32) code, requires a 112-bit (p_2=16×7) parity. To avoid the extra uniform storage overhead for the entire subarray, the extra parity associated with the hot zone can be mapped to a separate bank for parallel access by leveraging a multi-bank memory architecture. The parity bits of the first eight 32-bit blocks (8×7=56) can be accessed in parallel, as we have 64-bit space. The remaining 56-bits can also be accessed in parallel, however, from the separate bank.

(a) InterA-ECC with distinct ECC data block size for hot and cold zone.

(b) InterA-ECC with multi-bit ECC ((144,128) DEC code) for hot zone.

Fig. 1: Demonstration of proposed InterA-ECC strategy.

II.B. InterA-ECC with strong multi-bit ECC

In this variant, the hot zone is protected with (79,64) double error correction (t=2) and triple error detection (DEC-TED) code, which requires 120-bits of parity (8×15), while the cold zone uses the (72,64) SEC-DED, which requires 64-bit parity for a 512-bit cache block. The DEC-TED code necessitates extra parity storage in the hot zone. The 60-bit (4×15) parity corresponding to the first four 64-bit data blocks can be mapped to available 64-bit space, while the remaining 60-bits can be mapped to a subarray in a separate parity bank, similar to shown in Fig. 1(a). Furthermore, the extra memory overhead associated with the hot zone can be removed by considering a larger data block size (128-bit) with double error correction (DEC) code in the hot zone, as shown in Fig. 1(b).

978-3-9826741-0-0/DATE25/© 2025 EDAA

III. Results and Evaluation

III.A. Simulation details

We consider a $1MB$ memory organization with 16 banks, with a cache block size of 512-bits and a subarray size of 256 Rows and 256 Columns. To evaluate reliability, we assess the block error rate (block failure probability). For n-bits block, under a bit error rate (BER: P_{bit}) protected by t-bit ECC, the probability of success (P_S) can be given by Equation (1). The block error rate (P_{Block}) can be obtained by Equation (2), where a large cache block is divided into multiple blocks (n_B), and each block is safeguarded by its own (n, k) ECC.

$$P_S(n,t) = \sum_{i=0}^{t} \binom{n}{i} P_{bit}^{i}(1-P_{bit})^{n-i} \quad (1)$$

$$P_{Block}(n,t) = 1 - \prod_{j=1}^{n_B} P_{S_j}(n,t) \quad (2)$$

III.B. Performance evaluation of the proposed InterA-ECC using distinct ECC data block sizes in the hot and cold zone

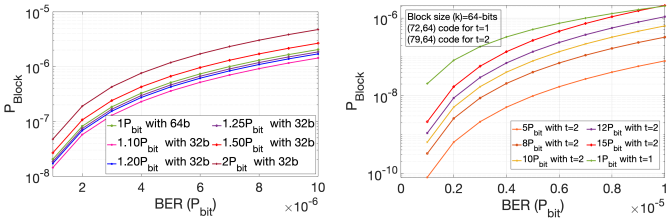
Fig. 2(a) shows the impact of ECC data block size on BER. As shown in Fig. 2(a), even with a 25% increase in BER, the SEC-DED with a 32-bit block maintains the lower block error rate as compared to SEC-DED with a 64-bit block size at the original BER (P_{bit}). Even up to BER of $1.5\times P_{bit}$, the SEC-DED with 32-bit nearly achieves the P_{BER} of a 64-bit block with the original BER (P_{bit}). Fig. 3(a) shows the average block error rate reduction in the memory using InterA-ECC with the smaller data block size for the hot zone compared to the (72,64) code while considering $N_{H_{Row}}$=32 as the number of rows under the hot zone. Even considering the bit error rate in the hot zone is five times ($5\times P_{bit}$) higher than in the cold zone, the proposed strategy can reduce the average block error rate by up to $\sim$33% while adding minimal memory overhead, only 1.56% higher than the (72,64) scheme, as shown in Table I.

III.C. Performance evaluation of the proposed InterA-ECC with multi-bit ECC for the hot zone

Fig. 2(b) shows that applying (79,64) DEC-TED to the hot zone maintains a block error rate in the hot zone nearly equal to that of the (72,64) SEC-DED at the original BER (P_{bit}), even when considering the BER in the hot zone $15\times P_{bit}$ higher. Protecting the hot zone with (79,64) DEC-TED also requires extra memory overhead compared to (72,64) code, as shown in Table. I ('Extra MO'). However, the total memory overhead ('Total MO') is still much lower (up to 40% reduction in ECC parity bits) than applying uniform DEC-TED. Furthermore, the DEC with a 128-bit block in the hot zone can achieve up to 10× higher BER resiliency while having the same 12.5% memory overhead as the (72,64) code. Fig. 3(b) shows the average block error rate reduction in the memory using InterA-ECC with multi-bit ECC while considering $N_{H_{Row}}$=32 number of rows under the hot zone.

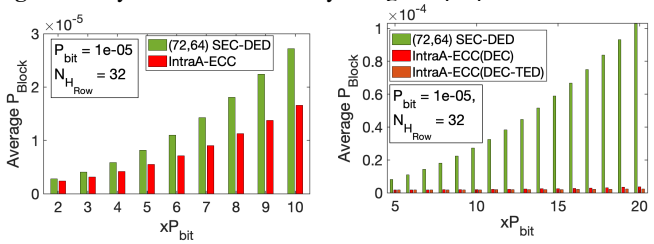
TABLE I: Memory overhead (MO) using InterA-ECC. $N_{H_{Row}}$: # rows under the hot zone, $N_{P_{Sub}}$: # subarrays in the parity bank.

Proposed: InterA-ECC				(72,64)	(39,32)	(79,64)
$N_{H_{Row}}$	$N_{P_{Sub}}$	Extra MO	Total MO	MO	MO	MO
32	2	1.56%	14.06%	12.5%	21.87%	23.43%
64	4	3.12%	15.62%			



(a) Distinct ECC data block size **(b)** Multi-bit ECC for the hot zone

Fig. 2: Analysis of BER resiliency using the proposed InterA-ECC.



(a) Smaller block size for the hot zone **(b)** Multi-bit ECC for the hot zone

Fig. 3: Average block error rate reduction vs multiplicands of P_{bit} in memory using the proposed InterA-ECC compared to (72,64) code. P_{bit}: BER in the cold zone, $\times P_{bit}$: BER in the hot zone.

IV. Conclusion

In this paper, we have proposed an efficient interconnect-aware error correction for STT-MRAM accounting for the increase of resistive-capacitive parasitics of the interconnect, specifically as technology scales down. The strategy divides the memory subarray into two zones: a cold zone (closer to the write driver) and a hot zone (farther from the driver), each with different error correction strengths. We have introduced two variants of the proposed InterA-ECC strategy: one using smaller data block sizes for ECC and the other employing strong multi-bit ECC for the hot zone. The proposed strategy improves memory reliability while avoiding uniform memory overhead by applying stronger error correction only where it is most needed. The proposed method is highly adaptable and can also be applied to other memory technologies.

References

[1] T. Evenblij *et al.*, "A comparative analysis on the impact of bank contention in stt-mram and sram based llcs," in *2019 IEEE 37th ICCD*, 2019, pp. 255–263.

[2] S. Sakhare *et al.*, "Enablement of stt-mram as last level cache for the high performance computing domain at the 5nm node," in *2018 IEEE IEDM*, 2018, pp. 18.3.1–18.3.4.

[3] S. Hemaram *et al.*, "Soft and hard error-correction techniques in stt-mram," *IEEE Design & Test*, vol. 41, pp. 65–82, 2024.

[4] G. Bonilla *et al.*, "Interconnect scaling challenges, and opportunities to enable system-level performance beyond 30 nm pitch," in *2020 IEEE IEDM*, 2020, pp. 20.4.1–20.4.4.

[5] V. B. Naik *et al.*, "Extended mtj tddb model, and improved stt-mram reliability with reduced circuit and process variabilities," in *2022 IEEE IRPS*, 2022, pp. 6B.3–1–6B.3–6.

[6] S. Hemaram *et al.*, "Hard error correction in stt-mram," in *2024 29th ASP-DAC*, 2024, pp. 752–757.

[7] Y.-D. Chih *et al.*, "13.3 a 22nm 32mb embedded stt-mram with 10ns read speed, 1m cycle write endurance, 10 years retention at 150°c and high immunity to magnetic field interference," in *2020 IEEE ISSCC*, 2020, pp. 222–224.

[8] Y.-C. Shih *et al.*, "Logic process compatible 40nm 16mb, embedded perpendicular-mram with hybrid-resistance reference, sub-μ a sensing resolution, and 17.5ns read access time," in *2018 IEEE Symposium on VLSI Circuits*, 2018, pp. 79–80.

[9] R. W. Hamming, "Error detecting and error correcting codes," *The Bell System Technical Journal*, vol. 29, pp. 147–160, 1950.