# Methodology

The simulator generates each cell's endurance (an endurance map) based on the distribution (mean=10^8) and CoV under test (0.15, 0.20, 0.25, 0.30, 0.35).

We model the memory as pages. The page is continuously written (one row/cacheline at a time), and the number of write cycles is recorded. The lifetime of a page reported is the number of write cycles to any row in the page when the page fails. To generalize to the entire memory from a page, the memory's lifetime is approximately the worst lifetime (1/1000) times the number of pages assuming the ideal page-level WL is applied. The reason is that the memory fails once the worst page fails.

The simulator also counts the number of writes to each cell, including the data cells and the auxiliary bits. When the number of writes to a cell exceeds its endurance, we regard the cell as a hard fault (wear-related).

We also provide an evaluation when there are fabrication defects (non-wear-related). In our experiments, there are 3, 5, 7, 9 faults per page.

Our system uses 4KB pages, 512-bit cache lines (64 lines per page), and employs a 4MB 16-way set associative LLC. The memory frequency is 1600 MHz.

We generate two types of endurance maps. Each type has 1000 maps. The first type is a normal distribution; the second type, based on the first type, also considers the local clusters of faults (due to wordline/bitline crosstalk).

Each row can tolerate 0-3 faults by attached fault tolerance capabilities. The results are shown for two (2G), four (4G) gaps, and eight (8G) gaps.

We use both Monte Carlo simulation and benchmark-based evaluation to simulate our memory. For Monte Carlo simulation, the writes are concentrated on 10 rows on each page to mimic a degree of write imbalance. For benchmark-based evaluation, we use a combination of address traces from nine representative SPEC CPU 2017 benchmarks: deepsjeng, gcc, lbm, mcf, nab, namd, omnetpp, perlbench, and wrf. These traces were recorded using Pintool. For row-level wear leveling schemes, the lifetime reported of each scheme are as follows:

A. Row sparing (RS) replaces the rows with unrecoverable faults with some spare rows (e.g., G2/G4/G8) to extend memory lifetime. In our experiments, for RS, a row is replaced by a spare row when there are any unrecoverable faults that cannot be tolerated by the attached fault tolerance scheme. A page fails when a new row with unrecoverable faults emerges, but there are no spare rows left.

B. Start Gap (SG) uses a “gap” row and algebraic mapping from logical to physical rows for wear-leveling [1]. For SG+RS, the writes to a page are balanced by the SG scheme. The swapping is conducted every 100 writes. On top of SG, RS is applied.

C. ARS is an aging-aware row-swapping scheme to balance the write count across all rows in crossbars [2]. For ARS+RS, the number of writes to each row is recorded by the counters. One-eighth of the hottest rows are swapped with one-eighth of the coldest rows every 1024 writes. On top of ARS, RS is applied.

D. WoLFRaM remaps the accessed row/subarray to a randomly selected row/subarray at a given probability to make the write distribution uniform [3]. The addresses for remapping have to be recorded. For WoLFRaM+RS, the accessed row has a probability of 1% being swapped with another randomly selected row. On top of WoLFRaM, RS is applied.

E. For RETROFIT, spare rows are utilized to quarantine weak rows with more faults according to RETROFIT design. A page fails when the number of rows with unrecoverable faults is larger than the number of spare rows. We set m=100 and n=10 for RETROFIT, which are the values of imbalanced write cycles.

F. For RETROFIT_GPS, GPS activates once RETROFIT fails to protect a page. A page fails when a new unrecoverable fault emerges, and the number of retired rows with PPPs equals the number of spare rows, and all the PPPs are used up.

G. For UWL+RS, the writes are issued to the rows in turn. On top of UWL, RS is applied.

In our experiments, we use three access patterns to mimic uneven bit flips within a row from previous works. These patterns include periodic high-/low-frequency bit flips, bit flips biased to LSBs rather than MSBs, and bit flip concentration on partially continuous cells. For these three patterns, the most frequently written bit has as much as 1.5x, 4.0x, and 8.0x average writes, respectively. We use these indicating numbers of unevenness as weights and report the weighted lifetimes of the schemes. Except for RETROFIT_CL_WL and RETROFIT_GPS_CL_WL, spare rows are only utilized to replace the rows with unrecoverable faults. The lifetime reported of each scheme are as follows:

A. Byte-level Rotation (BLR) shifts a row in a round-robin style in byte granularity [4]. In our experiments, for SG+BLR, the gap moves to its neighbor row every 100 writes, and each row is shifted every 256 writes. On top of SG+BLR, RS is applied.

B. Intra-row Flipping (ILF) utilizes a simple encoding method to flip the bits in several granularities [5]. For SG+ILF, the gap moves to its neighbor row every 100 writes, and each row is flipped, according to enhanced ILF, every 256 writes. On top of SG+ILF, RS is applied.

C. Horizontal Wear Leveling (HWL) shifts a row in bit level and achieves a very close result to uniform WL [6]. For SG+HWL, whenever the gap moves to its neighbor row, every 100 writes, the neighbor row is copied with the row rotated by 1 position. On top of SG+HWL, RS is applied.

D. For RETROFIT and its variants, whenever the moving gaps move to their neighbor rows, the neighbor rows are copied with the rows rotated by 1 position.

Reference:

[1] Qureshi, Moinuddin K., John Karidis, Michele Franceschini, Vijayalakshmi Srinivasan, Luis Lastras, and Bulent Abali. "Enhancing lifetime and security of PCM-based main memory with start-gap wear leveling." *In 2009 42nd Annual IEEE/ACM international symposium on microarchitecture (MICRO)*, pp. 14-23. IEEE, 2009.

[2] Cai, Yi, Yujun Lin, Lixue Xia, Xiaoming Chen, Song Han, Yu Wang, and Huazhong Yang. "Long live TIME: Improving lifetime and security for NVM-based training-in-memory systems." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 39, no. 12 (2020): 4707-4720.

[3] Yavits, Leonid, Lois Orosa, Suyash Mahar, João Dinis Ferreira, Mattan Erez, Ran Ginosar, and Onur Mutlu. "WoLFRaM: Enhancing wear-leveling and fault tolerance in resistive memories using programmable address decoders." *In 2020 IEEE 38th International Conference on Computer Design (ICCD)*, pp. 187-196. IEEE, 2020.

[4] Zhou, Ping, Bo Zhao, Jun Yang, and Youtao Zhang. "A durable and energy efficient main memory using phase change memory technology." *ACM SIGARCH computer architecture news* 37, no. 3 (2009): 14-23.

[5] Zhao, Mengying, Liang Shi, Chengmo Yang, and Chun Jason Xue. "Leveling to the last mile: Near-zero-cost bit level wear leveling for PCM-based main memory." In *2014 IEEE 32nd International Conference on Computer Design (ICCD)*, pp. 16-21. IEEE, 2014.

[6] Young, Vinson, Prashant J. Nair, and Moinuddin K. Qureshi. "DEUCE: Write-efficient encryption for non-volatile memories." *ACM SIGARCH Computer Architecture News* 43, no. 1 (2015): 33-44.