Understanding and Leveraging the Spatial Variation in Read Disturbance Vulnerability of Real DRAM Chips

1. Motivation and Limitations of State-of-the-Art

Read disturbance is a widespread weakness in modern DRAM chips, which can compromise memory security [1, 2]. RowHammer [1] and RowPress [2] are two prime examples where repeatedly accessing (hammering) or keeping active (pressing) data within a specific location may perturb nearby data, respectively. Unfortunately, shrinking manufacturing technology node size exacerbates read disturbance [3], and existing solutions exhibit limited effectiveness [4, 5]. To advance memory safety, an enhanced comprehension of read disturbance is critical. While prior work [6] explores different aspects of read disturbance, none rigorously analyzes its spatial variability. This paper investigates the variation in read disturbance across rows and leverages this understanding to enhance existing measures.

2. Experimental Methodology

Infrastructure. Our testing setup has four key components: 1) an FPGA development board (Xilinx Alveo U200 [7] for DIMMs or Bittware XUSP3S [8] for SODIMMs) programmed with DRAM Bender [9] for executing test programs. 2) a host machine for generating test programs and collecting results. 3) a thermocouple temperature sensor and heater pads in contact with DRAM chips to attain desired temperatures. 4) a PID temperature controller (MaxWell FT200 [10]) maintaining precise temperature control ($\pm 0.1\,^{\circ}$C).

Measurements. We measure two metrics: 1)the fraction of DRAM cells that exhibit bitflips, i.e., bit error rate (*BER*) and 2) the minimum hammer count required to induce the first bitflip (HC_{first}). We observe bitflips at the circuit level with minimal interference by adopting established methods [2, 3, 6, 11].

Tested DRAM Chips. Table 1 displays details of 136 real DDR4 DRAM chips (14 modules) from major manufacturers, encompassing diverse die densities and revisions. To account for in-DRAM row address mapping [12, 13], we employ prior methods [2, 3, 6] to reverse-engineer the physical layout.

Table 1: Tested DDR4 DRAM Chips.

Mfr.	DIMMs	Chips	Density	Die Rev.	Org.	Date
Mfr. H (SK Hynix)	1	8	16Gb	A	x8	2051
	3	8	16Gb	C	x8	2048
Mfr. M (Micron)	1	16	8Gb	B	x4	N/A
	1	4	16Gb	B	x16	2126
	2	16	16Gb	E	x4	2014
	1	4	16Gb	E	x16	2046
Mfr. S (Samsung)	2	8	8Gb	B	x8	2053
	1	16	8Gb	C	x4	2135
	1	8	8Gb	D	x8	2110
	1	8	4Gb	F	x8	N/A

3. Real DRAM Chip Characteristics

To understand the spatial variation of rows with high and low *BER*s, we investigate their positions within their respective banks. Fig. 1 shows the change in *BER* (y-axis) with increasing row addresses (x-axis). To make the variation trends more visible among the existence of large differences across DRAM modules, we normalize the *BER* of each DRAM row to the minimum *BER* observed in the same DRAM module. Each curve represents a different DRAM module, and each subplot is dedicated to a different manufacturer. Shades around the curves show the variation across different DRAM banks.

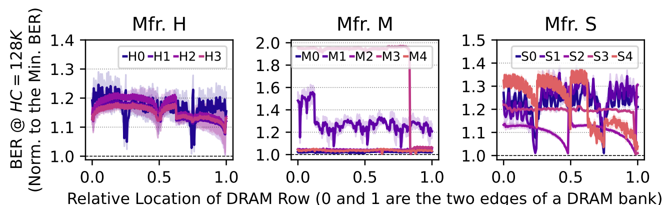

Figure 1: Distribution of *BER* across DRAM rows and bank groups

We make two observations from Fig. 1. First, *BER* curves exhibit a periodic pattern across rows, oscillating between distinct levels, e.g., 1.31 and 1.30 within the first 20% of the banks in S4. Second, the average (minimum/maximum) normalized *BER* across DRAM rows can significantly vary across chunks of the bank, e.g., 1.31 (1.10/1.39) and 1.12 (1.00/1.34) for S4's first 62% and last 38%, respectively.

Takeaway 1. *The variation in BER across different DRAM rows exhibits repeating patterns, and certain chunks of rows can exhibit higher BER than the rest of the rows.*

We investigate the variation in HC_{first} across DRAM rows. To do so, we repeat our tests at 14 different hammer counts from 1K to 128K and measure *BER*. Fig. 2 shows the distribution of HC_{first} values across DRAM rows. Each subplot shows the distribution for a different manufacturer. The x-axis shows the HC_{first} values, and the y-axis shows the fraction of the DRAM rows with the specified HC_{first} value. Different colors represent different modules, and the error bars mark the minimum and the maximum of a given value across tested banks. The red vertical dashed line marks the minimum HC_{first} that we observe across all rows in tested modules from a manufacturer.

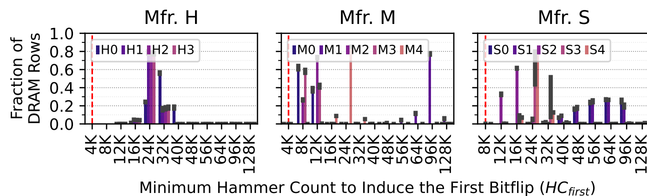

Figure 2: Distribution of HC_{first} across DRAM rows and banks

We make two observations from Fig. 2. First, HC_{first} significantly varies across rows in a bank but not across banks, e.g., S0 and S1 contain rows with HC_{first} values of 24K and 128K. Despite this large variation, the variation across banks is significantly low as shown by the error bars. Second, HC_{first} significantly varies across modules, e.g., HC_{first} values in M1 and M4 overlap at 32K, while the vast majority of their distributions concentrate at distinct values of 24K and 96K.

Takeaway 2. *HC_{first} significantly varies across different DRAM rows and different DRAM modules, while different banks in a DRAM module exhibit similar distributions.*

We investigate the effect of RowPress [2] on HC_{first} distribution across DRAM rows in Fig. 3. We find HC_{first} distributions (y-axis) for three levels of the time that an aggressor row stays active (t_{AggOn}) (x-axis).

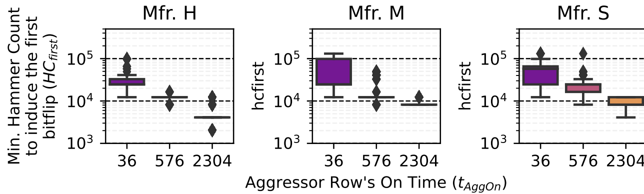

Figure 3: Effect of RowPress on HC_{first} Distribution across Rows

We make two observations from Fig. 3. First, bitflips occur at lower hammer counts for higher t_{AggOn} values across all manufacturers. Second, the standard deviation in HC_{first} across rows does *not* monotonically decrease, e.g., it increases from 1038.6 at 576ns to 3045.7 at 2304ns.

Takeaway 3. *HC_{first} reduces as t_{AggOn} increases but the variation in HC_{first} across DRAM rows does* not *always reduce.*

4. Leveraging the Variation in Read Disturbance

To reduce the performance overheads of existing read disturbance mitigation mechanisms, we leverage the variation in read disturbance vulnerability of different DRAM rows to tune the aggressiveness of existing read disturbance mitigation mechanisms dynamically. Upon activating a DRAM row, the read disturbance mechanism uses the row's HC_{first} value to decide to perform a preventive action by modifying its hammer count or probability threshold. Storing per-row HC_{first} metadata is feasible using a memory controller's SRAM scratchpad or DRAM's data integrity bits. In either case, *only* four bits per DRAM row suffice to cluster rows into one of 14 different HC_{first} bins, i.e., aggressiveness levels.

5. Performance Evaluation

We evaluate our proposal's performance benefits using Ramulator [14] to simulate a 16-bank dual-rank DDR4 DRAM module in cycle level. Our evaluation encompasses three mitigation mechanisms: Hydra [15], PARA [1], and RRS [16]. We simulate 8-core multiprogrammed workloads from SPEC CPU2006 and CPU2017, TPCWeb, MediaBench, and YCSB benchmark suites. We scale the vulnerability profiles of real DRAM chips to HC_{first} of 128.

We make two key observations. First, our proposal significantly enhances system performance, e.g., by 2.7%, 57.8%, and 2.98x on average across 50 workloads and variation profiles from three manufacturers, compared to Hydra [15], PARA [1], and RRS [16], respectively. Second, our proposal reduces RRS's performance overhead (74.3%) down to 54.6%, 24.2%, and 1.7% for Mfrs H, M, and S, respectively.

Takeaway 4. *Our proposal effectively mitigates performance degradation caused by read disturbance mitigation mechanisms in a system, depending on the read disturbance vulnerability variation of the DRAM module.*

6. Conclusion and Future Work

Our proposal reduces the performance degradation caused by existing RowHammer defense mechanisms by leveraging the spatial variation of read disturbance vulnerability across different memory locations within a memory module. To do so, we 1) characterize the spatial variation in read disturbance in 136 real DRAM chips, and 2) propose dynamically adapting the aggressiveness of existing solutions based on the variation profile. By learning and leveraging this large variation, our proposal reduces the performance and energy overheads of state-of-the-art RowHammer defenses. Our proposal's effectiveness and wide adoption can be improved in three ways: 1) fast and accurate online profiling techniques, 2) maintaining the read disturbance profile at a low area and latency overhead, and 3) showcasing our proposal on a wider range of existing mechanisms. We leave these for future work.

References

[1] Y. Kim, R. Daly, J. Kim, C. Fallin, J. H. Lee, D. Lee, C. Wilkerson, K. Lai, and O. Mutlu. Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors. In *ISCA*, 2014.

[2] Haocong Luo, Ataberk Olgun, Abdullah Giray Yağlıkcı, Yahya Can Tuğrul, Steve Rhyner, Meryem Banu Cavlak, Joël Lindegger, Mohammad Sadrosadati, and Onur Mutlu. RowPress: Amplifying Read Disturbance in Modern DRAM Chips. In *ISCA*, 2023.

[3] Jeremie S. Kim, Minesh Patel, Abdullah Giray Yağlıkcı, Hasan Hassan, Roknoddin Azizi, Lois Orosa, and Onur Mutlu. Revisiting RowHammer: An Experimental Analysis of Modern Devices and Mitigation Techniques. In *ISCA*, 2020.

[4] Pietro Frigo, Emanuele Vannacci, Hasan Hassan, Victor van der Veen, Onur Mutlu, Cristiano Giuffrida, Herbert Bos, and Kaveh Razavi. TRRespass: Exploiting the Many Sides of Target Row Refresh. In *S&P*, 2020.

[5] Hasan Hassan, Yahya Can Tugrul, Jeremie S. Kim, Victor van der Veen, Kaveh Razavi, and Onur Mutlu. Uncovering In-DRAM RowHammer Protection Mechanisms: A New Methodology, Custom RowHammer Patterns, and Implications. In *MICRO*, 2021.

[6] Lois Orosa, A Giray Yağlıkcı, Haocong Luo, Ataberk Olgun, Jisung Park, Hasan Hassan, Minesh Patel, Jeremie S. Kim, and Onur Mutlu. A Deeper Look into RowHammer's Sensitivities: Experimental Analysis of Real DRAM Chips and Implications on Future Attacks and Defenses. In *MICRO*, 2021.

[7] Xilinx Inc. Xilinx Alveo U200 FPGA Board. `https://www.xilinx.com/products/boards-and-kits/alveo/u200.html`.

[8] Bittware XUSP3S FPGA Board. `https://www.bittware.com/fpga/xus-p3s/`.

[9] Ataberk Olgun, Hasan Hassan, A. Giray Yağlıkcı, Yahya Can Tuğrul, Lois Orosa, Haocong Luo, Minesh Patel, Ergin Oğuz, and Onur Mutlu. DRAM Bender: An Extensible and Versatile FPGA-based Infrastructure to Easily Test State-of-the-art DRAM Chips. arXiv:2211.05838, 2022.

[10] Maxwell. FT20X User Manual. `https://www.maxwell-fa.com/upload/files/base/8/m/311.pdf`.

[11] A. Giray Yağlıkcı, Haocong Luo, Geraldo F De Oliviera, Ataberk Olgun, Minesh Patel, Jisung Park, Hasan Hassan, Jeremie S Kim, Lois Orosa, and Onur Mutlu. Understanding RowHammer Under Reduced Wordline Voltage: An Experimental Study Using Real DRAM Devices. In *DSN*, 2022.

[12] B. Keeth and R.J. Baker. *DRAM Circuit Design: A Tutorial*. Wiley, 2001.

[13] Kiyoo Itoh. *VLSI Memory Chip Design*. Springer, 2001.

[14] Y. Kim et al. Ramulator: A Fast and Extensible DRAM Simulator. *CAL*, 2015.

[15] Moinuddin Qureshi, Aditya Rohan, Gururaj Saileshwar, and Prashant J Nair. Hydra: Enabling Low-Overhead Mitigation of Row-Hammer at Ultra-Low Thresholds via Hybrid Tracking. In *ISCA*, 2022.

[16] Gururaj Saileshwar, Bolin Wang, Moinuddin Qureshi, and Prashant J Nair. Randomized Row-Swap: Mitigating Row Hammer by Breaking Spatial Correlation Between Aggressor and Victim Rows. In *ASPLOS*, 2022.