**An Experimental Analysis of RowHammer in HBM2 DRAM Chips**

Ataberk Olgun¹ Majd Osseiran¹,² A. Giray Yaşkıçlı¹ Yahya Can Tuğrul¹
Haocng Luo¹ Steve Rhynner¹ Behzad Salami¹ Juan Gomez Luna¹ Onur Mutlu¹
¹SAFARI Research Group, ETH Zürich ²American University of Beirut

RowHammer (RH) is a significant and worsening security, safety, and reliability issue of modern DRAM chips that can be exploited to break memory isolation. Therefore, it is important to understand RH vulnerabilities. Unfortunately, no prior work extensively studies the RH vulnerability of modern 3D-stacked high-bandwidth memory (HBM) chips, which are commonly used in modern GPUs.

In this work, we experimentally characterize the RH vulnerability of a real HBM2 DRAM chip. We show that different 3D-stacked channels of HBM memory exhibit significantly different levels of RH vulnerability. We present the first detailed experimental characterization of the RH vulnerability of modern HBM2 DRAM chips.

1. **Introduction**

   Modern DRAM chips suffer from the RowHammer (RH) phenomenon [1–5] where repeatedly opening (i.e., activating) and closing a DRAM row (i.e., aggressor row) induces bitflips in physically nearby rows (i.e., victim rows), breaking memory isolation [1,2,4–27]. Numerous studies experimentally demonstrate that a malicious attacker can reliably induce RH bitflips in a targeted manner to compromise system integrity, confidentiality, and availability [1,2,15,16,25,27–85]. RH worsens in newer DRAM chips with smaller technology nodes, where RH bitflips (up to 10× reduction in less than a decade [4]) and manifest in more DRAM cells, compared to older DRAM chips [2,4,5,14,17,25–27].

   To meet the high bandwidth requirements of modern data-intensive applications (e.g., GPU workloads) [86–89], DRAM designers develop High Bandwidth Memory (HBM) [90] DRAM chips, which contain multiple layers of 3D-stacked DRAM dies, using cutting-edge technology nodes. It is important to understand RH in HBM DRAM chips that have new architectural characteristics (e.g., multiple layers of DRAM dies and area- and energy-intensive through-silicon vias) which might potentially affect the RH vulnerability in various ways. Such understanding can help identify potential RH-induced security, safety, and reliability issues in HBM-based systems and allow system designers to develop effective and efficient defense mechanisms. Unfortunately, no prior work studies the RH vulnerability of modern HBM DRAM chips.

   **Our goal** in this work is to experimentally analyze how vulnerable HBM DRAM chips are to RH. To this end, we provide the first detailed experimental characterization of the RH vulnerability in a modern HBM2 DRAM chip. We provide two main analyses in our study. First, we analyze the spatial variation in RH vulnerability based on the physical location of victim rows in terms of two metrics: the fraction of DRAM cells that experience a bitflip in a DRAM row (i.e., Bit Error Rate, BER) and the minimum aggressor row activation count necessary to cause an RH bitflip (i.e., HC_first). §4 shows detailed results on how RH vulnerability varies across HBM2 channels, pseudo channels, banks, and rows. Second, we investigate undisclosed in-DRAM RH mitigations, triggered by periodic refreshes (e.g., TRR) in HBM2 (§5).

   We summarize the three major observations from our experimental analyses. First, different 3D-stacked channels of the HBM2 chip exhibit significantly different levels of RH vulnerability in BER (up to 79%) and HC_first (up to 20%). Second, DRM rows near the end of a DRAM bank (the last 832 out of 16k rows) exhibit substantially smaller BER than other DRAM rows. Third, the tested HBM2 DRAM chip implements an undisclosed in-DRAM RH defense mechanism (§5). Our experimental analyses also show that the RH vulnerability of a cell depends on i) the cell’s physical location within a DRAM bank and ii) data stored in the neighboring cells, similar to prior works’ findings in DDR3/4 DRAM chips [1,4,14].

   We make the following contributions:

   • We present the first detailed experimental characterization of the RowHammer (RH) vulnerability in a modern HBM2 DRAM chip and show that it is susceptible to RH bitflips.
   • We show that the RH vulnerability significantly varies across HBM2 channels and rows within each channel.
   • We show that an HBM2 DRAM chip implements an undisclosed in-DRAM RH mitigation mechanism, which resembles the one found in recent DDR4 DRAM chips manufactured by a major DRAM manufacturer [64].

2. **Background**

2.1. **HBM2 Organization & Operation**

Fig. 1 presents the organization of an HBM2 DRAM chip [90, 92] when used in an FPGA-based system. The FPGA (Φ) has a memory controller communicating with one or multiple stacks of HBM using the HBM2 interface via a silicon interposer. An HBM2 stack (Φ) contains multiple DRAM dies that are stacked on top of the buffer die and connected using through-silicon vias (TSVs). Each HBM2 die has one or more (shown two) independent HBM2 channels. An HBM2 channel (Φ) consists of two pseudo channels (Φ), each of which with multiple

---

¹The HBM2 standard [90] specifies a mode called Target Row Refresh (TRR). To enable TRR Mode, the memory controller issues a well-defined series of commands. Different from this mode, we investigate whether an undisclosed TRR mechanism is implemented in the tested chip.
We experimentally study an HBM2 chip using a modified version of the host machine to retrieve a target temperature and finish within 27 ms, which is significantly smaller than the standard refresh period of 32 ms where manufacturers guarantee no data retention errors will occur [90]. Fourth, we disable ECC by setting the corresponding HBM2 mode register bit to zero [90].

**RH Access Pattern.** We use double-sided RH [1, 4, 14, 37], whereby we alternate activations to each of the aggressor rows neighboring a victim row. To find which rows are physically adjacent, we reverse-engineer the logical (memory-controller-visible) to physical (in-DRAM) row address mapping [1, 14, 32, 50, 59, 102, 104, 108, 111–118], following the methodology described in prior work [14].

**RH Test Parameters.** We define one hammer as a pair of activations to the two aggressor rows. We measure two metrics in our tests: BER and $HC_{\text{first}}$, as defined in §1. We use 256K hammers (i.e., 512K activations) in our BER experiments and up to 256K hammers in our $HC_{\text{first}}$ experiments. We repeat both experiments for each of the four data patterns shown in Table 1. We define the worst-case data pattern ($WCDP$) as the data pattern that causes the smallest $HC_{\text{first}}$ for a given row. When multiple data patterns cause the smallest $HC_{\text{first}}$, we select $WCDP$ as the data pattern that causes the largest BER at a hammer count of 256K. To maintain a reasonable experiment time, we study the effects of RH on the first, middle, and last 3K rows in a bank in every channel and repeat all experiments five times. The HBM2 chip’s temperature is kept at 85°C, the maximum operating temperature at the nominal refresh rate, in all of our experiments.

### 4. Spatial Variation Analysis

We provide the first spatial variation analysis of RH across HBM2 channels, pseudo channels, banks, and rows. Fig. 3 shows a box-and-whiskers plot depicting the distribution of BER (y-axis) across different DRAM rows for a given data pattern (x-axis) in a channel (color-coded). In addition to the four data patterns, we choose and plot the BER for the $WCDP$ of each row.

We make four major observations from Fig. 3. First, RH bitflips occur in every tested DRAM row across all HBM channels. Second, BER varies across channels. A subset of channels

<table>
<thead>
<tr>
<th>Table 1: Data patterns used in our RH tests</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Row Addresses</strong></td>
</tr>
<tr>
<td>Victim (V)</td>
</tr>
<tr>
<td>Aggressors (V ± 1)</td>
</tr>
<tr>
<td>V ± [2:8]</td>
</tr>
</tbody>
</table>

2 The lower- and upper-bounds of the box are the first and the third quartiles, marking the medians of the first and second half of the ordered set of data points, respectively. Whiskers show the minimum and maximum values. The circle marker in each box shows the distribution’s mean.
(channels 6 and 7) exhibit significantly higher BER than other channels. For example, channel 7 (with the highest BER) has $2.03 \times$ higher BER than channel 0 (with the lowest BER) for WCDP. Third, channels can be classified into groups of two based on the number of bitflips they exhibit. We highlight these groups using different shades of the same color in the figure. We hypothesize that groups of channels are spread across different HBM2 DRAM dies. The variation in BER across the channel groups could be due to manufacturing process variation (similar to that in DDR3/4 DRAM chips [1, 4, 6, 7, 14]). Fourth, BER changes with data pattern, e.g., the maximum BER in channel 7 is 3.13% and 2.04% for data patterns Rowstripe1 and Checkered0, respectively.

Fig. 4 shows the distribution of $HC_{first}$ (y-axis) across different DRAM rows for a given data pattern (x-axis) in a channel (color-coded) using a box-and-whiskers plot. In addition to the four data patterns, we choose and plot the $HC_{first}$ for the WCDP of each row.

**Figure 4:** The minimum aggressor row activation count necessary to cause a RowHammer bitflip ($HC_{first}$) across different DRAM rows, channels, and data patterns.

We make three major observations from Fig. 4. First, $HC_{first}$ is as low as 14531 across all tested channels and data patterns. Second, different channels exhibit different $HC_{first}$ distributions. For example, channels 7 and 6 contain more rows with smaller $HC_{first}$ values than other channels. Because these channels also exhibit a higher number of RH bitflips than other channels, we hypothesize that these channels belong in the die with the worst RH vulnerability across all dies. Third, the $HC_{first}$ distribution in a channel depends on the data pattern used. For example, the mean $HC_{first}$ for Rowstripe0 and Rowstripe1 in channel 0 are 57925 and 79179, respectively. We conclude that testing with different data patterns is necessary to assess the RH vulnerability of HBM2 DRAM chips as no data pattern achieves the smallest $HC_{first}$ or BER (Fig. 3).

Fig. 5 shows the BER (y-axis) for tested DRAM rows (x-axis) when we use the per-row WCDP to initialize the rows. Different channels are color-coded and different subplots show the three regions (first, middle, and last 3K rows) we test.

**Figure 5:** The fraction of DRAM cells that experience a bitflip in a DRAM row (BER) for different rows across a bank in different channels. Highlighted regions show individual DRAM subarrays.

We make two major observations from Fig. 5. First, the BER periodically increases and decreases across DRAM rows such that it is higher in the middle of a subarray and lower towards either end of the subarray. We hypothesize that this pattern results from the physical organization of the DRAM banks. For example, the RH vulnerability of a row could increase with the row’s distance from the row buffer. Second, significantly fewer bitflips occur in the last subarray of the bank (e.g., the last 832 rows in SA Z) compared to the rest of the bank. We hypothesize that this is due to the DRAM bank’s physical placement. For example, assuming that proximity to the shared I/O circuitry on the DRAM die affects the RH vulnerability of a subarray, the last subarray might be placed near this shared I/O circuitry [119]. We leave further analysis on subarrays to future work.

We examine RH vulnerability differences across banks by comparing the RH BER distribution over 300 rows (the first, the middle, and the last 100 rows) in each of the 256 banks (across 8 channels and two pseudo channels per channel). Fig. 6 shows each bank in a scatter plot based on its BER distribution: 1) the coefficient of variation (CV) $^3$ on the x-axis and 2) the mean on the y-axis. A marker’s color and shape indicate the bank’s channel and pseudo channel, respectively.

**Figure 6:** BER variation across banks. Each bank is represented by the average BER (y-axis) and the coefficient of variation in BER (x-axis) across the rows within the bank.

We make two major observations. First, there is variation in BER across banks and pseudo channels. For example, there is up to 0.23% difference in mean BER across banks in channel 7. Second, BER variation across banks is dominated by variation across channels. Banks in different channels tend to have a larger BER difference than banks in the same channel (Fig. 3). We conclude that testing different channels is more important than testing different banks or pseudo channels to assess the RH vulnerability of a given HBM2 DRAM chip.

---

$^3$We reverse engineer subarray boundaries by performing single-sided RH [1, 4] that induces bitflips in only one of the victim rows if the aggressor row is at the edge of a subarray. We find that a subarray contains either 832 (SA X in Fig. 5) or 768 (SA Y in Fig. 5) DRAM rows.

$^4$CV is a distribution’s standard deviation, normalized to its mean [120].
Summary. We summarize the two key takeaways of our spatial variation study. First, different 3D-stacked channels of the HBM2 DRAM chip exhibit significantly different levels of RH vulnerability in terms of BER (up to 79%) and HC_{first} (up to 20%). Second, DRAM rows near the end of a DRAM bank exhibit substantially smaller BER than other DRAM rows on average across all HBM2 DRAM channels. Our key takeaways have two implications for RH attacks and defenses. First, an RH attack can use the most-RH-vulnerable HBM2 channel to reduce the time it spends on i) preparing for an attack, by finding exploitable RH bitflips faster (i.e., by accelerating memory templating), and ii) performing the attack, by benefiting from a small HC_{first} value. Second, an RH defense mechanism can adapt itself to the heterogeneous distribution of the RH vulnerability across channels and subarrays, which may allow the defense mechanism to more efficiently prevent RH bitflips.

5. Uncovering in-DRAM RH Mitigations

To prevent RH bitflips, DRAM manufacturers equip their chips with a mitigation mechanism broadly referred to as Target Row Refresh (TRR) [25, 64, 91]. Proprietary versions of TRR (e.g., in DDR4) operate transparently from the perspective of the memory controller. At a high level, TRR identifies potential aggressor rows and preventively refreshes their victim rows upon receiving a REF command. We demonstrate that the tested chip implements a proprietary TRR (similar to the ones used in DDR4 [64, 121]) in addition to the TRR mode documented in the HBM2 standard [90].

Methodology. We use the U-TRR methodology [64, 121] to uncover the proprietary TRR mechanism. The key idea of this methodology is to use retention failures as a side channel to infer whether or not TRR refreshes a DRAM row.

One iteration of our experiment consists of six steps. First, we profile a row (R) to find its retention time (T), after which retention errors accumulate in row R unless row R is refreshed. Second, we activate and precharge row R once (i.e., refresh it) and wait for T/2. Third, we activate and precharge row R+1. We hypothesize that if the HBM2 chip implements an undisclosed TRR mechanism, it will sample the activation of R+1 and refresh the neighboring rows (i.e., row R) once a periodic REF command triggers TRR. Fourth, to trigger the TRR mechanism, we issue a periodic REF command. Fifth, we wait for another T/2 such that if TRR does not refresh row R, it will exhibit retention errors. Sixth, we check row R for bitflips to see if any TRR refresh occurred (i.e., if there are no bitflips). We perform 100 iterations of this experiment to check if TRR refreshes the victim row R.

Results. We observe that the profiled row (R) is refreshed once every 17 iterations. Therefore, we conclude that i) the tested HBM2 DRAM chip implements a proprietary, undisclosed TRR mechanism, and ii) this TRR mechanism performs a victim row refresh once every periodic REF commands. We use the U-TRR [64, 121] to uncover a proprietary TRR mechanism that U-TRR [64, 121] uncovers in DRAM chips fromVendor C. We intend to uncover more details of the proprietary TRR mechanism as part of future work.

6. Future Work

This work presents the results of our RH characterization study on a real HBM2 DRAM chip. We plan to present more insights into how RH behaves in real HBM2 DRAM chips by strengthening our characterization study at least in the following three directions.

1) Testing more HBM chips. We intend to repeat our experiments on a larger number of HBM2 chips to improve the statistical significance of our observations.

2) More characterization for each HBM chip. We plan to understand RH’s sensitivities to multiple other factors. We will investigate how RH varies: 1) across different stacks in HBM2 chips, 2) based on the time an aggressor row remains active [14], 3) based on a richer set of data patterns used in initializing victim and aggressor rows, and 4) across different HBM2 voltage and temperature levels. We also intend to study the effect of RowPress [110].

3) Investigating cross-channel interference. HBM2 chips stack DRAM dies such that certain HBM channels are placed on top of each other. We aim to investigate if frequently accessing one or more aggressor channels can induce bitflips or worsen the reliability characteristics of other victim channels.

7. Related Work

We present the first experimental characterization of the RowHammer (RH) vulnerability in a modern HBM2 DRAM chip. Prior works [1, 2, 4–27] analyze new aspects of the RH vulnerability by testing real DDR3/4 DRAM chips. Other prior works [122–124] characterize real HBM chips to understand their i) soft error resiliency [124], ii) performance and reliability characteristics under reduced voltage [122], and iii) susceptibility to retention failures at different temperatures [123]. No prior work investigates the RowHammer vulnerability in a real HBM chip.

8. Conclusion

We present the results of our detailed characterization study of the RowHammer (RH) vulnerability in a modern HBM2 chip. We show that the RH vulnerability is heterogeneously distributed across various components in the HBM2 chip, which we believe has important implications for future RH attacks and defenses. We discover that the HBM2 chip implements a proprietary RH mitigation mechanism and explain how the mitigation mechanism works. We hope and expect that our findings will lead to a deeper understanding of and new solutions to the RH vulnerability in HBM-Based systems.

Acknowledgments

We thank the anonymous reviewers of DSN 2023 Disrupt for feedback. We thank the SAFARI Research Group members for valuable feedback and the stimulating scientific and intellectual environment they provide. We acknowledge the generous gift funding provided by our industrial partners (especially Google, Huawei, Intel, Microsoft, VMware), which has been instrumental in enabling the decade-long research we have been conducting on read disturbance in DRAM. This work was in part supported by the Google Security and Privacy Research Award and the Microsoft Swiss Joint Research Center.