BLASTER: Characterizing the Blast Radius of Rowhammer

Zhenrong Lang Patrick Jattke Michele Marazzi Kaveh Razavi
ETH Zurich

***Abstract*—A decade after Rowhammer was first exposed, we are still learning about the intricacies of this vulnerability inside DRAM. Making things worse, the shrinking of technology nodes seems to expose new effects with significant implications for both attackers and defenders. One of these effects, known as *Half-Double*, shows that Rowhammer can affect *victim* rows located two rows away from the *aggressor* row. Understanding the impact of Half-Double is essential for the design of secure mitigations.**

We characterize Half-Double using 24 commodity DDR4 DRAM chips from the three major DRAM vendors. Furthermore, we introduce BLASTER as a generalization of the Half-Double access patterns, encompassing aggressor rows located multiple rows away from the victim. In particular, we show for the first time that BLASTER significantly reduces the number of necessary activations to the victim-adjacent aggressors using other aggressor rows that are up to *four rows away* from the victim. We discuss the implications of BLASTER on the design of future Rowhammer mitigations.

I. INTRODUCTION

The Rowhammer vulnerability [1] remains an issue in contemporary DDR4 DRAM devices despite all deployed mitigations [2, 3]. As the process technology of DRAM cells continues to shrink, devices become increasingly susceptible to Rowhammer, as notable by the decreasing *Rowhammer threshold* [4]. This threshold refers to the number of activations required for a single row to induce the first bit flip in its neighboring rows. However, the impact of a higher DRAM cell density is more far-reaching than the reduction in Rowhammer thresholds. A higher cell density increases the *blast radius*, i.e., the maximum physical distance of victim rows affected by a single aggressor. The increase in blast radius adds another challenge in mitigating Rowhammer. Specifically, the *Half-Double* pattern [5] exploits this effect by targeting victim rows that are two rows away from the aggressor.

To circumvent Rowhammer, manufacturers have incorporated mitigations known as *Target Row Refresh* (TRR) into their chips. TRR identifies repeatedly activated rows as potential aggressors and issues extra refreshes to their neighboring rows on top of the regular refreshes [3]. Recent studies [6] show that some vendors implement TRR by issuing refreshes to up to four neighboring rows, which suggests the vendors' concern about the escalating impact of a single aggressor row.

A careful characterization is necessary to better understand the interaction between existing and emerging Rowhammer effects and address them appropriately when designing new mitigations. However, no rigorous experimental study has investigated yet how the growing blast radius and new Rowhammer patterns manifest in modern DDR4 DRAM devices. In this work, we conduct an experimental analysis of **BLASTER patterns**, which generalize all patterns involving multiple *far* aggressors located at distances greater than one row from the victim. We experiment on three DDR4 DRAM devices, encompassing a total of 24 DRAM chips, from the three major DRAM vendors: SK Hynix, Micron and Samsung. We employ an FPGA-based testing infrastructure with all in-DRAM mitigations disabled to ensure accurate observations and measurements.

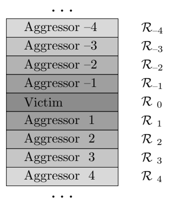


Fig. 1: **Row arrangement** of aggressors in our experiments. We conduct tests with aggressor rows positioned at up to four rows away from the victim, while varying the number of activations for each aggressor to determine their impact on the probability of bit flips in the victim row.

To address the expanding blast radius, we design experiments involving aggressor rows that are up to four rows apart from the victim row $\mathcal{R}_0$, i.e., including aggressors $\mathcal{R}_{-4}$ and $\mathcal{R}_4$, as illustrated in Fig. 1. We explore BLASTER with aggressors of varying distances to the victim to determine which patterns can trigger bit flips in the victim row. To investigate the effects of farther aggressors, we denote by HC^* the maximum number of activations to the near aggressors, that a victim row can sustain *without* triggering any bit flips. In other words, if the near aggressors ($\mathcal{R}_{-1}$ and/or $\mathcal{R}_1$) are activated HC^* times, depending on whether single- or double-sided patterns are used, *no* bit flips occur in the victim row. By bounding the number of activations to the near aggressors by HC^*, we can accurately analyze the influence of more distant aggressors on the victim row.

Our experimental results indicate that while HC^* activations on aggressor $\mathcal{R}_1$ do *not* cause any bit flips in the single-sided case, it is possible to trigger a bit flip with significantly fewer activations on aggressor $\mathcal{R}_1$ when assisted by activations to aggressor $\mathcal{R}_2$. Furthermore, we discovered that not all aggressor rows between the farthest aggressor and the victim need to be activated to induce bit flips in the victim row. For example, the effect of aggressor $\mathcal{R}_4$ ($\mathcal{R}_{-4}$) can propagate to the victim row without the need for aggressors $\mathcal{R}_2$ ($\mathcal{R}_{-2}$) and $\mathcal{R}_3$ ($\mathcal{R}_{-3}$) to be activated.

Contributions. The following summarizes our contributions:

1) We rigorously assess the Half-Double case, which only involves aggressors $\mathcal{R}_1$ ($\mathcal{R}_{-1}$) and $\mathcal{R}_2$ ($\mathcal{R}_{-2}$), of our more generalized BLASTER patterns across three DDR4 DRAM devices from the three major DRAM manufacturers.
2) We study BLASTER patterns involving aggressor rows positioned up to four rows away from the victim row (ranging from aggressors $\mathcal{R}_{-4}$ to $\mathcal{R}_4$) and demonstrate their impact on obtaining bit flips in the victim.
3) We systematically characterize different BLASTER patterns by varying the number of activations of aggressor rows.

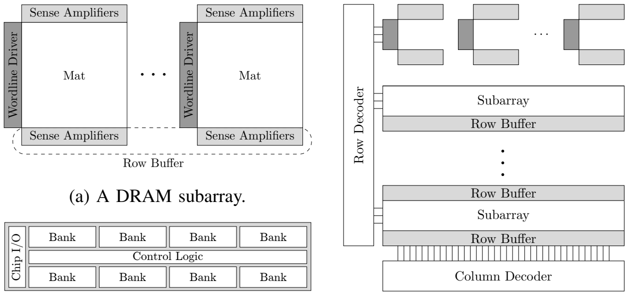
II. Background

In this section, we provide an overview of the DRAM organization (§II-A) in a bottom-up approach and explain the operation of DRAM chips. Then, we introduce Rowhammer (§II-B) while focusing on existing patterns and mitigations.

A. DRAM Organization

A DRAM *cell* consists of a capacitor and an access transistor. The cells are organized in grid-like DRAM *mats*. The cells within the same row are interconnected through a wordline, and those in the same column share a bitline. The DRAM mats with the same wordlines form a DRAM *subarray* (Fig. 2a). Each row passes through all the mats, and an array of sense amplifiers horizontally forms a row buffer, which separates different DRAM subarrays.

A DRAM *bank* (Fig. 2b) comprises multiple DRAM subarrays and a circuitry responsible for decoding DRAM addresses. Each DRAM *chip* (Fig. 2c) consists of several DRAM banks. Multiple chips are mounted on DIMMs, which are connected to the CPU's memory controller. All chips on a DDR4 DIMM share the same command/address (CA) bus. When data is read from or written to DRAM, all the chips are activated, with each chip being associated only with a portion of the data.



(a) A DRAM subarray.

(c) A DRAM chip.

(b) A DRAM bank.

Fig. 2: **Architectural view** of a DRAM subarray (a), DRAM bank (b), and DRAM chip (c).

B. Rowhammer

Rowhammer is a DRAM vulnerability where repeated *activations* of specific rows, known as *aggressor* rows, can cause bit flips in their physically neighboring rows, referred to as *victim* rows. The severity of this vulnerability has been exacerbated by the shrinking processes employed in manufacturing DRAM chips.

Rowhammer Patterns. The original Rowhammer work by Kim et al. [1] uses *single-sided* patterns (Fig. 3-❶), where a single aggressor row is activated, and the victims are the adjacent rows above and below it. Subsequent work [7] showed that the underlying mechanism of Rowhammer, namely charge leakage, is amplified when a victim row is subject to alternating activations from both of its direct neighboring aggressor rows. This is referred to as a *double-sided* pattern (Fig. 3-❷). More recently, the discovery of *Half-Double* patterns (Fig. 3-❸) revealed that the influence of an aggressor extends beyond its immediate neighboring rows [5]. In these patterns, in addition to the two near aggressor rows (N), far aggressors (F) that are two rows away from the victim are also activated.

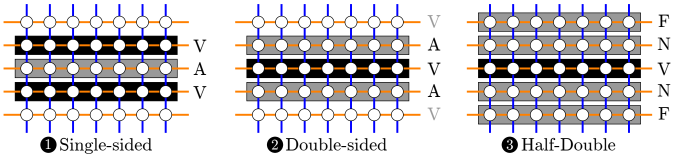


Fig. 3: **Rowhammer patterns** used by prior work with their victim (V), near (N/A) and far (F) aggressor rows.

Rowhammer Mitigations. Several in-DRAM Rowhammer mitigations have been proposed in existing literature [8–12]. However, to the best of our knowledge, none of them are currently deployed on real devices. Choosing a suitable mitigation largely depends on the device's vulnerability level, particularly the blast radius, as some mitigations do not scale well with its increase. As a result, it is imperative to conduct experimental characterization and analysis of the cumulative effects when multiple aggressor rows with varying activation ratios are considered.

Currently, DRAM devices implement mitigations collectively known as *Target Row Refresh* (TRR). These mitigations identify potential aggressor rows and *prematurely* refresh their adjacent rows before their periodic refresh. However, recent studies have shown that DRAM devices with TRR remain vulnerable to Rowhammer attacks [2, 3]. Kogler et al. [5] demonstrated that TRR refreshes could be misused to hammer victim rows. They exploited the fact that the mitigation of the targeted devices considered rows directly adjacent to the aggressors only [13]. Using this knowledge, they bypassed mitigations using Half-Double patterns, which involve aggressor rows located farther away from the victim row (i.e., at a distance larger than one row). As a result, their work shows that the blast radius of Rowhammer, which varies across devices, must be taken into account to refresh *all* victim rows correctly. Therefore, it is crucial to evaluate the blast radius of Rowhammer and its effects on mitigations to achieve security in the design of DRAM devices. However, no existing work systematically characterizes this effect by considering more distant aggressor rows and their distribution of activations.

III. Motivation

In 2021, the emergence of a new Rowhammer effect dubbed as *Half-Double* showed that we have yet to understand all underlying disturbance effects working together [14]. In the original report revealing Half-Double [13], the authors investigated selected instances of these new distance-2 patterns, namely single- and double-sided distance-2 assisted patterns. Later, Kogler et al. [5] analyzed their variations and showed that Rowhammer is not the sole root cause, but TRR refreshes assist the hammering process and amplify the effect. Therefore, it is necessary to study these effects systematically to improve our understanding and facilitate the development of principled mitigations that account for these factors.

While previous research has investigated the spatiality of aggressors, these characterization studies have primarily focused on LPDDR4(X) DRAM with limited coverage of DDR4 DRAM devices from different manufacturers [5, 13]. We seek to understand if any Half-Double effect exists on

DDR4 DRAM devices and if the effect is similarly strong compared to LPDDR4(X) devices. Therefore, we raise the following research question:

RQ1. Research Question

To what extent do Half-double patterns affect existing DDR4 DRAM devices?

To address this research question, we systematically explore the impact of Half-Double patterns on DDR4 devices from different vendors. We aim to investigate the effect of these patterns and compare our findings with prior work.

Another aspect that has not been rigorously analyzed yet in existing work are aggressors at distances exceeding two rows from the victim. Previous studies [4, 5, 13] have demonstrated that charge leakage can propagate across multiple rows, indicating that the blast radius of aggressors has expanded. To foster the design of secure mitigations, we must study the effects across a more extensive range of rows.

Motivated by these considerations, we introduce a new class of patterns, named **Blaster**, which generalizes the aforementioned patterns and extends beyond their limitation to distance-2 aggressors. Blaster patterns involve *multiple* aggressors positioned at various distances from the victim row. This leads us to the following research question:

RQ2. Research Question

What is the effect of Blaster patterns encompassing aggressors *up to four* rows away from the victim?

To fully comprehend the consequences of aggressors at increased distances, systematic experiments involving different numbers of far aggressor rows are necessary.

Another aspect arising from our generalized Blaster patterns is the distribution of activations among the aggressor rows. Including multiple aggressors in Blaster patterns creates a large design space. We seek to determine whether there are any variations in the effects when different patterns are employed by changing the number of activations between the various aggressors. We formulate this as follows:

RQ3. Research Question

How do varying activation ratios between aggressors at varying distances affect the victim row?

To answer this question, we systematically explore the parameter space of Blaster patterns by varying the overall amount of activations and the distribution of activations to each aggressor.

IV. Implementation

In this section, we describe the experiment setup (§IV-A) and the implementation of our experiments. First, we outline the necessary preparatory steps, including determining physically neighboring rows (§IV-B), the hammer count (§IV-C), and the retention time (§IV-D). Subsequently, we elaborate on the different experiments in detail (§IV-E). Finally, we explain the control experiments conducted to ensure the accuracy of our findings (§IV-F).

A. General Setup & Platform

Addressing our research questions poses challenges when using an off-the-shelf consumer CPU since its memory controller prohibits us from having fine-grained control over the exact commands sent. To overcome this challenge, we use the FPGA-based DRAM testing infrastructure known as *DRAM Bender* [15]. This platform allows us to exercise precise control over the DDR4 commands directed to the DRAM module. It enables us to bypass CPU caches for repeated row accesses and conduct experiments without the interference of DRAM refresh or Rowhammer mitigation mechanisms. In total, we evaluate three DDR4 UDIMMs obtained from three different DRAM vendors: SK Hynix, Micron and Samsung. We refer to Appendix A for the list of test devices we used.

B. Determining Physically Neighboring Rows

DRAM rows with logically consecutive rows may not be physically adjacent in the DRAM chip due to scrambling and row remapping [5, 16, 17]. However, for accurate results in our experiments, we require physically neighboring rows. As proposed by previous work [17, 18], we use the singled-sided pattern to identify the physically adjacent rows of an aggressor since its immediate neighbors are more likely to experience a higher number of bit flips compared to rows located farther apart. In cases where logical row numbers align with the physical row layout, activating row A leads to bit flips in rows $A-1$ and $A+1$. By repeating this test for multiple rows, we obtain a sample of physically adjacent rows equivalent to the number of rows required in later experiments. Our observations indicate that SK Hynix and Micron employ a linear row mapping, while Samsung remaps every 8th row.

C. Determining the Hammer Count

Each row has a distinct $\mathrm{HC}_{\mathit{first}}$ value, representing the minimum number of activations required by neighboring rows to induce the first bit flip in the target victim row. We note that $\mathrm{HC}_{\mathit{first}}$ differs depending on whether single-sided or double-sided patterns are used. To investigate the impact of far aggressors, it is essential to determine the $\mathrm{HC}_{\mathit{first}}$ value of the victim row, enabling the isolation of the effects caused by near aggressors from those located farther away. To approximate the $\mathrm{HC}_{\mathit{first}}$ of a row, we activate the victims' neighboring rows 300 K times, enough to trigger bit flips even in "strong" rows. We then gradually decrease the number of activations in steps of 1000 until *no more* bit flips occur in the victim row of interest. We define the resulting value as HC^*, which falls within the range $[\mathrm{HC}_{\mathit{first}} - 1000, \mathrm{HC}_{\mathit{first}})$. In the case of single-sided patterns, we denote the determined value as $\mathrm{HC}^*_{\mathrm{S}}$, which accounts for a single aggressor located on the same side as the aggressors used in subsequent experiments. For example, suppose the effects of far aggressors below the victim row should be investigated. In that case, $\mathrm{HC}^*_{\mathrm{S}}$ activations of the near aggressor below the victim should not cause any bit flips in the victim row. For double-sided patterns, $\mathrm{HC}^*_{\mathrm{D}}$ represents the *total* number of activations of both near aggressors above and below the victim.

Subsequently, to confirm that indeed *no* bit flips in the victim row are solely caused by the near aggressors, we repeat the process for 100 repetitions. Victim rows that do not yield consistent results are discarded. In the subsequent experiments, the near aggressors are *not* activated beyond HC^*, allowing us to distinguish the effects caused by more distant aggressors.

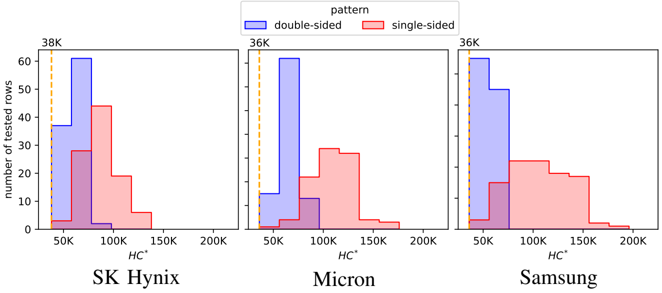


Fig. 4: **HC^* distribution** of all 100 tested rows for each test device.

In Fig. 4, we show the distribution of the determined HC^* values for all the devices. We tested a total of 100 rows for each test device. Consistent with previous findings [4, 19], double-sided Rowhammer requires fewer activations than single-sided Rowhammer to induce a bit flip, which is reflected in HC^*. Notably, $\mathrm{HC}^*_{\mathrm{D}}$ is roughly half of $\mathrm{HC}^*_{\mathrm{S}}$, and $\mathrm{HC}^*_{\mathrm{D}}$ encompasses a smaller range of values than $\mathrm{HC}^*_{\mathrm{S}}$.

D. *Profiling the Retention Time*

A DRAM cell must at least hold data for 64 ms, the standard refresh window (t_{REFW}) in DDR4. During this period, it is crucial to refresh each DRAM row at least once to avoid retention failures. Within the refresh window, a maximum of $t_{\mathrm{REFW}}/(t_{\mathrm{RAS}} + t_{\mathrm{RP}})$ activations can be conducted, typically amounting to roughly 1.36 M activations.

However, as the impact of the aggressor row diminishes with increasing distance between the victim and the aggressor, the effects of distant aggressors become apparent only when a significant number of activations is sent. Some experiments explore the impact of far aggressors and require more activations than can be accommodated within a single refresh window. Nevertheless, we consider these experiments valuable because they promote an understanding of BLASTER patterns and their connection to bit flips.

To minimize the influence of retention failures on bit flips in the victim row, we ensure that the experiment duration *never* exceeds the retention time of the victim row. To ensure a consistent retention time of the victim, we verify in 100 repeated trials that the victim row can accurately retain data throughout the entire experiment duration. In §V, we clearly differentiate between BLASTER patterns that fit within the standard refresh window and those that exceed it.

E. *Experiment Configurations*

Following, we present the design of our experiments. We start by investigating the Half-Double patterns on DDR4 DIMMs. We then study BLASTER patterns, considering aggressors located up to four rows away from the victim row. Fig. 1 on page 1 illustrates our notation denoting aggressor rows at different distances in our experiments. More generally, we refer to the victim-adjacent aggressor rows (i.e., rows $\mathcal{R}_{-1}$ and $\mathcal{R}_1$) as *near* aggressors, while all other more distant aggressors as *far* aggressors.

Half-Double on DDR4 DIMMs. To examine and compare the effect of Half-Double on DDR4 DIMMs, we experiment using the pattern presented in Alg. 1, excluding the highlighted part, by solely targeting aggressors $\mathcal{R}_1$ ($\mathcal{R}_{-1}$) and $\mathcal{R}_2$ ($\mathcal{R}_{-2}$). As we want to avoid the near aggressor $\mathcal{R}_1$ ($\mathcal{R}_{-1}$) from causing any bit flips, we further introduce $ratio_1$ to reduce the number of activations to the near aggressor. Consequently, the near aggressor $\mathcal{R}_1$ ($\mathcal{R}_{-1}$) is activated $ratio_1 \times \mathrm{HC}^*$ times, and after each activation of aggressor $\mathcal{R}_1$ ($\mathcal{R}_{-1}$), the far aggressor $\mathcal{R}_2$ ($\mathcal{R}_{-2}$) is activated $ratio_2$ times. The total number of activations is thus $\mathrm{HC}^* \times ratio_1 \times (1 + ratio_2)$.

We vary $ratio_1$ from 0.1 to 1.0 in increments of 0.1, and we vary $ratio_2$ over the range $\{0, 2^0, \ldots, 2^8\}$. For example, $ratio_1 = 1.0$ and $ratio_2 = 0$ means that only the near aggressor $\mathcal{R}_1$ ($\mathcal{R}_{-1}$) is activated HC^* times, which is insufficient to trigger a bit flip in the victim row. We conduct experiments by systematically sweeping through all possible parameter combinations on 100 randomly selected rows while recording whether a bit flip occurred in the victim row.

Alg. 1: BLASTER patterns. In single-sided patterns, we use $i \leftarrow 1$ ***to*** $ratio_1 \times \mathrm{HC}^*_{\mathrm{S}}$, and only aggressors below the victim row are activated and precharged.

```
inputs:
  • $\mathrm{HC}^*$: the victim's maximum bit-flip-free hammer count
  • $ratio_1$: the ratio relative to $\mathrm{HC}^*$ that $\mathcal{R}_1$ ($\mathcal{R}_{-1}$) is activated
  • $ratio_n$ ($n > 1$): the ratio relative to $\mathcal{R}_1$ ($\mathcal{R}_{-1}$) that $\mathcal{R}_n$ ($\mathcal{R}_{-n}$)
    is activated
for $i \leftarrow 1$ to $ratio_1 \times \mathrm{HC}^*_{\mathrm{D}}/2$ do          // $ratio_1 \times \mathrm{HC}^*_{\mathrm{S}}$
    activate and precharge aggressor $\mathcal{R}_1$
    activate and precharge aggressor $\mathcal{R}_{-1}$
    for $j \leftarrow 1$ to $ratio_2$ do                                // distance-2
        activate and precharge aggressor $\mathcal{R}_2$
        activate and precharge aggressor $\mathcal{R}_{-2}$
    for $j \leftarrow 1$ to $ratio_3$ do                                // distance-3
        activate and precharge aggressor $\mathcal{R}_3$
        activate and precharge aggressor $\mathcal{R}_{-3}$
    for $j \leftarrow 1$ to $ratio_4$ do                                // distance-4
        activate and precharge aggressor $\mathcal{R}_4$
        activate and precharge aggressor $\mathcal{R}_{-4}$
```

Far aggressors and activation ratios. To examine the influence of BLASTER patterns, which include aggressor rows located more than two rows away from the victim, we expand the previous experiment by consecutively activating multiple far aggressor rows, as highlighted in Alg. 1. We define the parameter $ratio_n$, which describes the proportion of activations to row $\mathcal{R}_n$ ($\mathcal{R}_{-n}$) relative to the hammer count of the near aggressor $\mathcal{R}_1$ ($\mathcal{R}_{-1}$). The total number of activations then becomes $\mathrm{HC}^* \times ratio_1 \times (1 + ratio_2 + ratio_3 + ratio_4)$. Like before, we vary $ratio_n$ in the range $\{0, 2^0, \ldots, 2^8\}$ for all aggressors to systematically explore the effects of different hammer counts.

F. Control Experiments

We conducted control experiments to eliminate other potential explanations that could contribute to an increased probability of bit flips when having farther aggressors, such as variations in the relative timing of the memory accesses. In these experiments, the activation of the far aggressors was replaced with a waiting period of $t_{\text{RAS}} + t_{\text{RP}}$. Our results indicate that prolonging the waiting time does *not* result in *any* significant increase in the probability of triggering a bit flip. This confirms our observation in §V: the observed increase in the bit flip probability is indeed attributable to the activations of the far aggressors.

V. Evaluation

This section presents the results of our experiments. We first discuss the results of the Half-Double pattern on DDR4 DIMMs (§V-A), followed by the effects of activating distance-3 aggressors (§V-B) and distance-4 aggressors (§V-C). The boxed or circled values in Figs. 5 and 6 represent patterns with less than 1.36 M activations, calculated based on the *average* HC* of the corresponding test device. Please note that as HC* varies across rows, some Blaster patterns might still fit into the standard refresh window, but they are not boxed or circled if the specific rows have low HC* values.

A. Effects of Half-Double on DDR4 DIMMs

The $ratio_3 = 0$ subplots in Fig. 5 on page 6 present the results of the Half-Double pattern, which includes only distance-1 and distance-2 aggressors. It is evident that the probability of bit flips increases with higher $ratio_1$ and $ratio_2$. Our findings demonstrate the presence of the Half-Double effect across all DDR4 devices obtained from the three vendors. Furthermore, in the case of single-sided patterns, we observe that while HC^*_{S} activations of aggressor $\mathcal{R}_1$ do not lead to any bit flips, the activations of aggressor $\mathcal{R}_1$ can be further reduced by at least 20% ($ratio_1 = 0.8$). This reduction is made possible by incorporating activations to aggressor $\mathcal{R}_2$.

B. Effects of Distance-3 Aggressor Rows

The results of our single- and double-sided experiments involving distance-3 aggressors are presented in Fig. 5 on page 6. We make three key observations from our results. First, the bit flip probability grows in the worst case by up to 50% when incorporating both aggressors $\mathcal{R}_2$ and $\mathcal{R}_3$ in the single-sided patterns. Second, we find that bit flips are possible even when $ratio_2$ is zero. This indicates that the effect of aggressor $\mathcal{R}_3$ ($\mathcal{R}_{-3}$) does not require activating aggressor $\mathcal{R}_2$ ($\mathcal{R}_{-2}$) to propagate to the victim row. Third, the probability of inducing bit flips mostly increases in patterns where $ratio_1$ is high. The increase is negligible for patterns where the near aggressors are not activated frequently. This highlights the importance of near aggressors in Blaster patterns.

Overall, the effects of distance-3 aggressor rows are less pronounced in the double-sided case compared to the single-sided case. This could be due to the limited number of activations performed in the double-sided case. As derived in §IV-E, the total number of activations is $\text{HC}^* \times ratio_1 \times (1+ratio_2+ratio_3)$, for both single- and double-sided patterns. As indicated in Fig. 4, HC^*_{D} is approximately half of HC^*_{S}, the total number of activations is thus reduced in double-sided patterns. Nonetheless, we consistently observe an increase in the probability of bit flips with higher values of $ratio_3$ in both single-sided and double-sided cases.

C. Effects of Distance-4 Aggressor Rows

We further explore Blaster patterns by investigating the impact of distance-4 aggressor rows.

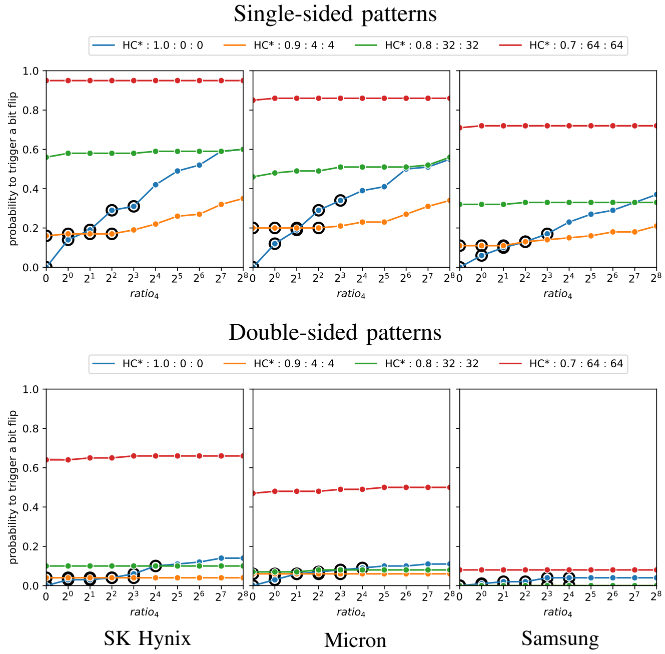


Fig. 6: **Results of distance-4 aggressors.** For each of our test devices, we report the probability of triggering a bit flip across all 100 tested rows for different patterns as denoted by $\text{HC}^* : ratio_1 : ratio_2 : ratio_3$, where $ratio_1 \times \text{HC}^*$ is the hammer count of aggressor $\mathcal{R}_1$ ($\mathcal{R}_{-1}$), and after each activation, we consecutively activate aggressors $\mathcal{R}_2$ ($\mathcal{R}_{-2}$), $\mathcal{R}_3$ ($\mathcal{R}_{-3}$) and $\mathcal{R}_4$ ($\mathcal{R}_{-4}$) exactly $ratio_2$, $ratio_3$, and $ratio_4$ times, respectively. The circled values show patterns within the standard refresh window.

Fig. 6 presents the results of the three test devices, with circled values denoting patterns falling into the standard refresh window. The line labeled as "$\text{HC}^* : 0.9 : 4 : 4$" corresponds to the pattern in which aggressor $\mathcal{R}_1$ ($\mathcal{R}_{-1}$) is activated $\text{HC}^* \times 0.9$ times, and in each iteration aggressors $\mathcal{R}_2$ ($\mathcal{R}_{-2}$) and $\mathcal{R}_3$ ($\mathcal{R}_{-3}$) are activated four times. The number of activations to aggressor $\mathcal{R}_4$ ($\mathcal{R}_{-4}$) is varied as indicated on the x-axis. Across all three vendors, a rise in the bit flip probability of the victim row is notable when the aggressor $\mathcal{R}_1$ ($\mathcal{R}_{-1}$) is activated $1.0 \times \text{HC}^*$ times while increasing the hammer count of aggressor $\mathcal{R}_4$ ($\mathcal{R}_{-4}$). In these patterns, the impact of aggressor $\mathcal{R}_4$ ($\mathcal{R}_{-4}$) propagated to the victim, even if aggressors $\mathcal{R}_2$ ($\mathcal{R}_{-2}$) and $\mathcal{R}_3$ ($\mathcal{R}_{-3}$) are not activated. For all other patterns, we observe only minor changes in the bit flip probability, again highlighting the importance of aggressor $\mathcal{R}_1$ ($\mathcal{R}_{-1}$). Similar to distance-3 aggressors, the increase in the bit flip probability caused by distance-4 aggressors in double-sided patterns is less pronounced than for single-sided patterns.

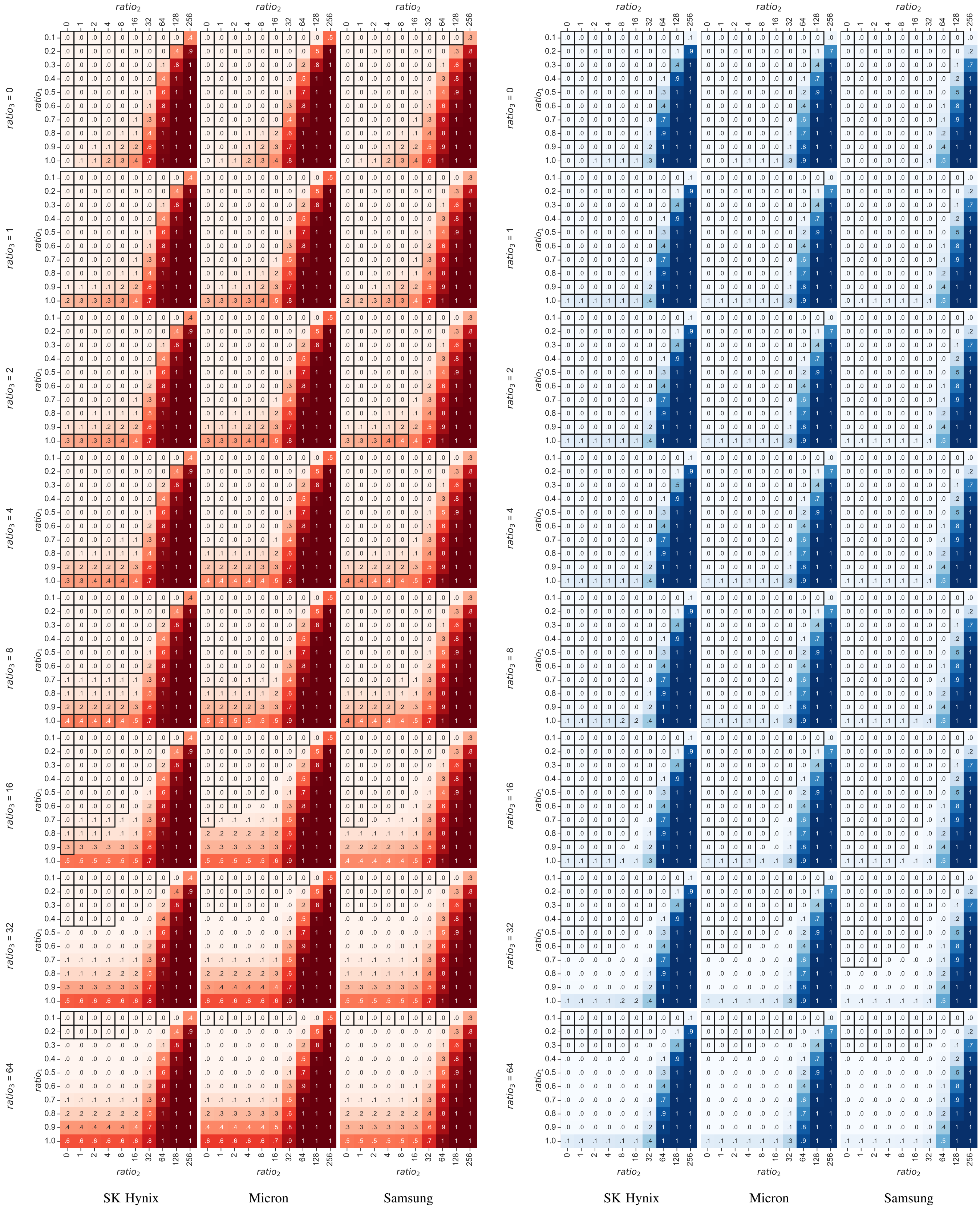


Fig. 5: **Results of distance-3 aggressors.** Red is used to illustrate the results of single-sided BLASTER patterns, and blue shows the outcomes of double-sided BLASTER patterns. For each of our test devices, we report the probability of triggering a bit flip across all 100 tested rows by varying $ratio_1$ for aggressor $\mathcal{R}_1$ ($\mathcal{R}_{-1}$), and relative to it, the $ratio_2$ of aggressor $\mathcal{R}_2$ ($\mathcal{R}_{-2}$), and $ratio_3$ of aggressor $\mathcal{R}_3$ ($\mathcal{R}_{-3}$). For example, $ratio_1 = 0.7$, $ratio_2 = 2$, and $ratio_3 = 8$ means we activate aggressor $\mathcal{R}_1$ ($\mathcal{R}_{-1}$) HC* $\times$ 0.7 times, and after every activation, we activate aggressor $\mathcal{R}_2$ ($\mathcal{R}_{-2}$) two times, followed by aggressor $\mathcal{R}_3$ ($\mathcal{R}_{-3}$) for eight times. The $\boxed{\text{boxed}}$ values show patterns with less than 1.36 M activations, i.e., respecting the standard t_{REFW} of 64 ms.

VI. Discussion

We discuss the practicality of our BLASTER patterns and explain extensions we would like to explore continuing this work in the future. Lastly, we explain how existing mechanisms could be changed to mitigate BLASTER patterns.

Practicality of attack patterns. In our evaluation, we showed that not all BLASTER patterns could be exploited in standard settings respecting periodic refreshes on average every $7.8\,\mu s$ (t_{REFI}). We aim to show that BLASTER patterns are practical and can be reproduced on a regular PC, thus showing that they pose a severe threat to real-world attacks.

Future extensions. In this work, we presented preliminary results on characterizing BLASTER patterns. Moving forward, we would like to extend our work with more DRAM devices from different manufacturing years. We would also like to extend our experiment scope to larger aggressor distances, investigate the impact of the access sequence, and consider the effect of TRR on BLASTER patterns.

Mitigating BLASTER patterns. Secure Rowhammer mitigations typically do not scale well with respect to an increase in blast radius, with exceptions noted in specific cases [10]. However, as emphasized by this paper, mitigations that seek to achieve long-term security need to be both designed and evaluated for blast radii higher than Half-Double.

VII. Conclusion

We introduced BLASTER patterns, a generalization of the recent Half-Double effect. We further characterized the impact of BLASTER patterns on 24 commodity DRAM chips from the three major DRAM vendors. Our results demonstrate that future Rowhammer mitigations should consider activations that are up to *four rows* apart from a potential victim row. Furthermore, hardware vendors must continuously consider the increasing blast radius in future devices.

VIII. Acknowledgments

We thank our anonymous reviewers for their valuable feedback. This work was supported by the Swiss National Science Foundation under NCCR Automation, grant agreement 51NF40 180545, the Swiss State Secretariat for Education, Research and Innovation under contract number MB22.00057 (ERC-StG PROMISE).

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, pp. 361–372. [Online]. Available: http://ieeexplore.ieee.org/document/6853210/

[2] P. Jattke, V. van der Veen, P. Frigo, S. Gunter, and K. Razavi, "BLACKSMITH: Scalable Rowhammering in the Frequency Domain," in *IEEE S&P*, 2022, pp. 716–734. [Online]. Available: https://ieeexplore.ieee.org/document/9833772/

[3] P. Frigo, E. Vannacc, H. Hassan, V. v. der Veen, O. Mutlu, C. Giuffrida, H. Bos, and K. Razavi, "TRRespass: Exploiting the Many Sides of Target Row Refresh," in *IEEE S&P*, 2020, pp. 747–762. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9152631

[4] J. S. Kim, M. Patel, A. G. Yaglikci, H. Hassan, R. Azizi, L. Orosa, and O. Mutlu, "Revisiting RowHammer: An Experimental Analysis of Modern DRAM Devices and Mitigation Techniques," in *ISCA*, 2020, pp. 638–651. [Online]. Available: https://ieeexplore.ieee.org/document/9138944/

[5] A. Kogler, J. Juffinger, S. Qazi, Y. Kim, M. Lipp, N. Boichat, E. Shiu, M. Nissler, and D. Gruss, "Half-Double: Hammering From the Next Row Over," in *USENIX Security*, 2022, pp. 3807–3824. [Online]. Available: https://www.usenix.org/conference/usenixsecurity22/presentation/kogler-half-double

[6] H. Hassan, Y. C. Tugrul, J. S. Kim, V. van der Veen, K. Razavi, and O. Mutlu, "Uncovering In-DRAM RowHammer Protection Mechanisms: A New Methodology, Custom RowHammer Patterns, and Implications," in *MICRO*, 2021, pp. 1198–1213. [Online]. Available: https://dl.acm.org/doi/10.1145/3466752.3480110

[7] M. Seaborn and T. Dullien, "Project Zero: Exploiting the DRAM Rowhammer Bug to Gain Kernel Privileges," Mar. 2015. [Online]. Available: https://googleprojectzero.blogspot.com/2015/03/exploiting-dram-rowhammer-bug-to-gain.html

[8] M. Marazzi, P. Jattke, F. Solt, and K. Razavi, "ProTRR: Principled yet Optimal In-DRAM Target Row Refresh," in *IEEE S&P*, 2022, pp. 735–753. [Online]. Available: https://ieeexplore.ieee.org/document/9833664

[9] T. Bennett, S. Saroiu, A. Wolman, L. Cojocar, and A.-G. Llc, "Panopticon: A Complete In-DRAM Rowhammer Mitigation," 2021. [Online]. Available: https://dramsec.ethz.ch/papers/panopticon.pdf

[10] M. Marazzi, F. Solt, P. Jattke, K. Takashi, and K. Razavi, "REGA: Scalable Rowhammer Mitigation with Refresh-Generating Activations," in *IEEE S&P*, 2023. [Online]. Available: https://comsec.ethz.ch/wp-content/files/rega_sp23.pdf

[11] M. Kim, J. Park, Y. Park, W. Doh, N. Kim, T. Ham, J. W. Lee, and J. Ahn, "Mithril: Cooperative Row Hammer Protection on Commodity DRAM Leveraging Managed Refresh," in *HPCA*, 2022, pp. 1156–1169. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/HPCA53966.2022.00088

[12] M. Wi, J. Park, S. Ko, M. J. Kim, N. S. Kim, E. Lee, and J. H. Ahn, "SHADOW: Preventing Row Hammer in DRAM with Intra-Subarray Row Shuffling," in *HPCA*, 2023, pp. 333–346. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10070966

[13] Google, "Half-Double: Next-Row-Over Assisted Rowhammer." [Online]. Available: https://github.com/google/hammer-kit

[14] O. Mutlu, A. Olgun, and A. G. Yağlıkcı, "Fundamentally Understanding and Solving RowHammer," in *ASPDAC*, 2023, pp. 461–468. [Online]. Available: https://dl.acm.org/doi/10.1145/3566097.3568350

[15] A. Olgun, H. Hassan, A. G. Yağlıkçı, Y. C. Tuğrul, L. Orosa, H. Luo, M. Patel, O. Ergin, and O. Mutlu, "DRAM Bender: An Extensible and Versatile FPGA-based Infrastructure to Easily Test State-of-the-art DRAM Chips," 2022. [Online]. Available: http://arxiv.org/abs/2211.05838

[16] S. Saroiu, A. Wolman, and L. Cojocar, "The Price of Secrecy: How Hiding Internal DRAM Topologies Hurts Rowhammer Defenses," in *IRPS*, 2022, pp. 2C.3–1–2C.3–6. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9764591

[17] A. Tatar, C. Giuffrida, H. Bos, and K. Razavi, "Defeating Software Mitigations Against Rowhammer: A Surgical Precision Hammer," in *RAID*, 2018, pp. 47–66. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-00470-5_3

[18] L. Cojocar, J. Kim, M. Patel, L. Tsai, S. Saroiu, A. Wolman, and O. Mutlu, "Are We Susceptible to Rowhammer? An End-to-End Methodology for Cloud Providers," in *IEEE S&P*, 2020, pp. 712–728. [Online]. Available: https://ieeexplore.ieee.org/document/9152654/

[19] L. Orosa, A. G. Yaglikci, H. Luo, A. Olgun, J. Park, H. Hassan, M. Patel, J. S. Kim, and O. Mutlu, "A Deeper Look into RowHammer's Sensitivities: Experimental Analysis of Real DRAM Chips and Implications on Future Attacks and Defenses," in *MICRO*, 2021, pp. 1182–1197. [Online]. Available: https://doi.org/10.1145/3466752.3480069

Appendix A

Vendor	**Date** *(yy-ww)*	**Size** *(GB)*	**Freq.** *(MHz)*	**Organization**			
				#Ranks	#Chips	#Banks	#Rows
SK Hynix	20-38	8	2400	1R	8	16	64 K
Micron	20-07	8	2400	1R	8	16	64 K
Samsung	20-07	8	2400	1R	8	16	64 K

Tbl. I: **DDR4 DRAM test devices** used in our experiments.