Understanding RowHammer Under Reduced Wordline Voltage: An Experimental Study Using Real DRAM Devices

A. Giray Yağlıkçı[1] Haocong Luo[1] Geraldo F. de Oliviera[1] Ataberk Olgun[1] Minesh Patel[1]
Jisung Park[1] Hasan Hassan[1] Jeremie S. Kim[1] Lois Orosa[1,2] Onur Mutlu[1]
[1]*ETH Zürich* [2]*Galicia Supercomputing Center (CESGA)*

RowHammer is a circuit-level DRAM vulnerability, where repeatedly activating and precharging a DRAM row, and thus alternating the voltage of a row's wordline between low and high voltage levels, can cause bit flips in physically nearby rows. Recent DRAM chips are more vulnerable to RowHammer: with technology node scaling, the minimum number of activate-precharge cycles to induce a RowHammer bit flip reduces and the RowHammer bit error rate increases. Therefore, it is critical to develop effective and scalable approaches to protect modern DRAM systems against RowHammer. To enable such solutions, it is essential to develop a deeper understanding of the RowHammer vulnerability of modern DRAM chips. However, even though the voltage toggling on a wordline is a key determinant of RowHammer vulnerability, no *prior work experimentally demonstrates the effect of wordline voltage (V_{PP}) on the RowHammer vulnerability. Our work closes this gap in understanding.*

This is the first work to experimentally demonstrate on 272 real DRAM chips that lowering V_{PP} reduces a DRAM chip's RowHammer vulnerability. We show that lowering V_{PP} 1) increases the number of activate-precharge cycles needed to induce a RowHammer bit flip by up to 85.8 % *with an average of* 7.4 % *across all tested chips and 2) decreases the RowHammer bit error rate by up to* 66.9 % *with an average of* 15.2 % *across all tested chips. At the same time, reducing V_{PP} marginally worsens a DRAM cell's access latency, charge restoration, and data retention time within the guardbands of system-level nominal timing parameters for 208 out of 272 tested chips. We conclude that reducing V_{PP} is a promising strategy for reducing a DRAM chip's RowHammer vulnerability without requiring modifications to DRAM chips.*

1. Introduction

Manufacturing process technology scaling continuously increases DRAM storage density by reducing circuit component sizes and enabling tighter packing of DRAM cells. Such advancements reduce DRAM chip cost but worsen DRAM reliability [1, 2]. Kim et al. [3] show that modern DRAM chips are susceptible to a read disturbance effect, called *RowHammer*, where repeatedly activating and precharging a DRAM row (i.e., *aggressor row*) many times (i.e., *hammering* the aggressor row) can cause bit flips in physically nearby rows (i.e., *victim rows*) at consistently predictable bit locations [3–15].

Many works [3, 4, 6–48] demonstrate that RowHammer is a serious security vulnerability that can be exploited to mount system-level attacks, such as escalating privilege or leaking private data. To make matters worse, recent experimental studies on real DRAM chips [3, 8, 9, 11, 12, 36, 37, 43] find that the RowHammer vulnerability is more severe in newer DRAM chip generations. For example, 1) the minimum aggressor row activation count necessary to cause a RowHammer bit flip (HC_{first}) is *only* 4.8K and 10K for some newer LPDDR4 and DDR4 DRAM chips (manufactured in 2019–2020), which is 14.4× and 6.9× lower than the HC_{first} of 69.2K for some older DRAM chips (manufactured in 2010–2013) [11]; and 2) the fraction of DRAM cells that experience a bit flip in a DRAM row (BER) after hammering two aggressor rows for $30K$ times is 2×10^{-6} for some newer DRAM chips from 2019–2020, which is 500× larger than that for some other older chips manufactured in 2016–2017 (4×10^{-9}) [11]. As the RowHammer vulnerability worsens, ensuring RowHammer-safe operation becomes more expensive across a broad range of system-level design metrics, including performance overhead, energy consumption, and hardware complexity [8, 9, 11, 12, 36, 43, 49–52].

To find effective and efficient solutions for RowHammer, it is essential to develop a deeper understanding of the RowHammer vulnerability of modern DRAM chips [8, 9, 12]. Prior works [3, 4, 6–12, 15] hypothesize that the RowHammer vulnerability originates from circuit-level interference between 1) wordlines that are physically nearby each other and 2) between a wordline and physically nearby DRAM cells. Existing circuit-level models [7, 10, 15] suggest that toggling of the voltage on a wordline is a key determinant of how much repeated aggressor row activations disturb physically nearby circuit components. However, it is still unclear 1) how the magnitude of the wordline voltage (V_{PP}) affects modern DRAM chips' RowHammer vulnerability and 2) whether it is possible to reduce RowHammer vulnerability by reducing V_{PP}, without significantly worsening other issues related to reliable DRAM operation. Therefore, **our goal** is to experimentally understand how V_{PP} affects RowHammer vulnerability and DRAM operation.

Our Hypothesis. We hypothesize that lowering V_{PP} can reduce RowHammer vulnerability without significantly impacting reliable DRAM operation. To test this hypothesis, we experimentally demonstrate how RowHammer vulnerability varies with V_{PP} by conducting rigorous experiments on 272 real DDR4 DRAM chips from three major DRAM vendors. To isolate the effect of V_{PP} and to avoid failures in DRAM chip I/O circuitry, we scale *only* V_{PP} and supply the rest of the DRAM circuitry using the nominal supply voltage (V_{DD}).

Key Findings. Our experimental results yield six novel observations about V_{PP}'s effect on RowHammer (§5). Our key observation is that a DRAM chip's RowHammer vulnerability reduces by scaling down V_{PP}: 1) HC_{first} increases by 7.4 % (85.8 %), and 2) the BER caused by a RowHammer attack reduces by 15.2 % (66.9 %), on average (at max) across all tested DDR4 DRAM chips.

To investigate the potential adverse effects of reducing V_{PP} on reliable DRAM operation, we conduct experiments using both real DDR4 DRAM chips and SPICE [53] simulations that measure how reducing V_{PP} affects a DRAM cells' 1) row activation latency, 2) charge restoration process, and 3) data retention time. Our measurements yield nine novel observations (§6). We make two key observations: First, V_{PP} reduction only *marginally* worsens the access latency, charge restoration process, and data retention time of most DRAM chips: 208 out of 272 tested DRAM chips reliably operate using nominal timing parameters due to the built-in *safety margins* (i.e., *guardbands*) in nominal timing parameters that DRAM manufacturers already provide. Second, 64 DRAM chips that exhibit erroneous behavior at reduced V_{PP} can reliably operate using 1) a longer row activation latency, i.e., 24 ns / 15 ns for 48 / 16 chips (§6.1), 2) simple single-error-correcting codes [54] (§6.3), or 3) doubling the refresh rate *only* for 16.4 % of DRAM rows.

We make the following major contributions in this paper.

- We present the first experimental RowHammer characterization study under reduced wordline voltage (V_{PP}).
- Our experiments on 272 real DDR4 DRAM chips show that when a DRAM module is operated at a reduced V_{PP}, an attacker 1) needs to hammer a row in the module more times (by 7.4 % / 85.8 %) to induce a bit flip, and 2) can cause fewer (15.2 % / 66.9 %) RowHammer bit flips in the module (on average / at maximum across all tested modules).
- We present the first experimental study of how reducing V_{PP} affects DRAM access latency, charge restoration process, and data retention time.
- Our experiments on real DRAM chips show that reducing V_{PP} slightly worsens DRAM access latency, charge restoration process, and data retention time. Most (208 out of 272) DRAM chips reliably operate under reduced V_{PP}, while the remaining 64 chips reliably operate using increased row activation latency, simple error correcting codes, or doubling the refresh rate *only* for 16.4 % of the rows.

2. Background

We provide a high-level overview of DRAM design and operation as relevant to our work. For a more detailed overview, we refer the interested reader to prior works [3, 55–79].

2.1. DRAM Background

DRAM Organization. Fig. 1 illustrates a DRAM module's hierarchical organization. At the lowest level of the hierarchy, a single DRAM cell comprises 1) a *storage capacitor* that stores a single bit of data encoded using the charge level in the capacitor and 2) an *access transistor* that is used to read from and write to the storage capacitor. The DRAM cell is connected to a *bitline* that is used to access the data stored in the cell and a *wordline* that controls access to the cell.

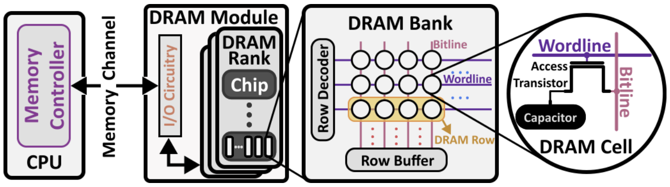

Figure 1: Organization of a typical modern DRAM module.

DRAM cells are organized as a two-dimensional array to form a *bank*. Each cell in a bank is addressed by its *row* and *column*. Each DRAM cell in a DRAM row is connected to a common *wordline* via its *access transistor*. A *bitline* connects a column of DRAM cells to a DRAM *sense amplifier* to read or write data. A row of sense amplifiers is called a *row buffer.* Multiple (e.g., 16 [80]) DRAM banks are put together to form a single DRAM *chip*. Multiple chips form a *rank*. Chips in a rank operate in lock-step such that each chip serves a portion of the data for each DRAM access. A DRAM module may have one or more ranks, communicating with the memory controller over the *memory channel*.

DRAM Operation. The memory controller services main memory requests using three key operations.

1) Row Activation. The memory controller sends an *ACT* command along with a row address to a bank, and the DRAM chip asserts the corresponding wordline to activate the DRAM row. Asserting a wordline connects each cell capacitor in the activated row to its corresponding bitline, perturbing the bitline voltage. Then, the sense amplifier senses and amplifies the voltage perturbation until the cell charge is restored. The data is accessible when the bitline voltage is amplified to a certain level. The latency from the start of row activation until the data is reliably readable is called *row activation latency (t_{RCD})*. A DRAM cell loses its charge during row activation, and thus its initial charge needs to be restored before the row is closed. The latency from the start of row activation until the completion of the DRAM cell's charge restoration is called *charge restoration latency (t_{RAS})*. DRAM manufacturers provide a built-in safety margin in the nominal timing parameters to account for the worst-case latency in t_{RCD} and t_{RAS} operations [58, 60, 81].

2) Read/Write. The memory controller sends a *RD*/*WR* command along with a column address to perform a read or write to the activated row in the DRAM bank. A *RD* command serves data from the row buffer to the memory channel. A *WR* command writes data into the row buffer, which subsequently modifies the data stored in the DRAM cell. The latency of performing a read/write operation is called *column access latency (t_{CL}) / column write latency (t_{CWL}).*

3) Precharge. The memory controller sends a *PRE* command to an active bank. The DRAM chip de-asserts the active row's wordline and precharges the bitlines to prepare the DRAM bank for a new row activation. The timing parameter for precharge is called *precharge latency (t_{RP})*, which is the latency between issuing a *PRE* command and when the DRAM bank is ready for a new row activation.

DRAM Refresh. A DRAM cell inherently leaks charge and thus can retain data for only a limited amount of time, called *data retention time*. To prevent data loss due to such leakage, the memory controller periodically issues *REF (refresh)* commands that ensure every DRAM cell is refreshed at a fixed interval, called *refresh window (t_{REFW})* (e.g., every 64 ms [80, 82, 83] or 32 ms [84]).

2.2. DRAM Voltage Control

Modern DRAM chips (e.g., DDR4 [80], DDR5 [83], GDDR5X [85], and GDDR6 [86] standard compliant ones) use two separate voltage rails: 1) supply voltage (V_{DD}), which is used to operate the core DRAM array and peripheral circuitry (e.g., the sense amplifiers, row/column decoders, precharge and I/O logic), and 2) wordline voltage (V_{PP}), which is exclusively used to assert a wordline during a DRAM row activation. V_{PP} is generally significantly higher (e.g., 2.5V [87–90]) than V_{DD} (e.g., 1.25–1.5V [87–90]) in order to ensure 1) full activation of all access transistors of a row when the wordline is asserted and 2) low leakage when the wordline is de-asserted. V_{PP} is internally generated from V_{DD} in older DRAM chips (e.g., DDR3 [82]). However, newer DRAM chips (e.g., DDR4 onwards [80, 83, 85, 86]) expose *both* V_{DD} and V_{PP} rails to external pins, allowing V_{DD} and V_{PP} to be independently driven with different voltage sources.

2.3. The RowHammer Vulnerability

Modern DRAM is susceptible to a circuit-level vulnerability known as RowHammer [3–15], where a cell's stored data can be corrupted by repeatedly activating physically nearby (aggressor) rows. RowHammer results in unwanted software-visible bit flips and breaks memory isolation [3, 8, 9]. RowHammer poses a significant threat to system security, reliability, and DRAM technology scaling. First, RowHammer leads to data corruption, system crashes, and security attacks if not appropriately mitigated. Many prior works [8, 9, 16–48] show that RowHammer can be exploited to mount system-level attacks to compromise system security (e.g., to acquire root privileges or leak private data). Second, RowHammer vulnerability worsens as DRAM technology scales to smaller node sizes [3, 8, 9, 11, 12, 36, 37, 43]. This is because process technology shrinkage reduces the size of circuit elements, exacerbating charge leakage paths in and around each DRAM cell. Prior works [11, 12, 36, 43] experimentally demonstrate with modern DRAM chips that RowHammer is and will continue to be an increasingly significant reliability, security, and safety problem going forward [8, 9], given that the minimum aggressor row activation count necessary to cause a RowHammer bit flip (HC_{first}) is *only* 4.8K in modern DRAM chips [11] and it continues to reduce.

We describe two major error mechanisms that lead to RowHammer, as explained by prior works [4, 6, 7, 10, 15, 91, 92]: 1) *electron injection / diffusion / drift* and 2) *capacitive crosstalk*. The *electron injection / diffusion / drift* mechanism creates temporary charge leakage paths that degrade the voltage of a cell's storage capacitor [6, 10, 15, 91]. A larger voltage difference between a wordline and a DRAM cell or between two wordlines exacerbates the electron injection / diffusion / drift error mechanism. The *capacitive crosstalk* mechanism exacerbates charge leakage paths in and around a DRAM cell's capacitor [4, 15, 91, 92] due to the parasitic capacitance between two wordlines or between a wordline and a DRAM cell.

2.4. Wordline Voltage's Impact on DRAM Reliability

RowHammer. As explained in §2.3, a larger V_{PP} exacerbates both electron injection / diffusion / drift and capacitive crosstalk mechanisms. Therefore, we hypothesize that the RowHammer vulnerability of a DRAM chip increases as V_{PP} increases. Unfortunately, there is no prior work that tests this hypothesis and quantifies the effect of V_{PP} on real DRAM chips' RowHammer vulnerability. §3 discusses this hypothesis in further detail, and §5 experimentally examines the effects of changing V_{PP} on the RowHammer vulnerability of real DRAM chips.

Row Activation and Charge Restoration. An access transistor turns on (off) when its gate voltage is higher (lower) than a threshold. An access transistor's gate is connected to a wordline (Fig. 1) and driven by V_{PP} (ground) when the row is activated (precharged).[1] Between V_{PP} and ground, a larger access transistor gate voltage forms a stronger channel between the bitline and the capacitor. A strong channel allows fast DRAM row activation and full charge restoration. Based on these properties, we hypothesize that a *larger* V_{PP} provides *smaller* row activation latency and increased data retention time, leading to more reliable DRAM operation.[2] Unfortunately, there is *no* prior work that tests this hypothesis and quantifies V_{PP}'s effect on real DRAM chips' reliable operation (i.e., row activation and charge restoration characteristics). §6 studies the effect of reduced V_{PP} on DRAM operation reliability using both real-device characterizations and SPICE [53, 95] simulations.

3. Motivation

RowHammer is a critical vulnerability for modern DRAM-based computing platforms [3–48]. Many prior works [3, 5, 13, 30, 45, 48, 50–52, 65, 80, 91, 96–114] propose RowHammer mitigation mechanisms that aim to prevent RowHammer bit flips. Unfortunately, RowHammer solutions need to consider a large number of design space constraints that include cost, performance impact, energy and power overheads, hardware complexity, technology scalability, security guarantees, and changes to existing DRAM standards and interfaces. Recent works [8, 9, 11, 12, 36, 43, 49–52] suggest that many existing proposals may fall short in one or more of these dimensions. As a result, there is a critical need for developing better RowHammer mitigation mechanisms.

To enable more effective and efficient RowHammer mitigation mechanisms, it is critical to develop a comprehensive understanding of how RowHammer bit flips occur [8, 9, 12]. In this work, we observe that although the wordline voltage (V_{PP}) is expected to affect the amount of disturbance caused by a RowHammer attack [3, 4, 6–15], *no* prior work experimentally studies its real-world impact on a DRAM chip's RowHammer vulnerability.[3] Therefore, **our goal** is to understand how V_{PP} affects RowHammer vulnerability and DRAM operation.

To achieve this goal, we start with the hypothesis that V_{PP} can be used to reduce a DRAM chip's RowHammer vulnerability without impacting the reliability of normal DRAM operations. Reducing a DRAM chip's RowHammer vulnerability via V_{PP} scaling has two key advantages. First, as a circuit-level RowHammer mitigation approach, V_{PP} scaling is *complementary* to existing system-level and architecture-level RowHammer mitigation mechanisms [3, 5, 13, 30, 45, 48, 50–52, 65, 80, 91, 96–114]. Therefore, V_{PP} scaling can be used *alongside* these mechanisms to increase their effectiveness and/or reduce their overheads. Second, V_{PP} scaling can be implemented with a *fixed hardware cost* for a given power budget, irrespective of the number and types of DRAM chips used in a system.

We test this hypothesis through the first experimental RowHammer characterization study under reduced V_{PP}. In this study, we test 272 real DDR4 DRAM chips from three major DRAM manufacturers. Our study is inspired by state-of-the-art analytical models for RowHammer, which suggest that the effect of RowHammer's underlying error mechanisms depends on V_{PP} [7, 10, 15]. §5 reports our findings, which yield valuable insights into how V_{PP} impacts the circuit-level RowHammer

[1] To increase DRAM cell retention time, modern DRAM chips may apply a negative voltage to the wordline [93, 94] when the wordline is not asserted. Doing so reduces the leakage current and this improves data retention.

[2] Increasing/decreasing V_{PP} does *not* affect the reliability of *RD*/*WR* and *PRE* operations since the DRAM circuit components involved in these operations are powered using *only* V_{DD}.

[3] Both V_{PP} and V_{DD} can affect a DRAM chip's RowHammer vulnerability. However, changing V_{DD} can negatively impact DRAM reliability in ways that are unrelated to RowHammer (e.g., I/O circuitry instabilities) because V_{DD} supplies power to *all* logic elements within the DRAM chip. In contrast, V_{PP} affects *only* the wordline voltage, so V_{PP} can influence RowHammer without adverse effects on unrelated parts of the DRAM chip.

characteristics of modern DRAM chips, both confirming our hypothesis and supporting V_{PP} scaling as a promising new dimension toward robust RowHammer mitigation.

4. Experimental Methodology

We describe our methodology for two analyses. First, we experimentally characterize the behavior of 272 real DDR4 DRAM chips from three major manufacturers under reduced V_{PP} in terms of RowHammer vulnerability (§4.2), row activation latency (t_{RCD}) (§4.3), and data retention time (§4.4). Second, to verify our observations from real-device experiments, we investigate reduced V_{PP}'s effect on *both* DRAM row activation and charge restoration using SPICE [53, 95] simulations (§4.5).

4.1. Real-Device Testing Infrastructure

We conduct real-device characterization experiments using an infrastructure based on SoftMC [64, 115], the state-of-the-art FPGA-based open-source infrastructure for DRAM characterization. We extensively modify SoftMC to test modern DDR4 DRAM chips. Fig. 2 shows a picture of our experimental setup. We attach heater pads to the DRAM chips that are located on both sides of a DDR4 DIMM. We use a MaxWell FT200 PID temperature controller [116] connected to the heaters pads to maintain the DRAM chips under test at a preset temperature level with the precision of ±0.1 °C. We program a Xilinx Alveo U200 FPGA board [117] with the modified version of SoftMC. The FPGA board is connected to a host machine through a PCIe port for running our tests. We connect the DRAM module to the FPGA board via a commercial interposer board from Adexelec [118] with current measurement capability. The interposer board enforces the power to be supplied through a shunt resistor on the V_{PP} rail. We remove this shunt resistor to electrically disconnect the V_{PP} rails of the DRAM module and the FPGA board. Then, we supply power to the DRAM module's V_{PP} power rail from an external TTi PL068-P power supply [119], which enables us to control V_{PP} at the precision of ±1 mV. We start testing each DRAM module at the nominal V_{PP} of 2.5 V. We gradually reduce V_{PP} with 0.1 V steps until the lowest V_{PP} at which the DRAM module can successfully communicate with the FPGA (V_{PPmin}).

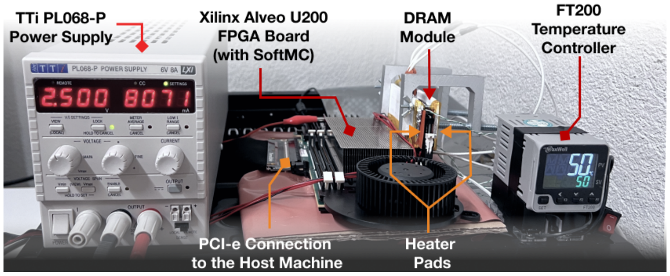
TTi PL068-P Power Supply
Xilinx Alveo U200 FPGA Board (with SoftMC)
DRAM Module
FT200 Temperature Controller
PCI-e Connection to the Host Machine
Heater Pads

Figure 2: Our experimental setup based on SoftMC [64, 115].

To show that our observations are *not* specific to a certain DRAM architecture/process but rather common across different designs and generations, we test DDR4 DRAM modules from all three major manufacturers with different die revisions, purchased from the retail market. Table 1 provides the chip density, die revision (Die Rev.), chip organization (Org.), and manufacturing date of tested DRAM modules.[4] We report the manufacturing date of these modules in the form of $week-year$. All tested modules are listed in Table 3 in Appendix A.

Table 1: Summary of the tested DDR4 DRAM chips.

Mfr.	#DIMMs	#Chips	Density	Die Rev.	Org.	Date
Mfr. A (Micron)	1	8	4Gb		×8	48-16
	4	64	8Gb	B	×4	11-19
	3	24	4Gb	F	×8	07-21
	2	16	4Gb		×8	
Mfr. B (Samsung)	2	16	8Gb	B	×8	52-20
	1	8	8Gb	C	×8	19-19
	3	24	8Gb	D	×8	10-21
	1	8	4Gb	E	×8	08-17
	1	8	4Gb	F	×8	02-21
	2	16	8Gb		×8	
Mfr. C (SK Hynix)	2	16	16Gb	A	×8	51-20
	3	24	4Gb	B	×8	02-21
	2	16	4Gb	C	×8	
	3	24	8Gb	D	×8	48-20

Temperature. We conduct RowHammer and t_{RCD} tests at 50 °C and retention tests at 80 °C to ensure both stable and representative testing conditions.[5] We conduct t_{RCD} tests at 50 °C because 50 °C is our infrastructure's minimum stable temperature due to cooling limitations.[6] We conduct retention tests at 80 °C to capture any effects of increased charge leakage [74] at the upper bound of regular operating temperatures [80].[7]

Disabling Sources of Interference. To understand fundamental device behavior in response to V_{PP} reduction, we make sure that V_{PP} is the only control variable in our experiments so that we can accurately measure the effects of V_{PP} on RowHammer, row activation latency (t_{RCD}), and data retention time. To do so, we follow four steps, similar to prior rigorous RowHammer [11, 12], row activation latency [58, 60, 81], and data retention time [74, 77] characterization methods. First, we disable DRAM refresh to ensure no disturbance on the desired access pattern. Second, we ensure that during our RowHammer and t_{RCD} experiments, *no* bit flips occur due to data retention failures by conducting each experiment within a time period of less than 30 ms (i.e., much shorter than the nominal t_{REFW} of 64 ms). Third, we test DRAM modules without error-correction code (ECC) support to ensure neither on-die ECC [121–127] nor rank-level ECC [32, 128] can affect our observations by correcting V_{PP}-reduction-induced bit flips. Fourth, we disable known on-DRAM-die RowHammer defenses (i.e., TRR [36, 43, 83, 88, 129, 130]) by not issuing refresh commands throughout our tests [11, 12, 36, 43] (as all TRR defenses require refresh commands to work).

Data Patterns. We use six commonly used data patterns [3, 11, 12, 60, 66, 67, 72, 74, 81, 131, 132]: row stripe (`0xFF/0x00`), checkerboard (`0xAA/0x55`), and thickchecker (`0xCC/0x33`). We identify the worst-case data pattern ($WCDP$) for each row among these six patterns at nominal V_{PP} separately for each of

[4] Die Rev. and Date columns are blank if undocumented.

[5] A recent work [12] shows a complex interaction between RowHammer and temperature, suggesting that one should repeat characterization at many different temperature levels to find the worst-case RowHammer vulnerability. Since such characterization requires many months-long testing time, we leave it to future work to study temperature, voltage, and RowHammer interaction in detail.

[6] We do not repeat the t_{RCD} tests at different temperature levels because prior work [60] shows small variation in t_{RCD} with varying temperature.

[7] DDR4 DRAM chips are refreshed at 2× the nominal refresh rate when the chip temperature reaches 85 °C [80]. Thus, we choose 80 °C as a representative high temperature within the regular operating temperature range. For a detailed analysis of the effect of temperature on data retention in DRAM, we refer the reader to [74, 77, 120].

RowHammer (§4.2), row activation latency (t_{RCD}) (§4.3), and data retention time (§4.4) tests. We use each row's corresponding *WCDP* for a given test, at reduced V_{PP} levels.

4.2. RowHammer Experiments

We perform multiple experiments to understand how V_{PP} affects the RowHammer vulnerability of a DRAM chip.

Metrics. We measure the RowHammer vulnerability of a DRAM chip using two metrics: 1) the minimum aggressor row activation count necessary to cause a RowHammer bit flip (HC_{first}) and 2) the fraction of DRAM cells that experience a bit flip in a DRAM row (*BER*) caused by a double-sided RowHammer attack with a fixed hammer count of 300K per aggressor row.[8]

WCDP. We choose *WCDP* as the data pattern that causes the *lowest* HC_{first}. If there are multiple data patterns that cause the lowest HC_{first}, we choose the data pattern that causes the *largest BER* for the fixed hammer count of $300K$.[9]

RowHammer Tests. Alg. 1 describes the core test loop of each RowHammer test that we run. The algorithm performs a *double-sided* RowHammer attack on each row within a DRAM bank. A double-sided RowHammer attack activates the two attacker rows that are physically adjacent to a victim row (i.e., the victim row's two immediate neighbors) in an alternating manner. We define hammer count (*HC*) as the number of times each physically-adjacent row is activated. In this study, we perform double-sided attacks instead of single- [3] or many-sided attacks (e.g., as in TRRespass [36], U-TRR [43], and BlackSmith [44]) because a double-sided attack is the most effective RowHammer attack when no RowHammer defense mechanism is employed: it reduces HC_{first} and increases *BER* compared to both single- and many-sided attacks [3, 11, 12, 36, 43, 44]. Due to time limitations, 1) we test $4K$ rows per DRAM module (four chunks of $1K$ rows evenly distributed across a DRAM bank) and 2) we run each test ten times and record the smallest (largest) observed HC_{first} (*BER*) to account for the worst-case.

Finding Physically Adjacent Rows. DRAM-internal address mapping schemes [37, 87] are used by DRAM manufacturers to translate *logical* DRAM addresses (e.g., row, bank, and column) that are exposed over the DRAM interface (to the memory controller) to physical DRAM addresses (e.g., physical location of a row). Internal address mapping schemes allow 1) post-manufacturing row repair techniques to repair erroneous DRAM rows by remapping these rows to spare rows and 2) DRAM manufacturers to organize DRAM internals in a cost-optimized way, e.g., by organizing internal DRAM buffers hierarchically [67, 133]. The mapping scheme can vary substantially across different DRAM chips [3, 12, 14, 37, 55, 67, 68, 72, 74, 124, 134–137]. For every victim DRAM row we test, we identify the two neighboring physically-adjacent DRAM row addresses that the memory controller can use to access the aggressor rows in a double-sided RowHammer attack. To do so, we reverse-engineer the physical row organization using techniques described in prior works [11, 12].

Alg. 1: Test for HC_{first} and BER for a Given V_{PP}

```
// $RA_{victim}$:  victim row address
// $WCDP$:  worst-case data pattern
// $HC$:  number of activations per aggressor row
Function measure_$BER$($RA_{victim}$, $WCDP$, $HC$):
    initialize_row ($RA_{victim}$, $WCDP$)
    initialize_aggressor_rows ($RA_{victim}$, bitwise_inverse($WCDP$))
    hammer_doublesided ($RA_{victim}$, $HC$)
    $BER_{row}$ = compare_data ($RA_{victim}$, $WCDP$)
    return $BER_{row}$

// $V_{pp}$:  wordline voltage for the experiment
// $WCDP\_list$:  the list of $WCDP$s (one $WCDP$ per row)
// $row\_list$:  the list of tested rows
Function test_loop($V_{pp}$, $WCDP\_list$):
    set_vpp ($V_{pp}$)
    foreach $RA_{victim}$ in $row\_list$ do
        $HC = 300K$ // initial hammer count to test
        $HC_{step} = 150K$ // how much to increment/decrement $HC$
        while $HC_{step} > 100$ do
            $BER_{rowmax} = 0$
            for $i \leftarrow 0$ to $num\_iterations$ do
                $BER_{row}$ = measure_BER ($RA_{victim}$, $WCDP$, $HC$)
                record_BER($V_{pp}$, $RA_{victim}$, $WCDP$, $HC$, $BER_{row}$, $i$)
                $BER_{rowmax} = max(BER_{rowmax}, BER_{row})$
            end
            if $BER_{rowmax} == 0$ then
                $HC+ = HC_{step}$ // Increase HC if no bit flips occur
            end
            else
                $HC- = HC_{step}$ // Reduce HC if a bit flip occurs
            end
            $HC_{step} = HC_{step}/2$
        end
        record_HCfirst($V_{pp}$, $RA_{victim}$, $WCDP$, $HC$)
    end
```

4.3. Row Activation Latency (t_{RCD}) Experiments

We conduct experiments to find how a DRAM chip's row activation latency (t_{RCD}) changes with reduced V_{PP}.

Metric. We measure the minimum time delay required (t_{RCDmin}) between a row activation and the following read operation to ensure that there are *no* bit flips in the entire DRAM row.

WCDP. We choose *WCDP* as the data pattern that leads to the *largest* observed t_{RCDmin}.

t_{RCD} Tests. Alg. 2 describes the core test loop of each t_{RCD} test that we run. The algorithm sweeps t_{RCD} starting from the nominal t_{RCD} of 13.5 ns with steps of 1.5 ns.[10] We decrement (increment) t_{RCD} by 1.5 ns until we observe at least one (no) bit flip in the entire DRAM row in order to pinpoint t_{RCDmin}. To test a DRAM row for a given t_{RCD}, the algorithm 1) initializes the row with the row's *WCDP*, 2) performs an access using the given t_{RCD} for each column in the row and 3) checks if the access results in any bit flips. After testing each column in a DRAM row, the algorithm identifies the row's t_{RCDmin} as the minimum t_{RCD} that does not cause any bit flip in the entire DRAM row. Due to time limitations, we 1) test the same set of rows as we use in RowHammer tests (§4.2) and 2) run each test ten times and record the *largest* t_{RCDmin} for each row across all

[8] We choose the 300K hammer count because 1) it is low enough to be used in a system-level RowHammer attack in a real system, and 2) it is high enough to provide us with a large number of bit flips to make meaningful observations in all DRAM modules we tested.

[9] To investigate if *WCDP* changes with reduced V_{PP}, we repeat *WCDP* determination experiments for different V_{PP} values for 16 DRAM chips. We observe that *WCDP* changes for *only* 2.4 % of tested rows, causing less than 9 % deviation in HC_{first} for 90 % of the affected rows. We leave a detailed sensitivity analysis of *WCDP* to V_{PP} for future work.

[10] Our version of SoftMC can send a DRAM command every 1.5 ns due to the clock frequency limitations in the FPGA's physical DRAM interface.

runs.[11]

Alg. 2: Test for Row Activation Latency for a Given V_{PP}

```
// $V_{pp}$:  wordline voltage for the experiment
// $WCDP\_list$:  the list of WCDPs (one WCDP per row)
// $row\_list$:  the list of tested rows
Function test_loop($V_{pp}$, $WCDP\_list$, $row\_list$):
    set_vpp ($V_{pp}$)
    foreach $RA$ in $row\_list$ do
        $t_{RCD}$ = 13.5 ns
        found_faulty, found_reliable = False, False
        while not found_faulty or not found_reliable do
            is_faulty = False
            for $i \leftarrow 0$ to $num\_iterations$ do
                foreach column $C$ in row $RA$ do
                    initialize_row ($RA$, $WCDP\_list[RA]$)
                    activate_row($RA$, $t_{RCD}$) //activate the row using $t_{RCD}$
                    read_data = read_col($C$)
                    close_row($RA$)
                    $BER_{col}$ = compare (WCDP_list[RA], read_data)
                    if $BER_{col} > 0$ then is_faulty=True
                end
            end
            if is_faulty then  {$t_{RCD}$ += 1.5 ns; found_faulty = True;}
            else  {$t_{RCDmin} = t_{RCD}$; $t_{RCD}$ -= 1.5 ns; found_reliable = True;}
        end
        record_$t_{RCDmin}$($RA$, $t_{RCDmin}$)
    end
```

4.4. Data Retention Time Experiments

We conduct data retention time experiments to understand the effects of V_{PP} on DRAM cell data retention characteristics. We test the same set of DRAM rows as we use in RowHammer tests (§4.2) for a set of fixed refresh windows from 16 ms to 16 s in increasing powers of two.

Metric. We measure the fraction of DRAM cells that experience a bit flip in a DRAM row (retention-*BER*) due to violating a DRAM row's data retention time, using a reduced refresh rate.

WCDP. We choose *WCDP* as the data pattern which causes a bit flip at the *smallest* refresh window (t_{REFW}) among the six data patterns. If we find more than one such data pattern, we choose the one that leads to the largest *BER* for t_{REFW} of 16 s.

Data Retention Time Tests. Alg. 3 describes how we perform data retention tests to measure retention-*BER* for a given V_{PP} and refresh rate. The algorithm 1) initializes a DRAM row with WCDP, 2) waits as long as the given refresh window, and 3) reads and compares the data in the DRAM row to the row's initial data.

4.5. SPICE Model

Alg. 3: Test for Data Retention Times for a Given V_{PP}

```
// $V_{pp}$:  wordline voltage for the experiment
// $WCDP\_list$:  the list of WCDPs (one WCDP per row)
// $row\_list$:  the list of tested rows
Function test_loop($V_{pp}$, $WCDP\_list$, $row\_list$):
    set_vpp ($V_{pp}$)
    $t_{REFW}$ = 16 ms
    while $t_{REFW} \leq 16$ s do
        for $i \leftarrow 0$ to $num\_iterations$ do
            foreach $RA$ in $row\_list$ do
                initialize_row ($RA$, $WCDP\_list[RA]$)
                wait($t_{REFW}$)
                read_data = read_row($RA$)
                $BER_{row}$ = compare_data (WCDP_list[RA], read_data)
                record_retention_errors($RA$, $t_{REFW}$, $BER_{row}$)
            end
        end
        $t_{REFW} = t_{REFW} \times 2$
    end
```

To provide insights into our real-chip-based experimental observations about the effect of reduced V_{PP} on row activation latency and data retention time, we conduct a set of SPICE [53, 95] simulations to estimate the bitline and cell voltage levels during two relevant DRAM operations: row activation and charge restoration. To do so, we adopt and modify a SPICE model used in a relevant prior work [60] that studies the impact of changing V_{DD} (but *not* V_{PP}) on DRAM row access and refresh operations. Table 2 summarizes our SPICE model, which we open-source [138]. We use LTspice [95] with the 22 nm PTM transistor model [139, 140] and scale the simulation parameters according to the ITRS roadmap [141, 142].[12] To account for manufacturing process variation, we perform Monte-Carlo simulations by randomly varying the component parameters up to 5 % for each simulation run. We run the simulation at V_{PP} levels from 1.5 V to 2.5 V with a step size of at 0.1 V 10K times, similar to prior works [65, 76].

Table 2: Key parameters used in SPICE simulations.

Component	Parameters
DRAM Cell	C: 16.8 fF, R: 698 Ω
Bitline	C: 100.5 fF, R: 6980 Ω
Cell Access NMOS	W: 55 nm, L: 85 nm
Sense Amp. NMOS	W: 1.3 um, L: 0.1 um
Sense Amp. PMOS	W: 0.9 um, L: 0.1 um

4.6. Statistical Significance of Experimental Results

To evaluate the statistical significance of our methodology, we investigate the variation in our measurements by examining the *coefficient of variation (CV)* across ten iterations. CV is a standardized metric to measure the extent of variability in a set of measurements, in relation to the mean of the measurements. CV is calculated as the ratio of standard deviation over the mean value [143]. A smaller CV shows a smaller variation across measurements, indicating higher statistical significance. The coefficient of variation is 0.08, 0.13, and 0.24 for 90^{th}, 95^{th}, and 99^{th} percentiles of all of our experimental results, respectively.

5. RowHammer Vulnerability Under Reduced V_{PP}

We provide the first experimental characterization of how wordline voltage (V_{PP}) affects the RowHammer vulnerability of a DRAM row in terms of 1) the fraction of DRAM cells that experience a bit flip in a DRAM row (*BER*) (§5.1) and 2) the minimum aggressor row activation count necessary to cause a RowHammer bit flip (HC_{first}) (§5.2). To conduct this analysis, we provide experimental results from 272 real DRAM chips, using the methodology described in §4.1 and §4.2.

5.1. Effect of V_{PP} on RowHammer BER

Fig. 3 shows the RowHammer *BER* a DRAM row experiences at a fixed hammer count of 300*K* under different voltage levels, normalized to the row's RowHammer *BER* at nominal V_{PP}

[11] To understand whether reliable DRAM row activation latency changes over time, we repeat these tests for 24 DRAM chips after one week, during which the chips are tested for RowHammer vulnerability. We observe that *only* 2.1 % of tested DRAM rows experience only a small variation (<1.5 ns) in t_{RCD}. This result is consistent with results of prior works [60, 69, 81].

[12] We do *not* expect SPICE simulation and real-world experimental results to be identical because a SPICE model *cannot* simulate a real DRAM chip's exact behavior without proprietary design and manufacturing information.

(2.5 V). Each line represents a different DRAM module. The band of shade around each line marks the 90 % confidence interval of the normalized BER value across all tested DRAM rows. We make Obsvs. 1 and 2 from Fig. 3.

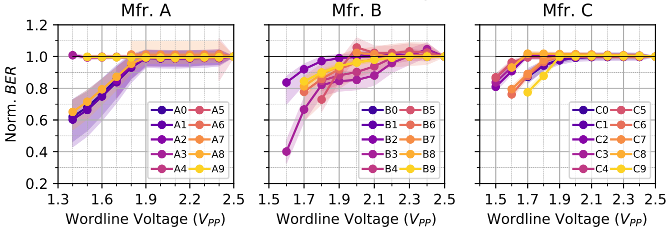

Figure 3: Normalized BER values across different V_{PP} levels. Each curve represents a different DRAM module.

Observation 1. *Fewer DRAM cells experience bit flips due to RowHammer under reduced wordline voltage.*

We observe that RowHammer BER *decreases* as V_{PP} reduces in 81.2 % of tested rows across all tested modules. This reduction in BER reaches up to 66.9 % (B3 at $V_{PP} = 1.6V$) with an average of 15.2 % (not shown in the figure) across all modules we test. We conclude that the disturbance caused by hammering a DRAM row becomes weaker, on average, with reduced V_{PP}.

Observation 2. *In contrast to the dominant trend, reducing V_{PP} can sometimes increase BER.*

We observe that BER increases in 15.4 % of tested rows with reduced V_{PP} by up to 11.7 % (B5 at $V_{PP} = 2.0V$). We suspect that the BER increase we observe occurs due to a weakened charge restoration process rather than an actual increase in read disturbance (due to RowHammer). §6.3 analyzes the impact of reduced V_{PP} on the charge restoration process.

Variation in BER Reduction Across DRAM Rows. We investigate how BER reduction with reduced V_{PP} varies across DRAM rows. To do so, we measure BER reduction of each DRAM row at V_{PPmin} (§4.1). Fig. 4 shows a population density distribution of DRAM rows (y-axis) based on their BER at V_{PPmin}, normalized to their BER at the nominal V_{PP} level (x-axis), for each manufacturer. We make Obsv. 3 from Fig. 4.

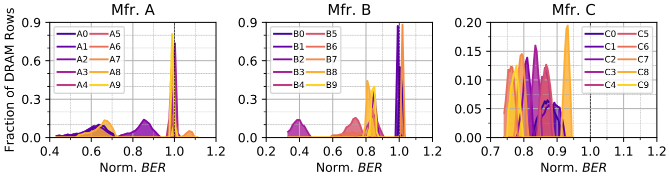

Figure 4: Population density distribution of DRAM rows based on their normalized BER values at V_{PPmin}.

Observation 3. *BER reduction with reduced V_{PP} varies across different DRAM rows and different manufacturers.*

DRAM rows exhibit a large range of normalized BER values (0.43–1.11, 0.33–1.03, and 0.74–0.94 in chips from Mfrs. A, B, and C, respectively). BER reduction also varies across different manufacturers. For example, BER reduces by more than 5 % for *all* DRAM rows of Mfr. C, while BER variation with reduced V_{PP} is smaller than 2 % in 49.6 % of the rows of Mfr. A.

Based on Obsvs. 1–3, we conclude that a DRAM row's RowHammer BER tends to decrease with reduced V_{PP}, while both the amount and the direction of change in BER varies across different DRAM rows and manufacturers.

5.2. Effect of V_{PP} on HC_{first}

Fig. 5 shows the HC_{first} a DRAM row exhibits under different voltage levels, normalized to the row's HC_{first} at nominal V_{PP} (2.5 V). Each line represents a different DRAM module. The band of shade around each line marks the 90 % confidence interval of the normalized HC_{first} values across all tested DRAM rows in the module. We make Obsvs. 4 and 5 from Fig. 3.

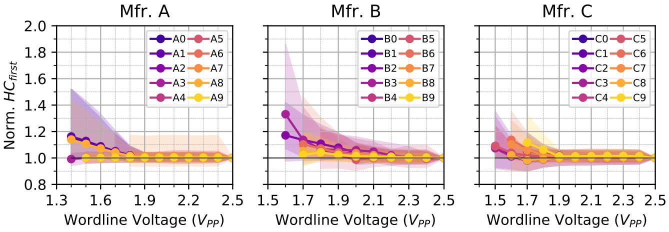

Figure 5: Normalized HC_{first} values across different V_{PP} levels. Each curve represents a different DRAM module.

Observation 4. *DRAM cells experience RowHammer bit flips at higher hammer counts under reduced wordline voltage.*

We observe that HC_{first} of a DRAM row increases as V_{PP} reduces in 69.3 % of tested rows across all tested modules. This increase in HC_{first} reaches up to 85.8 % (B3 at $V_{PP} = 1.6V$) with an average of 7.4 % (not shown in the figure) across all tested modules. We conclude that the disturbance caused by hammering a DRAM row becomes weaker with reduced V_{PP}.

Observation 5. *In contrast to the dominant trend, reducing V_{PP} can sometimes cause bit flips at lower hammer counts.*

We observe that HC_{first} reduces in 14.2 % of tested rows with reduced V_{PP} by up to 9.1 % (C8 at V_{PP}=1.6 V). Similar to Obsv. 2, we suspect that this behavior is caused by the weakened charge restoration process (see §6.3).

Variation in HC_{first} Increase Across DRAM Rows. We investigate how HC_{first} increase varies with reduced V_{PP} across DRAM rows. To do so, we measure HC_{first} increase of each DRAM row at V_{PPmin} (§4.1). Fig. 6 shows a population density distribution of DRAM rows (y-axis) based on their HC_{first} at V_{PPmin}, normalized to their HC_{first} at the nominal V_{PP} level (x-axis), for each manufacturer. We make Obsv. 6 from Fig. 6.

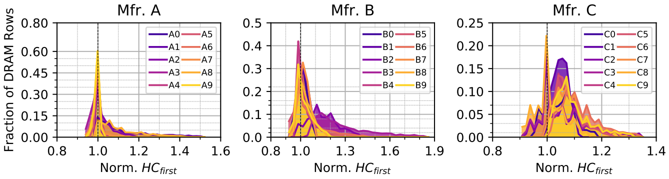

Figure 6: Population density distribution of DRAM rows based on their normalized HC_{first} values at V_{PPmin}.

Observation 6. *HC_{first} increase with reduced V_{PP} varies across different DRAM rows and different manufacturers.*

DRAM rows in chips from the same manufacturer exhibit a large range of normalized HC_{first} values (0.94–1.52, 0.92–1.86, and 0.91–1.35 for Mfrs. A, B, and C, respectively). HC_{first} increase also varies across different manufacturers. For example, HC_{first} increases with reduced V_{PP} for 83.5 % of DRAM rows in modules from Mfr. C, while 50.9 % of DRAM rows exhibit this behavior in modules from Mfr. A.

Based on Obsvs. 4–6, we conclude that a DRAM row's HC_{first} tends to increase with reduced V_{PP}, while both the amount and the direction of change in HC_{first} varies across different DRAM rows and manufacturers.

Summary of Findings. Based on our analyses on both BER and HC_{first}, we conclude that a DRAM chip's RowHammer vulnerability can be reduced by operating the chip at a V_{PP} level that is lower than the nominal V_{PP} value.

6. DRAM Reliability Under Reduced V_{PP}

To investigate the effect of reduced V_{PP} on reliable DRAM operation, we provide the first experimental characterization of how V_{PP} affects the reliability of three V_{PP}-related fundamental DRAM operations: 1) DRAM row activation (§6.1), 2) charge restoration (§6.2), and 3) DRAM refresh (§6.3). To conduct these analyses, we provide both 1) experimental results from real DRAM devices, using the methodology described in §4.1, §4.3, and §4.4 and 2) SPICE simulation results, using the methodology described in §4.5.

6.1. DRAM Row Activation Under Reduced V_{PP}

Motivation. DRAM row activation latency (t_{RCD}) should theoretically increase with reduced V_{PP} (§2.2). We investigate how t_{RCD} of real DRAM chips change with reduced V_{PP}.

Novelty. We provide the first experimental analysis of the isolated impact of V_{PP} on activation latency. Prior work [60] tests DDR3 DRAM chips under reduced supply voltage (V_{DD}), which may or may not change internally-generated V_{PP} level. In contrast, we modify only wordline voltage (V_{PP}) without modifying V_{DD} to avoid the possibility of negatively impacting DRAM reliability due to I/O circuitry instabilities (§2.2).

Experimental Results. Fig. 7 demonstrates the variation in t_{RCDmin} (§4.3) on the y-axis under reduced V_{PP} on the x-axis, across 30 DRAM modules. We annotate the nominal t_{RCD} value (13.5 ns) [80] with a black horizontal line. We make Obsv. 7 from Fig. 7.

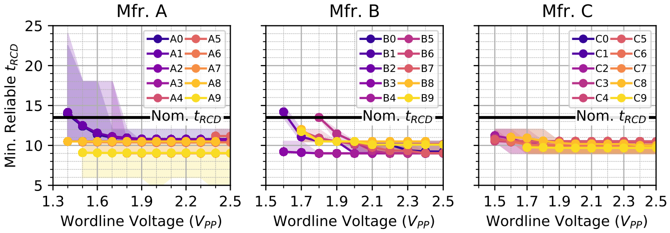

Figure 7: Minimum reliable t_{RCD} values across different V_{PP} levels. Each curve represents a different DRAM module.

Observation 7. *Reliable row activation latency generally increases with reduced V_{PP}. However, 208 (25) out of 272 (30) DRAM chips (modules) complete row activation before the nominal activation latency.*

The minimum reliable activation latency (t_{RCDmin}) increases with reduced V_{PP} across all tested modules. t_{RCDmin} exceeds the nominal t_{RCD} of 13.5 ns for *only* 5 of 30 tested modules (A0–A2, B2, and B5). Among these, modules from Mfr. A and B contain 16 and 8 chips per module. Therefore, we conclude that 208 of 272 tested DRAM chips do *not* experience bit flips when operated using nominal t_{RCD}. We observe that since t_{RCDmin} increases with reduced V_{PP}, the available t_{RCD} guardband reduces by 21.9 % with reduced V_{PP}, on average across all DRAM modules that reliably work with nominal t_{RCD}. We also observe that the three and two modules from Mfrs. A and B, which exhibit t_{RCDmin} values larger than the nominal t_{RCD}, reliably operate when we use a t_{RCD} of 24 ns and 15 ns, respectively.

To verify our experimental observations and provide a deeper insight into the effect of V_{PP} on activation latency, we perform SPICE simulations (as described in §4.5). Fig. 8a shows a waveform of the bitline voltage during the row activation process. The time in the x-axis starts when an activation command is issued. Each color corresponds to the bitline voltage at a different V_{PP} level. We annotate the bitline's supply voltage (V_{DD}) and the voltage threshold that the bitline voltage should exceed for the activation to be reliably completed (V_{TH}). We make Obsv. 8 from Fig. 8a.

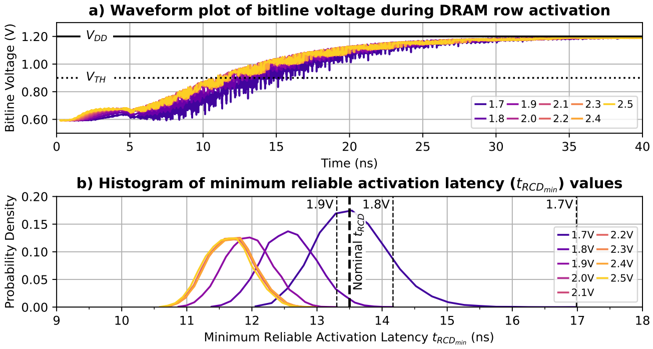

Figure 8: (a) Waveform of the bitline voltage during row activation and (b) probability density distribution of t_{RCDmin} values, for different V_{PP} levels.

Observation 8. *Row activation successfully completes under reduced V_{PP} with an increased activation latency.*

Fig. 8a shows that, as V_{PP} decreases, the bitline voltage takes longer to increase to V_{TH}, resulting in a slower row activation. For example, t_{RCDmin} increases from 11.6 ns to 13.6 ns (on average across 10^4 Monte-Carlo simulation iterations) when V_{PP} is reduced from 2.5 V to 1.7 V. This happens due to two reasons. First, a lower V_{PP} creates a weaker channel in the access transistor, requiring a longer time for the capacitor and bitline to share charge. Second, the charge sharing process (0–5 ns in Fig. 8a) leads to a smaller change in bitline voltage when V_{PP} is reduced due to the weakened charge restoration process that we explain in §6.2.

Fig. 8b shows the probability density distribution of t_{RCDmin} values under reduced V_{PP} across a total of 10^4 Monte-Carlo simulation iterations for different V_{PP} levels (color-coded). Vertical lines annotate the worst-case reliable t_{RCDmin} values across all iterations of our Monte-Carlo simulation (§4.5) for different V_{PP} levels. We make Obsv. 9 from Fig. 8b.

Observation 9. *SPICE simulations agree with our activation latency-related observations based on experiments on real DRAM chips: t_{RCDmin} increases with reduced V_{PP}.*

We analyze the variation in 1) the probability density distribution of t_{RCDmin}, and 2) the worst-case (largest) reliable t_{RCDmin} value when V_{PP} is reduced. Fig. 8b shows that the probability density distribution of t_{RCDmin} both shifts to larger values and becomes wider with reduced V_{PP}. The worst-case (largest) t_{RCDmin} increases from 12.9 ns to 13.3 ns, 14.2 ns, and 16.9 ns when V_{PP} is reduced from 2.5 V to 1.9 V, 1.8 V and 1.7 V, respectively.[13] For a realistic nominal value of 13.5 ns, t_{RCD}'s guardband reduces from 4.4 % to 1.5 % as V_{PP} reduces from 2.5 V to 1.9 V. As §4.5 explains, SPICE simulation results do *not* exactly match measured real-device characteristics (shown in Obsv. 7) because a SPICE model *cannot* simulate a real DRAM chip's exact behavior without proprietary design and manufacturing information.

From Obsvs. 7–9, we conclude that 1) the reliable row activation latency increases with reduced V_{PP}, 2) the increase in reliable row activation latency does *not* immediately require in-

[13] SPICE simulation results do not show reliable operation when $V_{PP} \leq 1.6$ V, yet real DRAM chips do operate reliably as we show in §6.1 and §6.3.

creasing the nominal t_{RCD}, but reduces the available guardband by 21.9 % for 208 out of 272 tested chips, and 3) observed bit flips can be eliminated by increasing t_{RCD} to 24 ns and 15 ns for erroneous modules from Mfrs. A and B.

6.2. DRAM Charge Restoration Under Reduced V_{PP}

Motivation. A DRAM cell's charge restoration process is affected by V_{PP} because, similar to the row activation process, a DRAM cell capacitor's charge is restored through the channel formed in the access transistor, which is controlled by the wordline. Due to access transistor's characteristics, reducing V_{PP} without changing V_{DD} reduces gate-to-source voltage (V_{GS}) and forms a weaker channel. To understand the impact of V_{PP} reduction on the charge restoration process, we investigate how charge restoration of a DRAM cell varies with reduced V_{PP}.

Experimental Results. Since our FPGA infrastructure cannot probe a DRAM cell capacitor's voltage level, we conduct this study in our SPICE simulation environment (§4.5). Fig. 9a shows the waveform plot of capacitor voltage (y-axis) over time (x-axis), following a row activation event (at t=0). Fig. 9b shows the probability density distribution (y-axis) of the minimum latency required (t_{RASmin}) to reliably complete the charge restoration process on the x-axis under different V_{PP} levels. We make Obsvs. 10 and 11 from Fig. 9a and 9b.

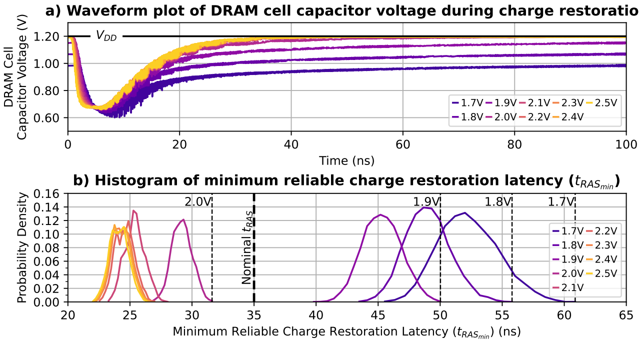

Figure 9: (a) Waveform of the cell capacitor voltage following a row activation and (b) probability density distribution of t_{RASmin} values, for different V_{PP} levels.

Observation 10. *A DRAM cell's capacitor voltage can saturate at a lower voltage level when V_{PP} is reduced.*

We observe that a DRAM cell capacitor's voltage saturates at V_{DD} (1.2 V) when V_{PP} is 2.0 V or higher. However, the cell capacitor's voltage saturates at a lower voltage level by 4.1 %, 11.0 %, and 18.1 % when V_{PP} is 1.9 V, 1.8 V, and 1.7 V, respectively. This happens because the access transistor turns off when the voltage difference between its gate and source is smaller than a threshold level. For example, when V_{PP} is set to 1.7 V, the access transistor allows charge restoration until the cell voltage reaches 0.98 V. When the cell voltage reaches this level, the voltage difference between the gate (1.7 V) and the source (0.98 V) is not large enough to form a strong channel, causing the cell voltage to saturate at 0.98 V. This reduction in voltage can potentially 1) increase the row activation latency (t_{RCD}) and 2) reduce the cell's retention time. We 1) already account for reduced saturation voltage's effect on t_{RCD} in §6.1 and 2) investigate its effect on retention time in §6.3.

Observation 11. *The increase in a DRAM cell's charge restoration latency with reduced V_{PP} can increase the t_{RAS} timing parameter, depending on the V_{PP} level.*

Similar to the variation in t_{RCD} values that we discuss in Obsv. 9, the probability density distribution of t_{RAS} values also shifts to larger values (i.e., t_{RAS} exceeds the nominal value when V_{PP} is lower than 2.0V) and becomes wider as V_{PP} reduces. This happens as a result of reduced cell voltage, weakened channel in the access transistor, and reduced voltage level at the end of the charge sharing process, as we explain in Obsv. 9.

From Obsvs. 10 and 11, we conclude that reducing V_{PP} can negatively affect the charge restoration process. Reduced V_{PP}'s negative impact on charge restoration can potentially be mitigated by leveraging the guardbands in DRAM timing parameters [58,60,69,72,81] and using intelligent DRAM refresh techniques, where a partially restored DRAM row can be refreshed more frequently than other rows, so that the row's charge is restored before it experiences a data retention bit flip [75,144,145]. We leave exploring such solutions to future work.

6.3. DRAM Row Refresh Under Reduced V_{PP}

Motivation. §6.2 demonstrates that the charge restored in a DRAM cell after a row activation can be reduced as a result of V_{PP} reduction. This phenomenon is important for DRAM-based memories because reduced charge in a cell might reduce a DRAM cell's data retention time, causing *retention bit flips* if the cell is *not* refreshed more frequently. To understand the impact of V_{PP} reduction on real DRAM chips, we investigate the effect of reduced V_{PP} on data retention related bit flips using the methodology described in §4.4.

Novelty. This is the first work that experimentally analyzes the isolated impact of V_{PP} on DRAM cell retention times. Prior work [60] tests DDR3 DRAM chips under reduced V_{DD}, which may or may not change the internally-generated V_{PP} level.

Experimental Results. Fig. 10 demonstrates reduced V_{PP}'s effect on data retention BER on real DRAM chips. Fig. 10a shows how the data retention BER (y-axis) changes with increasing refresh window (log-scaled in x-axis) for different V_{PP} levels (color-coded). Each curve in Fig. 10a shows the average BER across all DRAM rows, and error bars mark the 90 % confidence interval. The x-axis starts from 64 ms because we do *not* observe any bit flips at t_{REFW} values smaller than 64 ms. To provide deeper insight into reduced V_{PP}'s effect on data retention BER, Fig. 10b demonstrates the population density distribution of data retention BER across tested rows for a t_{REFW} of 4 s. Dotted vertical lines mark the average BER across rows for each V_{PP} level. We make Obsvs. 12 and 13 from Fig. 10.

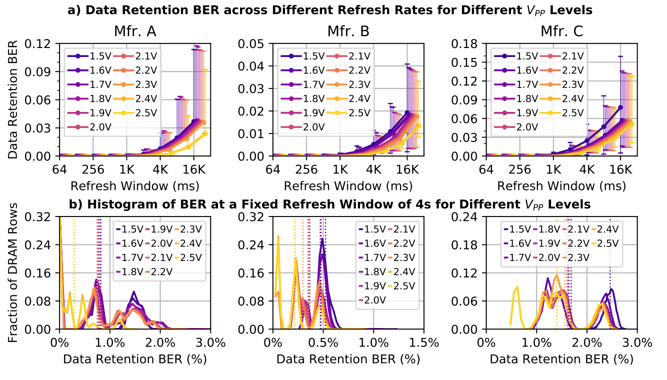

Figure 10: Reduced V_{PP}'s effect on a) data retention BER across different refresh rates and b) the distribution of data retention BER across different DRAM rows for a fixed t_{REFW} of 4 s.

Observation 12. *More DRAM cells tend to experience data retention bit flips when V_{PP} is reduced.*

Fig. 10a shows that data retention *BER* curve is higher (e.g., dark-purple compared to yellow) for smaller V_{PP} levels (e.g., 1.5 V compared to 2.5 V). To provide a deeper insight, Fig. 10b shows that average data retention *BER* across all tested rows when t_{REFW}=4 s increases from 0.3 %, 0.2 %, and 1.4 % for a V_{PP} of 2.5 V to 0.8 %, 0.5 %, and 2.5 % for a V_{PP} of 1.5 V for Mfrs. A, B, and C, respectively. We hypothesize that this happens because of the weakened charge restoration process with reduced V_{PP} (§6.2).

Observation 13. *Even though DRAM cells experience retention bit flips at smaller retention times when V_{PP} is reduced, 23 of 30 tested modules experience* no *data retention bit flips at the nominal refresh window (*64 ms*).*

Data retention *BER* is very low at the t_{REFW} of 64 ms even for a V_{PP} of 1.5 V. We observe that *no* DRAM module from Mfr. A exhibits a data retention bit flip at the 64 ms t_{REFW}, and *only* three and four modules from Mfrs. B (B6, B8, and B9) and C (C1, C3, C5, and C9) experience bit flips across all 30 DRAM modules we test.

We investigate the significance of the observed data retention bit flips and whether it is possible to mitigate these bit flips using error correcting codes (ECC) [54] or other existing methods to avoid data retention bit flips (e.g., selectively refreshing a small fraction of DRAM rows at a higher refresh rate [75, 144, 145]). To do so, we analyze the nature of data retention bit flips when each tested module is operated at the module's V_{PPmin} for two t_{REFW} values: 64 ms and 128 ms, which are the smallest refresh windows that yield non-zero BER for different DRAM modules.

To evaluate whether data retention bit flips can be avoided using ECC, we assume a realistic data word size of 64 bits [32, 123–125, 127, 128, 137]. We make Obsv. 14 from this analysis.

Observation 14. *Data retention errors can be avoided using simple single error correcting codes at the smallest t_{REFW} that yields non-zero BER.*

We observe that *no* 64-bit data word contains more than one bit flip for the smallest t_{REFW} that yield non-zero *BER*. We conclude that simple *single error correction double error detection (SECDED) ECC* can correct *all* erroneous data words.

To evaluate whether data retention bit flips can be avoided by selectively refreshing a small fraction of DRAM rows, we analyze the distribution of these bit flips across different DRAM rows. Fig. 11a (Fig. 11b) shows the distribution of DRAM rows that experience a data retention bit flip when t_{REFW} is 64 ms (128 ms) but *not* at a smaller t_{REFW}, based on their data retention bit flip characteristics. The x-axis shows the number of 64-bit data words with one bit flip in a DRAM row. The y-axis shows the fraction of DRAM rows in log-scale, exhibiting the behavior, specified in the x-axis for different manufacturers (color-coded). We make Obsv. 15 from Fig. 11.

Observation 15. *Only a small fraction (*16.4 % / 5.0 %*) of DRAM rows contain erroneous data words at the smallest t_{REFW} (*64 ms / 128 ms*) that yields non-zero BER.*

Fig. 11a shows that modules from Mfr. A do *not* exhibit any bit flips when t_{REFW} is 64 ms, while 15.5 % and 0.2 % of DRAM rows in modules from Mfrs. B and C exhibit four and one 64-bit words with a single bit flip, respectively; and 0.01 % of DRAM rows from Mfr. B contain 116 data words with one bit flip. Fig. 11b shows that 0.1 %, 4.7 %, and 0.2 % of rows from Mfrs. A, B, and C contain 1, 2, and 1 erroneous data words, respectively, when the refresh window is 128 ms. We conclude that *all* of these data retention bit flips can be avoided by doubling the refresh rate[14] *only* for 16.4 % / 5.0 % of DRAM rows [75, 144, 145] when t_{REFW} is 64 ms / 128 ms.

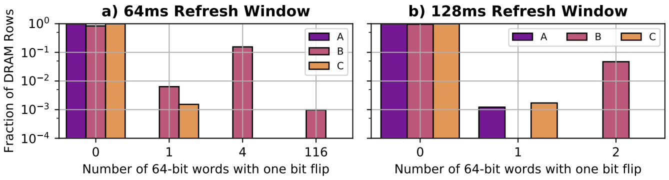

Figure 11: Data retention bit flip characteristics of DRAM rows in DRAM modules that exhibit bit flips at (a) 64 ms **and (b)** 128 ms **refresh windows but not at lower t_{REFW} values when operated at V_{PPmin}. Each subplot shows the distribution of DRAM rows based on the number of erroneous 64-bit words that the rows exhibit.**

From Obsvs. 12–15, we conclude that a DRAM row's data retention time can reduce when V_{PP} is reduced. However, 1) most of (i.e., 23 out of 30) tested modules do *not* exhibit any bit flips at the nominal t_{REFW} of 64 ms and 2) bit flips observed in seven modules can be mitigated using existing SECDED ECC [54] or selective refresh methods [75, 144, 145].

7. Limitations of Wordline Voltage Scaling

We highlight four key limitations of wordline voltage scaling and our experimental characterization.

First, in our experiments, we observe that none of the tested DRAM modules reliably operate at a V_{PP} lower than a certain voltage level, called V_{PPmin}. This happens because an access transistor cannot connect the DRAM cell capacitor to the bitline when the access transistor's gate-to-source voltage difference is *not* larger than the transistor's threshold voltage. Therefore, each DRAM chip has a minimum V_{PP} level at which it can reliably operate (e.g., lowest at 1.4 V for A0 and highest at 2.4 V for A5). With this limitation, we observe 7.4 % / 15.2 % average increase / reduction in HC_{first} / *BER* across all tested DRAM chips at their respective V_{PPmin} levels. A DRAM chip's RowHammer vulnerability can potentially reduce further if access transistors are designed to operate at smaller V_{PP} levels.

Second, we cannot investigate the root cause of all results we observe since 1) DRAM manufacturers do *not* describe the exact circuit design details of their commodity DRAM chips [14, 36, 127, 137] and 2) our infrastructure's physical limitations prevent us from observing a DRAM chip's exact internal behavior (e.g., it is *not* possible to directly measure a cell's capacitor voltage).

Third, this paper does *not* thoroughly analyze the three-way interaction between V_{PP}, temperature, and RowHammer. There is already a complex two-way interaction between RowHammer and temperature, requiring studies to test each DRAM cell at all allowed temperature levels [12]. Since a three-way interaction study requires even more characterization that would take several months of testing time, we leave it to future work to study the interaction between V_{PP}, temperature, and RowHammer.

Fourth, we experimentally demonstrate that the RowHammer vulnerability can be mitigated by reducing V_{PP} at the cost of a 21.9 % average reduction in the t_{RCD} guardband of

[14] We test our chips at fixed refresh rates in increasing powers of two (§4.4). Therefore, our experiments do *not* capture whether eliminating a bit flip is possible by increasing the refresh rate by less than 2×. We leave a finer granularity data retention time analysis to future work.

tested DRAM chips. Although reducing the guardband can hurt DRAM manufacturing yield, we leave studying V_{PP} reduction's effect on yield to future work because we do *not* have access to DRAM manufacturers' proprietary yield statistics.

8. Key Takeaways

We summarize the key findings of our experimental analyses of the wordline voltage (V_{PP})'s effect on the RowHammer vulnerability and reliable operation of modern DRAM chips. From our new observations, we draw two key takeaways.

Takeaway 1: Effect of V_{PP} on RowHammer. We observe that scaling down V_{PP} reduces a DRAM chip's RowHammer vulnerability, such that RowHammer *BER decreases* by 15.2 % (up to 66.9 %) and HC_{first} increases by 7.4 % (up to 85.8 %) on average across all DRAM rows. Only 15.4 % and 14.2 % of DRAM rows exhibit opposite *BER* and HC_{first} trends, respectively (§5.1 and §5.2).

Takeaway 2: Effect of V_{PP} on DRAM reliability. We observe that reducing V_{PP} 1) reduces the existing guardband for row activation latency by 21.9 % on average across tested chips and 2) causes DRAM cell charge to saturate at 1 V instead of 1.2 V (V_{DD}) (§6.2), leading 0 %, 15.5 %, and 0.2 % of DRAM rows to experience SECDED ECC-correctable data retention bit flips at the nominal refresh window of 64 ms in DRAM modules from Mfrs. A, B, and C, respectively (§6.3).

Finding Optimal Wordline Voltage. Our two key takeaways suggest that reducing RowHammer vulnerability of a DRAM chip via V_{PP} reduction can require 1) accessing DRAM rows with a slightly larger latency, 2) employing error correcting codes (ECC), or 3) refreshing a small subset of rows at a higher refresh rate. Therefore, one can define different Pareto-optimal operating conditions for different performance and reliability requirements. For example, a security-critical system can choose a lower V_{PP} to reduce RowHammer vulnerability, whereas a performance-critical and error-tolerant system might prefer lower access latency over higher RowHammer tolerance. DRAM designs and systems that are informed about the trade-offs between V_{PP}, access latency, and retention time can make better-informed design decisions (e.g., fundamentally enable lower access latency) or employ better-informed memory controller policies (e.g., using longer t_{RCD}, employing SECDED ECC, or doubling the refresh rate only for a small fraction of rows when the chip operates at reduced V_{PP}). We believe such designs are important to explore in future work. We hope that the new insights we provide can lead to the design of stronger DRAM-based systems against RowHammer along with better-informed DRAM-based system designs.

9. Related Work

To our knowledge, this is the first work that experimentally studies how reducing wordline voltage affects a real DRAM chip's 1) RowHammer vulnerability, 2) row activation latency, 3) charge restoration process, and 4) data retention time. We divide prior work into three categories: 1) explorations of reduced-voltage DRAM operation, 2) experimental characterization studies of the RowHammer vulnerability of real DRAM chips, and 3) RowHammer attacks and defenses.

Reduced-Voltage DRAM Operation. Prior works [60, 146, 147] propose operating DRAM with reduced V_{DD} to improve energy efficiency. [146] and [147] propose dynamic voltage and frequency scaling (DVFS) for DRAM chips and [146] provides results in a real system. [60] proposes to scale down V_{DD} without reducing DRAM chip frequency. To do so, [60] experimentally demonstrates the interaction between V_{DD} and DRAM row access latency in real DDR3 DRAM chips. These three works neither focus on the RowHammer vulnerability nor distinguish between V_{DD} and V_{PP}. Unlike these works, we focus on the impact of V_{PP} (isolated from V_{DD}) on RowHammer and reliable operation characteristics of real DDR4 DRAM chips.

Experimental RowHammer Characterization. Prior works extensively characterize the RowHammer vulnerability in real DRAM chips [3, 6, 11, 12, 36, 43]. These works experimentally demonstrate (using real DDR3, DDR4, and LPDDR4 DRAM chips how) a DRAM chip's RowHammer vulnerability varies with 1) DRAM refresh rate [3, 36, 43], 2) the physical distance between aggressor and victim rows [3, 11], 3) DRAM generation and technology node [3, 11, 12, 43], 4) temperature [6, 12], 5) the time the aggressor row stays active [6, 12], and 6) physical location of the victim DRAM cell [12]. None of these works analyze how reduced V_{PP} affects RowHammer vulnerability in real DRAM chips. Our characterization study furthers the analyses in these works by uncovering new insights into RowHammer behavior and DRAM operation.

RowHammer Attacks and Defenses. Many prior works [3, 4, 6–48] show that RowHammer can be exploited to mount system-level attacks to compromise system security and safety (e.g., to acquire root privileges or leak private data). To protect against these attacks, many prior works [3, 5, 13, 30, 45, 48, 50–52, 65, 80, 91, 96–114] propose RowHammer mitigation mechanisms that prevent RowHammer bit flips from compromising a system. The novel observations we make in this work can be leveraged to reduce RowHammer vulnerability and complement existing RowHammer defense mechanisms, further increasing their effectiveness and reducing their overheads.

10. Conclusion

We present the first experimental RowHammer characterization study under reduced wordline voltage (V_{PP}). Our results, using 272 real DDR4 DRAM chips from three major manufacturers, show that RowHammer vulnerability can be reduced by reducing V_{PP}. Using real-device experiments and SPICE simulations, we demonstrate that although the reduced V_{PP} slightly worsens DRAM access latency, charge restoration process and data retention time, most of (208 out of 272) tested chips reliably work under reduced V_{PP} leveraging already existing guardbands of nominal timing parameters and employing existing ECC or selective refresh techniques. Our findings provide new insights into the increasingly critical RowHammer problem in modern DRAM chips. We hope that they lead to the design of systems that are more robust against RowHammer attacks.

Acknowledgments

We thank our shepherd Karthik Pattabiraman and the anonymous reviewers of DSN 2022 for valuable feedback. We thank the SAFARI Research Group members for valuable feedback and the stimulating intellectual environment they provide. We acknowledge the generous gifts provided by our industrial partners, including Google, Huawei, Intel, Microsoft, and VMware, and support from the Microsoft Swiss Joint Research Center.

References

[1] O. Mutlu, “Memory Scaling: A Systems Architecture Perspective,” in *IMW*, 2013.

[2] J. Meza, Q. Wu, S. Kumar, and O. Mutlu, “Revisiting Memory Errors in Large-Scale Production Data Centers: Analysis and Modeling of New Trends from the Field,” in *DSN*, 2015.

[3] 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.

[4] M. Redeker, B. F. Cockburn, and D. G. Elliott, “An Investigation into Crosstalk Noise in DRAM Structures,” in *MTDT*, 2002.

[5] B. Aichinger, “DDR Memory Errors Caused by Row Hammer,” in *HPEC*, 2015.

[6] K. Park, C. Lim, D. Yun, and S. Baeg, “Experiments and Root Cause Analysis for Active-Precharge Hammering Fault in DDR3 SDRAM under 3xnm Technology,” *Microelectronics Reliability*, 2016.

[7] K. Park, D. Yun, and S. Baeg, “Statistical Distributions of Row-Hammering Induced Failures in DDR3 Components,” *Microelectronics Reliability*, 2016.

[8] O. Mutlu, “The RowHammer Problem and Other Issues We May Face as Memory Becomes Denser,” in *DATE*, 2017.

[9] O. Mutlu and J. S. Kim, “RowHammer: A Retrospective,” *TCAD*, 2019.

[10] T. Yang and X.-W. Lin, “Trap-Assisted DRAM Row Hammer Effect,” *EDL*, 2019.

[11] J. S. Kim, M. Patel, A. G. Yağlıkçı, H. Hassan, R. Azizi, L. Orosa, and O. Mutlu, “Revisiting RowHammer: An Experimental Analysis of Modern Devices and Mitigation Techniques,” in *ISCA*, 2020.

[12] L. Orosa, A. G. Yağlıkçı, 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.

[13] M. Qureshi, “Rethinking ECC in the Era of Row-Hammer,” *DRAMSec*, 2021.

[14] S. Saroiu, A. Wolman, and L. Cojocar, “The Price of Secrecy: How Hiding Internal DRAM Topologies Hurts Rowhammer Defenses,” in *IRPS*, 2022.

[15] A. J. Walker, S. Lee, and D. Beery, “On DRAM RowHammer and the Physics on Insecurity,” *IEEE TED*, 2021.

[16] M. Seaborn and T. Dullien, “Exploiting the DRAM Rowhammer Bug to Gain Kernel Privileges,” *Black Hat*, 2015.

[17] V. van der Veen, Y. Fratantonio, M. Lindorfer, D. Gruss, C. Maurice, G. Vigna, H. Bos, K. Razavi, and C. Giuffrida, “Drammer: Deterministic Rowhammer Attacks on Mobile Platforms,” in *CCS*, 2016.

[18] D. Gruss, C. Maurice, and S. Mangard, “Rowhammer.js: A Remote Software-Induced Fault Attack in Javascript,” arXiv:1507.06955 [cs.CR], 2016.

[19] K. Razavi, B. Gras, E. Bosman, B. Preneel, C. Giuffrida, and H. Bos, “Flip Feng Shui: Hammering a Needle in the Software Stack,” in *USENIX Security*, 2016.

[20] P. Pessl, D. Gruss, C. Maurice, M. Schwarz, and S. Mangard, “DRAMA: Exploiting DRAM Addressing for Cross-CPU Attacks,” in *USENIX Security*, 2016.

[21] Y. Xiao, X. Zhang, Y. Zhang, and R. Teodorescu, “One Bit Flips, One Cloud Flops: Cross-VM Row Hammer Attacks and Privilege Escalation,” in *USENIX Security*, 2016.

[22] E. Bosman, K. Razavi, H. Bos, and C. Giuffrida, “Dedup Est Machina: Memory Deduplication as an Advanced Exploitation Vector,” in *S&P*, 2016.

[23] S. Bhattacharya and D. Mukhopadhyay, “Curious Case of Rowhammer: Flipping Secret Exponent Bits Using Timing Analysis,” in *CHES*, 2016.

[24] R. Qiao and M. Seaborn, “A New Approach for RowHammer Attacks,” in *HOST*, 2016.

[25] Y. Jang, J. Lee, S. Lee, and T. Kim, “SGX-Bomb: Locking Down the Processor via Rowhammer Attack,” in *SOSP*, 2017.

[26] M. T. Aga, Z. B. Aweke, and T. Austin, “When Good Protections Go Bad: Exploiting Anti-DoS Measures to Accelerate Rowhammer Attacks,” in *HOST*, 2017.

[27] A. Tatar, C. Giuffrida, H. Bos, and K. Razavi, “Defeating Software Mitigations Against Rowhammer: A Surgical Precision Hammer,” in *RAID*, 2018.

[28] D. Gruss, M. Lipp, M. Schwarz, D. Genkin, J. Juffinger, S. O’Connell, W. Schoechl, and Y. Yarom, “Another Flip in the Wall of Rowhammer Defenses,” in *S&P*, 2018.

[29] M. Lipp, M. T. Aga, M. Schwarz, D. Gruss, C. Maurice, L. Raab, and L. Lamster, “Nethammer: Inducing Rowhammer Faults Through Network Requests,” arXiv:1805.04956 [cs.CR], 2018.

[30] V. van der Veen, M. Lindorfer, Y. Fratantonio, H. P. Pillai, G. Vigna, C. Kruegel, H. Bos, and K. Razavi, “GuardION: Practical Mitigation of DMA-Based Rowhammer Attacks on ARM,” in *DIMVA*, 2018.

[31] P. Frigo, C. Giuffrida, H. Bos, and K. Razavi, “Grand Pwning Unit: Accelerating Microarchitectural Attacks with the GPU,” in *S&P*, 2018.

[32] L. Cojocar, K. Razavi, C. Giuffrida, and H. Bos, “Exploiting Correcting Codes: On the Effectiveness of ECC Memory Against Rowhammer Attacks,” in *S&P*, 2019.

[33] S. Ji, Y. Ko, S. Oh, and J. Kim, “Pinpoint Rowhammer: Suppressing Unwanted Bit Flips on Rowhammer Attacks,” in *ASIACCS*, 2019.

[34] S. Hong, P. Frigo, Y. Kaya, C. Giuffrida, and T. Dumitraş, “Terminal Brain Damage: Exposing the Graceless Degradation in Deep Neural Networks Under Hardware Fault Attacks,” in *USENIX Security*, 2019.

[35] A. Kwong, D. Genkin, D. Gruss, and Y. Yarom, “RAMBleed: Reading Bits in Memory Without Accessing Them,” in *S&P*, 2020.

[36] P. Frigo, E. Vannacci, H. Hassan, V. van der Veen, O. Mutlu, C. Giuffrida, H. Bos, and K. Razavi, “TRRespass: Exploiting the Many Sides of Target Row Refresh,” in *S&P*, 2020.

[37] 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 *S&P*, 2020.

[38] Z. Weissman, T. Tiemann, D. Moghimi, E. Custodio, T. Eisenbarth, and B. Sunar, “JackHammer: Efficient Rowhammer on Heterogeneous FPGA–CPU Platforms,” arXiv:1912.11523 [cs.CR], 2020.

[39] Z. Zhang, Y. Cheng, D. Liu, S. Nepal, Z. Wang, and Y. Yarom, “PThammer: Cross-User-Kernel-Boundary Rowhammer through Implicit Accesses,” in *MICRO*, 2020.

[40] SAFARI Research Group, “RowHammer — GitHub Repository,” https://github.com/CMU-SAFARI/rowhammer, 2021.

[41] F. Yao, A. S. Rakin, and D. Fan, “DeepHammer: Depleting the Intelligence of Deep Neural Networks Through Targeted Chain of Bit Flips,” in *USENIX Security*, 2020.

[42] F. de Ridder, P. Frigo, E. Vannacci, H. Bos, C. Giuffrida, and K. Razavi, “SMASH: Synchronized Many-Sided Rowhammer Attacks from JavaScript,” in *USENIX Security*, 2021.

[43] H. Hassan, Y. C. Tugrul, J. S. Kim, V. v. d. Veen, K. Razavi, and O. Mutlu, “Uncovering in-DRAM RowHammer Protection Mechanisms: A New Methodology, Custom RowHammer Patterns, and Implications,” in *MICRO*, 2021.

[44] P. Jattke, V. van der Veen, P. Frigo, S. Gunter, and K. Razavi, “Blacksmith: Scalable Rowhammering in the Frequency Domain,” in *SP*, 2022.

[45] M. Marazzi, P. Jattke, F. Solt, and K. Razavi, “ProTRR: Principled yet Optimal In-DRAM Target Row Refresh,” in *S&P*, 2022.

[46] M. C. Tol, S. Islam, B. Sunar, and Z. Zhang, “Toward Realistic Backdoor Injection Attacks on DNNs using Rowhammer,” arXiv:2110.07683v2 [cs.LG], 2022.

[47] W. Burleson, O. Mutlu, and M. Tiwari, “Invited: Who is the Major Threat to Tomorrow’s Security? You, the Hardware Designer,” in *DAC*, 2016.

[48] F. Brasser, L. Davi, D. Gens, C. Liebchen, and A.-R. Sadeghi, “Can’t Touch This: Software-Only Mitigation Against Rowhammer Attacks Targeting Kernel Memory,” in *USENIX Security*, 2017.

[49] O. Mutlu and L. Subramanian, “Research Problems and Opportunities in Memory Systems,” *SUPERFRI*, 2014.

[50] A. G. Yağlıkçı, M. Patel, J. S. Kim, R. Azizibarzoki, A. Olgun, L. Orosa, H. Hassan, J. Park, K. Kanellopoullos, T. Shahroodi, S. Ghose, and O. Mutlu, “BlockHammer: Preventing RowHammer at Low Cost by Blacklisting Rapidly-Accessed DRAM Rows,” in *HPCA*, 2021.

[51] Y. Park, W. Kwon, E. Lee, T. J. Ham, J. H. Ahn, and J. W. Lee, “Graphene: Strong yet Lightweight Row Hammer Protection,” in *MICRO*, 2020.

[52] A. G. Yağlıkçı, J. S. Kim, F. Devaux, and O. Mutlu, “Security Analysis of the Silver Bullet Technique for RowHammer Prevention,” arXiv:2106.07084 [cs.CR], 2021.

[53] L. Nagel and D. O. Pederson, “SPICE (Simulation Program with Integrated Circuit Emphasis),” 1973.

[54] R. W. Hamming, “Error Detecting and Error Correcting Codes,” *The Bell System Technical Journal*, 1950.

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

[56] B. Keeth, R. J. Baker, B. Johnson, and F. Lin, *DRAM Circuit Design: Fundamental and High-Speed Topics*. John Wiley & Sons, 2007.

[57] D. Lee, Y. Kim, V. Seshadri, J. Liu, L. Subramanian, and O. Mutlu, “Tiered-Latency DRAM: A Low Latency and Low Cost DRAM Architecture,” in *HPCA*, 2013.

[58] D. Lee, Y. Kim, G. Pekhimenko, S. Khan, V. Seshadri, K. Chang, and O. Mutlu, “Adaptive-Latency DRAM: Optimizing DRAM Timing for the Common-Case,” in *HPCA*, 2015.

[59] V. Seshadri, Y. Kim, C. Fallin, D. Lee, R. Ausavarungnirun, G. Pekhimenko, Y. Luo, O. Mutlu, P. B. Gibbons, M. A. Kozuch, and T. Mowry, “RowClone: Fast and Energy-Efficient In-DRAM Bulk Data Copy and Initialization,” in *MICRO*, 2013.

[60] K. K. Chang, A. G. Yağlıkçı, S. Ghose, A. Agrawal, N. Chatterjee, A. Kashyap, D. Lee, M. O’Connor, H. Hassan, and O. Mutlu, “Understanding Reduced-Voltage Operation in Modern DRAM Devices: Experimental Characterization, Analysis, and Mechanisms,” in *SIGMETRICS*, 2017.

[61] K. K. Chang, P. J. Nair, D. Lee, S. Ghose, M. K. Qureshi, and O. Mutlu, “Low-Cost Inter-Linked Subarrays (LISA): Enabling Fast Inter-Subarray Data Movement in DRAM,” in *HPCA*, 2016.

[62] S. Ghose, A. G. Yaglikci, R. Gupta, D. Lee, K. Kudrolli, W. Liu, H. Hassan, K. Chang, N. Chatterjee, A. Agrawal, M. O’Connor, and O. Mutlu, “What Your DRAM Power Models Are Not Telling You: Lessons from a Detailed Experimental Study,” in *SIGMETRICS*, 2018.

[63] H. Hassan, G. Pekhimenko, N. Vijaykumar, V. Seshadri, D. Lee, O. Ergin, and O. Mutlu, “ChargeCache: Reducing DRAM Latency by Exploiting Row Access Locality,” in *HPCA*, 2016.

[64] H. Hassan, N. Vijaykumar, S. Khan, S. Ghose, K. Chang, G. Pekhimenko, D. Lee, O. Ergin, and O. Mutlu, “SoftMC: A Flexible and Practical Open-Source Infrastructure for Enabling Experimental DRAM Studies,” in *HPCA*, 2017.

[65] H. Hassan, M. Patel, J. S. Kim, A. G. Yağlıkçı, N. Vijaykumar, N. Mansouri Ghiasi, S. Ghose, and O. Mutlu, “CROW: A Low-Cost Substrate for Improving DRAM Performance, Energy Efficiency, and Reliability,” in *ISCA*, 2019.

[66] S. Khan, D. Lee, Y. Kim, A. R. Alameldeen, C. Wilkerson, and O. Mutlu, “The Efficacy of Error Mitigation Techniques for DRAM Retention Failures: A Comparative Experimental Study,” in *SIGMETRICS*, 2014.

[67] S. Khan, D. Lee, and O. Mutlu, “PARBOR: An Efficient System-Level Technique to Detect Data-Dependent Failures in DRAM,” in *DSN*, 2016.

[68] S. Khan, C. Wilkerson, Z. Wang, A. R. Alameldeen, D. Lee, and O. Mutlu, “Detecting and Mitigating Data-Dependent DRAM Failures by Exploiting Current Memory Content,” in *MICRO*, 2017.

[69] J. S. Kim, M. Patel, H. Hassan, and O. Mutlu, “Solar-DRAM: Reducing DRAM Access Latency by Exploiting the Variation in Local Bitlines,” in *ICCD*, 2018.

[70] J. S. Kim, M. Patel, H. Hassan, and O. Mutlu, “The DRAM Latency PUF: Quickly Evaluating Physical Unclonable Functions by Exploiting the Latency–Reliability Tradeoff in Modern Commodity DRAM Devices,” in *HPCA*, 2018.

[71] Y. Kim, W. Yang, and O. Mutlu, “Ramulator: A Fast and Extensible DRAM Simulator,” *CAL*, 2016.

[72] D. Lee, S. Khan, L. Subramanian, S. Ghose, R. Ausavarungnirun, G. Pekhimenko, V. Seshadri, and O. Mutlu, “Design-Induced Latency Variation in Modern DRAM Chips: Characterization, Analysis, and Latency Reduction Mechanisms,” in *SIGMETRICS*, 2017.

[73] D. Lee, L. Subramanian, R. Ausavarungnirun, J. Choi, and O. Mutlu, “Decoupled Direct Memory Access: Isolating CPU and IO Traffic by Leveraging a Dual-Data-Port DRAM,” in *PACT*, 2015.

[74] J. Liu, B. Jaiyen, Y. Kim, C. Wilkerson, O. Mutlu, J. Liu, B. Jaiyen, Y. Kim, C. Wilkerson, and O. Mutlu, “An Experimental Study of Data Retention Behavior in Modern DRAM Devices,” in *ISCA*, 2013.

[75] J. Liu, B. Jaiyen, R. Veras, and O. Mutlu, “RAIDR: Retention-Aware Intelligent DRAM Refresh,” in *ISCA*, 2012.

[76] H. Luo, T. Shahroodi, H. Hassan, M. Patel, A. G. Yaglikci, L. Orosa, J. Park, and O. Mutlu, “CLR-DRAM: A Low-Cost DRAM Architecture Enabling Dynamic Capacity-Latency Trade-Off,” in *ISCA*, 2020.

[77] M. Patel, J. S. Kim, and O. Mutlu, “The Reach Profiler (REAPER): Enabling the Mitigation of DRAM Retention Failures via Profiling at Aggressive Conditions,” in *ISCA*, 2017.

[78] M. Qureshi, D.-H. Kim, S. Khan, P. Nair, and O. Mutlu, “AVATAR: A Variable-Retention-Time (VRT) Aware Refresh for DRAM Systems,” in *DSN*, 2015.

[79] V. Seshadri, D. Lee, T. Mullins, H. Hassan, A. Boroumand, J. Kim, M. A. Kozuch, O. Mutlu, P. B. Gibbons, and T. C. Mowry, “Ambit: In-Memory Accelerator for Bulk Bitwise Operations Using Commodity DRAM Technology,” in *MICRO*, 2017.

[80] JEDEC, *JESD79-4C: DDR4 SDRAM Standard*, 2020.

[81] K. K. Chang, A. Kashyap, H. Hassan, S. Ghose, K. Hsieh, D. Lee, T. Li, G. Pekhimenko, S. Khan, and O. Mutlu, “Understanding Latency Variation in Modern DRAM Chips: Experimental Characterization, Analysis, and Optimization,” in *SIGMETRICS*, 2016.

[82] JEDEC, *JESD79-3: DDR3 SDRAM Standard*, 2012.

[83] JEDEC, *JESD79-5: DDR5 SDRAM Standard*, 2020.

[84] JEDEC, *JESD209-4B: Low Power Double Data Rate 4 (LPDDR4) Standard*, 2017.

[85] JEDEC, *JESD232A: Graphics Double Data Rate (GDDR5X) Standard*, 2016.

[86] JEDEC, *JESD250C: Graphics Double Data Rate 6 (GDDR6) Standard*, 2021.

[87] Y. Kim, V. Seshadri, D. Lee, J. Liu, O. Mutlu, Y. Kim, V. Seshadri, D. Lee, J. Liu, and O. Mutlu, “A Case for Exploiting Subarray-Level Parallelism (SALP) in DRAM,” in *ISCA*, 2012.

[88] Micron, “DDR4 SDRAM Datasheet,” in *Micron*, 2016, p. 380.

[89] Micron, “DDR4 SDRAM RDIMM MTA18ASF2G72PZ – 16GB,” 2016.

[90] Micron Technology, “SDRAM, 4Gb: x4, x8, x16 DDR4 SDRAM Features,” 2014.

[91] S.-W. Ryu, K. Min, J. Shin, H. Kwon, D. Nam, T. Oh, T.-S. Jang, M. Yoo, Y. Kim, and S. Hong, “Overcoming the Reliability Limitation in the Ultimately Scaled DRAM using Silicon Migration Technique by Hydrogen Annealing,” in *IEDM*, 2017.

[92] T. Sakurai, “Closed-Form Expressions for Interconnection Delay, Coupling, and Crosstalk in VLSIs,” *IEEE Transactions on Electron Devices*, 1993.

[93] D. Frank, R. Dennard, E. Nowak, P. Solomon, Y. Taur, and H.-S. P. Wong, “Device Scaling Limits of Si MOSFETs and Their Application Dependencies,” *Proceedings of the IEEE*, 2001.

[94] D.-S. Lee, Y.-H. Jun, and B.-S. Kong, “Simultaneous Reverse Body and Negative Word-Line Biasing Control Scheme for Leakage Reduction of DRAM,” *IEEE Journal of Solid-State Circuits*, 2011.

[95] Linear Technology Corp., “LTspice IV,” http://www.linear.com/LTspice.

[96] Apple Inc., “About the Security Content of Mac EFI Security Update 2015-001,” https://support.apple.com/en-us/HT204934, 2015.

[97] Z. B. Aweke, S. F. Yitbarek, R. Qiao, R. Das, M. Hicks, Y. Oren, and T. Austin, “ANVIL: Software-Based Protection Against Next-Generation Rowhammer Attacks,” in *ASPLOS*, 2016.

[98] D.-H. Kim, P. J. Nair, and M. K. Qureshi, “Architectural Support for Mitigating Row Hammering in DRAM Memories,” *CAL*, 2014.

[99] M. Son, H. Park, J. Ahn, and S. Yoo, “Making DRAM Stronger Against Row Hammering,” in *DAC*, 2017.

[100] E. Lee, I. Kang, S. Lee, G. Edward Suh, and J. Ho Ahn, “TWiCe: Preventing Row-Hammering by Exploiting Time Window Counters,” in *ISCA*, 2019.

[101] J. M. You and J.-S. Yang, “MRLoc: Mitigating Row-Hammering Based on Memory Locality,” in *DAC*, 2019.

[102] S. M. Seyedzadeh, A. K. Jones, and R. Melhem, “Mitigating Wordline Crosstalk Using Adaptive Trees of Counters,” in *ISCA*, 2018.

[103] R. K. Konoth, M. Oliverio, A. Tatar, D. Andriesse, H. Bos, C. Giuffrida, and K. Razavi, “ZebRAM: Comprehensive and Compatible Software Protection Against Rowhammer Attacks,” in *OSDI*, 2018.

[104] I. Kang, E. Lee, and J. H. Ahn, “CAT-TWO: Counter-Based Adaptive Tree, Time Window Optimized for DRAM Row-Hammer Prevention,” *IEEE Access*, 2020.

[105] K. Bains, J. Halbert, C. Mozak, T. Schoenborn, and Z. Greenfield, “Row Hammer Refresh Command,” 2015, U.S. Patent 9,117,544.

[106] K. S. Bains and J. B. Halbert, “Distributed Row Hammer Tracking,” 2016, U.S. Patent 9,299,400.

[107] K. S. Bains and J. B. Halbert, “Row Hammer Monitoring Based on Stored Row Hammer Threshold Value,” 2016, U.S. Patent 9,384,821.

[108] H. Gomez, A. Amaya, and E. Roa, “DRAM Row-Hammer Attack Reduction Using Dummy Cells,” in *NORCAS*, 2016.

[109] F. Devaux and R. Ayrignac, “Method and Circuit for Protecting a DRAM Memory Device from the Row Hammer Effect,” 2021, U.S. Patent 10,885,966.

[110] C. Yang, C. K. Wei, Y. J. Chang, T. C. Wu, H. P. Chen, and C. S. Lai, “Suppression of RowHammer Effect by Doping Profile Modification in Saddle-Fin Array Devices for Sub-30-nm DRAM Technology,” *TDMR*, 2016.

[111] C.-M. Yang, C.-K. Wei, H.-P. Chen, J.-S. Luo, Y. J. Chang, T.-C. Wu, and C.-S. Lai, “Scanning Spreading Resistance Microscopy for Doping Profile in Saddle-Fin Devices,” *IEEE Transactions on Nanotechnology*, 2017.

[112] S. Gautam, S. Manhas, A. Kumar, M. Pakala, and E. Yieh, “Row Hammering Mitigation Using Metal Nanowire in Saddle Fin DRAM,” *IEEE TED*, 2019.

[113] Z. Greenfield and T. Levy, “Throttling Support for Row-Hammer Counters,” 2016, U.S. Patent 9,251,885.

[114] G. Saileshwar, B. Wang, M. Qureshi, and P. J. Nair, “Randomized Row-Swap: Mitigating Row Hammer by Breaking Spatial Correlation Between Aggressor and Victim Rows,” in *ASPLOS*, 2022.

[115] SAFARI Research Group, “SoftMC — GitHub Repository,” https://github.com/CMU-SAFARI/softmc, 2021.

[116] Maxwell, “FT20X,” https://www.maxwell-fa.com/upload/files/base/8/m/311.pdf.

[117] Xilinx, “Xilinx Alveo U200 FPGA Board,” https://www.xilinx.com/products/boards-and-kits/alveo/u200.html, 2021.

[118] Adexelec, “DDR4-SOD-V1 260-pin 1.2V, DDR4 SODIMM Vertical Extender with CSR Option,” http://www.adexelec.com/ddr4-sod-v1.

[119] TTi, “PL & PL-P Series DC Power Supplies Data Sheet - Issue 5,” https://resources.aimtti.com/datasheets/AIM-PL+PL-P_series_DC_power_supplies_data_sheet-Iss5.pdf.

[120] T. Hamamoto, S. Sugiura, and S. Sawada, “On the Retention Time Distribution of Dynamic Random Access Memory (DRAM),” *IEEE TED*, 1998.

[121] Micron Technology Inc., “ECC Brings Reliability and Power Efficiency to Mobile Devices,” White Paper, 2017.

[122] P. J. Nair, V. Sridharan, and M. K. Qureshi, “XED: Exposing On-Die Error Detection Information for Strong Memory Reliability,” in *ISCA*, 2016.

[123] M. Patel, G. F. Oliveira, and O. Mutlu, “HARP: Practically and Effectively Identifying Uncorrectable Errors in Main Memory Chips That Use On-Die ECC,” in *MICRO*, 2021.

[124] M. Patel, J. Kim, T. Shahroodi, H. Hassan, and O. Mutlu, “Bit-Exact ECC Recovery (BEER): Determining DRAM On-Die ECC Functions by Exploiting DRAM Data Retention Characteristics (Best Paper),” in *MICRO*, 2020.

[125] M. Patel, J. S. Kim, H. Hassan, and O. Mutlu, “Understanding and Modeling On-Die Error Correction in Modern DRAM: An Experimental Study Using Real Devices,” in *DSN*, 2019.

[126] U. Kang, H.-S. Yu, C. Park, H. Zheng, J. Halbert, K. Bains, S. Jang, and J. S. Choi, “Co-Architecting Controllers and DRAM to Enhance DRAM Process Scaling,” in *The Memory Forum*, 2014.

[127] M. Patel, “Enabling Effective Error Mitigation in Modern Memory Chips that Use On-Die Error-Correcting Codes,” Ph.D. dissertation, ETH Zurich, 2021.

[128] J. Kim, M. Sullivan, S. Lym, and M. Erez, “All-Inclusive ECC: Thorough End-to-End Protection for Reliable Computer Memory,” in *ISCA*, 2016.

[129] JEDEC, *JESD209-5A: LPDDR5 SDRAM Standard*, 2020.

[130] J. Lee, “Green Memory Solution,” Investor’s Forum, Samsung Electronics, 2014.

[131] S. Khan, C. Wilkerson, D. Lee, A. R. Alameldeen, and O. Mutlu, “A Case for Memory Content-Based Detection and Mitigation of Data-Dependent Failures in DRAM,” *CAL*, 2016.

[132] L. Mukhanov, D. S. Nikolopoulos, and G. Karakonstantis, “DStress: Automatic Synthesis of DRAM Reliability Stress Viruses using Genetic Algorithms (Best Paper Nominee),” in *MICRO*, 2020.

[133] A. van de Goor and I. Schanstra, “Address and Data Scrambling: Causes and Impact on Memory Tests,” in *DELTA*, 2002.

[134] A. Barenghi, L. Breveglieri, N. Izzo, and G. Pelosi, “Software-Only Reverse Engineering of Physical DRAM Mappings for Rowhammer Attacks,” in *IVSW*, 2018.

[135] M. Horiguchi, “Redundancy Techniques for High-Density DRAMs,” in *ISIS*, 1997.

[136] K. Itoh, *VLSI Memory Chip Design*. Springer, 2001.

[137] M. Patel, T. Shahroodi, A. Manglik, A. G. Yaglikci, A. Olgun, H. Luo, and O. Mutlu, “A Case for Transparent Reliability in DRAM Systems,” cs.AR:2204.10378, 2022.

[138] SAFARI Research Group, “RowHammer Under Reduced Wordline Voltage — GitHub Repository,” https://github.com/CMU-SAFARI/RowHammer-Under-Reduced-Wordline-Voltage, 2022.

[139] PTM, “Predictive Technology Model,” http://ptm.asu.edu/.

[140] W. Zhao and Y. Cao, “New Generation of Predictive Technology Model for sub-45 nm Early Design Exploration,” *IEEE TED*, 2006.

[141] International Technology Roadmap for Semiconductors, “ITRS Reports,” http://www.itrs2.net/itrs-reports.html, 2015.

[142] T. Vogelsang, “Understanding the Energy Consumption of Dynamic Random Access Memories,” in *MICRO*, 2010.

[143] B. Everitt, *DRAM Circuit Design: Fundamental and High-Speed Topics*. Cambridge University Press, 1998.

[144] A. Das, H. Hassan, and O. Mutlu, “VRL-DRAM: Improving DRAM Performance via Variable Refresh Latency,” in *DAC*, 2018.

[145] Y. Wang, A. Tavakkol, L. Orosa, S. Ghose, N. M. Ghiasi, M. Patel, J. S. Kim, H. Hassan, M. Sadrosadati, and O. Mutlu, “Reducing DRAM Latency via Charge-Level-Aware Look-Ahead Partial Restoration,” in *MICRO*, 2018.

[146] H. David, C. Fallin, E. Gorbatov, U. R. Hanebutte, and O. Mutlu, “Memory Power Management via Dynamic Voltage/Frequency Scaling,” in *ICAC*, 2011.

[147] Q. Deng, D. Meisner, L. Ramos, T. F. Wenisch, and R. Bianchini, “MemScale: Active Low-Power Modes for Main Memory,” in *ASPLOS*, 2011.

[148] Micron, “DDR4 SDRAM RDIMM MTA18ASF2G72PZ – 16GB,” https://www.micro-semiconductor.com/datasheet/7c-MTA18ASF2G72PZ-2G9E1.pdf.

[149] Crucial, “CT4G4DFS8266,” https://www.crucial.com/memory/eol_ddr4/ct4g4dfs8266.

[150] CORSAIR, “SKU CMV4GX4M1A2133C15 Specification,” https://tinyurl.com/CMV4GX4M1A2133C15.

[151] Samsung, “288pin Unbuffered DIMM based on 8Gb D-die, Rev 1.1,” https://semiconductor.samsung.com/resources/data-sheet/DDR4_8Gb_D_die_Unbuffered_DIMM_Rev1.1_Jun.18.pdf, 2018.

[152] GSKill, “F4-2400C17S-8GNT Specifications,” https://www.gskill.com/product/165/186/1535961538/F4-2400C17S-8GNT.

[153] Samsung, “288pin Registered DIMM based on 8Gb B-die, Rev 1.91,” https://semiconductor.samsung.com/resources/data-sheet/20170731_DDR4_8Gb_B_die_Registered_DIMM_Rev1.91_May.17.pdf, 2017.

[154] Samsung, “M471A5143EB0-CPB Specifications,” https://semiconductor.samsung.com/dram/module/sodimm/m471a5143eb0-cpb/.

[155] CORSAIR, “CMK16GX4M2B3200C16,” https://www.corsair.com/eu/en/Categories/Products/Memory/VENGEANCE-LPX/p/CMK16GX4M2B3200C16.

[156] Samsung, “260pin Unbuffered SODIMM based on 8Gb C-die,” https://semiconductor.samsung.com/resources/data-sheet/DDR4_8Gb_C_die_Unbuffered_SODIMM_Rev1.5_Apr.18.pdf, 2018.

[157] Kingston, “KSM32RD8/16HDR Specifications,” https://www.kingston.com/dataSheets/KSM32RD8_16HDR.pdf, 2020.

[158] Memory.NET, “HMAA4GU6AJR8N,” https://memory.net/product/hmaa4gu6ajr8n-xn-sk-hynix-1x-32gb-ddr4-3200-udimm-pc4-25600u-dual-rank-x8-module/.

Appendix A. Tested DRAM Modules

Table 3 shows the characteristics of the DDR4 DRAM modules we test and analyze.[15] For each DRAM module, we provide the 1) DRAM chip manufacturer, 2) DIMM name, 3) DIMM model,[16] 4) die density, 5) data transfer frequency, 6) chip organization, 7) die revision, specified in the module's serial presence detect (SPD) registers, 8) manufacturing date, specified on the module's label in the form of $week - year$, and 9) RowHammer vulnerability characteristics of the module. Table 3 reports the RowHammer vulnerability characteristics of each DIMM under two wordline voltage (V_{PP}) levels: *i)* nominal V_{PP} (2.5 V) and *ii)* the lowest V_{PP} at which the DRAM module can successfully communicate with the FPGA (V_{PPmin}). We quantify a DIMM's RowHammer vulnerability characteristics at a given V_{PP} in terms of two metrics: *i)* the minimum aggressor row activation count necessary to cause a RowHammer bit flip (HC_{first}) and *ii)* the fraction of DRAM cells that experience a bit flip in a DRAM row (*BER*). Based on these two metrics at nominal V_{PP} and V_{PPmin}, Table 3 also provides a *recommended* V_{PP} level ($V_{PP_{Rec}}$) and the corresponding RowHammer characteristics in the right-most three columns.

Table 3: Tested DRAM modules and their characteristics when V_{PP}=2.5 V (nominal) and V_{PP}=V_{PPmin}. V_{PPmin} is specified for each module.

								V_{PP} = 2.5V			**$V_{PP} = V_{PP_{min}}$**			**$V_{PP} = V_{PP_{Rec}}$**	
DRAM Chip Mfr.	**DIMM Name**	**DIMM Model**	**Die Density**	**Frequency (MT/s)**	**Chip Org.**	**Die Revision**	**Mfr. Date**	**Minimum HC_{first}**	**BER**	**$V_{PP_{min}}$**	**Minimum HC_{first}**	**BER**	**Recommended $V_{PP}(V_{PP_{Rec}})$**	**Minimum HC_{first}**	**BER**
Mfr. A (Micron)	A0	MTA18ASF2G72PZ-2G3B1QK [148]	8Gb	2400	x4	B	11-19	39.8K	1.24e-03	1.4	42.2K	1.00e-03	1.4	42.2K	1.00e-03
	A1	MTA18ASF2G72PZ-2G3B1QK [148]	8Gb	2400	x4	B	11-19	42.2K	9.90e-04	1.4	46.4K	7.83e-04	1.4	46.4K	7.83e-04
	A2	MTA18ASF2G72PZ-2G3B1QK [148]	8Gb	2400	x4	B	11-19	41.0K	1.24e-03	1.7	39.8K	1.35e-03	2.1	42.1K	1.55e-3
	A3	CT4G4DFS8266.C8FF [149]	4Gb	2666	x8	F	07-21	16.7K	3.33e-02	1.4	16.5K	3.52e-02	1.7	17.0K	3.48e-02
	A4	CT4G4DFS8266.C8FF [149]	4Gb	2666	x8	F	07-21	14.4K	3.18e-02	1.5	14.4K	3.33e-02	2.5	14.4K	3.18e-02
	A5	CT4G4SFS8213.C8FBD1	4Gb	2400	x8	-	48-16	140.7K	1.39e-06	2.4	145.4K	3.39e-06	2.4	145.4K	3.39e-06
	A6	CT4G4DFS8266.C8FF [149]	4Gb	2666	x8	F	07-21	16.5K	3.50e-02	1.5	16.5K	3.66e-02	2.5	16.5K	3.50e-02
	A7	CMV4GX4M1A2133C15 [150]	4Gb	2133	x8	-	-	16.5K	3.42e-02	1.8	16.5K	3.52e-02	2.5	16.5K	3.42e-02
	A8	MTA18ASF2G72PZ-2G3B1QG [148]	8Gb	2400	x4	B	11-19	35.2K	2.38e-03	1.4	39.8K	2.07e-03	1.4	39.8K	2.07e-03
	A9	CMV4GX4M1A2133C15 [150]	4Gb	2133	x8	-	-	14.3K	3.33e-02	1.5	14.3K	3.48e-02	1.6	14.6K	3.47e-02
Mfr. B (Samsung)	B0	M378A1K43DB2-CTD [151]	8Gb	2666	x8	D	10-21	7.9K	1.18e-01	2.0	7.6K	1.22e-01	2.5	7.9K	1.18e-01
	B1	M378A1K43DB2-CTD [151]	8Gb	2666	x8	D	10-21	7.3K	1.26e-01	2.0	7.6K	1.28e-01	2.0	7.6K	1.28e-01
	B2	F4-2400C17S-8GNT [152]	4Gb	2400	x8	F	02-21	11.2K	2.52e-02	1.6	12.0K	2.22e-02	1.6	12.0K	2.22e-02
	B3	M393A1K43BB1-CTD6Y [153]	8Gb	2666	x8	B	52-20	16.6K	2.73e-03	1.6	21.1K	1.09e-03	1.6	21.1K	1.09e-03
	B4	M393A1K43BB1-CTD6Y [153]	8Gb	2666	x8	B	52-20	21.0K	2.95e-03	1.8	19.9K	2.52e-03	2.0	21.1K	2.68e-03
	B5	M471A5143EB0-CPB [154]	4Gb	2133	x8	E	08-17	21.0K	7.78e-03	1.8	21.0K	6.02e-03	2.0	21.1K	8.67e-03
	B6	CMK16GX4M2B3200C16 [155]	8Gb	3200	x8	-	-	10.3K	1.14e-02	1.7	10.5K	9.82e-03	1.7	10.5K	9.82e-03
	B7	M378A1K43DB2-CTD [151]	8Gb	2666	x8	D	10-21	7.3K	1.32e-01	2.0	7.6K	1.33e-01	2.0	7.6K	1.33e-01
	B8	CMK16GX4M2B3200C16 [155]	8Gb	3200	x8	-	-	11.6K	2.88e-02	1.7	10.5K	2.37e-02	1.8	11.7K	2.58e-02
	B9	M471A5244CB0-CRC [156]	8Gb	2133	x8	C	19-19	11.8K	2.68e-02	1.7	8.8K	2.39e-02	1.8	12.3K	2.54e-02
Mfr. C (SK Hynix)	C0	F4-2400C17S-8GNT [152]	4Gb	2400	x8	B	02-21	19.3K	7.29e-03	1.7	23.4K	6.61e-03	1.7	23.4K	6.61e-03
	C1	F4-2400C17S-8GNT [152]	4Gb	2400	x8	B	02-21	19.3K	6.31e-03	1.7	20.6K	5.90e-03	1.7	20.6K	5.90e-03
	C2	KSM32RD8/16HDR [157]	8Gb	3200	x8	D	48-20	9.6K	2.82e-02	1.5	9.2K	2.34e-02	2.3	10.0K	2.89e-02
	C3	KSM32RD8/16HDR [157]	8Gb	3200	x8	D	48-20	9.3K	2.57e-02	1.5	8.9K	2.21e-02	2.3	9.7K	2.66e-02
	C4	HMAA4GU6AJR8N-XN [158]	16Gb	3200	x8	A	51-20	11.6K	3.22e-02	1.5	11.7K	2.88e-02	1.5	11.7K	2.88e-02
	C5	HMAA4GU6AJR8N-XN [158]	16Gb	3200	x8	A	51-20	9.4K	3.28e-02	1.5	12.7K	2.85e-02	1.5	12.7K	2.85e-02
	C6	CMV4GX4M1A2133C15 [150]	4Gb	2133	x8	C	-	14.2K	3.08e-02	1.6	15.5K	2.25e-02	1.6	15.5K	2.25e-02
	C7	CMV4GX4M1A2133C15 [150]	4Gb	2133	x8	C	-	11.7K	3.24e-02	1.6	13.6K	2.60e-02	1.6	13.6K	2.60e-02
	C8	KSM32RD8/16HDR [157]	8Gb	3200	x8	D	48-20	11.4K	2.69e-02	1.6	9.5K	2.57e-02	2.5	11.4K	2.69e-02
	C9	F4-2400C17S-8GNT [152]	4Gb	2400	x8	B	02-21	12.6K	2.18e-02	1.7	15.2K	1.63e-02	1.7	15.2K	1.63e-02

[15] All tested DRAM modules implement the DDR4 DRAM standard [80]. We make our best effort in identifying the DRAM chips used in our tests. We identify the DRAM chip density and die revision through the original manufacturer markings on the chip. For certain DIMMs we tested, the original DRAM chip markings are removed by the DIMM manufacturer. In this case, we can only identify the chip manufacturer and density by reading the information stored in the SPD. However, these DIMM manufacturers also tend to remove the die revision information in the SPD. Therefore, we *cannot* identify the die revision of five DIMMs and the manufacturing date of six DIMMs we test, shown as '-' in the table.

[16] DIMM models CMV4GX4M1A2133C15 and F4-2400C17S-8GNT appear in more than one DRAM chip manufacturer because different batches of these modules use DRAM chips from different manufacturers (i.e., Micron-SK Hynix and Samsung-SK Hynix, respectively) across different batches.