Understanding Reduced-Voltage Operation in Modern DRAM Chips: Characterization, Analysis, and Mechanisms

Kevin K. Chang[†] Abdullah Giray Yağlıkçı[†] Saugata Ghose[†] Aditya Agrawal[¶] Niladrish Chatterjee[¶]
Abhijith Kashyap[†] Donghyuk Lee[¶] Mike O'Connor[¶,‡] Hasan Hassan[§] Onur Mutlu[§,†]

[†]Carnegie Mellon University [¶]NVIDIA [‡]The University of Texas at Austin [§]ETH Zürich

ABSTRACT

The energy consumption of DRAM is a critical concern in modern computing systems. Improvements in manufacturing process technology have allowed DRAM vendors to lower the DRAM supply voltage conservatively, which reduces some of the DRAM energy consumption. We would like to reduce the DRAM supply voltage more aggressively, to further reduce energy. Aggressive supply voltage reduction requires a thorough understanding of the effect voltage scaling has on DRAM access latency and DRAM reliability.

In this paper, we take a comprehensive approach to understanding and exploiting the latency and reliability characteristics of modern DRAM when the supply voltage is lowered below the nominal voltage level specified by DRAM standards. Using an FPGA-based testing platform, we perform an experimental study of 124 real DDR3L (low-voltage) DRAM chips manufactured recently by three major DRAM vendors. We find that reducing the supply voltage below a certain point introduces bit errors in the data, and we comprehensively characterize the behavior of these errors. We discover that these errors can be avoided by increasing the latency of three major DRAM operations (activation, restoration, and precharge). We perform detailed DRAM circuit simulations to validate and explain our experimental findings. We also characterize the various relationships between reduced supply voltage and error locations, stored data patterns, DRAM temperature, and data retention.

Based on our observations, we propose a new DRAM energy reduction mechanism, called *Voltron*. The key idea of Voltron is to use a performance model to determine by how much we can reduce the supply voltage without introducing errors and without exceeding a user-specified threshold for performance loss. Our evaluations show that Voltron reduces the average DRAM and system energy consumption by 10.5% and 7.3%, respectively, while limiting the average system performance loss to only 1.8%, for a variety of memory-intensive quad-core workloads. We also show that Voltron significantly outperforms prior dynamic voltage and frequency scaling mechanisms for DRAM.

1 INTRODUCTION

In a wide range of modern computing systems, spanning from warehouse-scale data centers to mobile platforms, energy consumption is a first-order concern [26, 32, 35, 45, 55, 87, 94, 100, 137]. In these systems, the energy consumed by the DRAM-based main memory system constitutes a significant fraction of the total energy. For example, experimental studies of production systems have shown that DRAM consumes 40% of the total energy in servers [45, 133] and 40% of the total power in graphics cards [107].

The energy consumed by DRAM is correlated with the *supply voltage* used within the DRAM chips. The supply voltage is distributed to the two major components within DRAM: the *DRAM array* and the *peripheral circuitry* [73, 131]. The DRAM array consists of thousands of capacitor-based DRAM cells, which store data as charge within the capacitor. Accessing data stored in the *DRAM array* requires a DRAM chip to perform a series of fundamental operations: activation, restoration, and precharge.[1] A memory controller orchestrates each of the DRAM operations while obeying the DRAM timing parameters. On the other hand, the *peripheral circuitry* consists of control logic and I/O drivers that connect the DRAM array to the memory channel, which is responsible for transferring commands and data between the memory controller and the DRAM chip. Since the DRAM supply voltage is distributed to both the DRAM array and the peripheral circuitry, changing the supply voltage would affect the energy consumption of both components in the entire DRAM chip.

To reduce the energy consumed by DRAM, vendors have developed low-voltage variants of DDR (Double Data Rate) memory, such as LPDDR4 (Low-Power DDR4) [52] and DDR3L (DDR3 Low-voltage) [51]. For example, in DDR3L, the internal architecture remains the same as DDR3 DRAM, but vendors lower the nominal supply voltage to both the DRAM array and the peripheral circuitry via improvements in manufacturing process technology. In this work, we would like to reduce DRAM energy by *further reducing DRAM supply voltage*. Vendors choose a conservatively high supply voltage, to provide a *guardband* that allows DRAM chips with the worst-case process variation to operate without errors under the worst-case operating conditions [32]. The exact amount of supply voltage guardband varies across chips, and lowering the voltage below the guardband can result in erroneous or even undefined behavior. Therefore, we need to understand how DRAM chips behave during reduced-voltage operation. To our knowledge, no previously published work examines the effect of using a wide range of different supply voltage values on the reliability, latency, and retention characteristics of DRAM chips.

Our goal in this work is to *(i)* characterize and understand the relationship between supply voltage reduction and various characteristics of DRAM, including DRAM reliability, latency, and data retention; and *(ii)* use the insights derived from this characterization and understanding to design a new mechanism that can aggressively lower the supply voltage to reduce DRAM energy consumption while keeping performance loss under a bound. To this end, we build an FPGA-based testing platform that allows us to tune the DRAM supply voltage [43]. Using this testing platform, we perform experiments on 124 real DDR3L DRAM chips [51] from

[1]We explain the detail of each of these operations in Section 2.

three major vendors, contained within 31 dual in-line memory modules (DIMMs). Our comprehensive experimental characterization provides four major observations on how DRAM latency, reliability, and data retention time are affected by reduced supply voltage.

First, we observe that we can reliably access data when DRAM supply voltage is lowered below the nominal voltage, *until a certain voltage value*, V_{min}, which is the minimum voltage level at which no bit errors occur. Furthermore, we find that we can reduce the voltage below V_{min} to attain further energy savings, but that errors start occurring in some of the data read from memory. As we drop the voltage further below V_{min}, the number of erroneous bits of data increases exponentially with the voltage drop.

Second, we observe that while reducing the voltage below V_{min} introduces bit errors in the data, we can prevent these errors if we *increase* the latency of the three fundamental DRAM operations (activation, restoration, and precharge). When the supply voltage is reduced, the capacitor charge takes a longer time to change, thereby causing these DRAM operations to become slower to complete. Errors are introduced into the data when the memory controller does *not* account for this slowdown in the DRAM operations. We find that if the memory controller allocates extra time for these operations to finish when the supply voltage is below V_{min}, errors no longer occur. We validate, analyze, and explain this behavior using detailed circuit-level simulations.

Third, we observe that when only a small number of errors occur due to reduced supply voltage, these errors tend to *cluster* physically in certain *regions* of a DRAM chip, as opposed to being randomly distributed throughout the chip. This observation implies that when we reduce the supply voltage to the DRAM array, we need to increase the fundamental operation latencies for *only* the regions where errors can occur.

Fourth, we observe that reducing the supply voltage does *not* impact the data retention guarantees of DRAM. Commodity DRAM chips guarantee that all cells can safely retain data for 64ms, after which the cells are *refreshed* to replenish charge that leaks out of the capacitors. Even when we reduce the supply voltage, the rate at which charge leaks from the capacitors is so slow that no data is lost during the 64ms refresh interval at 20℃ and 70℃ ambient temperature.

Based on our experimental observations, we propose a new low-cost DRAM energy reduction mechanism called *Voltron*. The key idea of Voltron is to use a performance model to determine by how much we can reduce the DRAM array voltage at runtime without introducing errors and without exceeding a user-specified threshold for acceptable performance loss. Voltron consists of two components: *array voltage scaling* and *performance-aware voltage control*.

Array voltage scaling leverages minimal hardware modifications within DRAM to reduce the voltage of *only* the DRAM array, without affecting the voltage of the peripheral circuitry. If Voltron were to reduce the voltage of the peripheral circuitry, we would have to reduce the operating frequency of DRAM. This is because the *maximum* operating frequency of DRAM is a function of the peripheral circuitry voltage [32]. A reduction in the operating frequency reduces the memory data throughput, which can significantly harm the performance of applications that require high memory bandwidth, as we demonstrate in this paper.

Performance-aware voltage control uses performance counters within the processor to build a piecewise linear model of how the performance of an application decreases as the DRAM array supply voltage is lowered (due to longer operation latency to prevent errors), and uses the model to select a supply voltage that keeps performance above a user/system-specified performance target.

Our evaluations of Voltron show that it significantly reduces both DRAM and system energy consumption, at the expense of very modest performance degradation. For example, at an average performance loss of only 1.8% over seven memory-intensive quad-core workloads from SPEC2006, Voltron reduces DRAM energy consumption by an average of 10.5%, which translates to an overall system energy consumption of 7.3%. We also show that Voltron effectively saves DRAM and system energy on even non-memory-intensive applications, with very little performance impact.

This work makes the following major contributions:

- We perform the first detailed experimental characterization of how the reliability and latency of modern DRAM chips are affected when the supply voltage is lowered below the nominal voltage level. We comprehensively test and analyze 124 real DRAM chips from three major DRAM vendors. Our characterization reveals four new major observations, which can be useful for developing new mechanisms that improve or better trade off between DRAM energy/power, latency, and/or reliability.
- We experimentally demonstrate that reducing the supply voltage below a certain point introduces bit errors in the data read from DRAM. We show that we can avoid these bit errors by increasing the DRAM access latency when the supply voltage is reduced.
- We propose Voltron, a mechanism that *(i)* reduces the supply voltage to only the DRAM array without affecting the peripheral circuitry, and *(ii)* uses a performance model to select a voltage that does not degrade performance beyond a chosen threshold. We show that Voltron is effective at improving system energy consumption, with only a small impact on performance.
- We open-source our FPGA-based experimental characterization infrastructure and DRAM circuit simulation infrastructure, used in this paper, for evaluating reduced-voltage operation [3].

2 BACKGROUND AND MOTIVATION

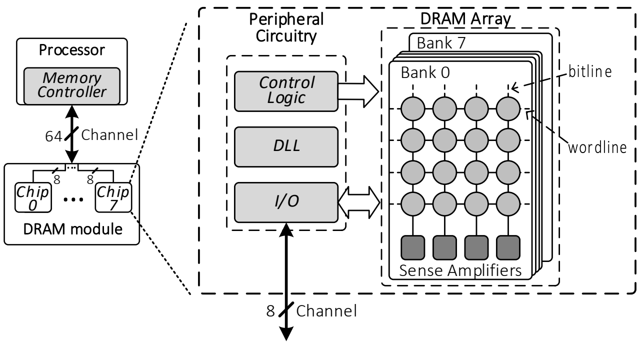
In this section, we first provide necessary DRAM background and terminology. We then discuss related work on reducing the voltage and/or frequency of DRAM, to motivate the need for our study.

2.1 DRAM Organization

Figure 1a shows a high-level overview of a modern memory system organization. A processor (CPU) is connected to a DRAM module via a *memory channel*, which is a bus used to transfer data and commands between the processor and DRAM. A DRAM module is also called a *dual in-line memory module* (DIMM) and it consists of multiple *DRAM chips*, which are controlled together.[2] Within each DRAM chip, illustrated in Figure 1b, we categorize the internal components into two broad categories: *(i)* the *DRAM array*, which

[2]In this paper, we study DIMMs that contain a single *rank* (i.e., a group of chips in a single DIMM that operate in lockstep).

consists of multiple banks of DRAM cells organized into rows and columns, and *(ii) peripheral circuitry*, which consists of the circuits that sit outside of the DRAM array. For a more detailed view of the components in a DRAM chip, we refer the reader to prior works [19–22, 44, 64, 68, 72–76, 78, 84, 113–116, 131].



(a) DRAM System **(b) DRAM Chip**

Figure 1: DRAM system and chip organization.

A DRAM array is divided into multiple banks (typically eight in DDR3 DRAM [50, 51]) that can process DRAM commands independently from each other to increase parallelism. A bank contains a 2-dimensional array of DRAM cells. Each cell uses a capacitor to store a single bit of data. Each array of cells is connected to a row of sense amplifiers via vertical wires, called *bitlines*. This row of sense amplifiers is called the *row buffer*. The row buffer senses the data stored in one row of DRAM cells and serves as a temporary buffer for the data. A typical row in a DRAM module (i.e., across all of the DRAM chips in the module) is 8KB wide, comprising 128 64-byte cache lines.

The peripheral circuitry has three major components. First, the I/O component is used to receive commands or transfer data between the DRAM chip and the processor via the memory channel. Second, a typical DRAM chip uses a delay-lock loop (DLL) to synchronize its data signal with the external clock to coordinate data transfers on the memory channel. Third, the control logic decodes DRAM commands sent across the memory channel and selects the row and column of cells to read data from or write data into.

2.2 Accessing Data in DRAM

To access data stored in DRAM, the *memory controller* (shown in Figure 1a) issues DRAM commands across the memory channel to the DRAM chips. Reading a cache line from DRAM requires three essential commands, as shown in Figure 2: `ACTIVATE`, `READ`, and `PRECHARGE`. Each command requires some time to complete, and the DRAM standard [51] defines the latency of the commands with a set of *timing parameters*. The memory controller can be programmed to obey different sets of timing parameters through the BIOS [4, 5, 47, 75].

Activate Command. To open the target row of data in the bank that contains the desired cache line, the memory controller first issues an `ACTIVATE` command to the target DRAM bank. During activation, the electrical charge stored in the target row starts to

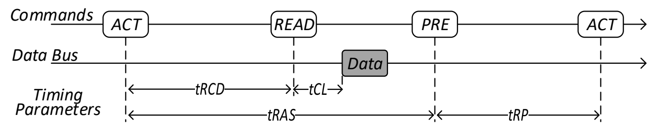


Figure 2: DRAM commands and timing parameters when reading one cache line of data.

propagate to the row buffer. The charge propagation triggers the row buffer to latch the data stored in the row after some amount of time. The latency of an `ACTIVATE` command, or the *activation latency*, is defined as the minimum amount of time that is required to pass from the issue time of an `ACTIVATE` until the issue time of a column command (i.e., `READ` or `WRITE`). The timing parameter for the activation latency is called tRCD, as shown in Figure 2, and is typically set to 13ns in DDR3L [92].

Since an activation drains charge from the target row's cells to latch the cells' data into the row buffer, the cells' charge needs to be restored to prevent data loss. The row buffer performs *charge restoration* simultaneously with activation. Once the cells' charge is fully restored, the row can be closed (and thus the DRAM array be prepared for the next access) by issuing a `PRECHARGE` command to the DRAM bank. The DRAM standard specifies the restoration latency as the minimum amount of time the controller must wait after `ACTIVATE` before issuing `PRECHARGE`. The timing parameter for restoration is called tRAS, as shown in Figure 2, and is typically set to 35ns in DDR3L [92].

Read Command. Once the row data is latched in the row buffer after the `ACTIVATE` command, the memory controller issues a `READ` command. The row buffer contains multiple cache lines of data (8KB), and the `READ` command enables all *n* DRAM chips in the DRAM module to select the desired cache line (64B) from the row buffer. Each DRAM chip on the module then drives $(1/n)^{th}$ of the cache line from the row buffer to the I/O component within the peripheral circuitry. The peripheral circuitry of each chip then sends its $(1/n)^{th}$ of the cache line across the memory channel to the memory controller. Note that the column access time to read and write the cache line is defined by the timing parameters tCL and tCWL, respectively, as shown in Figure 2. Unlike the activation latency (tRCD), tCL and tCWL are *DRAM-internal* timings that are determined by a clock inside DRAM [92]. Therefore, our FPGA-based experimental infrastructure (described in Section 3) *cannot* evaluate the effect of changing tCL and tCWL.

Precharge Command. After reading the data from the row buffer, the memory controller may contain a request that needs to access data from a *different* row within the same bank. To prepare the bank to service this request, the memory controller issues a `PRECHARGE` command to the bank, which closes the currently-activated row and resets the bank in preparation for the next `ACTIVATE` command. Because closing the activated row and resetting the bank takes some time, the standard specifies the precharge latency as the minimum amount of time the controller must wait for after issuing `PRECHARGE` before it issues an `ACTIVATE`. The timing parameter for precharge is called tRP, as shown in Figure 2, and is typically set to 13ns in DDR3L [92].

2.3 Effect of DRAM Voltage and Frequency on Power Consumption

DRAM power is divided into dynamic and static power. Dynamic power is the power consumed by executing the access commands: ACTIVATE, PRECHARGE, and READ/WRITE. Each ACTIVATE and PRECHARGE consumes power in the DRAM array and the peripheral circuitry due to the activity in the DRAM array and control logic. Each READ/WRITE consumes power in the DRAM array by accessing data in the row buffer, and in the peripheral circuitry by driving data on the channel. On the other hand, static power is the power that is consumed *regardless* of the DRAM accesses, and it is mainly due to transistor leakage. DRAM power is governed by both the supply voltage and operating clock frequency: $Power \propto Voltage^2 \times Frequency$ [32]. As shown in this equation, power consumption scales quadratically with supply voltage, and linearly with frequency.

DRAM supply voltage is distributed to both the DRAM array and the peripheral circuitry through respective power pins on the DRAM chip, dedicated separately to the DRAM array and the peripheral circuitry. We call the voltage supplied to the DRAM array, V_{array}, and the voltage supplied to the peripheral circuitry, V_{peri}. Each DRAM standard requires a specific nominal supply voltage value, which depends on many factors, such as the architectural design and process technology. In this work, we focus on the widely used DDR3L DRAM design that requires a nominal supply voltage of 1.35V [51]. To remain operational when the supply voltage is unstable, DRAM can tolerate a small amount of deviation from the nominal supply voltage. In particular, DDR3L DRAM is specified to operate with a supply voltage ranging from 1.283V to 1.45V [92].

The DRAM channel frequency value of a DDR DRAM chip is typically specified using the *channel data rate*, measured in mega-transfers per second (MT/s). The size of each data transfer is dependent on the width of the data bus, which ranges from 4 to 16 bits for a DDR3L chip [92]. Since a modern DDR channel transfers data on both the positive and the negative clock edges (hence the term *double data rate*, or DDR), the channel frequency is *half of the data rate*. For example, a DDR data rate of 1600 MT/s means that the frequency is 800 MHz. To run the channel at a specified data rate, the peripheral circuitry requires a certain minimum voltage (V_{peri}) for stable operation. As a result, the supply voltage scales directly (i.e., linearly) with DRAM frequency, and it determines the maximum operating frequency [32, 35].

2.4 Memory Voltage and Frequency Scaling

One proposed approach to reducing memory energy consumption is to scale the voltage and/or the frequency of DRAM based on the observed memory channel utilization. We briefly describe two different ways of scaling frequency and/or voltage below.

Frequency Scaling. To enable the power reduction that comes with reduced DRAM frequency, prior works propose to apply *dynamic frequency scaling* (DFS) by adjusting the DRAM channel frequency based on the memory bandwidth demand from the DRAM channel [14, 33–35, 107, 126]. A major consequence of lowering the frequency is the likely performance loss that occurs, as it takes a longer time to transfer data across the DRAM channel while operating at a lower frequency. The clocking logic within the peripheral circuitry requires a *fixed number of DRAM cycles* to transfer the data, since DRAM sends data on each edge of the clock cycle. For a 64-bit memory channel with a 64B cache line size, the transfer typically takes four DRAM cycles [50]. Since lowering the frequency increases the time required for each cycle, the total amount of time spent on data transfer, in nanoseconds, increases accordingly. As a result, not only does memory latency increase, but also memory data throughput decreases, making DFS undesirable to use when the running workload's memory bandwidth demand or memory latency sensitivity is high. The extra transfer latency from DRAM can also cause longer queuing times for requests waiting at the memory controller [48, 60, 61, 70, 124, 125], further exacerbating the performance loss and potentially delaying latency-critical applications [32, 35].

Voltage and Frequency Scaling. While decreasing the channel frequency reduces the peripheral circuitry power and static power, it does *not* affect the dynamic power consumed by the operations performed on the DRAM array (i.e., activation, restoration, precharge). This is because DRAM array operations are asynchronous, i.e., independent of the channel frequency [91]. As a result, these operations require a fixed time (in nanoseconds) to complete. For example, the activation latency in a DDR3L DRAM module is 13ns, regardless of the DRAM frequency [92]. If the channel frequency is doubled from 1066 MT/s to 2133 MT/s, the memory controller doubles the number of cycles for the ACTIVATE timing parameter (i.e., tRCD) (from 7 cycles to 14 cycles), to maintain the 13ns latency.

In order to reduce the dynamic power consumption of the DRAM array as well, prior work proposes *dynamic voltage and frequency scaling* (DVFS) for DRAM, which reduces the supply voltage along with the channel frequency [32]. This mechanism selects a DRAM frequency based on the current memory bandwidth utilization and finds the *minimum operating voltage* (V_{min}) for that frequency. V_{min} is defined to be the lowest voltage that still provides "stable operation" for DRAM (i.e., no errors occur within the data). There are two significant limitations for this proposed DRAM DVFS mechanism. The first limitation is due to a lack of understanding of how voltage scaling affects the DRAM behavior. No prior work provides experimental characterization or analysis of the effect of reducing the DRAM supply voltage on latency, reliability, and data retention in real DRAM chips. As the DRAM behavior under reduced-voltage operation is unknown to satisfactorily maintain the latency and reliability of DRAM, the proposed DVFS mechanism [32] can reduce supply voltage only *very conservatively*. The second limitation is that this prior work reduces the supply voltage only when it reduces the channel frequency, since a lower channel frequency requires a lower supply voltage for stable operation. As a result, DRAM DVFS results in the same performance issues experienced by the DRAM DFS mechanisms. In Section 6.3, we evaluate the main prior work [32] on memory DVFS to quantitatively demonstrate its benefits and limitations.

2.5 Our Goal

The goal of this work is to *(i)* experimentally characterize and analyze *real modern DRAM chips* operating at different supply voltage levels, in order to develop a solid and thorough understanding of how reduced-voltage operation affects latency, reliability, and data retention in DRAM; and *(ii)* develop a mechanism that can reduce DRAM energy consumption by reducing DRAM voltage, without having to sacrifice memory data throughput, based on the insights obtained from comprehensive experimental characterization. Understanding how DRAM characteristics change at different voltage levels is imperative not only for enabling memory DVFS in real systems, but also for developing other low-power and low-energy DRAM designs that can effectively reduce the DRAM voltage. We experimentally analyze the effect of reducing supply voltage of modern DRAM chips in Section 4, and introduce our proposed new mechanism for reducing DRAM energy in Section 5.

3 EXPERIMENTAL METHODOLOGY

To study the behavior of real DRAM chips under reduced voltage, we build an FPGA-based infrastructure based on SoftMC [43], which allows us to have precise control over the DRAM modules. This method was used in many previous works [20, 21, 43, 53, 54, 57–59, 64, 65, 72, 73, 75, 83, 89, 108] as an effective way to explore different DRAM characteristics (e.g., latency, reliability, and data retention time) that have not been known or exposed to the public by DRAM manufacturers. Our testing platform consists of a Xilinx ML605 FPGA board and a host PC that communicates with the FPGA via a PCIe bus (Figure 3). We adjust the supply voltage to the DRAM by using a USB interface adapter [127] that enables us to tune the power rail connected to the DRAM module directly. The power rail is connected to all the power pins of every chip on the module (as shown in Appendix A).

Figure 3: FPGA-based DRAM testing platform.

Characterized DRAM Modules. In total, we tested 31 DRAM DIMMs, comprising of 124 DDR3L (low-voltage) chips, from the three major DRAM chip vendors that hold more than 90% of the DRAM market share [13]. Each chip has a 4Gb density. Thus, each of our DIMMs has a 2GB capacity. The DIMMs support up to a 1600 MT/s channel frequency. Due to our FPGA's maximum operating frequency limitations, all of our tests are conducted at 800 MT/s. Note that the experiments we perform do *not* require us to adjust the channel frequency. Table 1 describes the relevant information about the tested DIMMs. Appendix E provides detailed information on each DIMM. Unless otherwise specified, we test our DIMMs at an ambient temperature of 20±1℃. We examine the effects of high ambient temperature (i.e., 70±1℃) in Section 4.5.

DRAM Tests. At a high level, we develop a test (Test 1) that writes/reads data to/from *every* row in the *entire* DIMM, for a given

Vendor	Total Number of Chips	Timing (ns) (tRCD/tRP/tRAS)	Assembly Year
A (10 DIMMs)	40	13.75/13.75/35	2015-16
B (12 DIMMs)	48	13.75/13.75/35	2014-15
C (9 DIMMs)	36	13.75/13.75/35	2015

Table 1: Main properties of the tested DIMMs.

supply voltage. The test takes in several different input parameters: activation latency (tRCD), precharge latency (tRP), and data pattern. The goal of the test is to examine if any errors occur under the given supply voltage with the different input parameters.

Test 1 Test DIMM with specified tRCD/tRP and data pattern.

```
VoltageTest(DIMM, tRCD, tRP, data, data̅)
  for bank ← 1 to DIMM.Bank_MAX
    for row ← 1 to bank.Row_MAX          ▷ Walk through every row within the current bank
      WriteOneRow(bank, row, data)       ▷ Write the data pattern into the current row
      WriteOneRow(bank, row + 1, data̅)   ▷ Write the inverted data pattern into the next row
      ReadOneRow(tRCD, tRP, bank, row)   ▷ Read the current row
      ReadOneRow(tRCD, tRP, bank, row + 1) ▷ Read the next row
      RecordErrors()                     ▷ Count errors in both rows
```

In the test, we iteratively test two consecutive rows at a time. The two rows hold data that are the inverse of each other (i.e., *data* and $\overline{data}$). Reducing tRP lowers the amount of time the precharge unit has to reset the bitline voltage from either *full voltage* (bit value 1) or *zero voltage* (bit value 0) to *half voltage*. If tRP were reduced too much, the bitlines would float at some other intermediate voltage value between *half voltage* and *full/zero voltage*. As a result, the next activation can potentially start before the bitlines are fully precharged. If we were to use the same data pattern in both rows, the sense amplifier would require *less* time to sense the value during the next activation, as the bitline is already *biased* toward those values. By using the *inverse* of the data pattern in the row that is precharged for the next row that is activated, we ensure that the partially-precharged state of the bitlines does *not* unfairly favor the access to the next row [21]. In total, we use three different groups of data patterns for our test: (0x00, 0xff), (0xaa, 0x33), and (0xcc, 0x55). Each specifies the *data* and $\overline{data}$, placed in consecutive rows in the same bank.

4 CHARACTERIZATION OF DRAM UNDER REDUCED VOLTAGE

In this section, we present our major observations from our detailed experimental characterization of 31 commodity DIMMs (124 chips) from three vendors, when the DIMMs operate under reduced supply voltage (i.e., below the nominal voltage level specified by the DRAM standard). First, we analyze the reliability of DRAM chips as we reduce the supply voltage without changing the DRAM access latency (Section 4.1). Our experiments are designed to identify if lowering the supply voltage induces bit errors (i.e., *bit flips*) in data.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | sales@docketalarm.com | 1-866-77-FASTCASE

fastcase®
Smarter legal research.