An Experimental Study of Data Retention Behavior in Modern DRAM Devices: Implications for Retention Time Profiling Mechanisms

Jamie Liu*
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA 15213
jamiel@alumni.cmu.edu

Ben Jaiyen*
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA 15213
bjaiyen@alumni.cmu.edu

Yoongu Kim
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA 15213
yoonguk@ece.cmu.edu

Chris Wilkerson
Intel Corporation
2200 Mission College Blvd.
Santa Clara, CA 95054
chris.wilkerson@intel.com

Onur Mutlu
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA 15213
onur@cmu.edu

ABSTRACT

DRAM cells store data in the form of charge on a capacitor. This charge leaks off over time, eventually causing data to be lost. To prevent this data loss from occurring, DRAM cells must be periodically refreshed. Unfortunately, DRAM refresh operations waste energy and also degrade system performance by interfering with memory requests. These problems are expected to worsen as DRAM density increases.

The amount of time that a DRAM cell can safely retain data without being refreshed is called the cell's *retention time*. In current systems, all DRAM cells are refreshed at the rate required to guarantee the integrity of the cell with the shortest retention time, resulting in unnecessary refreshes for cells with longer retention times. Prior work has proposed to reduce unnecessary refreshes by exploiting differences in retention time among DRAM cells; however, such mechanisms require knowledge of each cell's retention time.

In this paper, we present a comprehensive quantitative study of retention behavior in modern DRAMs. Using a temperature-controlled FPGA-based testing platform, we collect retention time information from 248 commodity DDR3 DRAM chips from five major DRAM vendors. We observe two significant phenomena: *data pattern dependence*, where the retention time of each DRAM cell is significantly affected by the data stored in other DRAM cells, and *variable retention time*, where the retention time of some DRAM cells changes unpredictably over time. We discuss possible physical explanations for these phenomena, how their magnitude may be affected by DRAM technology scaling, and their ramifications for DRAM retention time profiling mechanisms.

*Current affiliation: Google Inc., 1600 Amphitheatre Pkwy., Mountain View, CA 94043.

ISCA '13 Tel-Aviv, Israel

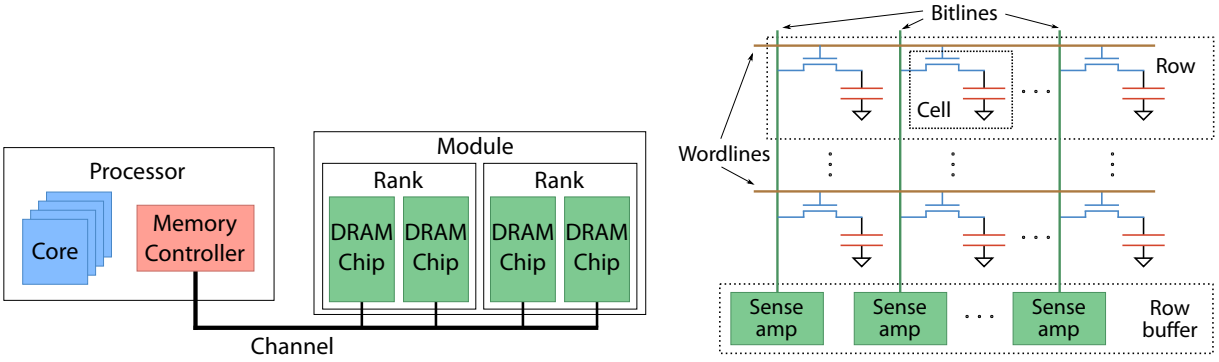
1. INTRODUCTION

Modern main memory systems are typically composed of dynamic random-access memory (DRAM). A DRAM cell stores one bit of information by storing charge on a capacitor. This charge leaks over time, causing data to be lost. In order to maintain data integrity, this lost charge must be restored periodically, a process referred to as *DRAM refresh*. These refresh operations cause both performance degradation and increased energy consumption, both of which are expected to worsen as DRAM density increases.

DRAM cells leak charge at different rates due to process variation. As a result, each DRAM cell's *retention time* — the time a cell can go without being refreshed while still storing enough charge to be able to be read correctly — is different. Taking advantage of this fact, prior work has aimed to reduce the effect of DRAM refresh on performance and energy by keeping track of *weak* cells with lower retention times [1, 17, 23, 28, 35, 38]. This is made possible by the fact that there are relatively few weak cells in a DRAM system. By knowing which cells are weak, the mechanisms proposed by prior work can lower the refresh rate for all other cells in the DRAM, which results in both performance and energy benefits over the existing mechanism in use today that refreshes all cells every 64 ms.

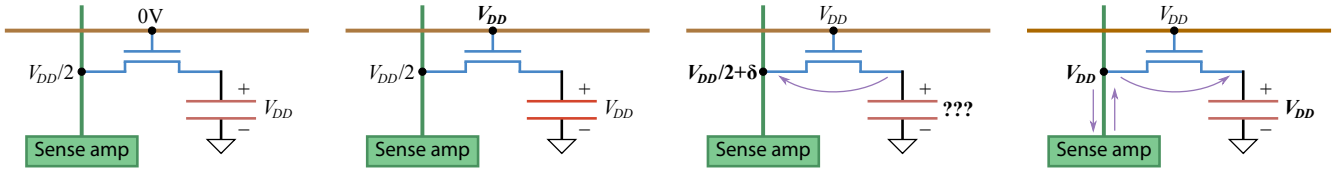
Prior work assumes that the retention time of DRAM cells can be easily measured or *profiled*, and the ability of such prior mechanisms to guarantee data integrity relies on having an accurate profile. In this work, we identify some difficulties in DRAM retention time profiling that challenge the assumptions made by prior work:

- Most prior work has assumed that writing simple data patterns such as "all 1s" and "all 0s" is sufficient to find all DRAM cells with a given retention time. However, the retention time of each DRAM cell is strongly affected by the value stored both in that cell and in nearby cells due to circuit-level crosstalk effects [21, 26]. We find that, in some devices, testing with all 1s and all 0s identifies less than 15% of all weak cells. Furthermore, the precise effect of this *data pattern dependence* effect varies between devices due to variation in DRAM array circuit design between manufacturers and DRAM processes.
- Prior work has identified a phenomenon called *variable retention time* (VRT) in which DRAM cells shift randomly between *multiple* retention time states [31, 39]. This complicates retention time



(a) Memory system hierarchy (b) DRAM bank structure

Figure 1: DRAM system organization



(a) Precharged state (b) Wordline raised (c) Charge sharing (d) Sensing

Figure 2: DRAM row activation (arrows indicate flow of charge)

profiling because VRT can cause a cell's retention time to drop significantly below its measured value; even a safety margin of 2x may not suffice to guarantee correctness in the presence of VRT. We demonstrate that VRT is ubiquitous in modern DRAMs, and that the timescale on which VRT occurs poses significant design challenges to retention time profiling mechanisms.

In this paper, our goal is to empirically show that both data pattern dependence and variable retention time are important issues that must be addressed in order to ensure retention time profile accuracy. To this end, we collected retention time information from 248 commodity DDR3 DRAM chips from five major vendors and demonstrate the problems that arise from each phenomenon. Our key contributions are as follows:

- We present quantitative data on the retention time behavior of DRAM cells in modern DRAM devices, including devices from a variety of manufacturers and generations.
- We quantitatively evaluate the impact of two significant factors, data pattern dependence and variable retention time, on the retention time of modern DRAM cells.
- We discuss the effects of data pattern dependence and variable retention time on future DRAM scaling and retention time measurement mechanisms, based both on our results and on an understanding of the underlying physical phenomena.

2. BACKGROUND AND MOTIVATION

2.1 DRAM Organization

A modern main memory DRAM system is organized hierarchically into channels, modules, and ranks, as shown in Figure 1a. Each *channel* contains an independent set of command, address, and data buses, and contains one or more DRAM modules. Each *module* contains a number of DRAM chips that are grouped into one or more ranks. A *rank* is a set of DRAM chips ganged together, responding to commands in parallel. In a typical configuration, a 64-bit wide rank may consist of 8 DRAM chips each 8 bits wide, such that a 64-bit transfer to or from the rank transfers 8 bits to or from each chip in parallel.

Within each chip, DRAM cells are organized into two-dimensional arrays called *banks*, as shown in Figure 1b. A DRAM cell consists of a capacitor and an access transistor. Each access transistor connects its associated capacitor to a wire called a *bitline* and is controlled by a wire called a *wordline*. Cells sharing a wordline form a *row*. Each bank also contains a row of sense amplifiers, where each sense amplifier is connected to a single bitline.[1] This row of sense amplifiers is called the bank's *row buffer*.

In order to access the data stored in the cell array, the row containing the data must first be *activated* to place the data on the bitlines. The process of activating a row is shown in Figure 2. Before a row can be activated, all bitlines in the bank must first be *precharged* to $V_{DD}/2$ (Figure 2a). The row's wordline is enabled (driven to V_{DD}), connecting all capacitors in that row to their respective bitlines (Figure 2b). This causes charge to flow from the capacitor to the bitline (if the capacitor is charged to V_{DD}) or vice versa (if the capacitor is at 0 V), in what is known as *charge sharing* (Figure 2c). Finally, in the sensing stage, the sense amplifier connected to that bitline detects the voltage change and amplifies it, driving the bitline fully to either V_{DD} or 0 V (Figure 2d). Cells in the activated row can then be read or written by sensing or driving the voltage on the appropriate bitlines. To activate another row, the bitlines must be returned to the precharged voltage of $V_{DD}/2$.

[1]This is a slight simplification that will be expanded on in Section 3.1.

2.2 DRAM Retention and Refresh

DRAM cells lose data because capacitors leak charge over time. Since the parasitic capacitance of a DRAM bitline is very large compared to the capacitance of a DRAM cell [13], the voltage change that occurs during charge sharing (referred to as the voltage *perturbation*) is small. As the amount of charge on the DRAM cell capacitor decreases, the resulting perturbation decreases until the sense amplifier is unable to reliably distinguish the perturbation from noise. At this point, the data stored on the DRAM cell has been lost. The amount of time that a DRAM cell can store data before data loss occurs is called the cell's *retention time*. Typically, all DRAM cells are required to have a retention time greater than 64 ms [13]; devices that cannot meet this requirement are discarded.

In order to preserve data integrity, the charge on each capacitor must be periodically restored or *refreshed*. Recall that when a row is activated, sense amplifiers drive each bitline fully to V_{DD} or 0 V. Since each bitline is still connected to a cell capacitor at this time, this causes the activated row's cell capacitors to be fully charged to V_{DD} or 0 V as well. Hence, a row is refreshed by activating it.

In current DRAM interfaces, the memory controller issues refresh commands to each DRAM rank at periodic time intervals. When a DRAM chip receives a refresh command, it refreshes a small number of its least-recently refreshed rows. For DDR3 DRAMs operating below 85 °C, the average time between refresh commands (called t_{REFI}) is 7.8 µs [10]. Since DRAM cells are required to have a retention time of at least 64 ms in this temperature range, and refresh logic in existing DRAM chips refreshes each row at the same rate for simplicity, all rows must be refreshed every 64 ms for correctness. This implies that each row is refreshed by every 8192$^{\text{nd}}$ refresh command, since $8192 \times 7.8\ \mu s \approx 64$ ms (the equality is inexact since the DDR3 specification allows some refresh commands to be issued early or late [10], increasing the maximum time between refreshes that a row may observe).

2.3 DRAM Retention Time Profiling

The retention time of a DRAM cell depends on the leakage current for that cell's capacitor and access transistor, which differs between cells due to manufacturing variation. To mitigate the substantial energy and performance overheads incurred by DRAM refresh [23], prior work [1, 17, 23, 28, 35, 38] has proposed mechanisms to refresh DRAM cells with longer retention times less frequently, decreasing the number of unnecessary refresh operations that are performed. These mechanisms measure and store the retention time of the DRAM cells in the system, then use these measured (or *profiled*) retention times to adjust the refresh rate, selectively or for the entire DRAM system. These prior works depend on the ability to accurately and reliably profile DRAM retention times. Once a profile has been created for a DRAM device, it is assumed that the profile stays the same, and refresh decisions are made assuming that the retention times in the profile are conservative enough to ensure data integrity.

However, the retention time of a DRAM cell can also be affected by factors that are *not* fixed at any single point in time, violating such an assumption. Two of these factors in particular have not been adequately explored by prior work:

- **Data pattern dependence.** A retention failure occurs when too much charge has leaked away from a DRAM cell capacitor, such that the voltage perturbation on the bitline that occurs during charge sharing can no longer be sensed reliably, as discussed in Section 2.2. However, the overall voltage change that occurs on the bitline is also affected by noise. When a row is activated, all bitlines in the bank are perturbed simultaneously, as described in Section 2.1. As a result, the following two major sources of bitline noise are *data-dependent* (that is, the magnitude of the noise produced depends on stored data):

  1. *Bitline-bitline coupling.* Electrical coupling between adjacent bitlines creates noise on each bitline that depends on the voltages of nearby bitlines, such that the noise experienced by each bitline is affected by the values stored in nearby cells [26, 33].
  2. *Bitline-wordline coupling.* Electrical coupling between each bitline and wordline creates data-dependent noise on each wordline, which in turn creates data-dependent noise in each coupled bitline. Hence, the noise experienced by each bitline is affected by the values stored in every other cell in the row [20, 33].

  These effects cause the retention time of each DRAM cell to depend on the data pattern stored in other cells. We refer to this phenomenon as *data pattern dependence (DPD)*.

- **Variable retention time.** Prior work on decreasing refresh overhead has assumed that each DRAM cell's leakage current is stable over time. However, many DRAM cells transition between multiple leakage current states, and consequently multiple retention time states, in a phenomenon referred to as *variable retention time (VRT)* [31, 39]. VRT is generally agreed to be caused by the presence of a charge trap in the gate oxide[2] of a DRAM cell's access transistor [27]. When this trap becomes occupied, charge leaks more readily from the access transistor's drain (to which the cell's capacitor is connected). This process is therefore referred to as *trap-assisted gate-induced drain leakage (TA-GIDL)* [3].

Our goal in this paper is to present a comprehensive investigation of these two phenomena. To this end, we implement an FPGA-based platform for studying the retention time of DDR3 DRAM modules and use it to collect quantitative data on the retention time of cells in a wide variety of modern DRAM chips.

3. METHODOLOGY

We developed a DDR3 DRAM testing platform based on the Xilinx ML605 FPGA development board [37], controlled by a commodity PC over a PCI Express channel.

We obtained a variety of commercial DDR3 chips in the ubiquitous small outline dual inline memory module (SO-DIMM) form factor from a variety of DRAM and DIMM manufacturers, shown in Table 1. We report capacity per rank since the FPGA board we used allows access to only one rank [36]. Manufacturer names have been anonymized.

We additionally categorize the tested DRAM chips into families. Part IDs ② and ③ used DRAM chips that were of the same generation, speed bin, etc. and differed only in per-chip I/O width, while part IDs ④ and ⑤ used identical DRAM chips. Since each of these pairs of DRAM chips did not show any distinguishable differences in behavior, we consider DRAMs from both ② and ③ as part of the `A 2Gb` chip family, and DRAMs from both ④ and ⑤ as the `B 2Gb` chip family.

To control the testing temperature, the testing platform was placed within an insulated enclosure equipped with fans for cooling and a radiative heat source for heating. Both the fans and the heater were controlled by a commercial closed-loop thermal controller that measured the DRAM module temperature via an attached thermocouple.

[2]The electrical insulator between the transistor's gate and its source, drain, and channel.

Table 1: Tested DRAM devices and modules

Part ID	# DIMMs	DRAM Manufacturer	DIMM Capacity/Rank	Ranks/DIMM	DRAM Capacity/Device	DRAM Devices/Rank	DRAM Family
①	4	A	512 MB	1	1 Gb	4	A 1Gb
②	2	A	1 GB	1	2 Gb	4	A 2Gb
③	2	A	2 GB	1	2 Gb	8	A 2Gb
④	10	B	2 GB	1	2 Gb	8	B 2Gb
⑤	8	B	2 GB	2	2 Gb	8	B 2Gb
⑥	2	C	2 GB	1	2 Gb	8	C 2Gb
⑦	2	D	1 GB	2	1 Gb	8	D 1Gb
⑧	2	D	2 GB	2	2 Gb	8	D 2Gb
⑨	2	E	2 GB	2	2 Gb	8	E 2Gb

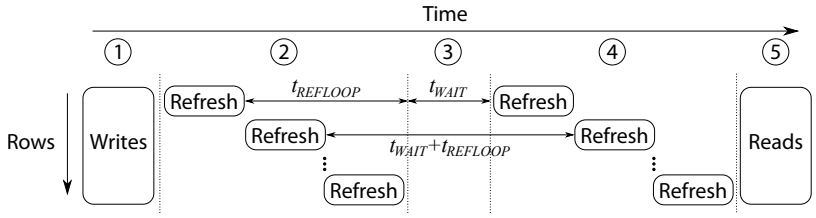


Figure 3: Test structure

3.1 Experiment Design

In our testing protocol, each module was used in one or more *experiments* consisting of many *tests*. Each test searches for the set of cells with a retention time less than a threshold value. Let $t_{REFLOOP} = 8192 \cdot 7.8\ \mu s = 63.8976\ \mu s$ be the *refresh loop time*. This is the time between refreshes to each row under normal refresh timing, as discussed in Section 2.2. Each test then consists of the following steps, as shown in Figure 3:

1. Data is written to the DRAM. Refreshes are initially enabled, to ensure that no data corruption occurs during the write.
2. Refreshes remain enabled for at least 64 ms while the DRAM is left idle. This ensures that the last time each row is activated is due to its periodic refresh.
3. Refreshes are disabled for a period of time t_{WAIT} while the DRAM is left idle. This time is controlled by the FPGA hardware in order to ensure that it is precisely timed.
4. Refreshes are re-enabled, and the DRAM is left idle for at least 64 ms. After this step, each row has experienced a length of time of exactly $t_{REFLOOP} + t_{WAIT}$ without a refresh.
5. Data is read back from the DRAM and checked for corruption. Any cell that has lost data is known to have a retention time less than $t_{REFLOOP} + t_{WAIT}$ during this test.

Our experiments follow the structure shown in Figure 4. Each test with a given t_{WAIT} and data pattern is immediately followed by a test with the complementary data pattern (i.e. all bits inverted) and the same t_{WAIT}, forming a *test pair*. An experiment that studies multiple data patterns runs test pairs for each data pattern at the same t_{WAIT} before moving on to the next t_{WAIT}. Finally, after a *round* of all t_{WAIT} values being tested has been completed for all test pairs, the experiment continues to the next round. The rationale behind this ordering of tests is as follows:

- It is necessary to test both a data pattern and its complement. Even though cell capacitors will always leak toward ground (charge does not leak *onto* the capacitor), some DRAM cells may store a value of 1 as V_{DD} and 0 as 0 V (*true cells*) while others store a value of 0 as V_{DD} and 1 as 0 V (*anti-cells*). This is because sense amplifiers (discussed in Section 2.1) are *differential*; to detect whether charge sharing has perturbed the bitline voltage away from $V_{DD}/2$ toward V_{DD} or 0 V, the sense amplifier must be connected to another wire initially holding a reference voltage of $V_{DD}/2$, and the sense amplifier amplifies the difference between the two voltages. To implement this cost-efficiently, each sense amplifier is actually connected to *two* bitlines, as shown in Figure 5. When either bitline is being activated, the other bitline, which must be in the precharged state, initially holds the reference (precharge) voltage of $V_{DD}/2$. During sensing, *both* bitlines are driven toward their final (complementary) values. Furthermore, as long as the sense amplifier is enabled, it maintains this complementary relationship between the two bitlines [13]. This means that of the two bitlines connected to each sense amplifier, only one requires an external driver and sensor; cells on the complementary bitline can be written and read by driving or sensing the complementary voltage on the connected bitline. Hence, cells on the complementary bitline have an inverted relationship between voltage and logic value, i.e. they are anti-cells, as shown in Figure 5b.

- Within a round, all tests at a given t_{WAIT} run together. This helps to decouple the apparent effects of data pattern dependence and variable retention time. Suppose a cell's retention time state changes due to VRT at one point during a round.[3] With all tests at a given t_{WAIT} running together, in the worst case, the retention time changes between data patterns, resulting in an anomalous result for a single t_{WAIT}. However, since the retention time state will still be consistent across data patterns for both the *previous* and *next* values of t_{WAIT}, the change in observed retention time between data patterns should still be minimal (in the absence of DPD). In contrast, if tests with the same t_{WAIT} were not run together, it could be difficult to identify whether a

[3] We show in Section 6.2 that VRT causes retention time states to change on sufficiently long timescales that each cell does not usually change states more than once per round.

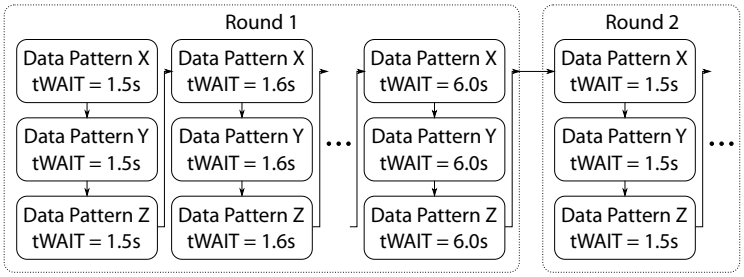


Figure 4: Experiment structure

(a) Activated true cell

(b) Activated anti-cell

Figure 5: True and anti-cells

drastic difference in the retention time is the result of DPD or VRT.

- Breaking experiments into repeated rounds allows the time between consecutive rounds of any given test to be approximately the same. That is, the time between rounds 1 and 2 of the test with data pattern X and $t_{WAIT} = \alpha$ is approximately the same as the time between rounds 1 and 2 of the test with data pattern Y and $t_{WAIT} = \beta$. This again helps to decouple DPD and VRT; as per the previous point, a single change in the retention time state of a cell in one round does not affect previous or following rounds.

Unless otherwise specified, all of our experiments were conducted at an ambient temperature of 45 °C. At this temperature, no cells were observed to have a retention time less than 1.5 second. As such, our experiments tested values of t_{WAIT} between $23t_{REFLOOP}$ and $95t_{REFLOOP}$, corresponding to retention times between $24t_{REFLOOP} \doteq 1.5$ s and $96t_{REFLOOP} \doteq 6.1$ s, in increments of $2t_{REFLOOP} \doteq 128$ ms. This gives us the ability to measure retention time differences of up to a factor of 4. (Higher maximum t_{WAIT} and smaller time increments are possible, but result in increasingly prohibitive experiment lengths.)

3.2 Data Patterns Tested

Most of our experiments examine four types of data patterns:

1. *All 0s/1s.* In this data pattern, the value 1 is written to all bits. (Since each test pair also tests the complement of the data pattern, as discussed in Section 3.1, this tests both 'all 1s' and 'all 0s'.) Prior work, e.g. [23, 35], has suggested that this intuitively simple data pattern is sufficient for measuring DRAM retention time.

2. *Checkerboard.* In this data pattern, consecutive bits alternate between 0 and 1. In DRAM circuit architectures where adjacent bits are mapped to adjacent cells, checkerboard patterns may induce worse retention time behavior than patterns of all 1s or 0s [21]. Intuitively, this is because bitline coupling noise increases with the voltage difference between the coupled bitlines, and storing alternating values in consecutive bitlines maximizes the voltage difference between each bitline and its immediate neighbors.

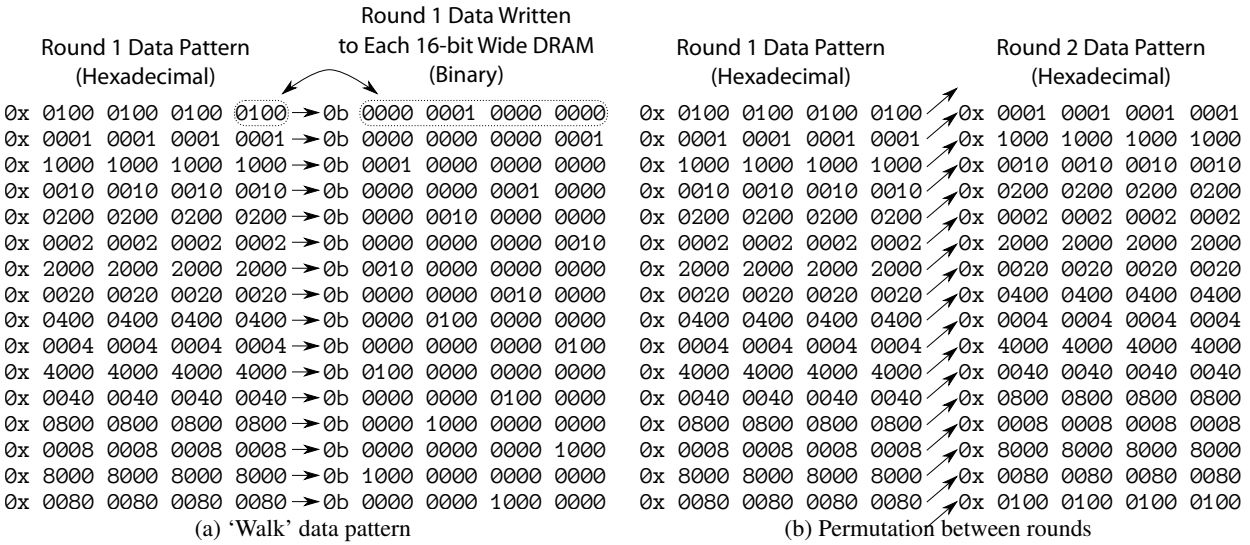
3. *Walk.* The checkerboard pattern attempts to maximize bitline coupling noise with respect to the two immediately neighboring bitlines. However, coupling with other nearby bitlines can also exist. Hence, even worse retention time behavior may be triggered when a single cell storing V_{DD} is surrounded by cells storing 0 V [21]. This data pattern attempts to efficiently approximate this scenario.[4] The 'walk' pattern consists of a 128-byte data pattern, shown in Figure 6a, chosen according to the following rationale:

   In our DDR3 memory system, each transfer to or from the DRAM is 64 bits in size. Of each of these 64-bit words, each consecutive 4, 8, or 16 bits goes to the same DRAM chip, depending on whether the 64-bit rank is made up of 16 DRAM chips each 4 bits wide (16×4), 8 DRAM chips each 8 bits wide (8×8), or 4 DRAM chips each 16 bits wide (4×16). The 'walk' pattern attempts to maximize the number of 0s between each 1 as seen by each chip, regardless of the rank configuration. Logically, the 'walk' pattern can be considered to consist of 16 16-bit patterns, where each pattern sets a single bit to 1, replicated 4 times across chips.

   In order to ensure that our experiments test with a 1 in each bit position, the 'walk' pattern is permuted by rotating it by 64 bits (1 word) after each round, effectively offsetting the data pattern by 1 word, as shown in Figure 6b. Hence, after 16 rounds, each bit has been tested with an isolated 1. This means the 'walk' data pattern repeats with a period of 16 rounds; as a result, the number of rounds performed by all of our experiments is a multiple of 16.

4. *Random.* The DRAM is filled with randomly generated data. (The complement of this data is still written by the other test in the test pair.) A new set of random data is generated for

[4] It is difficult to produce this scenario exactly for reasons discussed in Section 5.2.



(a) 'Walk' data pattern (b) Permutation between rounds

Figure 6: Design of the 'walk' data pattern

each round. Testing random data patterns allows us to efficiently explore the potential impact of other data patterns.

We refer to the first two of these patterns as *fixed* patterns (since they do not change from round to round) and the first three as *static* patterns (since their value in any given round is known).

4. FOUNDATIONAL RESULTS

In this section, we present results on temperature dependence and the distribution of retention time between DRAM cells. We then compare these results to prior work in order to confirm the validity of our methodology.

4.1 Temperature Dependence

Prior work has demonstrated that the retention time of DRAM cells decreases exponentially as temperature increases [6]. In this section, we confirm these results and establish quantitative bounds on the effect of temperature in order to disqualify temperature as a confounding factor in our later results.

DDR3 DRAM chips do not have user-accessible temperature sensors [10]. DDR3 DIMMs may include a temperature sensor, either as a separate chip or embedded in the DIMM's serial presence detect (SPD) chip, but this is an optional feature [9]. Of the DIMMs we tested, only the DIMMs with part ID ① included a temperature sensor. Therefore, the results in this section use only part ID ① DIMMs. (Note that this limitation does not affect the future viability of DRAM retention time profiling mechanisms in general, since the current-generation DDR4 [11] and LP-DDR3 [12] specifications require DRAM chips to include user-accessible temperature sensors.)

We first confirm that the temperature remains stable during testing. Figure 7a shows the measured temperature over time for each of the five temperatures we test in these experiments.[5] Temperature never deviates by more than 1 °C from its steady state value.

Next, we evaluate the effect of temperature on each cell's retention time. The 'all 0s/1s', 'checkerboard', and 'walk' patterns were each run for 16 rounds over the course of approximately 7 hours at each temperature. For each cell that exhibited a failure at the lowest temperature (50 °C) at any t_{WAIT}, we evaluated the retention time for that cell at each temperature, normalized to its retention time at 50 °C. (To account for the effects of DPD and VRT, we take the minimum retention time of each cell across all data patterns and rounds for each temperature.) Since the number of cells is large, instead of plotting each cell's normalized retention times as an individual point, we grouped normalized retention times into bins and plotted the density of each bin. The result is shown in Figure 7b. Best-fit exponential curves are drawn through the most populous bins ("peak") and the bins with the lowest normalized retention time ("tail").

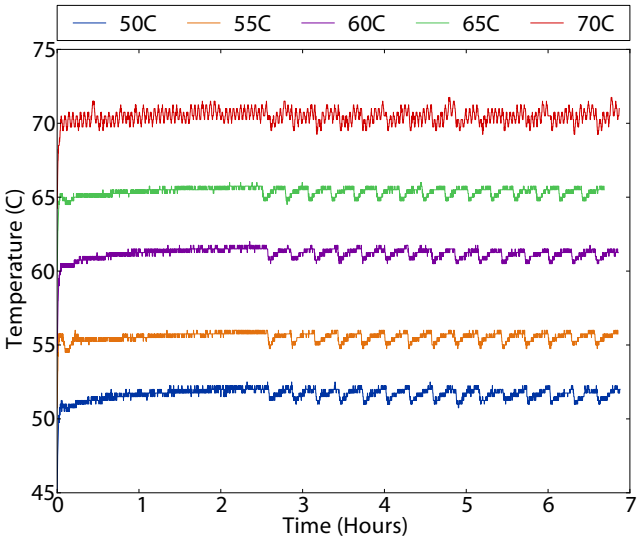
These exponential curves fit the data well, suggesting close correspondence with the exponential model proposed by prior work [6]. The best-fit "peak" curve is of the form $Ae^{-0.0498T}+C$, while the best-fit "tail" curve is of the form $Ae^{-0.0625T}+C$. This implies that every 10 °C increase in temperature results in a reduction in retention time of $1-e^{-0.498} \doteq 39.2\%$ in the "common case", and $1-e^{-0.625} \doteq 46.5\%$ in the worst case.

4.2 Retention Time Distribution

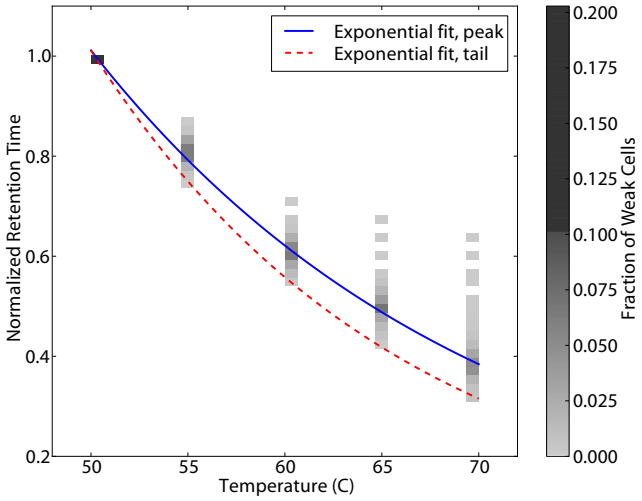
Figure 8 shows the cumulative distribution of retention times in the seven device families we tested. A point at (x,y) indicates that for the given device family, a fraction y of all DRAM cells in all tested devices of that family displayed a retention time less than x at 45 °C for some data pattern.

Retention time distributions in prior work are usually given at 85 °C. Therefore, in order to compare our measured retention time distributions to prior work, we adjust our retention times to account for the difference in temperature. Applying the worst-case formula for temperature we computed in Section 4.1, we determine that retention times should be reduced by $1-e^{-0.0625\cdot 40} \doteq 91.8\%$ to adjust for the 40 °C difference between the 45 °C temperature we tested at and the 85 °C temperature for which retention times are usually reported. Our minimum tested retention time of ≈ 1.5 s therefore translates to a retention time of ≈ 126 ms at 85 °C, indicating that all of the DRAM devices we tested were specified with a retention

[5]The heating element we used was unable to raise the DRAM temperature T_{case} to 75 °C or higher, even at maximum power and despite attempts to improve the insulation of our thermal testing chamber. We are investigating more powerful heating equipment.

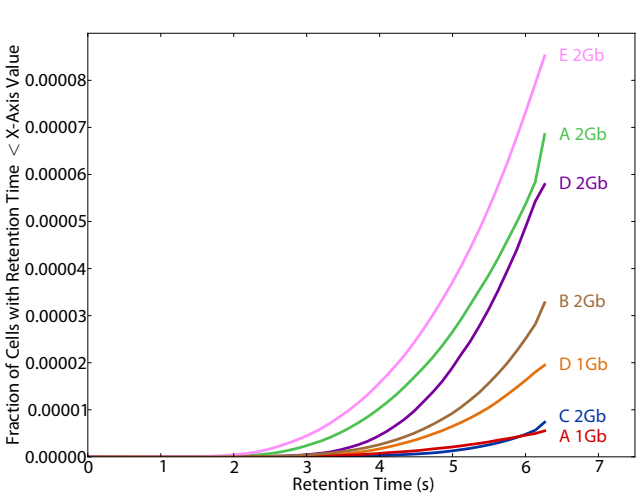


(a) Temperature stability during testing

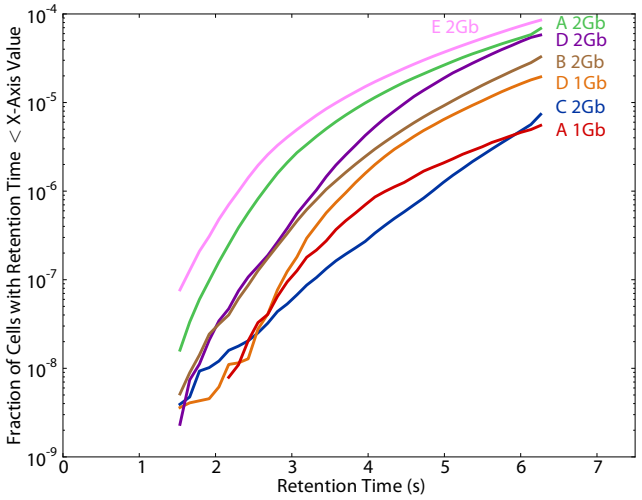


(b) Normalized retention time vs. temperature

Figure 7: Impact of temperature



(a) Linear y-axis



(b) Logarithmic y-axis

Figure 8: Cumulative distribution of retention times

time guard band[6] of at least 2x. Conversely, our maximum tested retention time of ≈ 6.1 s translates to a retention time of ≈ 504 ms at 85 °C.

With these adjusted retention times in mind, it is apparent that our measured retention time distribution corresponds closely with the distribution observed by prior work (Figure 2 in [18]), both qualitatively in the shape of the curve and — for the device families with relatively few weak cells — quantitatively. However, some device families have many more weak cells than any of the devices shown in [18]. This is likely to be a result of technology scaling. [18] presents measurements for DRAM chips fabricated using 100 nm, 60 nm, and 50 nm processes, while many of the chips we studied were produced in 2011 or 2012 and could therefore be fabricated using 36 nm or 32 nm processes [8]. This conclusion is supported by the observation that, in our results, higher-capacity, later-generation devices always have a larger number of retention failures than lower-capacity, older devices (compare `A 1Gb` vs. `A 2Gb`, and `D 1Gb` vs. `D 2Gb`, in Figure 8).

From the consistency of our foundational results with prior work, we conclude that our apparatus and methodology are sound.

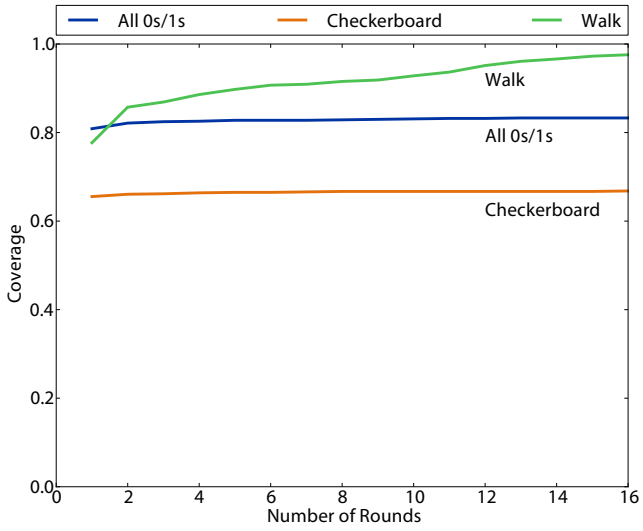
[6]The difference between the manufacturer-accepted minimum retention time and the specification-mandated minimum retention time of 64 ms.

5. DATA PATTERN DEPENDENCE

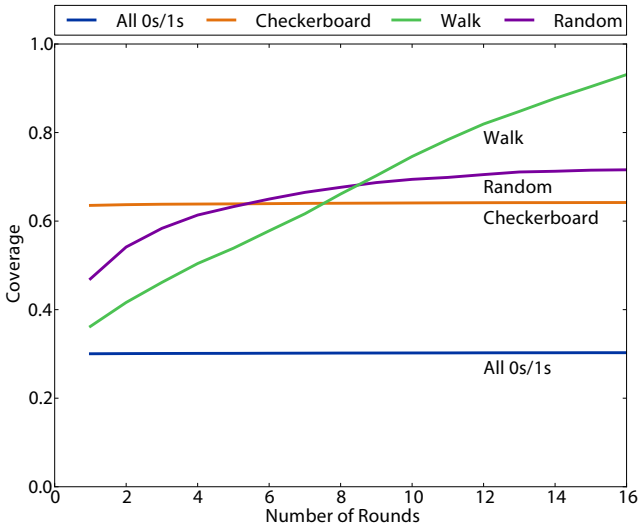
In this section, we investigate *data pattern dependence*, a phenomenon in which the retention time of DRAM cells changes depending on the data stored in other cells.

5.1 Coverage

Running all of the experiments for a given module produces a set of bit failures for each retention time, consisting of all of the

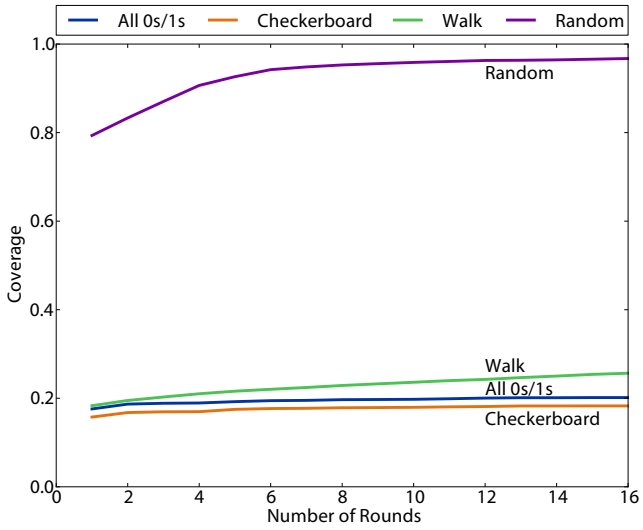


(a) A 1Gb chip family

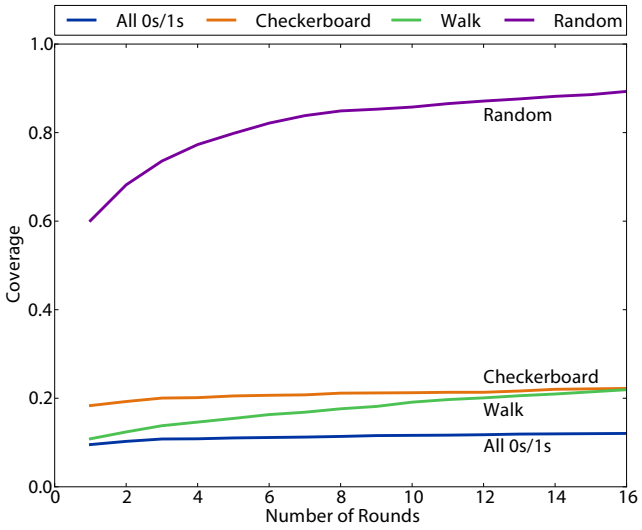


(b) A 2Gb chip family

Figure 9: Coverage in manufacturer A DRAMs



(a) B 2Gb chip family



(b) C 2Gb chip family

Figure 10: Coverage in manufacturer B and C DRAMs

cells that failed in any test for any data pattern at that retention time. We call this set the *failure population* of that retention time. We define the *coverage* of a given data pattern as the fraction of cells in the failure population that have been observed to fail in any test with that particular data pattern. In the absence of data pattern dependence, we would expect coverage to be nearly equal for all data patterns, since tests with each data pattern would observe the same set of bit failures (modulo some variation caused by VRT).

Figures 9 and 10 show the coverage of each tested data pattern for four selected DRAM chip families at the maximum tested retention time of 6.1 s, over experiments lasting 16 rounds.[7] From these results we draw several conclusions:

- *Data pattern dependence exists.* Coverage varies significantly between data patterns in each of the device families shown, indicating that the retention time of many DRAM cells depends on the data stored in other cells.

- *Data pattern dependence poses a significant obstacle to retention time profiling.* Figure 10b shows that in the worst case, a naive profiling mechanism that tested only data patterns of all 0s and 1s would miss approximately 90% of the actual bit failures that could occur in C 2Gb family DRAMs, suggesting that coupling effects are extremely strong in these devices. Furthermore, in no device does any single data pattern ever reach 100% coverage; static patterns rely on a set of assumptions about data mapping that may not be true in many devices and are vulnerable to remapping

[7] Other devices show similar trends; we omit them for space. The trends we discuss appear at all retention times, but are clearest at this retention time simply because the longest retention time has the largest number of bit failures. The random data pattern was not run on A 1Gb devices due to an error made while setting up experiments.

(both discussed in Section 5.2), while random patterns are simply statistically unlikely to reliably induce the worst-case data pattern for 100% of tested bits.

- *Repeated measurements using the same static data pattern are not effective in combating data pattern dependence.* The coverage of the 'walk' and 'random' patterns increases with number of rounds because these patterns vary from round to round, as discussed in Section 3.2. While the coverages of the fixed patterns also increase slightly with number of rounds, this is due to new bit failures appearing due to VRT, which we will discuss further in Section 6. (Since the 'walk' pattern repeats with a period of 16 rounds, further testing does not improve its coverage beyond the slight increase for all patterns due to VRT. This is why 16 rounds are shown for each experiment.)
- *The effectiveness of each data pattern varies significantly between DRAMs.* In `A 2Gb` DRAMs, the 'walk' pattern is the most effective pattern, suggesting a close correspondence between the circuit organization that the 'walk' pattern is optimized for and the actual circuit organization of `A 2Gb` devices. In both `B 2Gb` and `C 2Gb` DRAMs, none of the static patterns achieve greater than 30% coverage, suggesting that the circuit architecture of these devices is likely to be significantly different than that of `A 2Gb` devices. Furthermore, `B 2Gb` and `C 2Gb` devices are likely to be significantly different from each other, since the relative effectiveness of the static patterns is completely different between the two. This effect is not confined to differences between manufacturers; `A 1Gb` and `A 2Gb` DRAMs are both manufactured by manufacturer A, yet the relative effectiveness of the fixed data patterns switches between the two.
- *Technology scaling appears to increase the impact of data pattern dependence.* In the older `A 1Gb` DRAM, the lowest-coverage data pattern ('checkerboard') still achieves 67% coverage. In the newer `A 2Gb` DRAM, the lowest-coverage data pattern achieves approximately 30% coverage. A similar trend occurs between `D 1Gb` and `D 2Gb` DRAMs (not shown due to space constraints). This follows physically, since technology scaling reduces the physical distance between circuit elements, increasing the magnitude of coupling effects [8].

5.2 Implications for Retention Time Profiling

From the results in Section 5.1, it is apparent that any DRAM retention time profiling mechanism must handle data pattern dependence to be useful.

One intuitive approach to handling data pattern dependence could be to identify the data pattern that induces the worst-case retention time for a particular cell or device, based on information returned by additional testing or provided by the DRAM itself. However, three key issues complicate this approach:

1. *Opaque mapping of addresses to physical DRAM geometry.* Data pattern dependence is caused by circuit-level coupling effects, as discussed in Section 2.3. However, existing DRAM interfaces provide no method to communicate how bits are mapped to DRAM cells, nor where DRAM cells are located in the cell array. As a result, it is not obvious which bits actually interfere with one another.
2. *Remapping.* DRAM devices feature redundancy in order to tolerate a small number of faults in each chip [7]. Faulty elements may result in bitlines or wordlines being remapped to redundant bitlines or wordlines elsewhere in the array. This exacerbates the mapping problem by introducing irregularity into mappings.

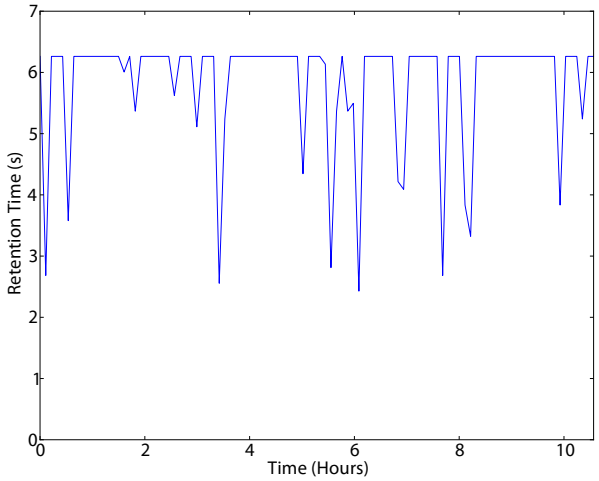


Figure 11: Retention time of a typical VRT cell

3. *Complexity of circuit organization.* Even if mappings are known, in sufficiently complex DRAM circuit organizations, the worst-case data patterns are not necessarily obvious due to second-order bitline coupling effects, where the coupling noise on one bitline is affected by coupling between other bitlines [2].

Clearly, a mechanism for identifying worst-case data patterns would need to be very sophisticated, and likely require support from the DRAM device itself. While these issues could potentially be mitigated if profiling was performed by the DRAM manufacturer rather than the end user's system, manufacturer profiling is likely to be limited by VRT for reasons discussed in Section 6.3.

Alternatively, since random data patterns are agnostic to the effects of data mapping, it may be more efficient to perform only enough random tests to establish a probability that the worst-case data pattern has been tested, and require the use of error-correcting codes (ECC) to handle a limited number of errors. The disadvantage of such a mechanism would be that it would incur the energy and capacity overheads of ECC at runtime. Future work might seek to alleviate this ECC overhead in such a system.

6. VARIABLE RETENTION TIME

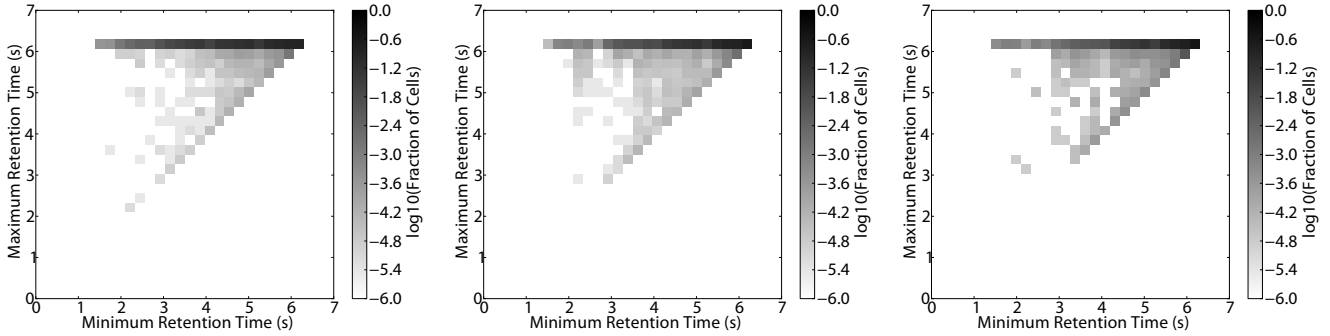
In this section, we investigate *variable retention time*, a phenomenon that causes the retention time of many DRAM cells to change randomly over time. Figure 11 shows how the retention time of a typical VRT cell shifts between multiple states over time, as observed in a cell from a `E 2Gb` DRAM.[8]

Throughout this section, we present results only for representative device families `A 2Gb`, `B 2Gb`, and `C 2Gb` for space; results for other device families are similar. Each of these devices was tested for 1024 rounds over approximately 24 hours. Results for each DRAM device family are presented only for the most effective fixed data pattern for that device family, as determined in Section 5.1.

6.1 Prevalence and Impact

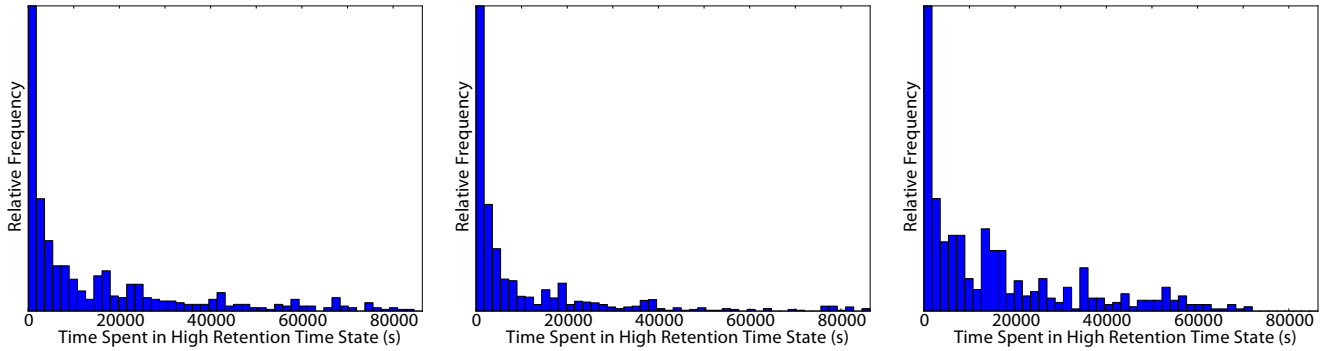
A cell suffering from variable retention time manifests as a cell whose retention time varies between tests, even with fixed temperature and data pattern. To illustrate both the frequency and magnitude

[8]The graph plateauing at 6.2 s indicates that for those rounds, no retention failure was observed up to our maximum tested retention time of 6.1 s.



(a) A 2Gb chip family (b) B 2Gb chip family (c) C 2Gb chip family

Figure 12: Maximum vs. minimum retention time



(a) A 2Gb chip family (b) B 2Gb chip family (c) C 2Gb chip family

Figure 13: Distribution of time between VRT events (each bin is 30 minutes)

of VRT, we construct a plot as follows. For each cell that fails any test at any retention time, we record both the *minimum* and *maximum* retention times across all tests that the cell fails. (If the cell does not fail any retention time tests in a given round, its retention time has increased beyond the maximum retention time we test of 6.1 s, and we record its maximum retention time as 6.2 s.) We would then like to plot each cell as a point on a scatterplot, where the x-coordinate is the cell's minimum retention time and the y-coordinate is the cell's maximum retention time. However, since the number of cells is very large, we instead plot the density of cells at each point. This is shown in Figure 12.

In the absence of VRT, all points would be expected to fall on the line $x = y$, i.e. tests would consistently observe the same retention time, such that the minimum would be equal to the maximum. Instead, in all tested devices, *most* cells (87% on average, and over 75% in all devices) fall into the horizontal line at $t_{MAX} = 6.2$ s indicating that at least one round did not observe a retention failure (note that the color scale is logarithmic). This has two ramifications:

- VRT is *ubiquitous* in modern DRAMs. Among cells with low minimum retention times (that is, cells of the greatest interest to mechanisms seeking to reduce refresh overhead), cells experiencing VRT are more common than cells not experiencing VRT.
- VRT can result in *significant* retention time changes. The points in the top-left hand of the plotted space indicates that a significant number of cells demonstrated a minimum retention time of approximately 1.5 s, and a maximum retention time of more than 6.1 s, a change by more than a factor of 4. This implies that handling VRT through guard banding alone would require a large guard band (greater than 4).

Both of these observations imply that mechanisms that use profiled retention times must be able to adapt to changes in the retention time profile.

6.2 Time Between State Changes

The length of time that DRAM cells spend in a given retention time state affects retention time profiling, since a cell has to leave higher retention time states and enter its lowest retention time state to be profiled accurately. Hence, if DRAM cells spend large amounts of time in high retention time states, profiling mechanisms may need to wait for long periods of time to observe low retention time states.

Each round, a VRT cell can be in either a low retention time state (which we define as having a measured retention time within 5% of the minimum retention time we observed for that cell in any test) or a high retention time state (which we define as having a measured retention time at least 5% greater than the cell's minimum retention time). For each VRT cell, whenever a state transition occurred (that is, the cell went from a low retention time state to a high retention time state or vice versa), we recorded the amount of time the cell spent in the previous state. Figure 13 shows the distribution of the amount of time spent in the high retention time state across all of the VRT cells in each of the three device families we present in this section.[9]

Ideally, all cells would spend very little time in high retention time states, so that a retention time profiling technique could detect the minimum retention time quickly. We find, however, that a substantial number of cells stay in high retention time states on the order of 15000 s (approximately 4 hours), and some cells were observed

[9]The distribution for the low retention time state, as well as distributions for other device families, are very similar.

to spend nearly the full length of the experiment (approximately 1 day) in high retention time states. This implies that retention time profiling mechanisms may need to continuously profile DRAM for periods of time on the order of *days* in order to reliably observe the lowest retention time of all cells.

Previous work has shown that each VRT cell spends an exponentially distributed amount of time in each state [14, 31], and that the distribution of time constants for these exponential distributions is itself exponentially distributed [15]. The shape of our observed distributions appear to be consistent with this prior work.

6.3 Implications for Retention Time Profiling

The greatest challenge posed by VRT for any retention time profiling mechanism is the fact that, at the architectural level, there does not appear to be any way to determine if a cell exhibits VRT without actually observing the cell undergoing VRT. This is because the charge trapping process that is believed to cause a retention time change is a memoryless random process [14]. (This is why time spent in each retention time state is exponentially distributed.) If previously undiscovered retention errors are a permanent possibility, then it seems inevitable that any approach to handling VRT will require some sort of error tolerance, possibly provided by the use of error-correcting codes (ECC). Future work will likely focus on *online profiling* (developing a mechanism to efficiently profile DRAM while it in use in order to mitigate the long profiling times implied by our results in Section 6.2 and on reducing the required ECC overhead).

As noted in Section 5.2, while DRAM manufacturers may be best equipped to perform retention time profiling due to DPD, the existence of VRT complicates retention time profiling for DRAM manufacturers. This is because exposure to very high temperatures, such as those used when soldering DRAM chips, can induce VRT in cells that were not previously susceptible to VRT [19, 34, 39]. As a result, any retention time profile collected by the manufacturer before module assembly is unlikely to accurately reflect the retention time profile of the DRAM in its final assembled state.

7. RELATED WORK

To our knowledge, this is the first work to comprehensively examine retention time behavior quantitatively in modern DRAM devices. In this section, we discuss prior work that has performed measurements on DRAM devices or examined either data pattern dependence (DPD) or variable retention time (VRT).

7.1 Measuring DRAM Device Behavior

Hamamoto et al. [6] measured the retention time distribution and circuit parameters for an experimental 16 Mbit DRAM chip. Kim et al. [18] performed similar measurements in three 1 Gbit chips. Both studies were limited in terms of the number and variety of DRAM devices they studied, and neither study discussed DPD or VRT.

Ahn et al. [1] propose to reduce refresh power in the low-power, high-wakeup-latency self-refresh mode found in modern DRAMs by profiling each DRAM row's retention time upon entering self-refresh to detect which rows can accept a lower refresh rate without losing data, then refreshing those rows less frequently accordingly. [1] avoids the problem of data pattern dependence because the DRAM cannot be accessed while in the self-refresh mode, so that the data pattern present during profiling is guaranteed to be the data pattern present throughout. This limits the scope of [1] to idle DRAMs. In addition, [1] does not appear to handle VRT.

7.2 Data Pattern Dependence

Nakagome et al. [26] first examined the sensing noise caused by interference between bitlines. They demonstrated the existence of this interference noise in an experimental DRAM array. Redeker et al. [30] performed simulation studies to attempt to predict the future impact of bitline coupling noise, and concluded that it would not be a serious issue. Al-Ars et al. [2] performed analysis and simulation studies to examine the interaction between different DRAM circuit architectures and bitline coupling noise. Li et al. [21] performed a similar analysis in their development of a DRAM reliability model. None of these works show the impact of bitline coupling on retention time. In addition, [30] and [2] studied only DRAMs with a *folded bitline* architecture, while modern DRAMs use an *open bitline* architecture [22, 24, 32, 33]. Open bitline architectures permit the use of smaller DRAM cells, improving DRAM density, but suffer from increased bitline-bitline coupling noise [33].

The memory testing literature discusses the similar issue of *neighborhood pattern-sensitive faults (NPSF)*, which refers to hard faults whose manifestation depends on the data stored in nearby memory cells [4]. Algorithms developed to test for NPSF could in theory be adapted to measure retention time in the presence of data pattern dependence. The most efficient algorithms for detecting NPSF are based on the March algorithm, which repeats a set of operations for each bit that must be tested and is hence linear in time complexity [4]. Unfortunately, March tests require knowledge of the mapping of addresses to DRAM array layout, which is problematic for reasons discussed in Section 5.2. Tests exist that do not require this knowledge [5], but these are exponential in time complexity, which is intractable.

7.3 Variable Retention Time

Variable retention time was first observed by Yaney et al. [39], and then confirmed and investigated in greater detail by Restle et al. [31]. These works established the key properties of VRT: the existence of multiple retention time states in VRT cells and the exponentially-distributed nature of the amount of time spent in each state. Since then, interest in VRT has focused on identifying its physical cause, primarily by measuring circuit-level features such as leakage currents [3, 14, 15, 16, 25, 27, 29]. No recent work we are aware of has evaluated the impact of VRT in modern DRAM devices.

8. CONCLUSION

In this paper, we present an analysis of key issues surrounding DRAM retention time profiling. We begin by laying out the structure of the DRAM system and providing a detailed explanation of our methodology for retention time profiling that we validate against prior works by illustrating retention time temperature dependence and the retention time distribution tail. We then describe and demonstrate the prevalence of two phenomena that complicate the profiling process: data pattern dependence and variable retention time. We explore the capabilities of different data patterns for detecting weak cells and discuss the role of device architecture in data pattern dependence. In addition, we investigate the propensity of variable retention time to manifest itself and analyze its implications on profiling. Using an FPGA-based testing platform, we demonstrate the effects these issues have on DRAM profiling on 248 commodity DDR3 DRAM chips across 5 different DRAM vendors.

To our knowledge, this paper is the first work of its kind to quantitatively study retention time behavior in modern DRAM devices. As DRAM scales to smaller technology nodes, it is expected that the problems we presented, alongside the refresh problem, will become exacerbated. Therefore, if mechanisms attempting to reduce refresh

count by profiling retention times are to be viable, these issues must be addressed in the near future. We hope that this paper will open up new research avenues for profiling techniques that can overcome the hurdles in retention time profiling that we have described.

9. ACKNOWLEDGMENTS

We thank the anonymous reviewers and members of the SAFARI research group for their feedback. We gratefully acknowledge Uksong Kang, Hak-soo Yu, Churoo Park, Jung-Bae Lee, and Joo Sun Choi at Samsung for feedback. We acknowledge the financial support of our industrial sponsors, including especially AMD, IBM, Intel, Oracle, and Samsung. Part of this work was done while Jamie Liu was an intern at Intel Corporation. This research was partially supported by grants from NSF (CAREER Award CCF-0953246) and Intel URO Memory Hierarchy Program.

10. REFERENCES

[1] J.-H. Ahn *et al.*, "Adaptive self refresh scheme for battery operated high-density mobile DRAM applications," in *ASSCC*, 2006.

[2] Z. Al-Ars *et al.*, "Influence of bit-line coupling and twisting on the faulty behavior of DRAMs," *IEEE TCAD*, vol. 25, no. 12, 2006.

[3] M. Chang *et al.*, "Si-H bond breaking induced retention degradation during packaging process of 256 Mbit DRAMs with negative wordline bias," *IEEE TED*, vol. 52, no. 4, 2005.

[4] K.-L. Cheng, M.-F. Tsai, and C.-W. Wu, "Neighborhood pattern-sensitive fault testing and diagnostics for random-access memories," *IEEE TCAD*, vol. 21, no. 11, 2002.

[5] D. Das and M. Karpovsky, "Exhaustive and near-exhaustive memory testing techniques and their BIST implementations," *Journal of Electronic Testing*, vol. 10, no. 3, 1997.

[6] T. Hamamoto, S. Sugiura, and S. Sawada, "On the retention time distribution of dynamic random access memory (DRAM)," *IEEE TED*, vol. 45, no. 6, 1998.

[7] M. Horiguchi and K. Itoh, *Nanoscale Memory Repair*. Springer, 2011.

[8] ITRS, "International Technology Roadmap for Semiconductors," 2011.

[9] JEDEC, "204-Pin DDR3 SDRAM Unbuffered SO-DIMM Design Specification," 2010.

[10] JEDEC, "DDR3 SDRAM Specification," 2010.

[11] JEDEC, "DDR4 SDRAM Specification," 2012.

[12] JEDEC, "LPDDR3 SDRAM Specification," 2012.

[13] B. Keeth *et al.*, *DRAM Circuit Design: Fundamental and High-Speed Topics*. Wiley-Interscience, 2008.

[14] H. Kim *et al.*, "Random telegraph signal-like fluctuation created by fowler-nordheim stress in gate induced drain leakage current of the saddle type dynamic random access memory cell transistor," *JJAP*, vol. 49, 2010.

[15] H. Kim *et al.*, "Characterization of the variable retention time in dynamic random access memory," *IEEE TED*, vol. 58, no. 9, 2011.

[16] H. Kim *et al.*, "Study of trap models related to the variable retention time phenomenon in DRAM," *IEEE TED*, vol. 58, no. 6, 2011.

[17] J. Kim and M. C. Papaefthymiou, "Dynamic memory design for low data-retention power," in *PATMOS-10*, 2000.

[18] K. Kim and J. Lee, "A new investigation of data retention time in truly nanoscaled DRAMs," *EDL*, vol. 30, no. 8, 2009.

[19] Y. I. Kim, K. H. Yang, and W. S. Lee, "Thermal degradation of DRAM retention time: Characterization and improving techniques," in *RPS-42*, 2004.

[20] M. J. Lee and K. W. Park, "A mechanism for dependence of refresh time on data pattern in DRAM," *EDL*, vol. 31, no. 2, 2010.

[21] Y. Li *et al.*, "DRAM yield analysis and optimization by a statistical design approach," *IEEE TCSI*, vol. 58, no. 12, 2011.

[22] K.-N. Lim *et al.*, "A 1.2V 23nm $6F^2$ 4Gb DDR3 SDRAM with local-bitline sense amplifier, hybrid LIO sense amplifier and dummy-less array architecture," in *ISSCC*, 2012.

[23] J. Liu *et al.*, "RAIDR: Retention-aware intelligent DRAM refresh," in *ISCA-39*, 2012.

[24] Y. Moon *et al.*, "1.2V 1.6Gb/s 56nm $6F^2$ 4Gb DDR3 SDRAM with hybrid-I/O sense amplifier and segmented sub-array architecture," in *ISSCC*, 2010.

[25] Y. Mori *et al.*, "The origin of variable retention time in DRAM," in *IEDM*, 2005.

[26] Y. Nakagome *et al.*, "The impact of data-line interference noise on DRAM scaling," *JSSC*, vol. 23, no. 5, 1988.

[27] B. Oh *et al.*, "Characterization of an oxide trap leading to random telegraph noise in gate-induced drain leakage current of DRAM cell transistors," *IEEE TED*, vol. 58, no. 6, 2011.

[28] T. Ohsawa, K. Kai, and K. Murakami, "Optimizing the DRAM refresh count for merged DRAM/logic LSIs," in *ISLPED*, 1998.

[29] K. Ohyu *et al.*, "Quantitative identification for the physical origin of variable retention time: A vacancy-oxygen complex defect model," in *IEDM*, 2006.

[30] M. Redeker, B. F. Cockburn, and D. G. Elliott, "An investigation into crosstalk noise in DRAM structures," in *MTDT*, 2002.

[31] P. J. Restle, J. W. Park, and B. F. Lloyd, "DRAM variable retention time," in *IEDM*, 1992.

[32] T. Schloesser *et al.*, "A $6F^2$ buried wordline DRAM cell for 40nm and beyond," in *IEDM*, 2008.

[33] T. Sekiguchi *et al.*, "A low-impedance open-bitline array for multigigabit DRAM," *JSSC*, vol. 37, no. 4, 2002.

[34] H. W. Seo *et al.*, "Charge trapping induced DRAM data retention time degradation under wafer-level burn-in stress," in *RPS-40*, 2002.

[35] R. K. Venkatesan, S. Herr, and E. Rotenberg, "Retention-aware placement in DRAM (RAPID): Software methods for quasi-non-volatile DRAM," in *HPCA-12*, 2006.

[36] Xilinx, "ML605 Hardware User Guide," 2012.

[37] Xilinx, "Virtex-6 FPGA ML605 Evaluation Kit," 2012. Available: http://www.xilinx.com/products/boards-and-kits/EK-V6-ML605-G.htm

[38] K. Yanagisawa, "Semiconductor memory," U.S. patent number 4736344, 1988.

[39] D. S. Yaney *et al.*, "A meta-stable leakage phenomenon in DRAM charge storage — variable hold time," in *IEDM*, 1987.