

HeatWatch: Improving 3D NAND Flash Memory Device Reliability by Exploiting Self-Recovery and Temperature Awareness

Yixin Luo[†] Saugata Ghose[†] Yu Cai[‡] Erich F. Haratsch[‡] Onur Mutlu^{§†}
[†]Carnegie Mellon University [‡]Seagate Technology [§]ETH Zürich

NAND flash memory density continues to scale to keep up with the increasing storage demands of data-intensive applications. Unfortunately, as a result of this scaling, the lifetime of NAND flash memory has been decreasing. Each cell in NAND flash memory can endure only a limited number of writes, due to the damage caused by each program and erase operation on the cell. This damage can be partially repaired on its own during the idle time between program or erase operations (known as the dwell time), via a phenomenon known as the self-recovery effect. Prior works study the self-recovery effect for planar (i.e., 2D) NAND flash memory, and propose to exploit it to improve flash lifetime, by applying high temperature to accelerate self-recovery. However, these findings may not be directly applicable to 3D NAND flash memory, due to significant changes in the design and manufacturing process that are required to enable practical 3D stacking for NAND flash memory.

In this paper, we perform the first detailed experimental characterization of the effects of self-recovery and temperature on real, state-of-the-art 3D NAND flash memory devices. We show that these effects influence two major factors of NAND flash memory reliability: (1) retention loss speed (i.e., the speed at which a flash cell leaks charge), and (2) program variation (i.e., the difference in programming speed across flash cells). We find that self-recovery and temperature affect 3D NAND flash memory quite differently than they affect planar NAND flash memory, rendering prior models of self-recovery and temperature ineffective for 3D NAND flash memory. Using our characterization results, we develop a new model for 3D NAND flash memory reliability, which predicts how retention, wearout, self-recovery, and temperature affect raw bit error rates and cell threshold voltages. We show that our model is accurate, with an error of only 4.9%.

Based on our experimental findings and our model, we propose HeatWatch, a new mechanism to improve 3D NAND flash memory reliability. The key idea of HeatWatch is to optimize the read reference voltage, i.e., the voltage applied to the cell during a read operation, by adapting it to the dwell time of the workload and the current operating temperature. HeatWatch (1) efficiently tracks flash memory temperature and dwell time online, (2) sends this information to our reliability model to predict the current voltages of flash cells, and (3) predicts the optimal read reference voltage based on the current cell voltages. Our detailed experimental evaluations show that HeatWatch improves flash lifetime by 3.85 \times over a baseline that uses a fixed read reference voltage, averaged across 28 real storage workload traces, and comes within 0.9% of the lifetime of an ideal read reference voltage selection mechanism.

1. Introduction

In many modern servers and mobile devices, solid-state drives (SSDs) containing NAND flash memory are used as the primary persistent storage media, due to their low access

latency compared to magnetic disk drives. As applications become more data intensive, the need for greater NAND flash memory density grows, to reduce the cost-per-bit of SSD storage. In the past decade, planar (i.e., 2D) NAND flash memory density has increased by more than 1000 \times , as a result of (1) aggressive manufacturing process technology scaling and (2) multi-level cell technology. Manufacturers have shrunk the planar NAND flash memory manufacturing process technology from 70 nm to 1X-nm (i.e., 15–19 nm) over the last decade [67], which has greatly decreased the size of each flash cell. At the same time, manufacturers use *multi-level cell* (MLC) and *triple-level cell* (TLC) technology to store more data in each cell [4, 5]. Older single-level cell (SLC) NAND flash memory stores a single bit of data per cell, while MLC and TLC NAND flash memory store two and three bits of data, respectively, per cell. Recently, manufacturers have turned to 3D integration to further increase the density of NAND flash memory by stacking flash memory cells vertically. State-of-the-art 3D NAND flash memory chips integrate 48–96 vertically-stacked layers of NAND flash memory [23, 34, 36, 54, 61, 66].

This rapid increase in NAND flash memory density has come at the cost of reduced reliability [4, 5, 11, 44, 50, 58]. NAND flash memory has a limited *lifetime*, which is defined as the number of program and erase operations (known as *P/E cycles*) that can be reliably performed on each flash cell while avoiding data loss for a minimum data retention time as guaranteed by vendors [4, 5, 11]. As the manufacturing process technology scales, the lifetime has reduced from 10,000 P/E cycles for 70 nm planar NAND flash memory to only 2,000 P/E cycles for a state-of-the-art 1X-nm planar NAND flash memory [67]. While 3D NAND flash memory currently has a higher lifetime than state-of-the-art planar NAND flash memory (e.g., 35,000 P/E cycles for 24-layer 3D MLC NAND), thanks to its use of a larger process technology node, its lifetime is expected to decrease in the future as manufacturers increase 3D NAND flash memory density more aggressively [4, 5, 26, 57].

The limited lifetime is a result of *wearout* that occurs as a flash cell is repeatedly programmed and erased. A flash cell consists of a transistor that can hold charge, where the amount of charge determines the *threshold voltage* at which the transistor turns on. The threshold voltage of the transistor is used to represent the data value stored within the cell. Unfortunately, after each additional P/E cycle, a greater number of electrons get *inadvertently trapped* within the flash cell, which changes the threshold voltage of the transistor [4, 5]. This threshold voltage change introduces errors and, thus, reduces the flash lifetime. Some of these inadvertently-trapped electrons gradually *escape* during the idle time between consecutive P/E cycles, i.e., the *dwell time*. The escape (i.e., *detrapping*) of the inadvertently-trapped elec-

trons is known as the *self-recovery effect* [46], as it partially undoes (i.e., *repairs*) the wearout of the cell. Prior works show that self-recovery can be accelerated by applying a high temperature to the flash cell during the dwell time [46], and propose to use this relationship to improve the lifetime of *planar* NAND flash memory [17, 64, 65].

While these proposals to exploit self-recovery and temperature effects are promising, they have two major shortcomings: they (1) do not demonstrate whether self-recovery mechanisms can practically reduce the effects of wearout on *real* NAND flash devices, (2) do not account for the design and manufacturing changes made for 3D NAND flash memory. 3D NAND flash memory cells predominantly use a different type of transistor (*charge trap transistor*) [5, 16, 22, 62] from planar NAND flash memory cells (*floating-gate transistor*), and require new manufacturing process technologies to successfully stack multiple dies within a chip [47, 54]. As a result, the observations and conclusions drawn for self-recovery in planar NAND flash memory may *not* be accurate for 3D NAND flash memory. **Our goal** is to (1) perform the first detailed experimental characterization of the effects of self-recovery and temperature on real, state-of-the-art 3D NAND flash memory devices, and (2) exploit our experimental findings by developing new mechanisms to improve the lifetime of 3D NAND flash memory.

We extensively characterize how self-recovery and temperature affect two major aspects of 3D NAND flash memory reliability: (1) retention loss and (2) program variation. *Retention loss* refers to the leakage of charge from a flash cell that contains valid data, which can introduce *retention errors* into the data [4, 5, 8, 9, 10, 40]. As a flash cell becomes more worn out, a greater number of electrons that are inadvertently trapped in the cell cause the cell’s *retention loss speed* (i.e., the rate at which charge leaks from the cell) to increase [19, 45]. *Program variation* refers to random variations that occur when a cell is being programmed to a target threshold voltage, which can cause the cell to be set to an incorrect voltage. The resulting errors are called *program variation errors* [4, 5, 12, 41, 45, 53, 55]. As a flash cell becomes more worn out, a greater number of electrons that get inadvertently trapped in the cell cause the cell’s programming speed (i.e., the rate at which the threshold voltage of a cell increases during a programming operation) to change [63], increasing program variation errors.

We make four key empirical findings on retention loss and program variation from our experimental characterization of self-recovery and temperature effects on real 3D NAND flash memory chips from a major manufacturer:

1. Increasing dwell time from 1 min to 137 min slows down retention loss by 40% (Section 3.2).
2. Lowering the temperature from 70 °C to 0 °C slows down retention loss by 58% (Section 3.3).
3. Increasing the temperature from 0 °C to 70 °C during programming increases the program variation by 21% (Section 3.3).
4. The effectiveness of self-recovery (i.e., the number of electrons that are successfully detrapped) is correlated with the dwell time experienced during the 20 most recent P/E cycles (Section 3.4).

Based on the results from our characterization, we find that prior models and model parameters for planar NAND flash

memory [46] are *not* accurate enough for 3D NAND flash memory. We use our characterization findings to construct a new unified model of self-recovery, temperature, retention loss, and wearout for 3D NAND flash memory, called *Unified Recovery and Temperature* (URT). URT consists of three components, which model (1) the impact of temperature and wearout on program variation; (2) the impact of temperature on data retention and self-recovery; and (3) the impact of time, wearout, and self-recovery on retention loss. URT combines these three components to accurately *predict* how changes in retention loss speed and program variation affect the raw bit error rate and cell threshold voltages. We find that URT is highly accurate, with a prediction error of only 4.9%.

Using URT, we develop *HeatWatch*, a new mechanism that exploits self-recovery and operating temperature information to improve the reliability of read operations and thus increase the lifetime of 3D NAND flash memory. *HeatWatch* efficiently tracks the amount of dwell time experienced (which depends on the storage access patterns of a workload) by each cell, the operating temperature of the memory device, and the retention time of data at runtime, requiring less than 1.6 MB of storage overhead within the SSD controller. Our mechanism uses the tracked information as inputs to URT, which accurately predicts and applies the best (i.e., optimal) read reference voltage to use for each read operation. This predicted optimal read reference voltage accounts for threshold voltage changes in the flash cells, reducing the number of raw bit errors in the data that is read from NAND flash memory. On average across a wide range of workloads, *HeatWatch* improves the overall lifetime of 3D NAND flash memory by $3.85\times$ compared to a baseline that uses a fixed read reference voltage, and comes within 0.9% of the lifetime provided by an *ideal* read reference voltage selection mechanism.

We make the following **key contributions**:

- Using real, state-of-the-art 3D charge trap NAND flash chips from a major vendor, we experimentally characterize the effects of self-recovery and temperature on retention loss speed and program variation. We show that 3D NAND flash memory exhibits different self-recovery and temperature effects than planar NAND flash memory.
- Based on our experimental characterization data, we construct URT, a unified model for retention loss, wearout, self-recovery, and temperature in 3D NAND flash memory. Our model quantifies these four effects to accurately predict the raw bit error rate and threshold voltage shift.
- We propose *HeatWatch*, a mechanism for 3D NAND flash memory that improves flash reliability and lifetime. *HeatWatch* (1) tracks the temperature, dwell time, and retention time online, and (2) sends this information to URT to accurately predict the optimal read reference voltage. By using the predicted optimal read reference voltage for flash read operations, *HeatWatch* reduces the raw bit error rate by 93.5%, and improves flash lifetime by $3.85\times$, over a baseline that uses a fixed read reference voltage.

2. Background

In this section, we provide necessary background on 3D NAND flash memory, NAND flash memory errors, and the self-recovery effect. A more detailed background is in [4, 5].

2.1. NAND Flash Memory Design and Operation

Flash Cell. NAND flash memory consists of multiple arrays of *flash cells*, where each cell stores a unit of data in the form of electric charge. In the majority of 3D NAND flash memory devices available today, a flash cell consists of a *charge trap transistor* [16, 22, 62].¹ Figure 1a shows the cross-section of a charge trap transistor in 3D NAND flash memory. The cylindrical *substrate* sits in the middle of the cell. One end of the substrate acts as the transistor *source*, and the other end acts as the transistor *drain*. There are three layers that wrap around the substrate: the *tunnel oxide*, the *charge trap* layer, and the *gate oxide*. The *control gate* wraps around the center of the cell, on top of the gate oxide.

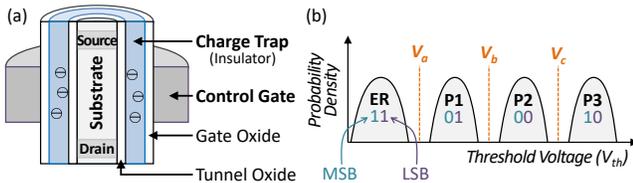


Figure 1: (a) Cross-sectional view of a 3D charge trap NAND flash memory cell. (b) Threshold voltage distribution for MLC NAND flash memory.

When a cell is programmed, charge is stored within the charge trap to represent data (see *Storing and Reading Data* below). The amount of charge stored within the charge trap changes the *threshold voltage* (V_{th}) of the cell. The flash cell turns *on* only if a voltage *higher* than the threshold voltage is applied to the control gate of the cell. When the cell turns on, current flows from the source to the drain.

Storing and Reading Data. In this work, we study *multi-level cell* (MLC) flash memory. Each multi-level cell stores *two* bits of data. To represent the different two-bit values (i.e., 00, 01, 10, 11), the range of all possible threshold voltages for the cell is split into four *voltage windows*, or *states*, as shown in Figure 1b. Before data can be programmed to a cell, the cell must be in the erased (ER) state. The NAND flash memory device programs a cell using *incremental step-pulse programming* (ISPP) [4, 5, 9]. During ISPP, the device repeatedly *pulses* a high voltage on the cell’s control gate, which injects additional charge into the charge trap. The pulses continue until the cell reaches its target state. Due to variation across program operations, the amount of charge injected during each pulse can differ. As a result, the threshold voltage of cells programmed to the same state is initially distributed across the voltage window. We represent the distribution of flash cells in each state using probability density functions.

To read the value currently stored within a cell, the NAND flash memory device applies a *read reference voltage* to the cell’s control gate. Figure 1b shows the three read reference voltages (V_a , V_b , and V_c) used to distinguish between the four states of an MLC NAND flash memory cell. The two bits stored within a cell are referred to as the *least significant bit* (LSB) and the *most significant bit* (MSB). To read the LSB

stored in a cell, the flash controller applies V_b to the cell control gate to determine whether the LSB is a 0 ($V_{th} > V_b$) or a 1 ($V_{th} < V_b$). To read the MSB stored in a cell, the flash controller first applies V_a , and then applies V_c , to the cell control gate, to determine whether the MSB is a 0 ($V_a < V_{th} < V_c$) or a 1 ($V_{th} < V_a$ or $V_{th} > V_c$).

Flash Organization. In a NAND flash memory chip, the cells are divided into multiple cell arrays known as *flash blocks*. Each block contains multiple rows of cells, where the control gates of cells in each row are connected together by a common *wordline*. All of the cells belonging to the same wordline are programmed and read at the same time. Each wordline in MLC NAND flash memory contains two *flash pages*, where the LSBs of all cells in the wordline form the *LSB page*, and the MSBs of all cells in the wordline form the *MSB page*. Each page is typically 4–16 kB in size. Within a block, all cells in the same column are connected in series to form a *bitline*. In 3D NAND flash memory, each cell on the same bitline belongs to a different stack layer, and the cells are connected together by a shared center cylinder (shown in Figure 1a), which includes the substrate, the charge trap, and oxide. For more detail on the physical organization of 3D NAND flash memory, we refer the reader to prior works [5, 21, 35].

2.2. NAND Flash Memory Errors

Errors can be induced in a NAND flash memory cell by several error sources, such as data retention [4, 5, 6, 8, 9, 10, 45], program or erase variation [4, 5, 12, 41, 45, 55], cell-to-cell program interference [4, 5, 6, 13, 14], and read disturbance [4, 5, 6, 7, 52, 55]. As shown in Figure 2, these errors can *shift* the threshold voltage distributions of each state across the original boundaries of each voltage window. As a result, the cells at the tail of each distribution are misread when we apply the originally-assigned read reference voltages (V_a , V_b , V_c), leading to *raw bit errors*.

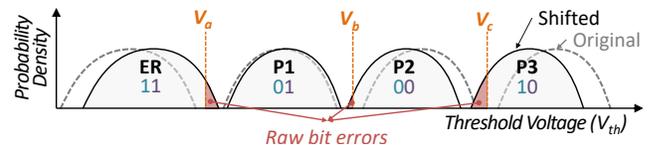


Figure 2: Shifted threshold voltage distribution leads to raw bit errors.

To mitigate the impact of raw bit errors and to avoid data corruption, flash controllers employ sophisticated *error-correcting codes* (ECC). ECC can correct up to a maximum *raw bit error rate* (RBER), known as the *error correction capability*. As the NAND flash memory is used, the RBER increases, and eventually exceeds the error correction capability, resulting in data loss (i.e., uncorrectable errors). The *lifetime* of a NAND flash memory device is determined by the number of program and erase (P/E) cycles that can be performed successfully while avoiding data loss for a minimum *retention guarantee* (i.e., the minimum amount of time after data has been written, that the data can still be read out without data loss). In this paper, we focus on two major sources of errors, for which self-recovery (see Section 2.3) can potentially lower the error rate and, thus, extend the flash lifetime: (1) retention errors and (2) program variation errors.

¹Note that this is different from planar (i.e., 2D) NAND flash memory, which typically uses a *floating-gate transistor* [33] as a flash cell. We refer the reader to prior works [4, 5, 47, 54] for detail on the differences between NAND flash memory based on charge trap transistors and NAND flash memory based on floating-gate transistors.

Retention errors are errors that are induced by charge leakage from a programmed flash cell while the cell is idle. Charge can leak either through the tunnel oxide layer to the substrate, or through the charge trap layer to a neighboring cell, as both the tunnel oxide and charge trap are imperfect insulators. Due to this leakage, the threshold voltage of a flash cell *decreases* over time, shifting the threshold voltage distributions of high-voltage states (i.e., states P1, P2, and P3) to lower voltages. Over a longer time period, an increasing fraction of the left tail of each distribution shifts across the read reference voltages, leading to more raw bit errors.

Program variation errors are errors that are induced by the random variations present in the flash *program* operation. In NAND flash memory, cells in the same page are programmed together. As we discuss in Section 2.1, during ISPP, a random amount of charge is injected during each programming pulse, due to variation across programming operations. When cells are programmed to a certain target state, their initial threshold voltages are naturally distributed across the state’s voltage window, as we show in Figure 1b. A number of factors can increase programming randomness, which can cause the tail of each threshold voltage distribution to extend across the read reference voltages, leading to program variation errors.

As more program and erase operations are performed to a flash cell, the number of retention and program variation errors increases due to **wearout**. Wearout is mainly caused by the electrical stress on the tunnel oxide when a flash cell is repeatedly programmed and erased. Due to this stress, a greater amount of charge is *inadvertently trapped* within the tunnel oxide. The inadvertently-trapped charge leads to two problems. First, it can form pathways for the charge stored in the charge trap layer to leak [8], which increases the speed of retention loss. Second, the NAND flash memory device *cannot* reliably erase a flash cell with inadvertently-trapped charge [41], and the cell may *not* always be in the ER state as a result, which increases program variation.

2.3. Self-Recovery

The self-recovery effect repairs the damage caused by flash wearout during the time between two P/E cycles, by *detrapping* some of the inadvertently-trapped charge [17, 37, 46, 48, 64, 65]. In this paper, we refer to the delay between consecutive program operations as the *dwel time*. The amount of repair done by self-recovery is affected by two factors: (1) dwell time and (2) operating temperature.

During the dwell time of a flash cell, a fraction of the charge that was inadvertently trapped in the tunnel oxide is slowly detrapped [46]. The reduction of inadvertently-trapped charge in a cell reduces the number of retention and program variation errors, and thus can extend the NAND flash memory lifetime. For a fixed retention time, a larger dwell time reduces the number of retention errors [46]. A *recovery cycle* refers to a P/E cycle where the program operation is followed by an extended dwell time. Since 3D NAND flash memory errors are dominated by retention errors [18, 47, 66], reducing the retention error rate by performing recovery cycles can increase flash lifetime significantly.

A higher operating temperature for NAND flash memory increases electron mobility [8, 46]. As a result, a short reten-

tion time at high temperature has the *same* retention loss effect as a longer retention time at room temperature [8], which we call the *effective retention time*. Similarly, a short dwell time at high temperature has the *same* self-recovery effect as a longer dwell time at room temperature [46], called the *effective dwell time*. The equivalence between time elapsed at a certain temperature and the corresponding effective time at room temperature can be modeled using Arrhenius’ Law [1, 8, 28, 46]:

$$AF(T_1, T_2) = \frac{t_1}{t_2} = \exp\left(\frac{E_a}{k_B} \cdot \left(\frac{1}{T_1} - \frac{1}{T_2}\right)\right) \quad (1)$$

In Equation 1, AF is the *acceleration factor* between t_1 and t_2 , where t_1 is the retention or dwell time under temperature T_1 , and t_2 is the retention or dwell time under temperature T_2 . k_B is the Boltzmann constant, which is 8.62×10^{-5} eV/K. E_a is the activation energy, which is a manufacturing-process-dependent constant. For a planar NAND flash memory device, $E_a = 1.1$ eV [29]. To our knowledge, there is no public literature that reports the value of E_a for 3D NAND flash memory.

3. Characterizing the Self-Recovery Effect

To understand the behavior of the self-recovery effect in 3D NAND flash memory, we perform an extensive characterization of the effect using *real*, state-of-the-art 3D NAND flash memory chips. Our goal in this characterization is to answer the following research questions:

- How does the dwell time affect retention and program variation errors?
- What is the correlation between dwell time and the magnitude of the self-recovery effect?
- How does the operating temperature affect the number of retention and program variation errors?
- How do the benefits of self-recovery change based on the number of performed *recovery cycles*?

We make all of our characterization data, including results not shown in this paper for brevity, available in an extended report [42] and online [56].

We use the observations and analysis from our characterization to drive the design of a new model of 3D NAND flash memory reliability, as described in Section 4.

3.1. Characterization Methodology

To answer the above research questions, we design new experiments to test how flash reliability changes with different dwell times and temperatures. In these experiments, we use state-of-the-art 30- to 40-layer 3D charge trap NAND flash chips from a major vendor.² We use a NAND flash memory characterization platform that fits in the SSD form factor, and contains a special version of the firmware in the SSD controller. We use a server machine to issue remote procedure calls (RPC) [3] to the firmware over a Serial ATA (SATA) [60] connection. These RPCs trigger commands to be sent directly to the flash chips, and can transfer the raw data (i.e., data with raw bit errors) directly from the flash chips to the server *without* applying error correction (ECC). This allows us to

²We do not disclose the exact number of stacked layers in the chips, to protect the anonymity of the flash vendor, and we cannot disclose the *exact* voltage values, as this is proprietary information.

observe the effect of dwell time and operating temperature on the raw bit error rate of each flash chip.

We use two metrics to evaluate flash reliability. First, we measure the *raw bit error rate* (RBER), which is the rate at which errors occur in the data *before error correction*. To calculate RBER, we read data from a NAND flash memory chip using the default read reference voltage, and compare the data using a pristine server-side copy of the data that was originally written to the chip. Second, we show the statistical mean of the threshold voltage distribution of each high-voltage state (i.e., P1, P2, P3). As we mention in Section 2.2, retention loss and program variation cause the threshold voltages of cells to shift, which leads to the raw bit errors.³ To obtain the threshold voltage distribution of a flash page, we perform multiple read operations to sweep the range of all positive read reference voltages, using the *read-retry* command on the NAND flash memory chip [8, 12, 24].⁴ Read-retry allows us to fine-tune the read reference voltage used for each read operation. The smallest amount by which we can change the read reference voltage is called a *voltage step*. We normalize each threshold voltage value to the number of voltage steps needed to reach that particular voltage value.²

Limitations. In our experiments, we characterize 3D NAND flash memory chips of the same model from one major vendor. Our approach ensures that any variation that we observe in our characterization is the result of only manufacturing process variation, and not a result of differences in flash chip architecture or different manufacturing techniques used by different vendors. While we do not take vendor-related variation into account, we believe that our general qualitative findings on the effects of self-recovery and temperature can be generalized to 3D charge trap NAND flash memory manufactured by other vendors. This is because the underlying structure of 3D charge trap cells (see Section 2.1) is similar across different vendors [18, 47, 54]. Thus, while the exact numbers reported in this work may differ from vendor to vendor, our qualitative findings, which are a result of charge detrapping from the tunnel oxide (see Section 2.3), should be similar across vendors.

We are unable to perform repeated runs of our test procedures on the *same* block, as each run of a test procedure induces further wearout on a block. The amount of wearout affects the error rate of NAND flash memory [4, 5, 12, 41, 45, 55]. To ensure an accurate comparison between multiple test procedure runs, we use eight *target* wordlines in the same stack layer from eight randomly-selected flash blocks that are set to the *same* level of wearout for the same test procedure. By selecting wordlines in the same layer, we eliminate the potential impact of cross-layer process variation. Note that we do not characterize *chip-to-chip* process variation, as an accurate study of such variation requires a large-scale study of a large number (e.g., hundreds) of 3D NAND flash memory chips, which we do not have access to. Hence, we leave such a large-scale study for future work.

³We are unable to show the full threshold voltage distribution for the ER state, because the ER state threshold voltages are negative, and our platform cannot measure negative voltage values. This is similar to prior works [4, 7, 8, 12].

⁴Due to space limitations, we refer the reader to prior works [41, 55] for a detailed methodology on how to obtain the threshold voltage distribution.

3.2. Characterizing the Dwell Time Effect

To measure the effect of dwell time on flash reliability (see Section 2.3), we characterize the RBER and the threshold voltage distribution. We wear out each of our target blocks by repeatedly writing pseudorandom data until the block reaches 3,000 P/E cycles. For the last 300 P/E cycles, we use a different dwell time for each block, spanning a range of 64–8192 s. Prior work shows that the magnitude of the self-recovery effect is correlated with the dwell time for only the last 10% of P/E cycles performed on a block [46]. We show in Section 3.4 that the dwell time used during only the *last 20 P/E cycles* affects self-recovery.

We measure how the dwell time affects retention loss speed and program variation, by performing a *retention test* on each target wordline *immediately* after the block containing the wordline reaches 3,000 P/E cycles. In this test, we program pseudorandom data to the target wordline, and repeatedly measure the threshold voltage distribution using the methodology described in Section 3.1 for up to 71 min (i.e., 4260 s) after the data was written. We conduct this experiment at an environmental temperature of 70 °C, which accelerates both self-recovery and retention loss to reduce the experiment duration to a reasonable length [46].⁵ We later repeat a small portion of the test under room temperature (20 °C), and verify that all of our observed trends remain the same.

Effect on RBER. First, we study how self-recovery affects the raw bit error rate. Figure 3 shows the RBER as retention time (t_r) increases, for different dwell times (t_d) used for the last 300 P/E cycles. We use a color gradient for the curves, where the reddest (topmost) curve has the shortest dwell time, and the blackest (bottommost) curve has the longest dwell time. Note that the x-axis and y-axis are both in log scale.

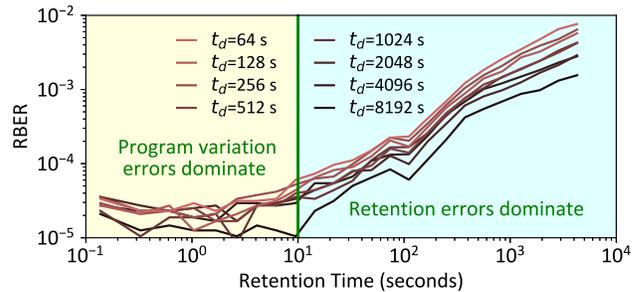


Figure 3: Change in RBER over retention time for flash pages that were programmed using different dwell times.

We make two observations from Figure 3. First, when the retention time is short (i.e., $t_r < 10$ s), the RBER is *similar* across different dwell times. During this time, the RBER is dominated by program variation errors [11, 41, 45]. Recall that a longer dwell time increases the amount of detrapped charge during self-recovery. However, since the RBER is similar across different curves regardless of the dwell time, this means that self-recovery does *not* mitigate program variation errors. Therefore, unlike previous works [17, 48, 65], we conclude that self-recovery does *not* repair *all* of the errors that occur due to wearout in 3D NAND flash memory. Second, when

⁵Based on Arrhenius’ Law [1], the same experiment would take more than 11 years to finish had we performed it at room temperature (20 °C).

the retention time is large (i.e., $t_r > 10$ s), there is a strong correlation between a longer dwell time and a lower RBER. During this time, the RBER is dominated by retention errors (hence the growth in RBER as the retention time increases). We conclude that a longer dwell time after an erase operation mitigates retention errors, but not program variation errors, in 3D NAND flash memory.

Effect on Threshold Voltage. Next, we study the threshold voltage distribution of the flash pages under test, to understand how self-recovery affects the threshold voltage shift (and thus the RBER) due to retention loss. Figure 4 shows the threshold voltage distribution *before* (black dots, $t_r = 1$ min) and *after* (red dots, $t_r = 71$ min) a large retention time elapses, for a flash page programmed using a 64 s dwell time (top plot), and for a flash page programmed using a 8192 s dwell time (bottom plot). We observe from the figure that when the dwell time is higher, the threshold voltage distribution shift due to retention loss is significantly smaller.

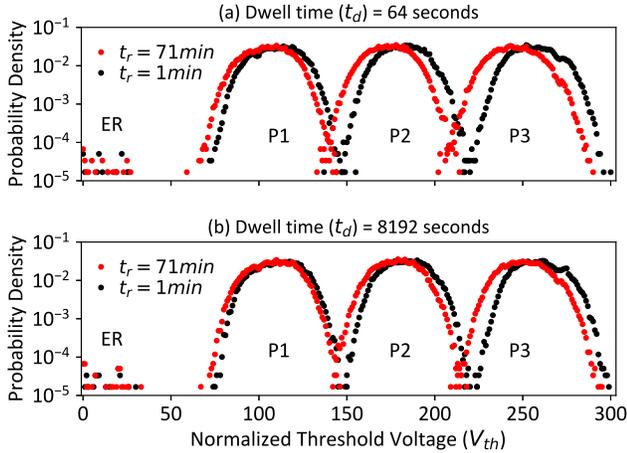


Figure 4: Threshold voltage distribution before and after a long retention time, for different dwell times.

To quantify the threshold voltage shift as a function of the dwell time, we plot the statistical means of the threshold voltage distributions of cells programmed to the P1, P2, and P3 states in Figure 5. We use the same color gradient that we used in Figure 3 to represent the different dwell times. Note that for these experiments, the smallest retention time that we show on the x-axis ($t_r = 64$ s) is much larger than the smallest retention time shown in Figure 3 ($t_r = 0.5$ s), because it takes significantly longer for us to sweep the threshold voltage of the cells in a wordline (as in Figure 5), compared to simply measuring the RBER of the wordline (as in Figure 3). We make two observations from the figure. First, for all three states, the mean threshold voltage changes by a smaller amount when the dwell time is higher, corroborating the threshold voltage distribution shifts shown in Figure 4. Second, for a fixed dwell time, the change in voltage is linearly related with the log of retention time.⁶ We use this relationship to develop a simple model that can quantify how retention loss speed changes with dwell time (see below).

⁶A similar linear relationship between the change in threshold voltage and the log of the retention time is observed for planar NAND flash memory in past works [29, 46].

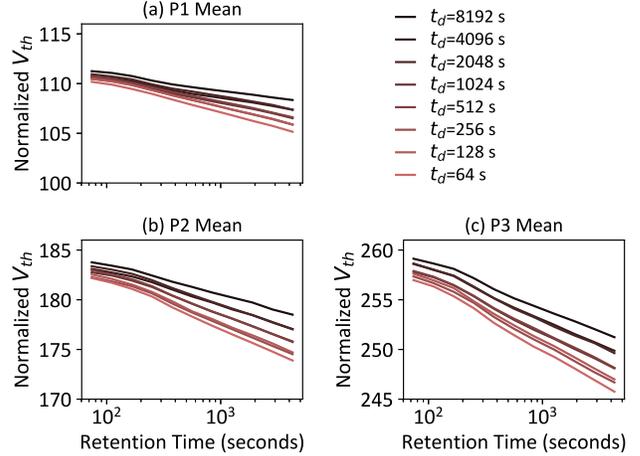


Figure 5: Threshold voltage distribution mean vs. retention time for different dwell times.

We also calculate how the width of the distribution changes due to retention loss for different dwell times (not plotted). We observe that the change in the distribution width is relatively small, and thus choose to ignore it to simplify the analysis.

Effect on Retention Loss Speed and Program Variation. To quantify how self-recovery changes (1) retention loss speed and (2) program variation, we construct a simple model of how the threshold voltage and RBER change due to these two factors. As we observe in Figure 5, the threshold voltage distribution mean is linearly correlated with the logarithm of the retention time (t_r). Thus, we fit our measurements to the following linear model, for a given dwell time:

$$Y(t_r) = \alpha \cdot \ln(t_r) + \beta \quad (2)$$

In this model, Y can represent either (1) the mean of the threshold voltage distribution of each high-voltage state (i.e., P1/P2/P3); or (2) the logarithm of the RBER, i.e., $\log(RBER)$,⁷ based on the values chosen for α and β . α represents the retention loss speed. β represents the *program offset*, which is the initial value of Y immediately after programming.

We use the *absolute value* of the program offset (i.e., $|\beta|$) to quantify the impact of program variation. For the threshold voltage distribution mean of each high-voltage state, Y and β are positive, and a *more positive* program offset results in a *higher* initial mean. As we observe under *Effect on Threshold Voltage* in Section 3.3, when the mean voltages of neighboring distributions increase, the overlap between the distributions decreases, which in turn reduces the number of program variation errors. For $\log(RBER)$, Y and β are negative, because the RBER is always less than 1. A *more negative* program offset (i.e., a greater $|\beta|$) corresponds to a *more negative* initial value of $\log(RBER)$ (i.e., *fewer* errors).

For each dwell time, we fit Equation 2 to our experimental characterization data in order to calculate the values of

⁷We model the *logarithm* of the RBER, because when retention loss linearly shifts the threshold voltage distribution, which roughly follows a Gaussian distribution [41], the linear distribution shift results in a logarithmic change in RBER, which is quantified as the overlapping area between two neighboring distributions.

α and $|\beta|$ when Y represents (1) the mean voltage of each higher-voltage state, or (2) $\log(RBER)$. Figure 6 (left) illustrates the relation between dwell time and retention loss speed (α), normalized to the greatest observed retention loss speed. Figure 6 (right) illustrates the relation between dwell time and program offset ($|\beta|$), normalized to the greatest observed program offset. Note that the x-axis (i.e., the dwell time) is in log scale. In both figures, the markers represent our measured data points from real NAND flash memory chips, and the dashed lines show a linear trend line for the data.

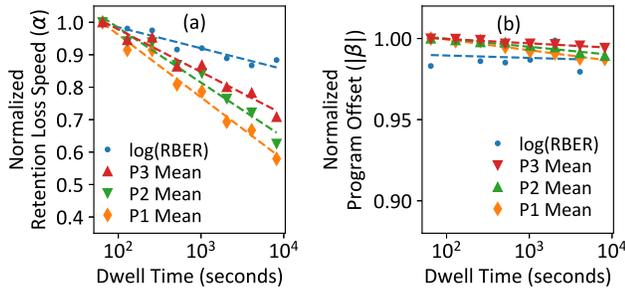


Figure 6: Retention loss speed (left) and program offset (right), for different dwell times.

We make two key observations from Figure 6. First, the self-recovery effect reduces the retention loss speed linearly with the logarithm of dwell time. We observe, however, that the change in retention loss speed is *different for each state*. As Figure 6 (left) shows, our data points follow the linear trend line closely (with an R^2 value larger than 90% for each state and for the RBER). Second, as we concluded from Figure 3, self-recovery has little effect on program variation within the tested dwell time range. As Figure 6 (right) shows, the maximum difference in program offset for any of our threshold voltage states is less than 2.1%. Note that our second finding is *new*, and it shows that, unlike previous findings for planar NAND flash memory [17, 48, 64, 65], self-recovery does *not* reduce the number of program variation errors, and hence the amount of wear, in 3D NAND flash memory.

3.3. Characterizing the Temperature Effect

Next, we measure the effect of temperature on self-recovery and flash reliability (see Section 2.3). Similar to the experiment in Section 3.2, we use eight target wordlines in the same stack layer from randomly-selected flash blocks. First, for each block, we wear out the block in 1,000 P/E cycle intervals up to a total of 10,000 P/E cycles, writing pseudorandom data and using a fixed dwell time of 0.5 s. We then put the test chip in a temperature-controlled box, and set a target temperature. After the temperature of the test chip settles to the target temperature, we perform 20 more P/E cycles to each target wordline at the target temperature, using a 0.5 s dwell time. Though these P/E cycles are performed at different temperatures for each test, the dwell time we use is small, and thus we assume that the difference between the equivalent dwell times at room temperature are small across our tests. Then, we perform the retention test described in Section 3.2 for all target wordlines up to a retention time of 71 min. We repeat the retention test under a range of target temperatures in each round: 0, 10, 20, 28, 35, 50, 60, and 70 °C.

During the retention test, data is both programmed and read under the target temperature.

Effect on RBER. First, we study how the RBER changes with retention time under different temperatures, as shown in Figure 7 for a wordline with 10,000 P/E cycles. Each curve represents the RBER under a different programming temperature. We use a color gradient to indicate the temperature: the reddest color represents the highest temperature (70 °C) and the blackest represents the lowest temperature (0 °C).

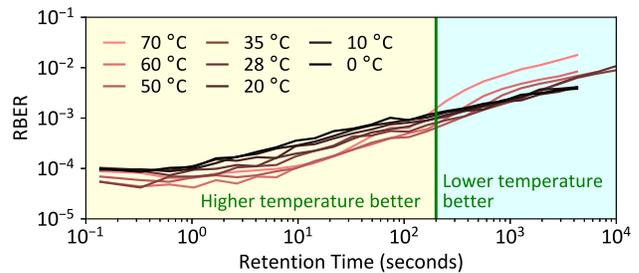


Figure 7: RBER over retention time at 10,000 P/E cycles under different programming temperatures.

We make two key observations from the figure. First, when the retention time is small (i.e., $t_r < 2 \cdot 10^2$), the RBER is lower for higher temperatures (i.e., the red curves). Recall that when the retention time is small, the RBER is dominated by program variation errors [11, 41, 45]. Thus, we expect that the RBER decreases with higher temperatures because higher programming temperatures *decrease* the number of program variation errors (we discuss this in more detail under *Effect on Threshold Voltage* below). Second, when the retention time is larger (i.e., $t_r > 2 \cdot 10^2$), the RBER becomes higher for higher programming temperatures. This is because as the temperature increases, the retention errors increase at a faster rate. Due to this faster growth, the RBER for a higher temperature overtakes the RBER for a lower temperature at a retention time between 10^2 – 10^3 s. This indicates that the threshold voltage shift due to high-temperature retention is faster than that for low-temperature retention, which is in line with Arrhenius’ Law [1] (see Section 2.3).

Effect on Threshold Voltage. Next, we study how the programming temperature affects the threshold voltage of a flash cell. We begin by studying how the initial threshold voltage (i.e., the threshold voltage immediately after programming) changes with temperature. We measure the threshold voltage distribution under different programming temperatures, and then fit our data to Equation 2 to compensate for any retention loss that occurs during the measurements. Figure 8 shows the resulting threshold voltage distribution for each state, at 0 °C (the black dotted curves) and at 70 °C (the red curves). Note that the ER state distribution at 70 °C completely falls below the lowest voltage that we can measure, and hence is not shown.

We make two observations from Figure 8. First, the higher programming temperature shifts the P1, P2, and P3 states to higher threshold voltages, and the ER state to lower threshold voltages. The threshold voltage shifts are likely due to the increased electron mobility under high temperature, which improves the speed of the program operation (and likely the

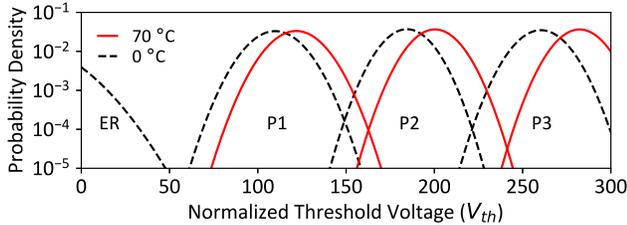


Figure 8: Threshold voltage distribution right after programming at different programming temperatures, predicted by our retention loss model (Equation 2).

erase operation as well [46, 63]). As a result, each programming pulse adds a greater amount of charge. Second, due to the threshold voltage distribution shifts, the amount of overlap between the P1 and P2 distributions, and between the P2 and P3 distributions, decreases at a higher programming temperature. This is because while the distribution means shift to higher voltages at a higher programming temperature, the distribution widths do not change. Because of the smaller amount of overlap between two neighboring distributions, there are fewer program variation errors at higher temperatures, as we have shown in Figure 7.

Next, we study how threshold voltage shifts due to retention loss change with the programming temperature. For brevity, we do not plot these results. We observe that when the retention time is large ($t_r > 2 \cdot 10^2$), the retention loss speed increases due to high temperature, which is similar in nature to the effect of programming temperature on RBER.

Effect on Retention Loss Speed and Program Variation. We use our model from Equation 2 to calculate the retention loss speed and program offset for each programming temperature, based on our characterization data. Figure 9 illustrates how the programming temperature affects retention loss speed (left) and program offset (right). We fit a quadratic trend line for retention loss speed, and a linear trend line for program offset (shown as dashed lines).

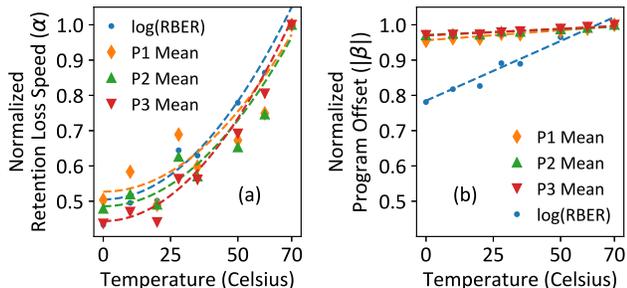


Figure 9: Retention loss speed (left) and program offset (right) across different programming temperatures.

We make two observations from Figure 9. First, a higher temperature accelerates retention loss at a superlinear rate. We show in Section 4 that this trend complies with Arrhenius’ Law [1]. Second, we find that the programming temperature affects program variation. This effect has *not* been accounted for in prior work, which usually assumes that program operations are performed at room temperature [29]. In Figure 9 (right), we find that the program offset is higher at higher

programming temperatures. As already shown in Figure 8, the higher initial threshold voltage at higher programming temperatures reduces the amount of overlap between neighboring threshold voltage distributions, which in turn *reduces* the number of program variation errors. We conclude that higher temperature *increases* retention errors but *reduces* program variation errors.

3.4. Characterizing the Recovery Cycle Effect

We measure the effect of *recovery cycles*, i.e., P/E cycles performed with a long dwell time, on self-recovery and flash reliability. We measure how the *number* of recovery cycles affects retention loss speed. We focus on retention loss speed in this experiment because, as we saw in Section 3.2, the dwell time does *not* affect program variation. We first wear out each block by repeatedly writing pseudorandom data with a dwell time of 0.5 s, until the block reaches 3,000 P/E cycles. Then, we start self-recovery, performing recovery cycles using a 6-hour dwell time. During the idle time of each recovery cycle, we perform our 71-minute retention test at an operating temperature of 70 °C to measure the current retention loss speed. We keep performing recovery cycles until the change in retention loss speed is less than 1%, which indicates that an additional recovery cycle does *not* significantly increase the effect of self-recovery.

Effect on Retention Loss Speed. Based on our characterization data, we calculate the retention loss speed (α) after each recovery cycle. We use Equation 2 to calculate α , as we did in Figures 6 and 9. Figure 10 shows how the retention loss speed changes as a function of the number of recovery cycles performed. We fit power law trend lines to the data, shown as a dashed line in the figure.

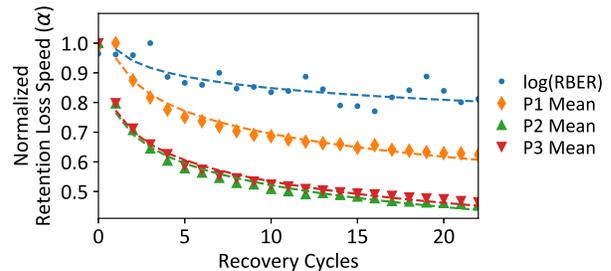


Figure 10: Retention loss speed vs. recovery cycles.

We make two key observations from Figure 10. First, to our surprise, the reduction in retention loss speed due to self-recovery becomes insignificant very quickly. We find that, for wordlines that have endured 3,000 P/E cycles, most of the benefits of self-recovery occur within 22 recovery cycles for 3D NAND flash memory. This is much smaller than the number of cycles required according to prior work [46], which finds that most of the benefits of self-recovery occur only when the number of recovery cycles is 10% of the total P/E cycle count. In other words, according to prior work, it should have taken at least 300 recovery cycles to achieve most of the benefits of self-recovery. Importantly, this implies that we can reap the benefits of self-recovery with a much lower overhead (i.e., significantly fewer recovery cycles) than previously-proposed mechanisms [20, 30, 46]. Thus,

we can improve the overall performance of NAND flash memory devices that perform self-recovery. Second, the RBER does *not* change significantly *until after* the first three recovery cycles. To reduce the latency of self-recovery, prior works [17, 48, 64, 65] distribute recovery cycles throughout the flash lifetime, and periodically perform only a single recovery cycle. Based on our observation, performing only one recovery cycle may *not* change the RBER significantly, and these mechanisms may *not* significantly improve the flash lifetime as currently designed.

3.5. Summary of Key Observations

We conclude that (1) the self-recovery effect reduces retention loss speed linearly with the logarithm of dwell time, and has little effect on program variation; (2) the temperature effect increases retention loss speed at a superlinear rate, and increases program variation; and (3) the reduction in retention loss speed due to self-recovery becomes insignificant after 20 recovery cycles.

4. Self-Recovery Effect Modeling

We use our observations from Section 3 to construct *Unified Recovery and Temperature* (URT), a comprehensive analytical model of the impact of retention, wearout, self-recovery, and temperature on two output parameters: (1) the threshold voltage of a flash cell, and (2) the raw bit error rate (RBER) of a flash page. URT calculates each output parameter Y as:

$$Y = Y_0 + \Delta Y(t_r \cdot AF, t_d \cdot AF, PEC) \quad (3)$$

In the equation, Y_0 is the initial value of the output parameter immediately after a cell is programmed, and ΔY is the change in the output parameter due to retention loss. ΔY is a function of the (1) retention time (t_r) and (2) the dwell time (t_d), both of which are scaled by an *acceleration factor* AF (see below), and (3) the P/E cycle count (PEC).

URT consists of three components. The *program variation component* (PVM; Section 4.1) predicts Y_0 based on the amount of program variation that took place. The *effective retention/dwell time component* (RDTM; Section 4.2) computes AF , which scales the retention or dwell time at the current temperature of the NAND flash memory to the *effective time* at room temperature that has the same impact on Y . The *self-recovery and retention component* (SRRM; Section 4.3), predicts ΔY based on the effective retention/dwell time and the P/E cycle count. We show how URT can be used to improve flash reliability in Section 5.

4.1. Program Variation Component

First, we build a *program variation model* (PVM) to predict the *initial values* (Y_0 in Equation 3) of the threshold voltage and RBER immediately after a cell is programmed. The initial values are determined by (1) the target programming voltage, which is fixed for each state, and (2) the program offset (Section 3). Recall that program offset is affected by the programming temperature (Section 3.3). Prior work shows that the P/E cycle count significantly affects program offset as well [12, 13, 41, 45].

To account for both variables (i.e., programming temperature and P/E cycle count), we use a multivariate linear regression model to model program variation:

$$Y_0 = A \cdot T_p \cdot PEC + B \cdot T_p + C \cdot PEC + D \quad (4)$$

In PVM, Y_0 is a function of the P/E cycle count of the flash cell (PEC) and the programming temperature (T_p), which are input parameters. A , B , C , and D are model constants that change based on which value we model (e.g., initial threshold voltage, initial log of RBER). We fit PVM to our characterization data using the ordinary least squares implemented in Statsmodels [59], and conclude that the model fits well, with an R^2 value of 91.7%. We provide the values of all the fitted model parameters online [42].

4.2. Effective Retention/Dwell Time Component

Next, we build an *effective retention/dwell time model* (RDTM) to calculate the *acceleration factor* (AF in Equation 3), which scales the retention time or dwell time under *any* temperature (t_{real}) to the effective time under room temperature (t_{room}). Arrhenius' Law [1] (see Section 2.3) is commonly used by prior works to scale the retention time and dwell time of flash memory across different temperatures (e.g., [8, 9, 10, 28, 46]). RDTM uses the same general equation as Arrhenius' Law (Equation 1):

$$AF = \frac{t_{real}}{t_{room}} = \exp\left(\frac{E_a}{k_B} \cdot \left(\frac{1}{T_{real}} - \frac{1}{T_{room}}\right)\right) \quad (5)$$

In RDTM, AF is a function of the room temperature (T_{room}), the current temperature of the NAND flash memory (T_{real}), and the activation energy (E_a). k_B is the Boltzmann constant. To accurately model the amplification factor, we (1) experimentally calculate a new value of E_a that we can use for 3D NAND flash memory; and (2) verify the accuracy of Arrhenius' Law through experimental characterization, which no previous work has done for 3D NAND flash memory.

While E_a is commonly accepted to be 1.1 eV for *planar* NAND flash memory [8, 28], we cannot use the same value of E_a for 3D NAND flash memory, due to changes in materials and manufacturing process. Fortunately, we have extensive real device characterization data on retention loss at different temperatures (Section 3.3), which enables us to determine the correct E_a for 3D NAND flash memory. We define t_1 as the time required for a 3D NAND flash memory device to experience a fixed amount of retention loss, ΔY , at temperature T_1 . We define t_2 as the time required for the *same* amount of retention loss to occur at temperature T_2 . Using Equation 1, the activation energy (E_a) can be calculated as:

$$E_a = \frac{k_B \cdot \ln\left(\frac{t_1}{t_2}\right) \cdot T_1 T_2}{T_2 - T_1} \quad (6)$$

We define t_1 as the time required for 3D NAND flash memory to experience a fixed amount of retention loss, ΔY , at temperature T_1 . We define t_2 as the time required for the *same* amount of retention loss to occur at temperature T_2 .

We choose $T_2 = 343.15$ K (70 °C) as the reference temperature, and $t_2 = 3600$ s as the reference retention time, as our model is more resilient to noise at a larger retention time. Using our characterization data from Section 3.3, we find the equivalent t_1 for different temperature values of T_1 , spanning 20–70 °C, and for different P/E cycle counts, spanning 1,000–

10,000 P/E cycles. We use ordinary least squares estimates to fit Equation 6 to these data points. From the fit, we determine that for the best fit, $E_a = 1.04$ eV for the 3D NAND flash memory chips that we test. The 95% confidence interval for E_a is 1.01–1.08 eV. The value of E_a is based on the materials used for the cell, and should be similar for 3D charge trap cells manufactured by other vendors [18, 31, 47].

Next, we verify that Arrhenius’ Law holds for 3D NAND flash memory, by calculating the coefficient of determination (R^2) of the fit to the equation for Arrhenius’ Law. We find that $R^2 = 0.76$. This means that Arrhenius’ Law explains 76% of the variations due to temperature. This is a good fit given that we use a single value for E_a (best fit) across *all* of our data points, because it is known that activation energy changes across different temperatures and P/E cycle counts [2, 43]. We use a single value of E_a regardless of temperature and P/E cycle count to simplify RDTM. We leave more accurate activation energy modeling for future work.

4.3. Self-Recovery and Retention Component

We build a *self-recovery and retention model* (SRRM) to accurately predict the *threshold voltage shift and change in RBER* (ΔY in Equation 3) due to retention loss. SRRM predicts the effect of (1) effective retention time, (2) effective dwell time, and (3) P/E cycle count, which all directly affect retention loss speed (see Section 3.2).

To construct SRRM, we repeat our dwell time experiments from Section 3.2 at *room temperature*. We cover a slightly larger dwell time range than our previous experiments, testing from 32 s to 4.6 h. To include the effect of the P/E cycle count, we perform the retention test described in Section 3.2 for up to a 24-day retention time under room temperature, using eight randomly-selected flash blocks, and spanning a range between 1,000 and 10,000 P/E cycles at 1,000-P/E-cycle intervals. We observe similar trends in terms of retention time, dwell time, and temperature sensitivity as the findings we observe at a higher temperature in Section 3. For brevity, we do not plot these results, but we provide them online [42].

From an analysis of the results of these experiments, we find that the threshold voltage shift in 3D NAND flash memory is much less sensitive to the P/E cycle count than planar NAND flash memory. Thus, we develop a new model that predicts the impact of retention and self-recovery on 3D NAND flash memory, instead of relying on prior models for planar NAND flash memory. Our SRRM model is as follows:

$$\Delta Y(t_{er}, t_{ed}, PEC) = b \cdot (PEC + c) \cdot \ln \left(1 + \frac{t_{er}}{t_0 + a \cdot t_{ed}} \right) \quad (7)$$

In SRRM, ΔY is a function of three input parameters: (1) the effective retention time of the data stored in the cell (t_{er}), (2) the effective dwell time (t_{ed}), and (3) the P/E cycle count for a flash cell (PEC). The model has four constants, whose values change depending on which output parameter (ΔY) we are evaluating: b and c control the impact of P/E cycle count on retention loss speed, and t_0 and a control the impact of dwell time on retention loss speed. To determine the values for these constants, we use the non-linear least squares algorithm implemented in SciPy [32, 39] to fit SRRM to the characterization data we collected.

Figure 11 illustrates how predictions from SRRM compare with our measured characterization data. Figure 11a compares the SRRM predictions and measured values of the threshold voltage shift for cells in state P1, P2, or P3. Figure 11b compares the SRRM predictions and measured values of the change in the log of RBER. Each cross (‘x’) in the figure represents a data point, where the x-coordinate of each data point is the value predicted by SRRM, and the y-coordinate of each data point is the value measured during our characterization. The dashed line shows the perfect fit (i.e., where the predicted and measured values are equal). We observe that for both the threshold voltage shift and the change in RBER, all of the data points are very close to the perfect fit line. Thus, SRRM accurately predicts both output parameters. Overall, the percentage root mean square error ($\%RMSE$) for SRRM is 4.9%. As a comparison, if we were to predict these two output parameters using UDM [46], a previously-proposed model for retention loss in planar NAND flash memory, the average $\%RMSE$ of the predictions would be 17.8%, which is much less accurate than the predictions from our model. We conclude that SRRM is highly accurate for predicting the change in RBER and the threshold voltage shift at room temperature in 3D NAND flash memory.

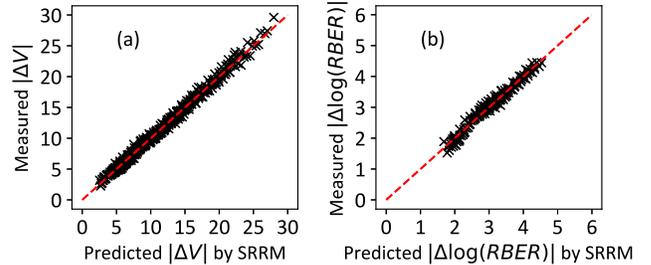


Figure 11: SRRM prediction accuracy.

5. Improving 3D NAND Reliability

Our *goal* in this section is to improve flash reliability and lifetime by developing a new mechanism that makes use of our findings (Section 3) and our new model (Section 4). Our new mechanism is called *HeatWatch*.

5.1. Observations

We make three key observations from the following three experiments that lead to the design of HeatWatch. First, we observe that *SSD write intensity and the SSD environmental temperature significantly impact flash lifetime*. The write intensity of an SSD is measured as the number of full drive writes per day. Given a fixed SSD size, the write intensity is inversely proportional to the average dwell time, thus affecting flash lifetime (Section 3.2). This is because modern SSDs use *wear-leveling* techniques to keep all flash blocks in the SSD at a similar P/E cycle count [4, 5, 15, 25]. The environmental temperature affects the flash lifetime (Section 3.3), because it changes the temperature of the underlying NAND flash memory.

Figure 12 shows the flash lifetime under different write intensities and environmental temperatures, assuming that the vendor guarantees a three-month retention time for the data, which is typical for enterprise SSDs [4, 5, 8, 9, 51]. The

SSD write intensity is shown on the x-axis in log scale. We plot the results by using separate curves for each temperature that we evaluate, which ranges from 0 °C to 70 °C.

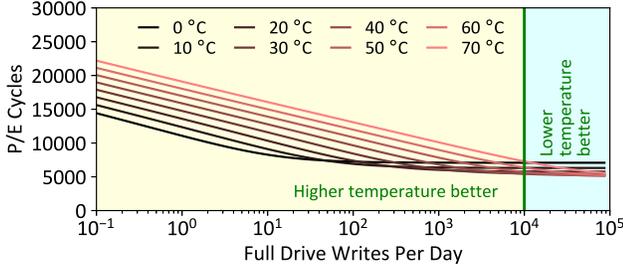


Figure 12: Change in flash lifetime due to write intensity and environmental temperature ($t_r = 3$ months).

From the figure, we see that the flash lifetime initially decreases as SSD write intensity increases, but stops decreasing at around 5,000 P/E cycles. When the write intensity is low ($< 10^4$ drive writes/day, which covers the range of write intensities of most contemporary workloads [9]), a higher temperature is desirable, as it improves both the effective dwell time and program variation and thus leads to a longer lifetime. In contrast, when the write intensity is high, a lower temperature is better due to an improved effective retention time. Note that these curves drastically shift (not shown) (1) with different assumptions about retention time, or (2) when the temperature is not constant. Thus, we find that no single temperature or temperature range is ideal.

Second, we observe that *the choice of the read reference voltage (V_{ref}) significantly affects RBER and flash lifetime.* The voltage at which the lowest RBER can be achieved is called the *optimal read reference voltage (V_{opt})*. V_{opt} changes based on conditions such as retention time and P/E cycle count. Adapting the optimal read reference voltage to these changing conditions significantly increases flash lifetime [4, 5, 8, 12, 13, 41, 55]. Based on our experiments under room temperature, Figure 13 shows how the RBER increases as the applied read reference voltage becomes further away from the optimal read reference voltage (which we refer to as the $|V_{ref} - V_{opt}|$ distance). We find that the RBER increases exponentially as the $|V_{ref} - V_{opt}|$ distance increases. We conclude, as prior works have [4, 5, 8, 13, 41], that it is important to accurately predict the optimal read reference voltage, as even a small $|V_{ref} - V_{opt}|$ distance can greatly increase the error rate (and thus hurt the flash lifetime).

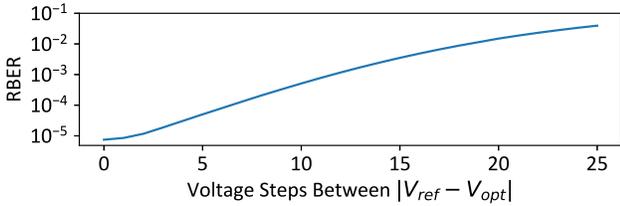


Figure 13: RBER vs. $|V_{ref} - V_{opt}|$ distance.

Third, we observe that *the optimal read reference voltage in 3D NAND flash memory changes over time, and can be accurately predicted as the value that falls in the middle of two*

neighboring threshold voltage distributions. Figure 14 shows the *measured V_{opt}* from our characterization (blue dots in the figure), and the value of V_{opt} calculated by using our URT model to predict the threshold voltage distributions of each state (orange curve), for read reference voltages V_b and V_c (see Section 2.1). The x-axis shows the retention time of the data in log scale. We see that after 4000 s of retention time, the measured optimal read reference voltages for V_b and V_c change by 8 and 11 voltage steps, respectively.⁸ Comparing the blue dots with the orange curves, we find that our URT-based V_{opt} prediction is *always* within 1 voltage step of the empirical optimal read reference voltage. We conclude that URT accurately predicts the optimal read reference voltage.

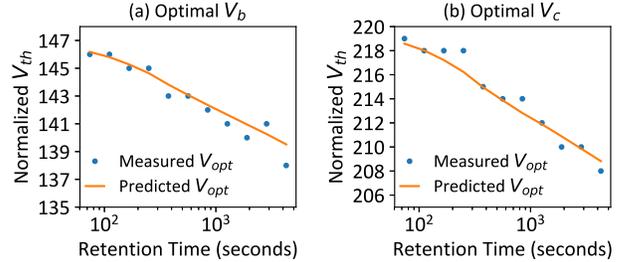


Figure 14: Measured and URT-predicted V_{opt} .

5.2. HeatWatch Mechanism

Motivated by our observations in Section 5.1, we propose *HeatWatch*, a mechanism that improves flash reliability and lifetime by predicting and applying the optimal read reference voltage using our URT model from Section 4. HeatWatch consists of (1) three *tracking components*, which monitor and efficiently record the SSD temperature, dwell time, retention time, and P/E cycle count; and (2) two *prediction components*, which compute the URT model using this tracked information to accurately predict the optimal read reference voltage for each read operation.

Tracking Component 1: SSD Temperature. Modern SSDs already contain multiple temperature sensors to prevent overheating [38, 44]. HeatWatch *efficiently* logs the readings from these sensors, which the RDTM component of URT (see Section 4.2) uses to scale the dwell time and retention time. HeatWatch records the temperature every second, and *pre-computes* the acceleration factors (AF_i) for every *logarithmic time interval i* . HeatWatch uses logarithmic intervals because the effects of dwell time and retention time are logarithmic with respect to their duration (see Section 3.2). HeatWatch uses RDTM to calculate AF_0 . For all other intervals, AF_i is calculated as a piecewise integral, by summing up the *two* most recent values of AF_{i-1} , since interval i covers *double* the amount of time as interval $i - 1$. Therefore, for each interval, HeatWatch stores the current *and* previous values of AF_i , in an *acceleration factor log*.

Tracking Component 2: Dwell Time. The self-recovery effect is dependent on the average dwell time across *multiple* P/E cycles (Section 5.1). The average dwell time is

⁸Our characterization shows that V_a does *not* change significantly based on retention time, so we do not plot it. We find that URT accurately predicts the lack of change in V_a .

determined by the workload write intensity. Thus, we use the SSD controller to (1) monitor the workload write intensity online, and (2) calculate the average dwell time for each flash block. HeatWatch does *not* need to track the variation in dwell time across different flash pages within the *same* block, as we find from our experimental characterization that the effect of page-to-page dwell time variation is negligible (Section 5.1).

HeatWatch approximates the effective dwell time by taking the average unscaled dwell time across the last 20 P/E cycles, and scaling it using RDTM. The SSD controller keeps a counter that tracks the amount of data written to the SSD, and logs the timestamps of the last 20 full drive writes. When a flash block is programmed during drive write n , the SSD controller calculates the average unscaled dwell time as the difference between the current time and the timestamp of drive write $n - 20$. Then, the SSD controller computes and stores the effective room temperature dwell time by scaling it using the aforementioned acceleration factor log.

Tracking Component 3: P/E Cycles and Retention Time. The SSD controller already records the P/E cycle count of each block to use in wear-leveling algorithms [4, 5, 15, 25]. To track the retention time of each flash block, HeatWatch simply logs the timestamp when the block is selected for programming. Then, HeatWatch calculates the effective retention time for each read operation as the difference between the program time and read time, scaled by RDTM.

Prediction Component 1: Optimizing the Read Reference Voltage. The optimal read reference voltage between two states can be predicted accurately by averaging the means of the threshold voltage distributions for each state (Section 5.1). As HeatWatch knows the P/E cycle count, programming temperature, effective dwell time, and effective retention time, it can use the URT model from Section 4 to predict the means of the threshold voltage distributions for each state, and thus the optimal read reference voltage. For each read operation, the SSD controller simply gathers all the metadata for the flash block that is to be read, and then predicts and applies the optimal read reference voltage. The information gathering and prediction happen *after* the FTL translates the logical address of the read to a physical address (see [4, 5] for detail), since the information for the flash block is indexed using the physical address.

Prediction Component 2: Fine-Tuning URT Model Parameters Online. To accommodate for chip-to-chip variation, URT learns its model parameters *online*. We initialize the URT model parameters using a set of parameters that have been learned offline, which the vendor can provide at manufacture time. Over time, URT fine-tunes its model parameters by (1) sampling a number of flash wordlines in the chip (10 in our evaluations), which are selected at random from blocks that span a range of different P/E cycles, effective dwell/retention time, and programming temperatures; (2) learning the optimal read reference voltages for the sampled flash wordlines online, using a technique similar to Retention Optimized Reading (ROR) [8]; and (3) using the sampled data to fit the fine-tuned URT model parameters for each chip, which can be done relatively easily in the firmware with little overhead [41]. The overall latency for online training

is dominated by the latency to predict the optimal read reference voltage for each wordline. In the worst case, ROR performs 300 read operations per wordline, using a different voltage step for each read. For the 10 wordlines sampled by URT model tuning, assuming a read latency of 100 μ s, the total worst-case latency of URT model tuning is 0.3 s. Note that this tuning procedure needs to be performed only infrequently (e.g., every 1,000 P/E cycles), and can be performed in the background (i.e., when the chip is idle), thus incurring negligible performance overhead.

Storage Overhead. HeatWatch needs to store three sets of information. (1) HeatWatch stores the acceleration factor for only logarithmic time intervals from 0.5 s to 1 year (26 intervals in total). HeatWatch stores the current and previous value of each acceleration factor, as described in *Tracking SSD Temperature* above. Assuming that each acceleration factor is stored as a 4 B floating-point number, the total log requires 208 B of storage. (2) HeatWatch stores the programming temperature, dwell time, and program time for each flash block. Assuming that each piece of information uses 4 B, for a 1 TB SSD with an 8 MB flash block size, HeatWatch uses 1.5 MB of storage in total to store this information. (3) HeatWatch needs a 32-bit counter, and must store the timestamps for the last 20 full disk writes (Section 3.4), which requires 84 B of storage. In total, the three sets of information require less than 1.6 MB of storage. To minimize the performance overhead of accessing this data, HeatWatch buffers the data in the on-board DRAM in the SSD controller [4, 5]. The storage overhead is very small, as a 1 TB SSD *typically* contains 1 GB of DRAM for caching [4, 5].

Latency Overhead. HeatWatch performs two operations that contribute to its latency. (1) Every second, HeatWatch updates the acceleration factor log with the latest temperature reading. This update can be done in the background, and, thus, its performance overhead is negligible. (2) HeatWatch computes the URT model during each read operation, which involves performing *only* 16 arithmetic computations in the SSD controller (Section 4). The model computation latency is negligible compared to the flash read latency ($>25 \mu$ s [27]).

5.3. Evaluation

To evaluate HeatWatch, we examine the raw bit error rate (RBER) and lifetime of four different configurations:

- *Fixed V_{ref}* , which always uses the default read reference voltage to read the data.
- *Retention-Only*, which predicts the optimal read reference voltage based on a model that considers only P/E cycle count and retention time [8, 12, 13, 41, 53, 55]. Note that this model always assumes a *fixed* retention loss speed, regardless of the dwell time and temperature.
- *HeatWatch*, our proposed mechanism to accurately predict the optimal read reference voltage by tracking dwell time and temperature and using our URT model.
- *Oracle*, which always *ideally* uses the measured optimal read reference voltage, and does *not* incur any performance overhead. Note that this is *not* implementable.

We evaluate these four configurations using 28 commonly-used real storage traces [49], which have varying write intensities. Each trace represents seven days of workload data, and

contains timestamps we can use to calculate the dwell time and retention time of each access. We simulate temperature variation over the course of a day as the superposition of a sinusoidal function and some Gaussian noise. The sinusoidal model has a mean of 35 °C, an amplitude of 15 °C, and a 1-day period, representing how the temperature changes during a daily cycle. The Gaussian noise model that we use has a standard deviation of 3 °C.

Figure 15 shows how the RBER increases with P/E cycle count for our four evaluated configurations, using a workload that appends *all 28* disk traces together to mimic an SSD that runs multiple workloads back-to-back. The dotted line shows an error correction capability (see Section 2.2) of $2 \cdot 10^{-3}$ errors per bit [4, 5]. We determine the lifetime for each configuration using the point at which the RBER intersects the error correction capability. We use *Fixed V_{ref}* , which has the highest RBER, as our baseline. From the figure, we see that *Retention-Only* reduces the RBER by 83.1%, on average across all P/E cycles, compared to the baseline, *HeatWatch* reduces the RBER by 93.5% compared to the baseline. This is very close to the average RBER improvement under *Oracle* (93.9%). *HeatWatch* significantly improves lifetime due to its RBER improvement. The lifetime with *HeatWatch* is 21,065 P/E cycles, which is $3.19\times$ and $1.29\times$ the lifetime with *Fixed V_{ref}* and *Retention-Only*, respectively. *HeatWatch* comes within only 200 P/E cycles of the *Oracle* lifetime.

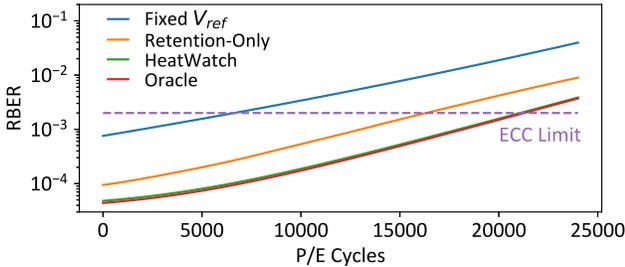


Figure 15: RBER vs. P/E cycle count.

We repeat the same experiment for each workload individually, and determine the lifetime for each workload under the four configurations, as shown in Figure 16. On average across all of our workloads, The lifetime under *Retention-Only* is $3.11\times$ the lifetime of the *Fixed V_{ref}* baseline. *HeatWatch* improves the lifetime further, with a lifetime that is $3.85\times$ the baseline lifetime. Again, this is very close to the lifetime improvement of *Oracle* ($3.89\times$).

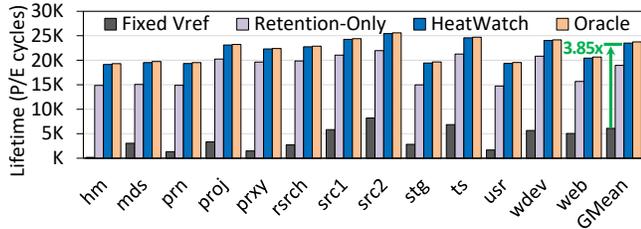


Figure 16: P/E cycle lifetime for each workload.

We conclude that by incorporating dwell time and temperature information to predict the optimal read reference voltage, *HeatWatch* improves the lifetime of 3D NAND flash memory

devices over a state-of-the-art mechanism [53], and approaches the lifetime of an ideal mechanism that has perfect knowledge of the optimal read reference voltage.

6. Related Work

To our knowledge, this paper is the first to (1) experimentally characterize and accurately model the self-recovery and temperature effects in 3D NAND flash memory; and (2) devise a mechanism that improves 3D NAND flash memory lifetime by comprehensively taking into account retention time, wearout, self-recovery, and temperature. We discuss the closely-related works.

Data Retention in NAND Flash Memory. Many prior works focus on flash memory retention loss and retention errors, and show that retention loss is the most dominant source of errors in modern NAND flash memory [4, 5, 8, 9, 19, 45, 51, 53]. These works do *not* consider the effects of self-recovery and temperature on retention loss. Our work investigates these effects through an extensive characterization of state-of-the-art 3D NAND flash memory chips.

3D NAND Flash Memory Characterization. Recent works study the error characteristics of 3D NAND flash memory, and identify differences between 3D and planar NAND flash memory due to memory cell design and architectural changes [4, 5, 18, 47, 54]. None of these works provide a detailed characterization of the impact of self-recovery, retention, P/E cycle count, and temperature on real 3D NAND flash memory.

Retention Loss Models. Our URT model is inspired by and improves upon the Unified Detrapping Metric (UDM) model [46]. There are three reasons why prior models developed for planar NAND flash memory, such as UDM, are insufficient for 3D NAND flash memory. First, 3D charge trap cells are more resilient to P/E cycling than the floating-gate cells used by planar NAND flash memory [18, 47, 54]. Thus, the PEC component in our model (Equation 7) is different from the equivalent component in UDM. Second, 3D NAND flash memory has a different activation energy than planar NAND flash memory, as we experimentally show in Section 4.2. Third, 3D NAND flash memory reliability is affected by the programming temperature, as we show in Section 4.1. Because UDM does *not* accurately capture these changes, its error rate for 3D NAND flash memory is $3.6\times$ greater than the error rate for the SRRM component of our new URT model, as we show in Section 4.3.

Improving Flash Reliability. Many prior works propose mechanisms to improve flash lifetime and reduce raw bit errors (see [4, 5] for a detailed survey). For example, flash refresh techniques limit the number of retention errors to achieve higher P/E cycle lifetime [9, 10, 40, 51]. Prior work also adjusts the read reference voltage according to P/E cycle count and retention time to reduce the RBER [8, 12, 13, 41, 53, 55]. Different from prior work, we develop a new mechanism that tracks workload write intensity and SSD temperature online and adjusts read reference voltage accordingly to improve flash lifetime. We compare our mechanism to prior mechanisms that are agnostic to these factors and show that it can significantly reduce RBER and improve flash lifetime.

7. Conclusion

We perform the first detailed *experimental* characterization of the impact of self-recovery and temperature on the reliability of 3D NAND flash memory. We find that due to significant changes in the memory design and manufacturing process, prior findings and models for planar NAND flash memory are *not* accurate for 3D NAND flash memory. We use key findings from our characterization to develop URT, a unified and accurate cell threshold voltage and raw bit error rate model that takes into account the combined effects of self-recovery, temperature, retention loss, and wearout. We develop a new mechanism, HeatWatch, that uses URT to dynamically adapt the read reference voltage to the data retention time, dwell time, SSD temperature, and wearout. We show that HeatWatch greatly reduces the raw bit error rate and improves flash lifetime. We conclude that the effects of self-recovery and temperature in 3D NAND flash memory can be accurately modeled and successfully used to improve flash reliability. We hope that our data, model, and new mechanism inspire others to develop other new mechanisms that take advantage of the self-recovery and temperature effects in 3D NAND flash memory.

Acknowledgments

We thank the anonymous shepherd and reviewers for their feedback. This work is partially supported by grants from Huawei and Seagate, and gifts from Intel and Samsung.

References

- [1] S. Arrhenius, "Über die Dissociationswärme und den Einfluß der Temperatur auf den Dissociationsgrad der Elektrolyte," *Z. Phys. Chem.*, Jul. 1889.
- [2] H. P. Belgal *et al.*, "A New Reliability Model for Post-Cycling Charge Retention of Flash Memories," in *IRPS*, 2002.
- [3] A. D. Birrell and B. J. Nelson, "Implementing Remote Procedure Calls," *TOCS*, Feb. 1984.
- [4] Y. Cai *et al.*, "Error Characterization, Mitigation, and Recovery in Flash-Memory-Based Solid-State Drives," *Proc. IEEE*, Sep. 2017.
- [5] Y. Cai *et al.*, "Errors in Flash-Memory-Based Solid-State Drives: Analysis, Mitigation, and Recovery," arXiv:1711.11427 [cs.AR], 2017.
- [6] Y. Cai *et al.*, "Vulnerabilities in MLC NAND Flash Memory Programming: Experimental Analysis, Exploits, and Mitigation Techniques," in *HPCA*, 2017.
- [7] Y. Cai *et al.*, "Read Disturb Errors in MLC NAND Flash Memory: Characterization and Mitigation," in *DSN*, 2015.
- [8] Y. Cai *et al.*, "Data Retention in MLC NAND Flash Memory: Characterization, Optimization, and Recovery," in *HPCA*, 2015.
- [9] Y. Cai *et al.*, "Flash Correct and Refresh: Retention Aware Management for Increased Lifetime," in *ICCD*, 2012.
- [10] Y. Cai *et al.*, "Error Analysis and Retention-Aware Error Management for NAND Flash Memory," *Intel Technol. J.*, May 2013.
- [11] Y. Cai *et al.*, "Error Patterns in MLC NAND Flash Memory: Measurement, Characterization, and Analysis," in *DATE*, 2012.
- [12] Y. Cai *et al.*, "Threshold Voltage Distribution in MLC NAND Flash Memory: Characterization, Analysis, and Modeling," in *DATE*, 2013.
- [13] Y. Cai *et al.*, "Program Interference in MLC NAND Flash Memory: Characterization, Modeling, and Mitigation," in *ICCD*, 2013.
- [14] Y. Cai *et al.*, "Neighbor-Cell Assisted Error Correction for MLC NAND Flash Memories," in *SIGMETRICS*, 2014.
- [15] L.-P. Chang, "On Efficient Wear Leveling for Large-Scale Flash-Memory Storage Systems," in *SAC*, 2007.
- [16] P. C. Y. Chen, "Threshold-Alterable Si-Gate MOS Devices," *TED*, May 1977.
- [17] R. Chen *et al.*, "DHeating: Dispersed Heating Repair for Self-Healing NAND Flash Memory," in *CODES+ISSS*, 2013.
- [18] B. Choi *et al.*, "Comprehensive Evaluation of Early Retention (Fast Charge Loss Within a Few Seconds) Characteristics in Tube-Type 3-D NAND Flash Memory," in *VLSIT*, 2016.
- [19] W. Choi *et al.*, "Exploiting Data Longevity for Enhancing the Lifetime of Flash-Based Storage Class Memory," in *SIGMETRICS*, 2017.
- [20] C. M. Compagnoni *et al.*, "Investigation of the Threshold Voltage Instability After Distributed Cycling in Nanoscale NAND Flash Memory Arrays," in *IRPS*, 2010.
- [21] L. Crippa and R. Micheloni, "3D Charge Trap NAND Flash Memories," in *3D Flash Memories*. Dordrecht, Netherlands: Springer, 2016.
- [22] B. Eitan, "Non-Volatile Semiconductor Memory Cell Utilizing Asymmetrical Charge Trapping," U.S. Patent 5,768,192, 1998.
- [23] J. Elliott and J. Jeong, "Advancements in SSDs and 3D NAND Reshaping Storage Market," Keynote presentation at *Flash Memory Summit*, 2017.
- [24] A. Fukami *et al.*, "Improving the Reliability of Chip-Off Forensic Analysis of NAND Flash Memory Devices," *Digital Investigation*, Mar. 2017.
- [25] E. Gal and S. Toledo, "Algorithms and Data Structures for Flash Memories," *CSUR*, Jun. 2005.
- [26] A. Grossi *et al.*, "Reliability of 3D NAND Flash Memories," in *3D Flash Memories*. Springer, 2016.
- [27] L. M. Grupp *et al.*, "The Bleak Future of NAND Flash Memory," in *FAST*, 2012.
- [28] JEDEC Solid State Technology Assn., *Method for Developing Acceleration Models for Electronic Component Failure Mechanisms*, Publication JESD91A, 2003.
- [29] JEDEC Solid State Technology Assn., *Solid-State Drive (SSD) Requirements and Endurance Test Method*, Publication JESD218, 2010.
- [30] JEDEC Solid State Technology Assn., *Electrically Erasable Programmable ROM (EEPROM) Program/Erase Endurance and Data Retention Stress Test*, Publication JESD22-A117C, 2011.
- [31] W. Jeong *et al.*, "A 128 Gb 3b/Cell V-NAND Flash Memory with 1 Gb/s I/O Rate," *JSSC*, Jan. 2016.
- [32] E. Jones *et al.*, "SciPy: Open Source Scientific Tools for Python," <http://www.scipy.org/>.
- [33] D. Kahng and S. M. Sze, "A Floating Gate and Its Application to Memory Devices," *Bell Syst. Tech. J.*, Aug. 1967.
- [34] D. Kang *et al.*, "7.1 256Gb 3b/Cell V-NAND Flash Memory with 48 Stacked WL Layers," in *ISSCC*, 2016.
- [35] R. Katsumata *et al.*, "Pipe-Shaped BiCS Flash Memory with 16 Stacked Layers and Multi-Level-Cell Operation for Ultra High Density Storage Devices," in *VLSIT*, 2009.
- [36] C. Kim *et al.*, "A 512Gb 3b/Cell 64-Stacked WL 3D V-NAND Flash Memory," in *ISSCC*, 2017.
- [37] S. Lee *et al.*, "Lifetime Management of Flash-Based SSDs Using Recovery-Aware Dynamic Throttling," in *FAST*, 2012.
- [38] T. Lenny, "The Maturity of NVM Express™," in *Flash Memory Summit*, 2014.
- [39] K. Levenberg, "A Method for the Solution of Certain Non-Linear Problems in Least Squares," *Quarterly of Applied Mathematics*, 1944.
- [40] Y. Luo *et al.*, "WARM: Improving NAND Flash Memory Lifetime With Write-Hotness Aware Retention Management," in *MSSST*, 2015.
- [41] Y. Luo *et al.*, "Enabling Accurate and Practical Online Flash Channel Modeling for Modern MLC NAND Flash Memory," *JSAC*, 2016.
- [42] Y. Luo *et al.*, "Exploiting Self-Recovery and Temperature Awareness to Improve 3D NAND Flash Memory Reliability," Carnegie Mellon Univ., SAFARI Research Group, Tech. Rep. 2018-001, 2018.
- [43] T. Marquart, "Practical Approach to Determining SSD Reliability," in *Flash Memory Summit*, 2015.
- [44] J. Meza *et al.*, "A Large-Scale Study of Flash Memory Failures in the Field," in *SIGMETRICS*, 2015.
- [45] N. Mielke *et al.*, "Bit Error Rate in NAND Flash Memories," in *IRPS*, 2008.
- [46] N. Mielke *et al.*, "Recovery Effects in the Distributed Cycling of Flash Memories," in *IRPS*, 2006.
- [47] K. Mizoguchi *et al.*, "Data-Retention Characteristics Comparison of 2D and 3D TLC NAND Flash Memories," in *IMW*, 2017.
- [48] V. Mohan *et al.*, "How I Learned to Stop Worrying and Love Flash Endurance," *HotStorage*, 2010.
- [49] D. Narayanan *et al.*, "Write Off-Loading: Practical Power Management for Enterprise Storage," *TOS*, Nov. 2008.
- [50] I. Narayanan *et al.*, "SSD Failures in Datacenters: What? When? And Why?" in *SYSTOR*, 2016.
- [51] Y. Pan *et al.*, "Quasi-Nonvolatile SSD: Trading Flash Memory Nonvolatility to Improve Storage System Performance for Enterprise Applications," in *HPCA*, 2012.
- [52] N. Papandreou *et al.*, "Effect of Read Disturb on Incomplete Blocks in MLC NAND Flash Arrays," in *IMW*, 2016.
- [53] N. Papandreou *et al.*, "Using Adaptive Read Voltage Thresholds to Enhance the Reliability of MLC NAND Flash Memory Systems," in *GLSVLSI*, 2014.
- [54] K. Park *et al.*, "Three-Dimensional 128 Gb MLC Vertical NAND Flash Memory with 24-WL Stacked Layers and 50 MB/s High-Speed Programming," *JSSC*, Jan. 2015.
- [55] T. Parnell *et al.*, "Modelling of the Threshold Voltage Distributions of Sub-20nm NAND Flash Memory," in *GLOBECOM*, 2014.
- [56] SAFARI Research Group, "3D NAND Flash Memory Characterization Data," <https://github.com/CMU-SAFARI/HeatWatch>.
- [57] Samsung Electronics Co., Ltd., "Samsung V-NAND Technology," white Paper: <http://tinyurl.com/mprrdxs>, 2014.
- [58] B. Schroeder *et al.*, "Flash Reliability in Production: The Expected and the Unexpected," in *FAST*, 2016.
- [59] S. Seabold and J. Perktold, "Statsmodels: Econometric and Statistical Modeling with Python," in *SciPy*, 2010.
- [60] Serial ATA International Organization, *Serial ATA Revision 3.3 Specification*, 2016.
- [61] Toshiba Corp., "3D Flash Memory: Scalable, High Density Storage for Large Capacity Applications," <http://www.toshiba.com/taec/adinfo/technologymoves/3d-flash.jsp>, 2017.
- [62] H. A. Wegener *et al.*, "The Variable Threshold Transistor, A New Electrically-Alterable, Non-Destructive Read-Only Storage Device," in *IEDM*, 1967.
- [63] E. H. Wilson *et al.*, "ZombieNAND: Resurrecting Dead NAND Flash for Improved SSD Longevity," in *MASCOTS*, 2014.
- [64] Q. Wu *et al.*, "A First Study on Self-Healing Solid-State Drives," in *IMW*, 2011.
- [65] Q. Wu *et al.*, "Exploiting Heat-Accelerated Flash Memory Wear-Out Recovery to Enable Self-Healing SSDs," in *HotStorage*, 2011.
- [66] J. H. Yoon, "3D NAND Technology: Implications to Enterprise Storage Applications," in *Flash Memory Summit*, 2015.
- [67] J. Yoon, "Flash Quality Management in Enterprise Storage Applications," in *Flash Memory Summit*, 2014.