electronics

MDPI

Article

REO: Revisiting Erase Operation for Improving Lifetime and Performance of Modern NAND Flash-Based SSDs †

Beomjun Kim and Myungsuk Kim *

School of Computer Science and Engineering, Kyungpook National University, Daegu 37224, Republic of Korea; beomjun0816@knu.ac.kr

* Correspondence: ms.kim@knu.ac.kr

† This article is a revised and expanded version of a paper entitled AERO: Adaptive Erase Operation for Improving Lifetime and Performance of Modern NAND Flash-Based SSDs, which was presented at ASPLOS '24: 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, La Jolla, CA, USA, 27 April–1 May 2024.

Abstract: This work investigates a new erase scheme in NAND flash memory to improve the lifetime and performance of modern solid-state drives (SSDs). In NAND flash memory, an erase operation applies a high voltage (e.g., >20 V) to flash cells for a long time (e.g., >3.5 ms), which degrades cell endurance and potentially delays user I/O requests. While a large body of prior work has proposed various techniques to mitigate the negative impact of erase operations, no work has yet investigated how erase latency and voltage should be set to fully exploit the potential of NAND flash memory; most existing techniques use a fixed latency and voltage for every erase operation, which is set to cover the worst-case operating conditions. To address this, we propose Revisiting Erase Operation, (REO) a new erase scheme that dynamically adjusts erase latency and voltage depending on the cells' current erase characteristics. We design REO by two key apporaches. First, REO accurately predicts such near-optimal erase latency based on the number of fail bits during an erase operation. To maximize its benefits, REO aggressively yet safely reduces erase latency by leveraging a large reliability margin present in modern SSDs. Second, REO applies near-optimal erase voltage to each WL based on its unique erase characteristics. We demonstrate the feasibility and reliability of REO using 160 real 3D NAND flash chips, showing that it enhances SSD lifetime over the conventional erase scheme by 43% without change to existing NAND flash chips. Our system-level evaluation using eleven real-world workloads shows that an REO-enabled SSD reduces average I/O performance and read tail latency by 12% and 38%, respectivley, on average over a state-of-the-art technique.

Keywords: solid-state drives (SSDs); NAND flash memory; erase operation; SSD lifetime; I/O performance

check for updates

Academic Editor: Spyridon Nikolaidis

Received: 5 January 2025
Revised: 3 February 2025
Accepted: 10 February 2025
Published: 13 February 2025

Citation: Kim, B.; Kim, M. REO: Revisiting Erase Operation for Improving Lifetime and Performance of Modern NAND Flash-Based SSDs. *Electronics* **2025**, *14*, 738. https://doi.org/10.3390/electronics14040738

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

NAND flash memory has emerged as the dominant memory technology in the design of modern storage systems. NAND flash-based solid-state drives (SSDs) offer many advantages over traditional hard disk drives (HDDs), such as higher performance, low power consumption, smaller size, and larger storage capacity; individual SSDs can hold several tens of terabytes of data [1–4]. Despite the development of various novel non-volatile memory technologies (e.g., [5–8]), NAND flash memory is expected to be the key technology for storage systems, addressing the growing capacity requirements of modern data-intensive applications.

The efficiency of the erase operation performed by NAND flash chips significantly affects SSD lifetime and I/O performance for two key reasons. First, the high voltage

required for the erase operation causes physical damage to the flash memory cells. After a certain number of repetitive program and erase (P/E) cycles, a flash cell cannot reliably store data, thereby limiting the SSD's lifetime. Second, erase latency is significantly higher (e.g., 3.5 ms [9,10]) than read and write latencies (e.g., 40 µs and 350 µs, respectively [9]). From the lifetime perspective, such a long latency leads an erase operation to have a more significant impact on flash cell's endurance compared to a program operation that also requires a high voltage to target cells [11]. From an I/O performance perspective, an erase operation can result in substantial delays in processing I/O requests, potentially lasting several milliseconds, thereby exacerbating the tail latency of SSDs [12,13].

An erase operation on modern SSDs often requires *multiple erase loops*, further increasing the performance/lifetime impact of erase operations. As a flash cell experiences more program/erase (P/E) cycles, it becomes more difficult to complete the erase operation [11,14,15]. Therefore, an erase operation with the default erase voltage may fail to sufficiently erase every cell in the block, which we call an *erase failure*. To ensure data reliability, modern NAND flash memory commonly employs the Incremental Step Pulse Erasure (ISPE) scheme [16]; when an erase failure occurs, the ISPE scheme *retries* an erase loop with incrementally higher voltages until it can successfully erase all the cells in the block. For example, our characterization study using 160 real 3D triple-level cell (TLC) NAND flash chips reveals that every erase operation needs at least two erase loops (up to five loops) after the target block experiences 2K P/E cycles.

Even though a large body of prior work [12,13,16–31] has investigated various optimizations to alleviate the negative impact of erase operations, no work has yet investigated *how erase latency should be set* to fully exploit the potential of NAND flash memory. Specifically, most previous approaches employ a *fixed* latency for *every* erase operation, which is decided by the manufacturers at design time based on the *worst-case* operating conditions. However, similar to other memory technologies such as DRAM, modern NAND flash memory also exhibits high process variation, which introduces significant differences in physical/electrical characteristics across flash cells [32–40]. For example, our real-device characterization study in Section 5 demonstrates that it is possible to completely and reliably erase a majority of blocks (e.g., 79–90%) with much lower latency (e.g., by 17–29%) than the default erase latency under many operating conditions. This observation indicates that most flash cells frequently suffer from unnecessary damage induced by excessive erase operation, which prevents modern SSDs from taking full advantage of their potential lifetime and I/O performance.

Our goal in this work is to improve the lifetime and performance of modern SSDs by mitigating the negative impact of erase operations. To achieve this, we introduce REO, an innovative block erasure mechanism specifically designed for modern 3D NAND flash memory. Our technique is developed based on precise device modeling obtained by comprehensive characterization using real 3D TLC NAND flash chips. To this end, we propose Revisiting Erase Operation (REO), a new block erasure mechanism for NAND flash memory that is redesigned based on characterizations of real 3D TLC NAND flash chips.

Our REO attempts to optimize the erase operation by dynamically adjusting erase latency and voltage to reliably erase the target cells, depending on the current erase characteristics of the cells. We implement the optimal erase conditions by two simple but challenging approaches. The first approach aims to accurately predict the optimal erase latency for a block in modern NAND flash memory. Due to the high process variation, the optimal erase latency for individual flash blocks is quite different. Our real-device characterization results from 160 real 3D TLC NAND flash chips (Section 5) show high erase latency variations even across blocks with the same PEC, e.g., a standard deviation of 2.7 ms in the erase latency across blocks with 3.5 K PEC. Therefore, the existing erase

mechanism cannot prevent most blocks from being over-erased. However, due to high process variation, flash blocks have different erase characteristics, making it challenging to accurately estimate the optimal erase latency of individual blocks under different operating environments (e.g., diverse P/E cycles).

To overcome such a challenge, REO introduces *Fail-bit-count-based Erase Latency Prediction (FELP)* that accurately predicts the near-optimal latency for an erase loop based on the number of fail bits that remain in the previous loop. At the end of each erase loop, the ISPE scheme senses all the cells in the target block simultaneously and counts the number of *fail bits*, i.e., the number of insufficiently erased cells, so as to perform the next loop if the fail-bit count is larger than a threshold. We find that the fail-bit count can be an accurate proxy for the minimum latency of the *next* loop, as the more sufficiently the cells are erased, the lower the fail-bit count. We construct a model between the fail-bit count in an erase loop and the optimal erase latency required for the next loop, thereby allowing REO to safely reduce erase latency depending on the block's characteristics.

The second approach is to apply optimal erase voltage to individual WLs in a block. Since the existing erase mechanism is performed in a block granularity, all WLs in a block experience the same erase voltage. However, due to the the high process variation during 3D manufacturing, WLs in a single 3D flash block have quite different erase speeds. Our real-device characterization study reveals that the erase speed of a fast WL is 40% higher than that of the slowest WL, on average. It strongly indicates that flash cells on most WLs experience unnecessary excessive stress during erase operation. In order to mitigate the over-erase stress, REO introduces *Selectivve Erase Votage Adjustment (SEVA)* that applies the near-optimal erase voltage for each WL by reflecting the erase speed difference between WLs in a block. Based on a precise model for erase speed characteristics, we redesign the NAND peripheral circuit condition for the erase mechanism to differentiate the WL gate voltage during the erase operation.

We further optimize REO to maximize its effectiveness. We leverage the high error-correction capability of modern SSDs to further reduce erase latency without compromising reliability. To cope with the low reliability of NAND flash memory, modern SSDs commonly adopt error-correction codes (ECCs), which leads to a large ECC-capability margin in many cases [14,41]. Aggressive erase latency reduction would inevitably cause insufficient erasure of some flash cells, potentially incurring more bit errors. To ensure data reliability, REO carefully reduces erase latency for certain operating conditions we find via extensive real-device characterization.

REO provides high lifetime and performance benefits with small overheads. REO requires only small changes to existing SSD firmware, controllers, or NAND flash chips, thereby achieving high applicability and practicality. Our real-device characterization and system-level evaluation with a state-of-the-art SSD simulator [42] show that REO enhances SSD lifetime by 13% and reduces the 99.9999th percentile read latency by 38% on average compared to state-of-the-art techniques [16,29–31]. Furthermore, REO also significantly enhances the average I/O performance because it can effectively reduce read-retry procedures during read operation.

The key contributions of this work are as follows:

- We introduce REO, a new block erasure mechanism that dynamically adjusts the erase latency and voltage based on varying erase characteristics of the target flash blocks.
- We validate the feasibility and reliability of REO via extensive characterization of real 3D NAND flash chips.
- We evaluate the effectiveness of REO using real-world workloads, showing large lifetime and performance benefits over state-of-the-art techniques [16,29–31].

2. Background

We provide a brief background on NAND flash memory necessary to understand the rest of this paper.

2.1. NAND Flash Basics

NAND Flash Organization: The NAND flash memory consists of flash cells, the basic unit of storing data, and peripheral circuits, which support flash commands such as read, write, and erase. Individual flash cells are organized into a hierarchical structure in an NAND flash chip (or die). Each flash chip has several planes and hundreds to thousands of blocks within each plane. Figure 1 depicts a typical flash block organization. The block is represented as matrices with rows and columns composed of flash cells. These horizontal rows, known as wordlines (WLs), connect the flash cells' control gates, whereas the vertical columns, known as bitlines (BLs), connect the cells' drain and source terminals. In our example, there are m WLs in the flash block, and each WL consists of n flash cells (i.e., n BLs). The flash cells on the same WL share the common WL_i gate. Therefore, when a WL is activated, the same voltage is applied to all cells of the target WL, allowing for simultaneous read and write operations across all cells on the WL.

Data Storage Mechanism: A flash cell stores bit data as a function of its threshold voltage (V_{TH}) level, which highly depends on the amount of charge in the cell's charge trap layer; the more electrons in the charge trap layer, the higher the cell's V_{TH} level. Depending on the amount of electrons in the cell's charge trap layer, the flash cell works as an off switch or an on switch under a given control gate voltage (i.e., WL gate voltage), thus effectively storing bit data. For example, we can assign a '0' state when the flash cell has a high V_{TH} and a '1' state when the flash cell has a low V_{TH}.

NAND Flash Operation: There are three basic operations to access or modify the data stored in NAND flash memory: (i) program, (ii) read, and (iii) erase operations. The program operation, which increases V_{TH} of selected flash cells, transfers electrons from the substrate into the charge trap layer of the selected flash cells using FN tunneling [43] by applying a high voltage (>20 V) to WL gates. As a set of flash cells are connected to a single WL, NAND flash memory writes data at *page* granularity (e.g., 16 KiB) On the other hand, to erase programmed cells, a high voltage (>20 V) is applied to the substrate (while WL gates are set to a low voltage such as 0V) to remove electrons from the charge trap layer, which decreases V_{TH} of the flash cells. Since the program operation can change the bit value of a flash cell only from '1' to '0', all the flash cells of a page should be erased to program data on the page (*erase-before-program*). The unit of erase operations is a *block*, which causes the erase latency `tBERS` to be much longer (e.g., 3–5 ms) than the program latency `tPROG` (e.g., 200–700 µs).

To read the stored data from flash cells, the V_{TH} level of the flash cells on the selected WL is sensed by using a read reference voltage, V_{REF}. In Figure 1, when WL_k is selected as the target WL for read, if V_{TH} of the target i-th flash cell in WL_k is higher than V_{REF}, the i-th flash cell turns off (i.e., operates as an off switch), so the cell current of BL_i is blocked (i.e., the flash cell is identified as '0'). (Since other WLs (e.g., WL_{k+1} or WL_{k-1}) should not affect the read operation of WL_k, all the flash cells in other WLs should behave like pass transistors, therefore, their gate voltage is set to V_{READ} (>6 V), which is much higher than the highest V_{TH} value of a flash cell.) On the other hand, if the V_{TH} of the i-th flash cell is lower than V_{REF}, the i-th flash cell turns on, so the cell current can flow through BL_i (i.e., the flash cell is identified as '1'). By checking BL_i's current from the selected WL_k, the stored data are read out to the page buffer.

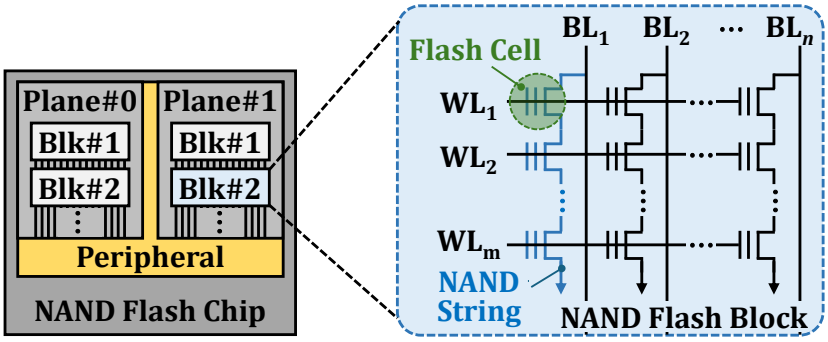


Figure 1. An organizational overview of NAND flash memory [44].

Multi-level Cell Flash Memory: To overcome the *cost per bit* challenge in flash memory, multi-leveling techniques have been widely used. Figure 2 illustrates V_{TH} distributions for 2^m-state NAND flash memory, which stores m bits within a single flash cell by using 2^m distinct V_{TH} states (i.e., m is 2 and 3 for MLC and TLC, respectively). As m is increased to store more bits within a flash cell, more V_{TH} states should be packed into the limited V_{TH} window, which is determined at the flash design time. In higher m-bit NAND flash memory, a V_{TH} margin (i.e., a gap between two neighboring V_{TH} states) inevitably becomes narrower, as shown in Figure 2a,b. A narrow V_{TH} margin makes NAND flash memory more vulnerable to various noise effects (i.e., two neighboring V_{TH} states are more likely to be overlapped), significantly degrading the flash reliability. For example, the MLC flash memory can tolerate up to 3000 program and erase (P/E) cycles, while the TLC flash memory can tolerate only 1000 P/E cycles. Therefore, more careful management is required for multi-level flash memory to form a finer V_{TH} state.

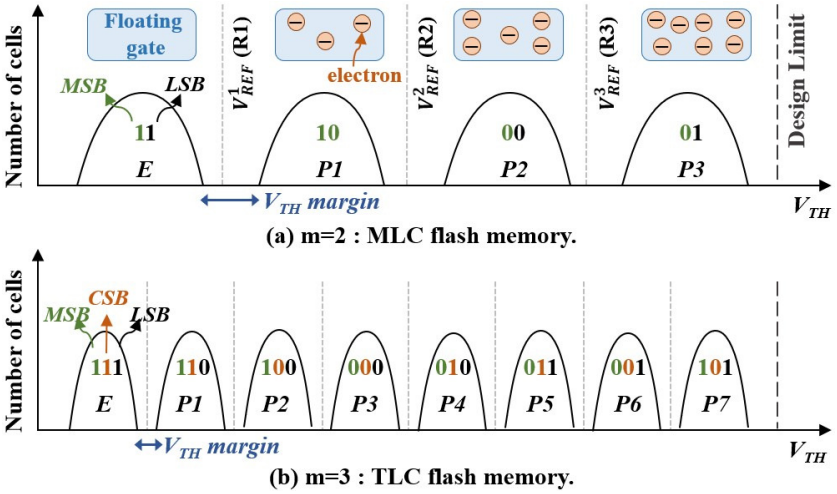


Figure 2. V_{TH} distributions of 2^m-state multi-level cell NAND flash memory.

2.2. Organizational Basics of 3D Vertical NAND Flash Memory

The 3D NAND flash memory [45] enables the continuous growth in the flash capacity by vertically stacking the memory cell to overcome various technical challenges in scaling 2D NAND flash memory. For example, 2D NAND flash memory technologies had encountered the fundamental limits to scaling below the 10 nm process technology [46] because of the low device reliability (due to severe cell-to-cell interference) and high manufacturing complexity (e.g., high-resolution patterning). However, since 3D NAND flash memory can integrate more memory cells by exploiting the vertical dimension even with a lower-resolution patterning technology, the flash capacity can be successfully increased by 50% annually while avoiding reliability degradation.

Compared to the conventional 2D NAND flash memory, there are two significant innovations in 3D NAND flash memory: *an architectural innovation* and *a material innovation*. Figure 3 illustrates the organizational difference between 2D and 3D NAND flash memory in the NAND flash block. Firstly, 2D NAND flash memory has a 2D matrix structure in which four WLs and three BLs intersect at 90 degrees, while 3D NAND flash memory has a cube-like 3D structure. The 3D NAND flash block consists of four vertical layers (v-layers) along the y-axis. Each v-layer consists of four vertically stacked WLs distinguished by select-line (SSL and GSL) transistors. (In NAND flash memory, there are two select transistors at the top and bottom of a BL, which compose the source select line (SSL) and ground select line (GSL) of a block, respectively. By applying proper voltages to the SSL and GSL of a block, we can activate the block for flash operations.) In the other perspective, the 3D NAND flash block consists of four horizontal layers (h-layers) composed of four WLs. By increasing the number of v-layers of 3D NAND flash memory (i.e., stacking more h-layers along the z-axis), the total number of WLs in a flash block is effectively increased. This scalability advantage enables 3D NAND flash memory to continuously increase its capacity by breaking through the manufacturing limit (e.g., lithography and patterning).

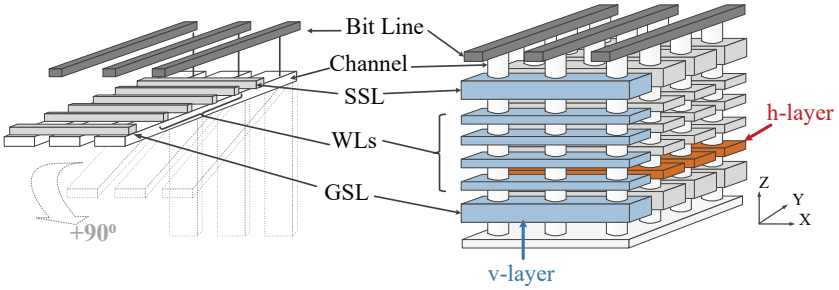


Figure 3. Illustration of organizational difference between 2D and 3D NAND flash memory.

Another key innovation in 3D NAND flash memory is a change in the type of NAND cells where charge is stored. Figure 4 shows a new flash cell structure called a charge trap (CT) type flash cell, which is the most significant difference compared to the 2D NAND flash memory. In a 3D NAND flash cell, electric charges are stored in the charge trap layer (non-conductive material) instead of the floating gate (conductor) of a 2D NAND flash cell.

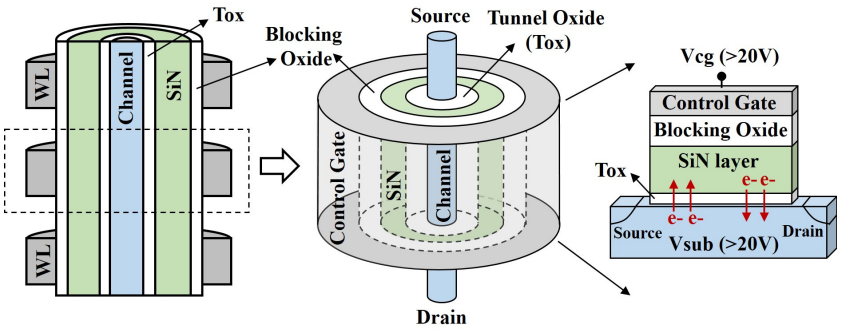


Figure 4. Schematic diagram of the flash cell of 3D NAND flash memory.

The working principles of both cell types are similar; however, a CT-type cell significantly reduces electrostatic interference between neighboring cells as the trap layer is an insulator (usually silicon nitride, SiN). As shown in Figure 4, the poly-silicon channel is surrounded by a three-layered structure composed of tunnel oxide, charge trap layer (SiN layer), and blocking oxide. Then, the control gates are formed by a poly-silicon horizontal layer (or metal-like composite of tungsten) wrapping the vertical channel.

2.3. The 3D NAND Manufacturing Process

Figure 5 illustrates an organization of a vertical structure in 3D NAND flash memory with a cross-sectional view (along the x-z plane) and a top-down view (along the x-y plane). The stacked WLs are connected by channel holes formed by the etching process [47] at the early stage of the 3D NAND flash memory manufacturing process. The etching process is one of the most crucial steps in the manufacturing process of 3D NAND flash memory, as it is essential for its revolutionary structure.

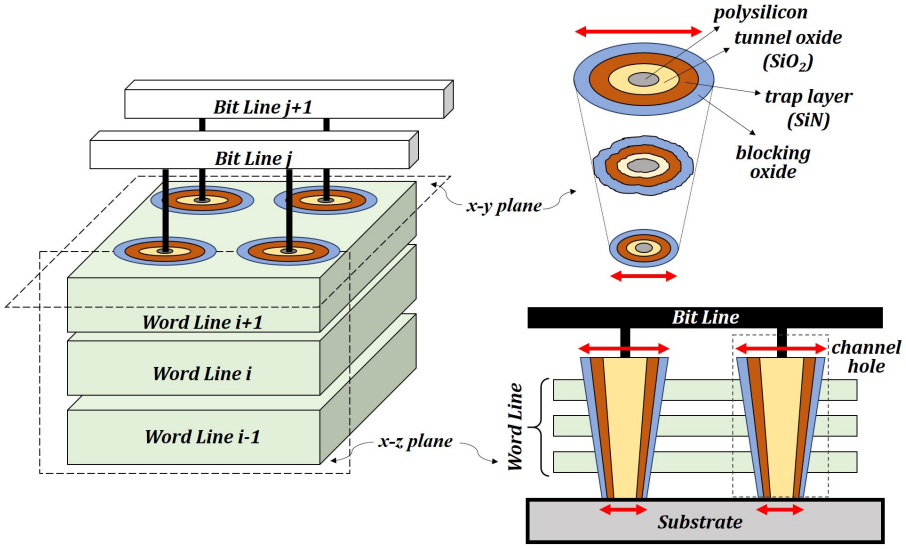


Figure 5. Schematic diagram of a 3D NAND flash cell and its operations.

Conceptually, each channel hole should have a uniform structure among all the WLs (i.e., every x-y plane and x-z plane). However, as shown in Figure 5, channel holes have structural variations depending on their vertical locations. This is because, during the etching process, the channel holes closer to the substrate are less exposed to the etchant ions, resulting in less etching. Therefore, the diameter of channel holes in the bottom layer is narrower than that of the top-most layers. Furthermore, the shape of channel holes in some layers is ellipse or rugged (i.e., not a circle shape) due to complex etchant fluid dynamics with a vertical location. These structural variations give every WL in a 3D NAND flash block different electrical characteristics (i.e., intra-block process variation).

2.4. NAND Flash Reliability

NAND flash memory is highly error-prone due to its imperfect physical characteristics. A flash cell leaks its charge (i.e., its V_{TH} level decreases) over time, which is called retention loss. Reading or programming cells slightly increases the V_{TH} levels of other cells in the same block (e.g., read/program disturbance [40,48–53]). If a cell's V_{TH} level shifts beyond the V_{REF} values (i.e., to adjacent V_{TH} ranges corresponding to different bit values), reading the cell causes a bit error.

There are two major factors that significantly increase the raw bit-error rate (RBER) of NAND flash memory. First, the high voltage used in program/erase operations physically damages flash cells, making the cells more error-prone. Second, the MLC technique increases RBER because packing more V_{TH} states within a limited V_{TH} window narrows the margin between adjacent V_{TH} states, as shown in Figure 2.

To ensure data reliability, it is common practice to employ strong error-correction codes (ECCs). ECCs store redundant bits called ECC parity, which enables detecting and correcting raw bit errors in the codeword. To cope with the high RBER of modern NAND

flash memory, modern SSDs use sophisticated ECCs that can correct several tens of raw bit errors per 1 KiB data (e.g., low-density parity-check (LDPC) codes [54]).

2.5. V_{REF} Adjustment Techniques to Handle Read Errors: Read-Retries

When an ECC engine fails to correct raw bit errors in a page, the flash controller adjusts V_{REF} values (i.e., a read-retry procedure) so that the number of raw bit errors from the subsequently sensed page can be reduced below the correction capability of the ECC engine. To facilitate the read-retry procedure, flash manufacturers often employ a predetermined sequence of V_{REF} values. When the current V_{REF} values fail to correct errors, the next V_{REF} values in the sequence (following the current V_{REF} values) are selected for re-sensing the failed page. As raw bit errors of flash memory become higher, read-retry occurs more frequently and often necessitates multiple iterations along the V_{REF} sequence, which results in a decline in read performance. In general, the page-read latency tREAD can be formulated as follows:

$$\texttt{tREAD} = (\texttt{tR} + \texttt{tDMA} + \texttt{tECC}) \times (N_{\text{RR}} + 1) \quad (1)$$

where `tR`, `tDMA`, `tECC`, and N_{RR} are the latencies of sensing the page data, transferring the sensed data from the chip to the SSD controller, decoding the data with the ECC engine, and the number of read-retries, respectively. (To retrieve the stored data, at least one read operation (called default read) is performed even through read-retry does not occur. Therefore, +1 should be included in calculating the read latency.) To mitigate the performance penalty associated with multiple read-retries, it is essential to reduce N_{RR} by making raw bit errors low.

2.6. Negative Impact of Erase Operation on I/O Performance

In modern flash-based SSDs, the I/O request latency can be severely affected when a read request from the host conflicts with an ongoing read, write, or erase operation that is currently being performed. First, a read request can be delayed due to ongoing write (about 1 ms) or erase (more than 3.5 ms and up to 20 ms) operations with relatively long response times. Such conflict can frequently occur because of the unique management tasks in modern SSDs, such as garbage collection or wear-leveling [55]. In particular, frequent GC is regarded as the root cause of the delay in SSD read response time [23,56]. Unlike HDDs, NAND flash memory does not support data overwriting, so free blocks where all data were erased must be secured before performing data write operations. If there are no free blocks, only valid data are copied from target blocks and collected into one new block, and then an erase operation should be performed on the target blocks to create free blocks. Therefore, GC requires a very long execution time (more than 100 ms). If a read operation conflicts with GC, the read requests cannot avoid a long delay until GC is completed, leading to abnormally long tail latency. Second, the read request can also be delayed by ongoing read operations. When the previous read operation suffers from multiple read-retries due to high RBER, the read request cannot be issued until the ongoing read-retries are completed, which significantly degrades I/O performance.

3. Motivation

We discuss (i) the negative effect from erase operations and (ii) limitations of the existing erase mechanism. Table A1 in Appendix A summarizes new terminologies defined in this work.

3.1. Negative Impact of Erase Operations

The erase operation has a substantial impact on both the lifetime and I/O performance of SSD. First, an erase operation is the primary determinant of SSD lifetime. The high voltage used in program and erase operations damages flash cells, which makes a block *unusable* after experiencing a certain number of program and erase (P/E) cycles (e.g., 5K P/E cycles [15]). Electrical stress during P/E cycles could have a detrimental effect on the tunnel oxide layer of flash cells. As the amount of damage to the tunnel oxide layer increases, flash cells eventually wear out and can no longer reliably store data.

Figure 6a,b show how erase and program operations affect the lifetime of a flash block in 2D 2x nm TLC and 3D 48-layer TLC NAND flash memory, respectively. The ordinate of Figure 6 demonstrates the percentage by which erase and write operations contribute to flash memory wear, respectively. When flash memory experiences repetitive program (or write) and erase operations (called P/E cycles), a high operating voltage in erase and write operations damages flash cells. Such damage results in flash memory wear, thus increasing RBER, which is called endurance error. The percentage of endurance error expresses how much the erase and write operations increase the endurance errors. Since the erase latency (e.g., 3.5 ms) is much longer than the program latency (e.g., 350 µs [9]), the endurance impact of an erase operation was significantly larger than that of a program operation in NAND flash memory. The endurance stress of erase operations was responsible for almost 80% of the total stress of flash cells in 3D NAND flash memory. Note that the impact of erase operations on flash endurance is about 33% higher in 3D NAND flash memory over 2D NAND flash memory. Since the amount of damage to the tunnel oxide layer increases exponentially as the erase voltage increases [57], lowering the erase voltage is essential in improving the SSD lifetime.

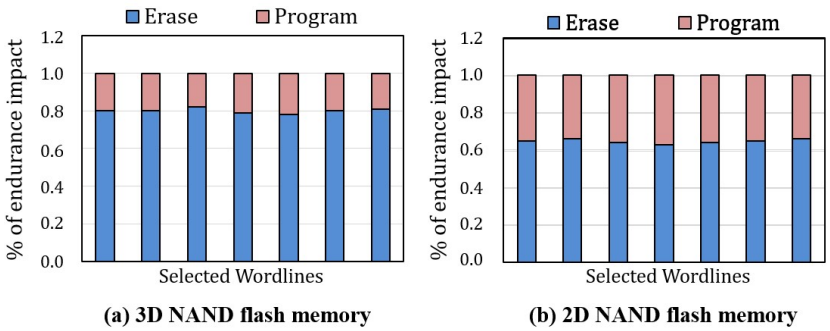


Figure 6. Endurance impact of erase and program operations.

Second, the long erase latency often increases the tail latency of user reads significantly, which is critical for modern applications such as data center applications [12,13]. In modern SSDs, the effect of erase operations on average I/O performance is relatively trivial [13], because the erase operations are performed much less frequently than read or program operations. For example, a block in modern 3D NAND flash memory consists of more than 2K pages [10,58,59], so one erase operation occurs after at least 2K page writes (and even more page reads potentially). However, when the read request conflicts with an erase operation under heavy user writes, it can be delayed or blocked for an order of magnitude longer time, thus significantly degrading the I/O performance of SSDs.

3.2. Incremental Step Pulse Erasure (ISPE) and Its Limitations

ISPE in modern SSDs: The lifetime and performance impact of erase operations increases even further in modern SSDs, as an erase operation often requires *multiple erase loops*. As a flash cell experiences more P/E cycles, the cell becomes more difficult to erase [11,15,60]. Consequently, an erase operation may fail to sufficiently erase all the cells

in a target block, which we call an *erase failure*. To secure data reliability, it is common practice to employ the Incremental Step Pulse Erasure (ISPE) scheme [16,61,62], which retries to erase the block with an increased erase voltage until completely erasing the block.

Figure 7 shows how an NAND flash chip erases a block via multiple erase loops, each of which consists of two steps: (i) an erase-pulse (EP) step and (ii) a verify-read (VR) step. An EP step (e.g., ❶ in Figure 7) applies V_{ERASE} to the target block for a fixed amount of time $\texttt{tEP}$ (e.g., 3.5 ms) that is predefined by NAND manufacturers at design time. After each EP step, a VR step (e.g., ❷) checks if all the cells in the block are sufficiently erased (i.e., the number of fail bits after EP(i), $F(i)$, is less than the redefined erase pass threshold, F_{PASS}). When EP(i) (the i-th EP step, $i \geq 1$) fails to do so, the ISPE scheme performs EP($i+1$) while progressively increasing V_{ERASE} by a fixed amount ΔV_{ISPE}. The ISPE scheme repeats this until completely erasing the block, leading to erasure latency $\texttt{tBERS}$ as follows:

$$\texttt{tBERS} = (\texttt{tEP} + \texttt{tVR}) \times N_{ISPE}, \tag{2}$$

where $\texttt{tVR}$ is the VR latency (~100 µs), and N_{ISPE} is the number of erase loops required to completely erase the block.

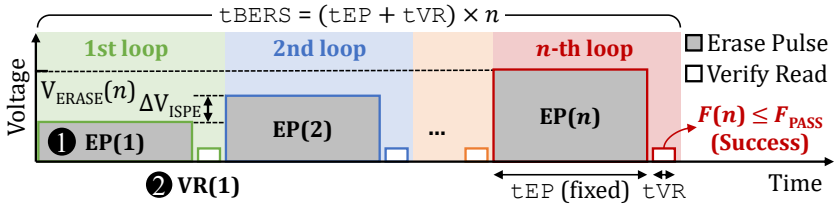
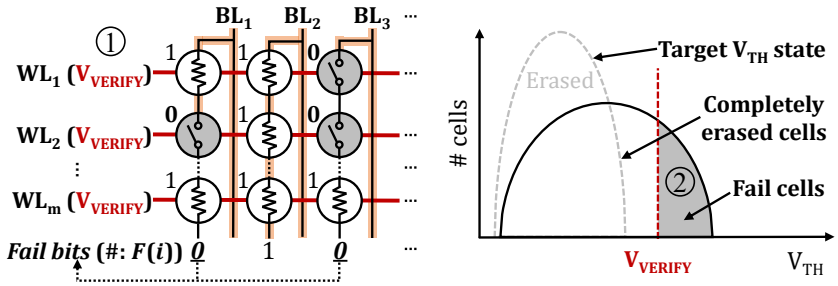


Figure 7. Incremental Step Pulse Erasure (ISPE) scheme.

Figure 8a,b describe how an NAND flash chip performs a VR step. It simultaneously senses (or activates) all the WLs in the block using a verify voltage V_{VERIFY} (① in Figure 8a) that is between the erase state and the first program state. If EP(i) fails, it means that the target block has some cells whose V_{TH} levels are still higher than V_{VERIFY} (② in Figure 8b). Such cells would operate as an off switch during VR(i) (the i-th VR step) because $V_{TH} > V_{VERIFY}$, making the corresponding BLs read as '0' bits, called *fail bits*. Then, VR(i) counts $F(i)$, the number of fail bits after EP(i), using *on-chip* counter logic [60,63]. It judges that EP(i) succeeds only when $F(i)$ is lower than a predefined threshold F_{PASS}.



(a) WL-voltage control in VR (b) V_{TH} distribution after EP

Figure 8. Details of verify-read (VR) step in ISPE scheme.

Limitations of the ISPE Scheme: To understand the potential of optimizing erase latency, we characterize 160 real 3D 48-layer triple-level cell (TLC) NAND flash chips (see Section 5 for our characterization methodology). We measure $m_{\texttt{tBERS}}$, the minimum $\texttt{tBERS}$ to completely erase a block, for 19,200 blocks randomly selected from the 160 chips. Figure 9 illustrates the cumulative distribution function (CDF) of $m_{\texttt{tBERS}}$ (PEC) across the evaluated blocks, under different P/E cycle counts (*PEC*).

From the evaluation results, we draw three key observations. First, an erase operation in modern SSDs needs multiple erase loops, which considerably exacerbates both erase-induced cell stress and erase latency. While all evaluated blocks can be erased within a single loop at zero PEC, every block requires multiple (2∼4) loops after reaching 2K PEC, resulting in a 2∼4× increase in tBERS when employing the ISPE scheme. Second, $m_{\texttt{tBERS}}$ significantly varies even across blocks that require the same N_{ISPE}. Note that $m_{\texttt{tBERS}}$ and N_{ISPE} indicate the minimum tBERS for completing erasure and the number of erase loops in the ISPE scheme. This clearly shows that a significant number of blocks are *over-erased* under the ISPE scheme, suffering from more erase-induced stress than necessary. If we used the ISPE scheme, 40% of the blocks at 3K PEC would require $N_{\text{ISPE}} = 3$ and thus experience the high erase voltage for 10.5 ms (purple dots in Figure 9), while their $m_{\texttt{tBERS}}$ values significantly vary. Third, there is considerable erase latency variation when $N_{\text{ISPE}} = 1$. More than 30% of the blocks at 1K PEC require only 2.5 ms to be completely erased, which is 29% lower than tBERS in the ISPE scheme (3.5 ms).

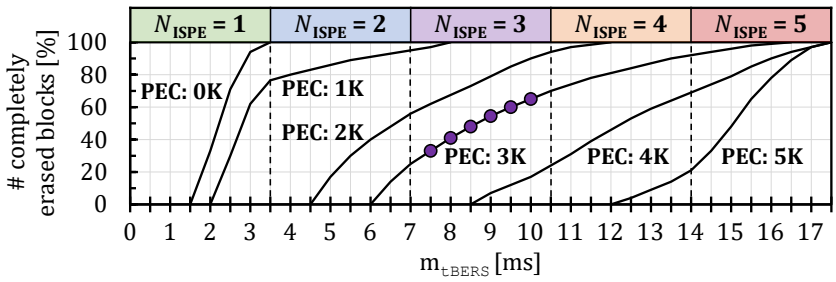


Figure 9. Erase latency variation under different P/E cycles.

From these observations, we draw two conclusions. First, modern NAND flash memory suffers from unnecessarily longer erase latency (and thus more erase-induced cell stress) than actually needed. Second, if it is possible to accurately predict and use $m_{\texttt{tBERS}}$ for a block, such an approach would significantly mitigate the wear-out and long tail latency problems.

3.3. Per-WL Erase Speed Variability

The high process variation during 3D manufacturing introduces not only inter-block variability in erase characteristics (i.e., different erase speed shown in Figure 9) but also per-WL variability in erase characteristics (i.e., different erase speed within WLs in a single 3D flash block). To analyze variability in erase characteristics per WL, we measure the erase speed for every WL in different vertical locations in a block. Figure 10 illustrates how to measure the erase speed of target WLs. First, all flash cells in a block are programmed to a specific state (e.g., the P7 state) to create a uniform precondition. Second, the block is erased with a predefined erase voltage, which is lower than the normal erase voltage (i.e., a block is softly erased). Subsequently, we can quantify the ΔV_{TH}, which indicates the extent of erasure for each WL.

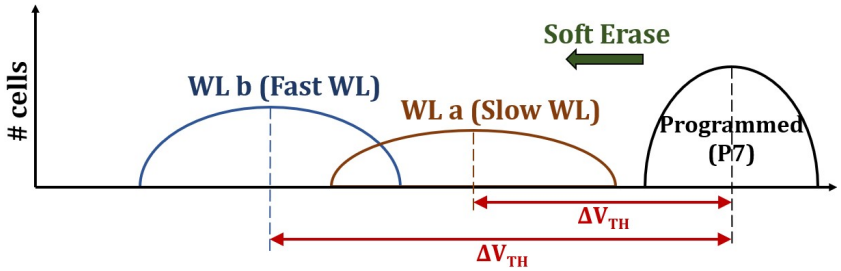


Figure 10. Erase speed measurement of flash cells in different WLs.

Figure 11 illustrates the per-WL erase speed for different WLs within a block under different P/E cycles. The evaluation results strongly indicate a huge erase speed variability between WLs in a block. The erase speed of the best WL (i.e., erase-prone WL) is 40% faster than that of the worst WLs. Furthermore, such per-WL variability in erase characteristics exacerbates as a flash block experiences more P/E cycles. In the ISPE scheme, the erase voltage is determined by the worst WL of the target block. Therefore, to mitigate the negative effect of the erase operation, it is crucial to alleviate the unnecessary erase stress by applying the per-WL optimal erase voltage depending on the erase speed difference between WLs.

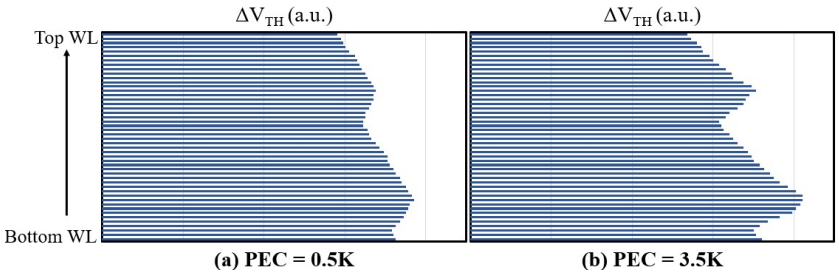


Figure 11. Erase speed variation within the block under different P/E cycles.

3.4. Limitations of the State of the Art

To our knowledge, only a few prior works on *2D* NAND flash memory [16,29–31] propose to dynamically adjust ISPE parameters. Figure 12 illustrates the fundamental concepts of (a) *Dynamic Program and Erase Scaling (DPES)* [29–31] and (b) *intelligent ISPE* (*i*-ISPE) [16]. DPES reduces erase-induced stress by lowering V_{ERASE}, which consequently narrows the voltage window for program states. However, it cannot avoid longer program latency to achieve the same level of reliability as the original ISPE scheme by forming much narrower program V_{TH} states. In contrast, *i*-ISPE monitors the N_{ISPE} of each block to execute only the final erase loop EP(N_{ISPE}) while bypassing the previous loops (e.g., EP(1) and EP(2) as depicted in Figure 12b). This approach can potentially reduce not only the erase-induced stress but also `tBERS`.

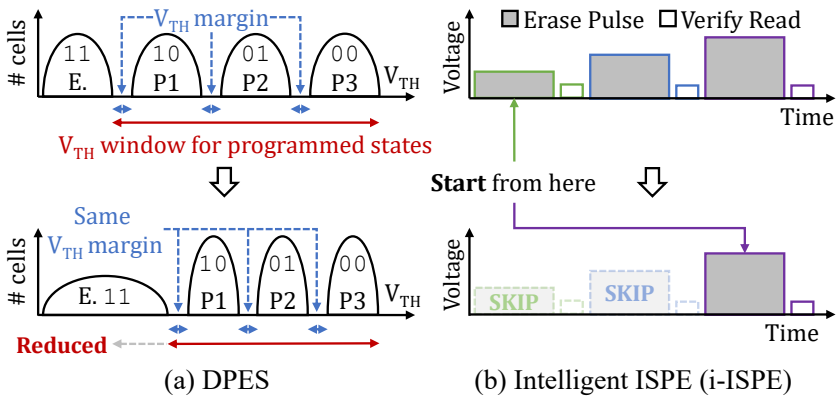


Figure 12. High-level overview of existing ISPE optimizations.

Unfortunately, it is challenging to implement DPES and *i*-ISPE in modern 3D NAND flash memory due to two reasons. First, erasing 3D flash cells is more complex than 2D flash cells due to their differences in cell physics and erase mechanisms [11]. Second, 3D NAND flash memory exhibits higher process variation across cells compared to 2D NAND flash memory [34,35,49]. Such characteristics significantly limit the effectiveness of both DPES and *i*-ISPE in modern NAND flash memory: (i) for DPES, securing the voltage window wide enough for the program states becomes more challenging; (ii) for *i*-ISPE, skipping the first erase loops makes incurring an erase failure more likely, which, in turn, rather requires

the next erase loop with a higher V_{ERASE} (i.e., more erase-induced stress) compared to the conventional ISPE scheme. We quantitatively evaluate the effectiveness of DPES and *i*-ISPE in modern SSDs in Section 7.

4. REO: Revisiting Erase Operation

In this work, we introduce REO, which improves SSD lifetime and I/O performance by implementing near-optimal erase latency and voltage for each target block. The two key ideas of REO are straightforward. First, unlike the ISPE scheme that performs all erase-pulse (EP) steps with fixed latency tEP for every block, REO dynamically adjusts tEP to be just long enough for complete erasure of a target block, called FELP. When EP(i) in REO may also fail to completely erase the block, REO also performs the next EP($i+1$) while trying to reduce tEP *if possible*, i.e., when it expects EP($i+1$) to completely erase the block with reduced tEP. This means that REO reduces tEP in the *final* erase loop (i.e., EP(N_{ISPE})) that completely erases the block, thereby reducing the total erase latency tBERS as follows:

$$\texttt{tBERS} = (\texttt{tEP} + \texttt{tVR}) \times N_{ISPE} - \Delta\texttt{tEP}, \tag{3}$$

where ΔtEP represents the reduction in tEP during EP(N_{ISPE}). Second, unlike the ISPE scheme that applies the same erase voltage to every WL in a block, REO differentiates near-optimal erase voltage depending on WLs, called SEVA. As explained in Section 3.2, in the conventional ISPE scheme, the erase voltage is determined by the worst WL of the target block. REO selectively controls the effective amount of erase voltage for each WL by setting different WL gate voltages (V_{WG}) during the erase operation, thus minimizing the erase stress of most WLs in a block (i.e., improving SSD lifetime by delaying the wear-out of flash cells).

Fail-bit-count-based Erase Latency Prediction (FELP): A key challenge in REO is to accurately identify $m_{\texttt{tEP}}(i)$ for a block, i.e., the minimum value of tEP in each EP step (i.e., EP(i)) just long enough to fully erase all the cells in the block. Even though prior work [64,65] has experimentally demonstrated a strong correlation between a block's PEC and erase latency, i.e., the higher the block's PEC, the longer the latency for erasing the block [31], PEC alone is insufficient for accurate prediction of near-optimal tEP due to high process variation in modern NAND flash memory. As shown in Figure 9, $m_{\texttt{tEP}}(N_{ISPE})$ significantly varies even across blocks at the same PEC. This suggests that REO needs a way more effective metric than PEC to figure out more precise erase characteristics (i.e., $m_{\texttt{tEP}}(i)$) for individual flash blocks.

REO addresses the challenge via *Fail-bit-count-based Erase Latency Prediction (FELP)* that predicts $m_{\texttt{tEP}}(i+1)$ based on $F(i)$, the number of fail bits incurred by the previous EP(i). Our key intuition is that $F(i)$ can likely be an *accurate proxy* of $m_{\texttt{tEP}}(i+1)$ because the more sufficiently the cells are erased by an EP step, the lower the fail-bit count. Commodity NAND flash chips already calculate $F(i)$ for the ISPE scheme as explained in Section 3.2, so the implementation overhead of FELP is trivial (see Section 6 for more detailed overhead analysis).

Figure 13 depicts how REO safely reduces tEP based on FELP. Like the ISPE scheme, REO also performs a verify-read step VR(i) after each EP(i), which results in $F(i)$. If $F(i)$ is higher than a threshold F_{HIGH} ($\gg F_{PASS}$), REO uses the default tEP for the next EP($i+1$) considering that there is no room for tEP reduction (e.g., ❶ until EP($N_{ISPE}-1$) in Figure 13). When $F_{PASS} < F(i) \leq F_{HIGH}$, REO ❷ reduces tEP, i.e., it predicts and uses $m_{\texttt{tEP}}(i+1)$ for EP($i+1$), such that the lower the value of $F(i)$, the lower the value of $m_{\texttt{tEP}}(i+1)$. Note that REO provides effectively the same reliability as the ISPE scheme as long as $F(N_{ISPE}) \leq F_{PASS}$ (❸).

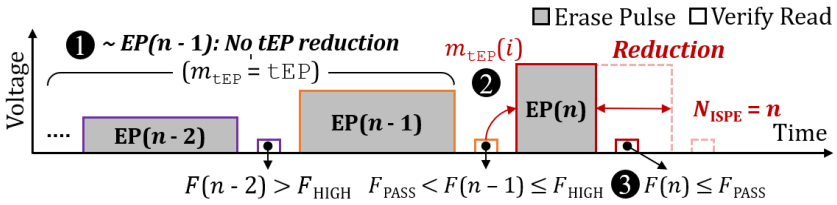


Figure 13. Fail-bit-count-based Erase Latency Prediction (FELP).

Leveraging ECC-Capability Margin: Prior work [32,41] has demonstrated that a large ECC-capability margin (the difference between the maximum number of bit errors per codeword that a given ECC can correct and the number of bit errors in a codeword) exists in modern SSDs due to two reasons. First, modern SSDs commonly employ strong ECCs to cope with the high RBER of NAND flash memory in the worst-case operating conditions (e.g., 1-year retention time at 5K PEC). Second, it is common practice to employ *read-retry* in modern SSDs [41] to ensure data reliability. When a read page's RBER exceeds the ECC capability, read-retry *repeats* reading of the page with adjusted V_{REF} until it sufficiently lowers RBER, thereby leading to a large ECC-capability margin.

REO leverages the large ECC-capability margin to reduce `tEP` *more aggressively* by increasing the pass threshold F_{PASS} in the ISPE scheme. Doing so would likely cause incomplete erasure of a target block, which introduces additional bit errors. However, we hypothesize that REO can still ensure data reliability in many cases due to three key reasons. First, a block's reliability degrades as it experiences P/E cycling, so an even larger ECC-capability margin to tolerate the additional errors exists at low PEC. Second, aggressive `tEP` reduction mitigates erase-induced cell stress, which can compensate for the additional errors as a long-term impact. When REO aggressively reduces `tEP`, bit errors just after the write operation can temporarily increase due to insufficiently erased cells, which is called an initial error. However, as P/E cycles increase (i.e., long-term usage of SSD), endurance error is more critical in determining SSD lifetime. Therefore, although aggressive `tEP` reduction can cause the initial error, it is beneficial to SSD lifetime because it can effectively mitigate erase-induced cell stress. In addition, the increment in the initial error does not cause any issues due to strong ECC in modern SSDs. Third, a majority of additional fail cells due to increased F_{PASS} would be programmed to higher V_{TH} states under the data randomization technique [66,67], e.g., 87.5% in TLC NAND flash memory. It means that such fail cells are unlikely to cause bit errors, significantly decreasing the reliability impact of aggressive `tEP` reduction. To maximize REO's benefits without compromising data reliability, we enhance FELP to also consider the expected ECC-capability margin for a target block by keeping the number of additional errors caused by aggressive `tEP` reduction below the current ECC-capability.

Selective Erase Voltage Adjustment (SEVA): Prior work [11,32] has demonstrated that the high process variation along the vertical direction (i.e., between WLs) within a block has increased in 3D NAND flash memory due to its unique manufacturing process. Such high process variation causes quite different erase speeds among WLs in a block, as shown in Section 3.3, making most WLs suffer from unnecessary erase stress. Excessive erase stress increases raw bit error rates (RBERs) of NAND flash memory by accelerating the wear-out of flash cells. Since higher RBERs can trigger read-retry procedures more frequently, SSD I/O performance can be significantly degraded. Therefore, mitigating excess erase stress is a key challenge of our technique, REO, for better SSD lifetime and I/O performance.

Unfortunately, the conventional erase mechanism of NAND flash memory does not allow for differentiated erase voltages depending on WL. When initiating the erase operation, a high erase voltage, V_{ERASE}, is applied to the substrate that all flash cells in a block

share. Since the WL gate voltage (V_{WG}) is maintained at 0V during the erase operation, flash cells in a block experience the effective erase voltage of ($V_{ERASE} - 0$)V, which leads the cell's V_{TH} to lower by removing electrons from the trap layer into the substrate. Therefore, regardless of their erase characteristics, cells in a block are erased using the same voltage.

REO overcomes the challenge through *Selectivve Erase Votage Adjustment (SEVA)* that dynamically adjusts V_{ERASE} considering different erase characteristics of WLs. Our key idea is derived from the fact that the effective erase voltage is determined by two factors: V_{ERASE} and V_{WG}. Since V_{ERASE} cannot be controlled for each WL, SEVA differentiates the effective erase voltage by selectively modifying V_{WG}. For example, when 20 V of erase voltage is applied to the substrate, we can reduce the effective erase voltage of WL30 to 19.7 V by raising the gate voltage of WL30 from 0 V to 0.3 V.

Figure 14 illustrates how to reduce the effective erase voltage on WL0 and WL46 while the rest of the WLs are erased using the default erase voltage. During the erase operation, the gate voltage of WL0 and WL46 are set to 0.1 V and 0.2 V, respectively, instead of 0 V. Through this action, SEVA can selectively reduce the effective erase voltage only for WL0 and WL46 by 0.1 V and 0.2 V. Consequently, since the erase stress is exponentially proportional to an effective erase voltage, the erase stress can be significantly reduced to the NAND flash cells of WL0 and WL46 during an erase operation. Furthermore, controlling V_{WG} in SEVA can be easily implemented without hardware modifications. By using low-level test commands such as `GET/SET FEATURE` commands [68], SEVA can modify V_{WG} from 0.0 V to 1.0 V in steps of 0.05 V.

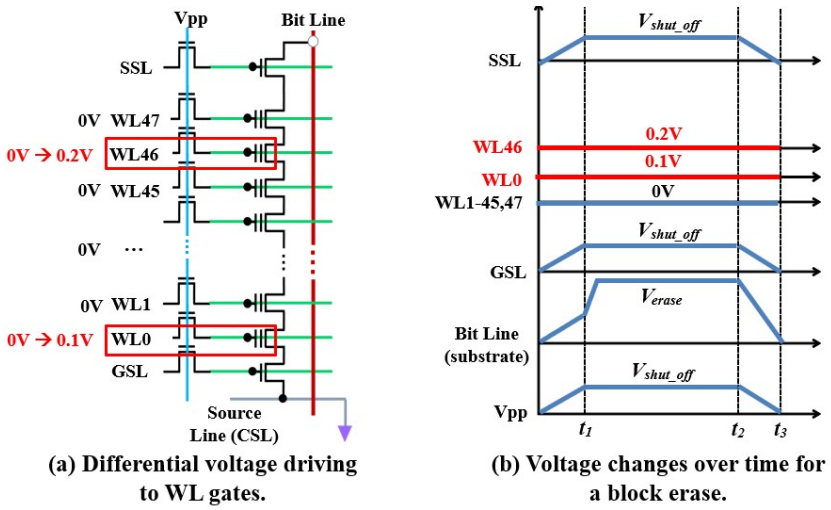


Figure 14. An overview of Selective Erase Voltage Adjustment (SEVA).

5. Device Characterization Study

To validate our key ideas and hypothesis in Section 4, we conduct an extensive real-device characterization study.

5.1. Characterization Methodology

Infrastructure: We use an FPGA-based testing platform with a custom flash controller and temperature controller. The flash controller can perform not only basic NAND flash commands (i.e., for read, program, and erase operations) but also low-level test commands such as `GET/SET FEATURE` commands [68]. The modification of NAND operations (e.g., counting fail bits or controlling WL gate voltage) can be implemented by low-level test commands such as `SET/GET FEATURE` commands. The `SET/GET FEATURE` commands used for REO are device-level commands defined by Open NAND Flash Interface Specification (ONFI), which are different from the NVMe commands with the same names. The ONFI

SET/GET FEATURE commands enable modifying/accessing an NAND flash chip's device-internal parameters; hence, REOFTL exploits the commands to modify/access erase-timing parameters/fail-bit count or per-WL gate voltage. Therefore, our technique, REO, does not require a hardware modification, thus minimizing the increase in hardware implementation complexity. This feature allows us to modify `tEP` of each EP(i) at a granularity of 0.5 ms and obtain $F(i)$ from the chip after VR(i). We can also change V_{WG} with 0.05 V resolution. The temperature controller can maintain the operating temperature of the tested chips within ±1 °C of the target temperature, thus minimizing unintended RBER variations potentially caused due to unstable temperature.

Methodology: We characterize 160 real 48-layer 3D TLC NAND flash chips from Samsung [69], in which the default `tEP` = 3.5 ms and V_{WG} = 0.0 V. To minimize the potential distortions in our results, for each test scenario, we evenly select 120 blocks from each chip at different physical locations and test all WLs in each selected block. We test a total of 3,686,400 WLs (11,059,200 pages) to obtain statistically significant results.

We test the chips while varying PEC and retention time. Unless specified otherwise, we increase a block's PEC by programming every page in the block using a random pattern and erasing the block with the default `tEP` in every erase loop. We follow the JEDEC industry standard [70] for an accelerated lifetime test to analyze the reliability under the worst-case operating conditions. For example, to emulate a 1-year retention time at 30 °C, we bake the chips at 85 °C for 13 h following the Arrhenius' law [71].

To identify a block's $m_{\texttt{tBERS}}$ (i.e., N_{ISPE} and $m_{\texttt{tEP}}(N_{\text{ISPE}})$), we erase the block using a *modified* ISPE scheme (m-ISPE) that we design by modifying the original ISPE scheme in two aspects. First, we reduce the fixed latency `tEP` for each EP(i) from 3.5 ms to 0.5 ms, i.e., we split an erase loop in the ISPE scheme into seven shorter loops. Second, we increase V_{ERASE} for every *seven* erase loops (not for every loop) to effectively emulate the ISPE scheme. If a block requires n loops under m-ISPE, we estimate $N_{\text{ISPE}} = \lceil n/7 \rceil$ and $m_{\texttt{tEP}}(N_{\text{ISPE}}) = 0.5 \times (1 + ((n-1) \bmod 7))$ of the block under the ISPE scheme. Even though the m-ISPE scheme requires six additional ramping-up/down steps to charge/discharge voltage and VR steps for each erase loop compared to the original ISPE scheme, its reliability impact is negligible; for our 160 tested chips, the m-ISPE scheme hardly increases the average RBER (by less than 1%) compared to the original ISPE scheme, under 1-year retention time at 5K PEC.

5.2. Fail-Bit Count vs. Near-Optimal Erase Latency

To validate the feasibility of FELP, we analyze the relationship between the minimum erase latency and fail-bit count. We measure each block's N_{ISPE} and $m_{\texttt{tEP}}(N_{\text{ISPE}})$ while tracking $F(i)$ in every EP(i) under the m-ISPE scheme. Figure 15 depicts the maximum value of $F(N_{\text{ISPE}})$ within the blocks that have the same $m_{\texttt{tEP}}(N_{\text{ISPE}})$, when we progressively increase `tEP` by 0.5 ms in the final EP step (i.e., EP(N_{ISPE})).

We make two key observations. First, the fail-bit count *almost linearly* decreases as `tEP` increases. While the negative correlation between $F(i)$ and `tEP` is expected in Section 4, the correlation is significantly high and consistent; increasing `tEP` by 0.5 ms decreases $F(N_{\text{ISPE}})$ by almost the same amount δ ($\simeq$5000) in all tested blocks with different N_{ISPE} and $m_{\texttt{tEP}}(N_{\text{ISPE}})$ values. This suggests that (i) erase latency has a linear impact on the degree of erasure under the same erase voltage and (ii) NAND manufacturers carefully set V_{ERASE} values (i.e., $V_{ERASE}(i)$ and ΔV_{ISPE}) in the ISPE scheme to avoid excessive increases in erase-induced cell stress and erase latency. Second, when $m_{\texttt{tEP}}(N_{\text{ISPE}}) = 0.5$ ms, $F(N_{\text{ISPE}})$ is quite consistent at a certain value γ ($\ll \delta$) in all test scenarios. The result suggests that the lower the cell's V_{TH} level, the more difficult it becomes to further reduce the V_{TH} level.

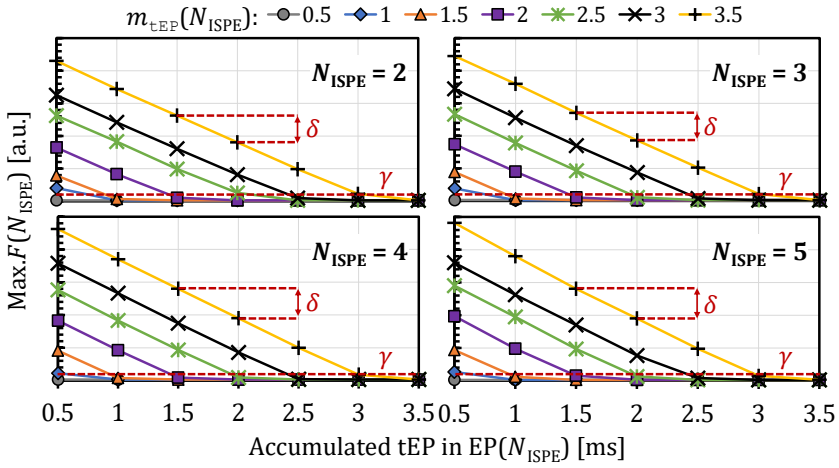


Figure 15. Impact of erase latency on the fail-bit count.

Our observations highlight the high potential of FELP. To confirm this, we analyze how a block's $m_{tEP}(N_{ISPE})$ varies depending on its $F(N_{ISPE}-1)$. Figure 16 depicts the probability of $m_{tEP}(N_{ISPE})$ at different N_{ISPE} across the fail-bit ranges that we set based on γ and δ from Figure 15. A box at (x, y) in Figure 16 represents the probability (in grayscale) that a block requires $m_{tEP}(N_{ISPE}) = y$ ms for complete erasure when its $F(N_{ISPE}-1)$ belongs to x-th fail-bit range. We also plot the fraction of blocks that belong to the x-th fail-bit range (top). Note that for the same N_{ISPE}, the sum of all fractions (top) is 100%, and the sum of all probabilities at the same x-th fail-bit range (bottom) is 100%.

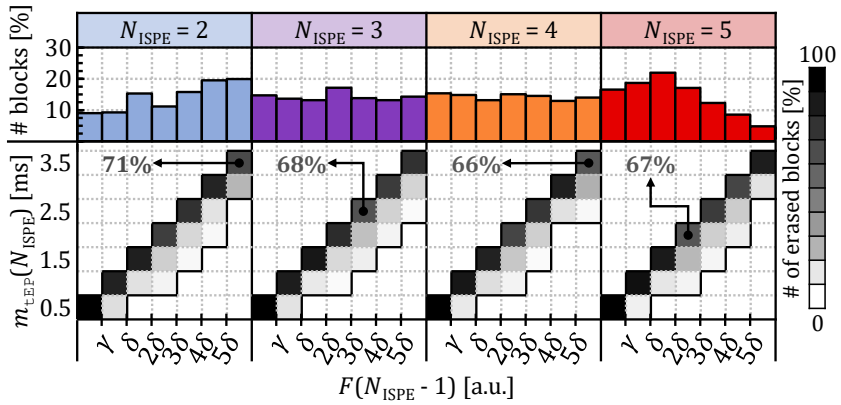


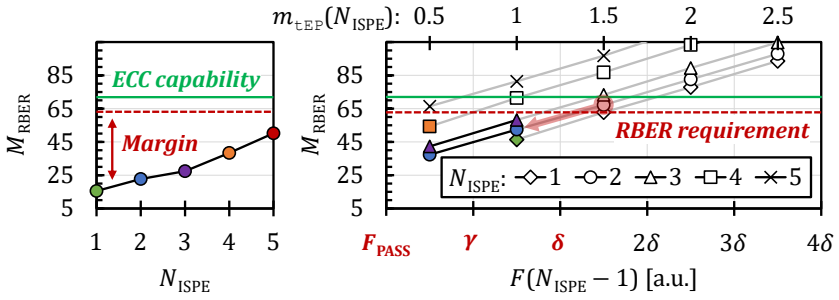
Figure 16. Erase-pulse latency depending on the fail-bit count.

We make two key observations. First, FELP is highly effective at predicting $m_{tEP}(N_{ISPE})$. A majority of blocks (e.g., ≥66% in $N_{ISPE} = 4$) in the same fail-bit range require the same $m_{tEP}(N_{ISPE})$ under all different N_{ISPE} cases. Even though every fail-bit range contains some blocks whose $m_{tEP}(N_{ISPE})$ is lower compared to the majority of blocks, the fraction is quite low (e.g., < 34% in $N_{ISPE} = 4$) in all the fail-bit ranges and N_{ISPE} cases. Second, $F(N_{ISPE}-1)$ is distributed across blocks quite evenly in all N_{ISPE} cases. This highlights again that $F(N_{ISPE}-1)$ is an accurate proxy of $m_{tEP}(N_{ISPE})$, given that $m_{tEP}(N_{ISPE})$ also significantly varies across blocks in a wide range as shown in Figure 9.

Based on our observations, we draw two conclusions. First, REO can accurately predict the minimum erase latency using FELP, even for blocks that have varying erase characteristics. Second, the implementation of FELP requires only identifying two values for fail-bit ranges, e.g., γ and δ in Figures 15 and 16, which can be performed using our characterization methodology.

5.3. Reliability Margin for Aggressive tEP Reduction

To identify the ECC-capability margin for more aggressive tEP reduction, we analyze the reliability impact of insufficient erasure. We measure M_{RBER} of each block, i.e., the maximum RBER within the pages in the block under 1-year retention time at 30 °C, when we erase the block in two different ways. First, we *completely* erase the block by performing N_{ISPE} erase loops with $m_{\mathtt{tEP}}(N_{\mathrm{ISPE}})$. Second, we *insufficiently* erase the block by performing only $(N_{\mathrm{ISPE}} - 1)$ erase loops, which results in $F_{\mathrm{PASS}} < F(N_{\mathrm{ISPE}} - 1) \leq F_{\mathrm{HIGH}}$. Figure 17 depicts the maximum value of M_{RBER} within the tested blocks when we program the blocks (a) after complete erasure and (b) after insufficient erasure. For Figure 17b, we group the tested blocks depending on their N_{ISPE} and fail-bit range. We also plot (i) the *ECC capability* at 72 bits per 1 KiB and (ii) the *RBER requirement* at 63 bits per 1 KiB to reflect sampling errors (i.e., a block is considered unusable if its $M_{\mathrm{RBER}} > 63$ to incorporate a safety margin into the ECC capability).



(a) Completely erased (b) Incompletely erased

Figure 17. Reliability margin depending on erase status.

We make two key observations. First, we observe a large reliability margin (i.e., *RBER requirement* $- M_{\mathrm{RBER}}$) that can potentially be used to further reduce tEP, especially in the early lifetime stage of blocks. The reliability margin can be calculated based on the difference between the current RBER and the amount of ECC capability. In general, the ECC capability would not significantly change because manufacturers employ stronger ECCs (e.g., BCH or LDPC) to address reliability degradation. Therefore, the reliability margin is mainly determined by the current RBER of the block. The one way to decide on the current RBER is offline error profiling based on comprehensive NAND device characterization. Such a method is simple but it cannot sufficiently consider process variations in 3D NAND flash memory, leading to less optimized results. Another approach is to measure the current RBER online dynamically. Once a read operation is performed, the RBER value of the target block can be monitored, and SSD tracks each value using an additional data structure. Since the DRAM capacity of modern SSDs reaches several GBs, the space overhead is not significant (i.e., less than 1 byte per block in SSDs incorporated with 72-bit ECC capability). As shown in Figure 17a, when a block is completely erased, there always exists a positive reliability margin for all N_{ISPE} values up to 47 bit errors ($N_{\mathrm{ISPE}} = 1$). Second, it is possible to further reduce tEP without compromising reliability in many operating conditions. As shown in Figure 17b, using an insufficiently erased block still meets the RBER requirement (i.e., $M_{\mathrm{RBER}} < 63$) if either of the following two conditions is met: **[C1:** $N_{\mathrm{ISPE}} \leq 3$ and $F(N_{\mathrm{ISPE}} - 1) < \delta$**]** and **[C2:** $N_{\mathrm{ISPE}} = 4$ and $F(3) < \gamma$**]**. This means that we can skip the final erase loop in such cases, thereby further increasing the amount of tEP reduction even higher than the default tEP in the ISPE scheme, i.e., ΔtEP can be larger than tEP in (3). Note that REO can further reduce $m_{\mathtt{tEP}}(N_{\mathrm{ISPE}})$ even if neither **[C1]** or **[C2]** are met because increasing tEP by 0.5 ms in EP(i) decreases $F(i)$ by δ as demonstrated in Section 5.2. For example, a block requires $m_{\mathtt{tEP}}(N_{\mathrm{ISPE}}) = 1.5$ ms for complete erasure when $N_{\mathtt{ISPE}} = 3$ and

$\delta < F(2) \leq 2\delta$, but using $\mathtt{tEP} = 0.5$ ms in EP(3) can still meet the RBER requirement since doing so would decrease M_{RBER} below 63 (see the arrow in Figure 17b).

We conclude that REO can significantly mitigate the negative impact of erase operations by leveraging not only the high process variation but also the large ECC-capability margin present in modern SSDs. Table 1 shows the final $m_{\mathtt{tEP}}(N_{\mathrm{ISPE}})$ model that we constructed; each cell's value 't_1 / t_2' indicates $m_{\mathtt{tEP}}(N_{\mathrm{ISPE}})$ when REO leverages only the process variation (t_1) and when also leveraging the ECC-capability margin (t_2).

Table 1. Final model of $m_{\mathtt{tEP}}(N_{\mathrm{ISPE}})$ based on $F(N_{\mathrm{ISPE}} - 1)$.

N_{ISPE}	$F(N_{\mathrm{ISPE}} - 1)$							
	$\leq \gamma$	$\leq \delta$	$\leq 2\delta$	$\leq 3\delta$	$\leq 4\delta$	$\leq 5\delta$	$\leq 6\delta$	$\leq 7\delta$
1	0.5/0	1/0	1.5/0.5	2/1	2.5/1.5	2.5/2	2.5/2.5	2.5/2.5
2	0.5/0	1/0	1.5/0.5	2/1	2.5/1.5	3/2	3.5/2.5	3.5/3
3	0.5/0	1/0	1.5/0.5	2/1	2.5/1.5	3/2	3.5/2.5	3.5/3
4	0.5/0	1/0.5	1.5/1	2/1.5	2.5/2	3/2.5	3.5/3	3.5/3.5
5	0.5/0.5	1/1	1.5/1.5	2/2	2.5/2.5	3/3	3.5/3.5	3.5/3.5

5.4. WL Gate Voltage vs. Erase Speed

To validate the feasibility of SEVA, we analyze the impact of V_{WG} on the erase speed of each WL. Figure 18a shows how the average ΔV_{TH} value of each WL changes during the erase operation by gradually increasing V_{WG} from 0.0 V to 1.0 V in steps of 0.05 V. The evaluation results confirm that the ΔV_{TH} value exponentially decreases as the V_{WG} increases. When V_{WG} is raised from 0 V to 0.1 V, the amount of ΔV_{TH} decreases approximately by 21%, while when V_{WG} increases from 0.4 V to 0.5 V, the change in ΔV_{TH} is about 9%.

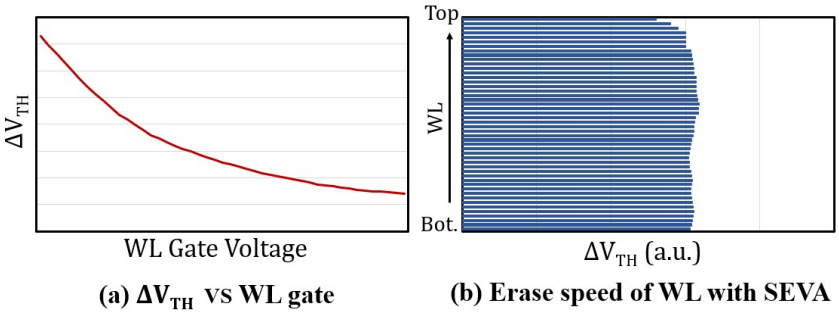


Figure 18. Impact of WL gate voltage on erase speed.

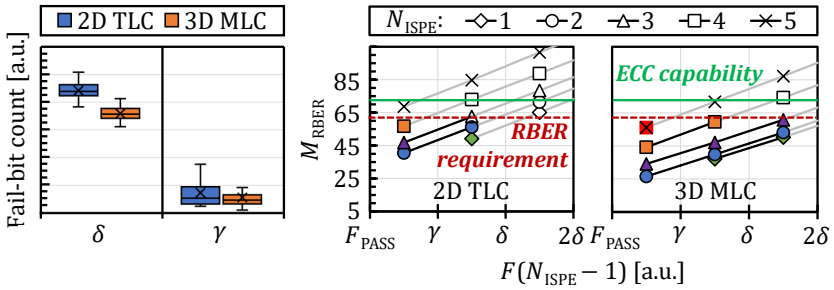
Based on the results of Figures 11 and 18a, we apply different V_{WG} to each WL depending on its erase characteristics to make the erase stress of all WLs uniform. Figure 18b verifies that SEVA can successfully mitigate unnecessary stress resulting from over-erased cells.

5.5. Applicability of REO *for Other Types of Chips*

We expect that the key ideas of REO are generally applicable to a wide range of NAND flash chips other than the chips used for our characterization study due to three reasons. First, our chips represent modern 3D NAND flash memory well because most commercial chips including SMArT/TCAT/BiCS have similar structures and cell types, e.g., vertical channel structures, gate-all-around cell transistors, and charge-trap type flash cells [60,62,72], sharing key device characteristics like operation mechanisms and reliability characteristics. Second, the erase mechanism of NAND flash memory has not changed

significantly for more than a decade. For example, the ISPE scheme has been used since 2D SLC NAND flash memory [60,62]. Third, AERO does *not* rely on chip-specific behaviors but leverages inherent erase characteristics, e.g., the more completely the cells within a block are erased, the lower the fail-bit count of the block.

To support our hypothesis, we characterize two additional types of NAND flash chips, (i) 2x nm 2D TLC NAND flash chips [73] and (ii) 48-layer 3D MLC NAND flash chips [74] from Samsung. We use the same methodology as other device characterizations. Figure 19 shows (a) the values of δ and γ for all tEP and N_{ISPE} cases (box plot) and (b) the maximum value of M_{RBER} within the tested block after insufficient erasure.



(a) Fail-bit count (b) M_{RBER} after insufficient erasure

Figure 19. Erase characteristics of other chip types.

We make two key observations. First, although the exact values of δ and γ slightly vary depending on the chip type, they are quite consistent within the same type of chips across all tested cases as shown in Figure 19a. This clearly shows that the strong linear relationship between the fail-bit count and accumulate tEP also holds in the chips additionally tested. Second, as shown in Figure 19b, the reliability impacts of insufficient erasure in the 2D TLC and 3D MLC chips exhibit similar trends to those in the 3D TLC chips (cf. Figure 17b), suggesting the high feasibility of aggressive tEP reduction is also present in the two types of chips. We conclude that REO can be used for a wide range of chips with the same methodology we use to construct the tEP model for our tested chips (Table 1).

6. Design and Implementation

We design REOFTL, an REO-enabled flash translation layer (FTL), by extending the conventional page-level FTL [75] with two key data structures: (i) *Erase-timing Parameter Table (EPT)* and (ii) *Erase-Voltage Parameter Tables (VPT)*. The EPT is a simple table to store $m_{\mathtt{tEP}}(i)$ for each EP(i) depending on $F(i-1)$, which can be built via offline profiling of target chips as in Section 5 (Table 1). The VPT is a table that keeps track of the erase speed of WLs and stores the control gate voltage of each WL.

Figure 20 illustrates how REOFTL dynamically adjusts erase latency and effective erase voltage. For erasing a block whose index (or ID) is k, it first looks up the VPT in the (❶ in Figure 20) and applies corresponding V_{WG} to each WL. Subsequently, REOFTL ❷ performs first erase loop of the target chip's with default tEP with a SET FEATURE command. Then it ❸ queries the EPT with $F(1)$, which can be obtained via a GET FEATURE command. If an erase loop fails, ❹ performs the next loop while adjusting the chip's tEP based on the value obtained from the EPT.

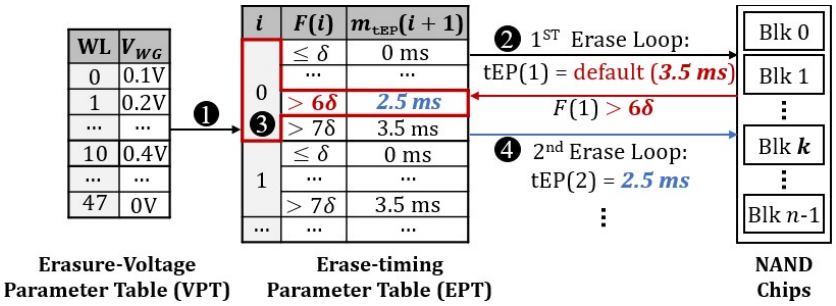


Figure 20. Operational overview of REOFTL.

Implementation Overhead: REOFTL requires only two small changes to conventional SSDs. First, REOFTL can use `GET/SET FEATURE` commands to obtain $F(i)$ and adjust `tEP` for each EP(i), respectively, thereby requiring no change to commodity NAND flash chips. Second, the storage overhead for maintaining the EPT and VPT table is trivial. The EPT needs to keep $T \times L$ entries, where T and L indicate the number of possible `tEP` values and the maximum number of erase loops, respectively. In our current design, the EPT has 35 entries ($T = 7$ and $L = 5$), which requires only 140 bytes even when using a 32-bit value per entry. The VPT needs N entries, which are determined by the number of vertical layers of 3D NAND flash memory. In our design, N is 48 because we design REO based on 48-layer 3D TLC NAND flash memory. Each entry needs 5-bit information to differentiate 21 WL gate voltages.

In a large body of prior work on improving SSD lifetime or performance, it has been challenging to avoid the inherent trade-off relationships between lifetime, performance, and hardware resources [76]. For example, if we aim to design a novel technique for improving SSD lifetime, performance or hardware overhead is a critical constraint in making the technique practical. However, by exploiting various device-oriented characteristics, our technique effectively addresses the trade-off relationship between lifetime, performance, and resource overhead. In addition, REO modifies NAND operations (e.g., counting fail bits or controlling WL gate voltage) by using well-known low-level test commands such as SET/GET FEATURE commands. Therefore, our technique, REO, does not require a hardware modification, thus minimizing the increase in hardware/software implementation complexity. For example, REOFTL requires additional metadata, but the overhead is negligible (0.00000125% of SSD capacity) considering the typical size of internal DRAM in modern SSDs (0.1% of SSD capacity).

7. Evaluation

We evaluate the effectiveness of REO at improving the lifetime and performance of modern NAND flash-based SSDs.

7.1. Evaluation Methodology

We evaluate REO in two ways. First, we characterize 160 real 3D TLC NAND flash chips to assess the lifetime enhancement of REO. Unless specified otherwise, we follow the real-device characterization methodology explained in Section 5.1. Second, we evaluate the impact of REO on I/O performance using MQSim [42], a state-of-the-art SSD simulator.

We compare six different erase schemes: (i) Baseline, (ii) i-ISPE, (iii) DPES, (iv) REO-, (v) REO, and (vi) REO+. Baseline is the conventional ISPE scheme explained in Section 3.2. I-ISPE is the intelligent ISPE scheme [16] explained in Section 3.4, which directly performs EP(n) while skipping the previous EP steps if the block has been completely erased by EP(n) in the most recent erase operation. DPES (explained in Section 3.4) mitigates erase stress by reducing erase voltage V_{ERASE} by 8–10% at a cost of 10–30% increase in write latency tPROG [31]. Since DPES is only applicable until 3K PEC in our tested chips (i.e., no matter how much tPROG increases, reducing V_{ERASE} can no longer meet the reliability requirements), we use the same V_{ERASE} and tPROG values as in Baseline after 3K PEC. REO- and REO dynamically adjust tEP for each EP(i) at a granularity of 0.5 ms based on $F(i-1)$. REO- does not exploit the ECC-capability margin, i.e., it is more *conservative* in tEP reduction compared to REO, which adopts optimization in Section 4. REO+ additionally adopts SEVA in Section 4 on REO.

Simulation Methodology: We extend MQSim in two aspects to model the behavior of modern SSDs more faithfully. First, we modify the NAND flash model of MQSim to emulate the erase characteristics of our 160 tested chips. To this end, during our real-device characterization study in Section 5, we keep track of erase-related metadata such as the minimum erase latency, fail-bit count, and PEC for every tested block. For simulation, we then randomly select tested blocks and assign their metadata to each of the simulated blocks in MQSim. Because MQSim already tracks PEC, a simulated block can accurately emulate the erase characteristics of the corresponding real block at a given PEC by simply looking up the metadata. Second, we optimize the request scheduling algorithm of MQSim to service user I/O requests with a higher priority over SSD-internal read/write/erase operations, e.g., suspending an ongoing erase operation [13]. Table 2 summarizes the configurations of the simulated SSDs. We configure the architecture and timing parameters of simulated SSDs to be close to commodity high-end SSDs (e.g., [1]).

Table 2. Configurations of simulated SSDs.

SSD	Capacity: 1024 GB GC policy: greedy [77] # of channels: 8	Interface: PCIe 4.0 (4 lanes) Overprovisioning ratio: 20% # of chips per channel: 2
NAND Flash Chip	# of planes per chip: 4 # of pages per block: 2,112 MLC technology: TLC tEP (REO): 0.5 ms–3.5 ms tPROG: 350 µs [9]	# of blocks per plane: 497 Page size: 16 KB tR: 40 µs [9] tEP: 3.5 ms [9]
	tPROG: 385 µs (DPES, 0.5K PEC), 455 µs (DPES, 2.5 K PEC)	

For our performance evaluation, we study eleven workloads selected from two benchmark suits, Alibaba Cloud Traces [78] and Microsoft Research Cambridge (MSRC) Traces [79], which are collected from real data center and enterprise servers. For MSRC traces, we reduce the inter-request arrival time by 10×, as similarly performed in a large body of prior work to evaluate more realistic workloads [11,29–31,37,42,80–84]. Table 3 summarizes the I/O characteristics of the workloads used for our evaluation.

Table 3. I/O characteristics of evaluated workloads.

Benchmark	Trace	Abbr.	Read Ratio	Avg. Req. Size	Avg. Inter Req. Arrival Time
Alibaba Cloud [78]	ali_32	ali.A	7%	54 KB	16.3 ms
	ali_3	ali.B	52%	26 KB	111.8 ms
	ali_12	ali.C	69%	38 KB	57.9 ms
	ali_121	ali.D	78%	18 KB	13.8 ms
	ali_124	ali.E	95%	36 KB	5.1 ms
MSR Cambridge [79]	rsrch_0	rsrch	9%	9 KB	421.9 ms
	stg_0	stg	15%	12 KB	297.8 ms
	hm_0	hm	36%	8 KB	151.5 ms
	prxy_1	prxy	65%	13 KB	3.6 ms
	proj_2	proj	88%	42 KB	20.6 ms
	usr_1	usr	91%	49 KB	13.4 ms

7.2. Impact on SSD Lifetime

To evaluate the lifetime enhancement of REO, we measure $M_{RBER}(PEC)$ for each real tested block, the maximum RBER within the pages in the block, while varying PEC under 1-year retention at 30 °C. We construct five sets of 120 blocks randomly selected from 160 real 3D NAND flash chips and test each set while increasing PEC using one of the five erase schemes. Figure 21 depicts the average M_{RBER} across the tested blocks under different PEC.

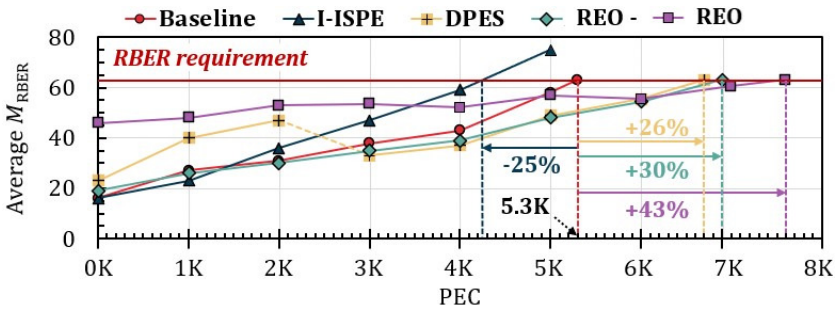


Figure 21. Comparison of SSD lifetime and reliability.

We make four key observations. First, both REO and REO- significantly improve SSD lifetime over Baseline. The average M_{RBER} increases at a much slower rate with PEC in REO- and REO compared to in Baseline, which clearly shows the high effectiveness of the erase latency reduction in REO at lowering erase-induced cell stress. The slower increase in M_{RBER}, in turn, enables a block to meet the RBER requirement at higher PEC in REO (7.6K) and REO- (6.9K) compared to Baseline (5.3K), significantly enhancing SSD lifetime by 43% and 30%, respectively.

Second, REO further improves SSD lifetime considerably over REO- (by 10%) without compromising data reliability. This highlights the high effectiveness of leveraging the reliability margin to reduce erase latency more aggressively. The aggressive `tEP` reduction causes high M_{RBER} even for fresh blocks (i.e., $M_{RBER}(0) = 46$ in REO) but greatly slows down M_{RBER} increase ($M_{RBER}(6K) - M_{RBER}(0) = 9.5$), showing its high long-term benefits.

Third, DPES also improves SSD lifetime considerably (by 26%) compared to Baseline, but its benefits are limited compared to both REO- and REO. Like REO, DPES exhibits

rather high M_{RBER} due to V_{ERASE} reduction until 3K PEC, which, in turn, enables its M_{RBER} to increase more slowly later. However, DPES's benefits are limited due to (i) its limited applicability (until 3K PEC) as well as (ii) its write-performance overheads.

Fourth, i-ISPE *accelerates* RBER increase, which, in turn, rather *decreases* SSD lifetime. In fact, i-ISPE provides the lowest M_{RBER} among the compared SSDs at $PEC = \langle 0\mathrm{K}, 1\mathrm{K} \rangle$ where it is relatively easy to completely erase a block compared to at high PEC. However, it frequently incurs an erase failure as PEC increases, causing more erase-induced cell stress as explained in Section 3.4. Consequently, i-ISPE leads to shorter SSD lifetime by 25% even compared to Baseline, which shows its limited applicability in modern SSDs. Note that REO+ does not improve SSD lifetime over REO because the lifetime is determined by RBER of the worst WL in each block.

7.3. Impact on I/O Performance

Average I/O Performance: To evaluate the performance impact of REO, we first compare *average* I/O performance of the six SSDs in three aspects: (i) read latency, (ii) write latency, and (iii) I/O throughput (IOPS, input/output operations per second). Figure 22 shows the three average performance values in i-ISPE, DPES, REO-, REO, and REO+ that are normalized to Baseline, on average across all the workloads. Table 4 summarizes the evaluation results quantitatively. We observe that all the evaluated SSDs *except for* DPES and REO+ show almost the same average performance for all the workloads and $PEC = \langle 0.5\mathrm{K}, 2.5\mathrm{K}, 4.5\mathrm{K} \rangle$. This is because modern SSDs perform erase operations much less frequently compared to reads and writes as explained in Section 3.1. Unlike the other SSDs, DPES significantly increases average write latency and IOPS at $PEC = \langle 0.5\mathrm{K}, 2.5\mathrm{K} \rangle$, i.e., when the DPES scheme is applicable. Note that we do *not* evaluate i-ISPE at 4.5K PEC, as it *cannot* meet the RBER requirement before PEC reaches 4.5K as shown in Figure 21. REO+ significantly enhanced average read latency and IOPS across all PECs, exhibiting better performance at higher PECs. Figure 23 shows distribution of RR occurrence in REO and REO+ at 4.5K PEC. REO+ performs read-retry less frequently than REO by adopting SEVA, which leads to improving I/O performance.

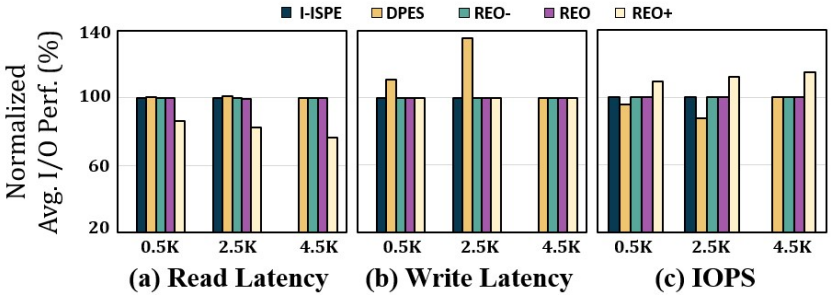


Figure 22. Average I/O performance of erase schemes normalized to baseline.

Table 4. Comparison of average I/O performance.

Erase Scheme	Geomean of Norm. Avg. Perf. at $PEC = \langle 0.5\mathrm{K}, 2.5\mathrm{K}, 4.5\mathrm{K} \rangle$		
	Norm. Avg. Read Latency [%]	Norm. Avg. Write Latency [%]	Norm. Avg. IOPS [%]
I-ISPE	⟨100.0, 99.8, N/A⟩	⟨100.0, 100.0, N/A⟩	⟨100.0, 100.1, N/A⟩
DPES	⟨100.4, 101.3, 99.9⟩	⟨110.8, 135.6, 100.0⟩	⟨95.7, 87.8, 100.0⟩
REO-	⟨99.9, 99.7, 99.7⟩	⟨99.8, 99.9, 99.8⟩	⟨100.2, 100.3, 100.3⟩
REO	⟨99.9, 99.6, 99.7⟩	⟨99.8, 99.8, 99.9⟩	⟨100.2, 100.4, 100.3⟩
REO+	⟨86.3, 82.6, 76.3⟩	⟨99.8, 99.7, 99.7⟩	⟨109.4, 111.9, 114.9⟩

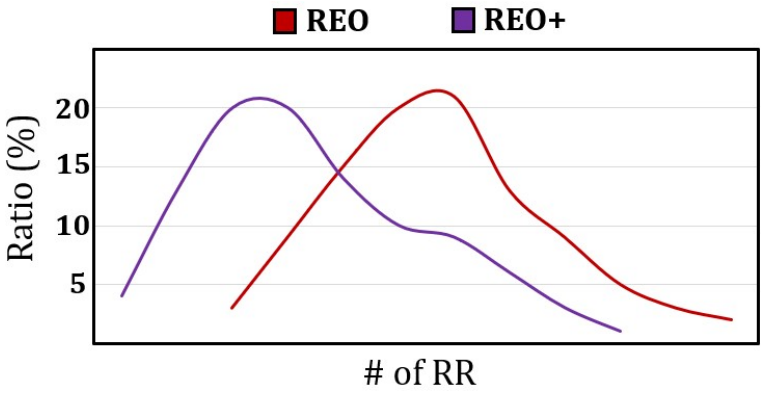


Figure 23. Distribution of the number read retires in REO and REO+ at PEC = 4.5 K.

Read Tail Latency: We evaluate the impact of REO on SSD read tail latency that is critical in modern enterprise or data center server systems [85,86]. Figure 24 represents the 99.99th and 99.9999th percentile read latencies ($\tau_{99.99P}$ and $\tau_{99.9999P}$, respectively) in the five simulated SSDs at $PEC = \langle 0.5K, 2.5K, 4.5K \rangle$ (all values are normalized to Baseline).

We make seven observations from Figure 24. First, REO (REO-) significantly reduces $\tau_{99.99P}$ and $\tau_{99.9999P}$ compared to Baseline by 22% (18%) and 26% (20%), respectively, on average across all the evaluated workloads and PEC. Second, REO achieves higher performance benefits at lower PEC while still providing considerable performance improvements at high PEC. REO (REO-) outperforms Baseline by $\langle 35\%, 24\%, 9\% \rangle$ ($\langle 32\%, 15\%, 7\% \rangle$) when $PEC = \langle 0.5K, 2.5K, 4.5K \rangle$, reducing $\tau_{99.99P}$ and $\tau_{99.9999P}$ compared to Baseline by $\langle 26\%, 25\%, 13\% \rangle$ ($\langle 26\%, 16\%, 11\% \rangle$) and $\langle 43\%, 23\%, 5\% \rangle$ ($\langle 39\%, 14\%, 2\% \rangle$) when $PEC = \langle 0.5K, 2.5K, 4.5K \rangle$. This is because REO only reduces `tEP` in EP(N_{ISPE}), which has a higher impact when N_{ISPE} is low. The high benefits at 0.5K PEC (Figure 24a) also clearly show the high effectiveness of shallow erasure, given that $N_{ISPE} = 1$ for most blocks. Third, REO improves I/O performance also when the workload is read-dominant (e.g., ali.E and usr). This highlights the importance of optimizing the latency of erase operations that dictate read tail latency. Fourth, at 2.5K PEC, REO considerably reduces $\tau_{99.99P}$ and $\tau_{99.9999P}$ over REO- by 11% on average (up to 34% and 22%, respectively), which shows the effectiveness of leveraging the ECC-capability margin for further `tEP` reduction. Fifth, REO also reduces $\tau_{99.99P}$ and $\tau_{99.9999P}$ over i-ISPE by $\langle 26\%, 20\% \rangle$ and $\langle 43\%, 23\% \rangle$ at $PEC = \langle 0.5K, 2.5K \rangle$, respectively, on average across all the evaluated workloads. At 0.5K PEC, both REO and i-ISPE can completely erase almost every block via a single loop for which only REO can reduce `tBERS` using shallow erasure. Even though i-ISPE can also decrease N_{ISPE} (and thus `tBERS`) to less than 2 (7 ms) at 2.5K PEC by skipping the first erase loops, REO achieves higher benefits due to aggressive `tEP` reduction and slower N_{ISPE} increase (92% of the blocks can be erased within two erase loops at 2.5K PEC). Sixth, REO reduces $\tau_{99.99P}$ and $\tau_{99.9999P}$ over DPES by 22% and 25%, respectively, on average across all the evaluated workloads and PEC. In particular, at 2.5K PEC, $\tau_{99.99P}$ of DPES rather often *increases* by 5% compared to Baseline due to the increase in `tPROG`, whereas REO causes no performance degradation but provides 25% benefits on average across all workloads. Seventh, REO+ outperforms $\tau_{99.99P}$ and $\tau_{99.9999P}$ over REO by 4% and 6%, respectively, on average across all the evaluated workloads and PEC. As PEC increases, the benefit of REO+ is more evident because REO+ efficiently suppresses the occurrence of read-retry during read operations.

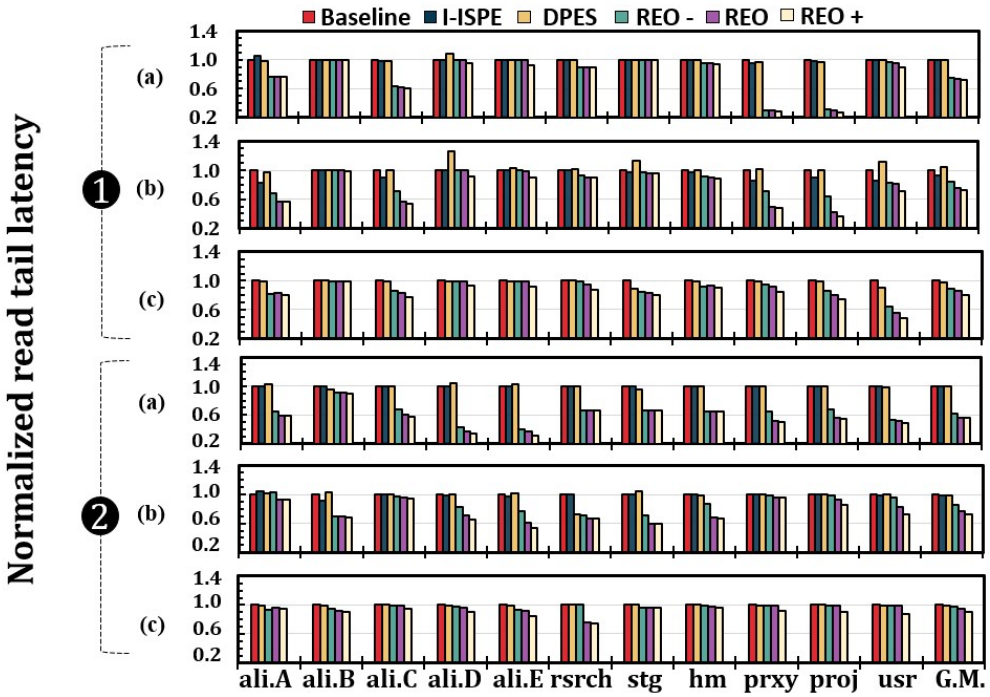


Figure 24. The 99.99th (❶) and 99.9999th (❷) percentile read latency at $PEC = \langle$(**a**) 0.5K, (**b**) 2.5K, and (**c**) 4.5K$\rangle$.

7.4. Sensitivity Analysis

Impact of Misprediction: Even though we have observed *no* misprediction in our real-device characterization, we analyze the performance and lifetime impact of misprediction in REO since it is improbable but not impossible to happen. To this end, we make two assumptions on REO's misprediction behavior based on our real-device characterization results. First, we consider each erase latency prediction of REO as an independent trial with a constant failure (i.e., misprediction) rate for all blocks and operating conditions. This is because, although reliability characteristics significantly vary across blocks and operating conditions, REO can accurately predict the minimum erase latency for all tested chips, blocks, and operating conditions as demonstrated in Section 5 (i.e., we observe nothing suggesting that certain chips, blocks, or operating conditions are more prone to REO's misprediction). Second, we assume that REO performs an additional 0.5 ms EP step for each misprediction. We believe that 0.5 ms is long enough for REO to handle a misprediction because the target block must be largely (though not completely) erased even when a misprediction happens (otherwise, REO would not have reduced erase latency). Figure 25 shows how REO's misprediction rate affects its benefits in SSD lifetime (left) and read tail latency (right).

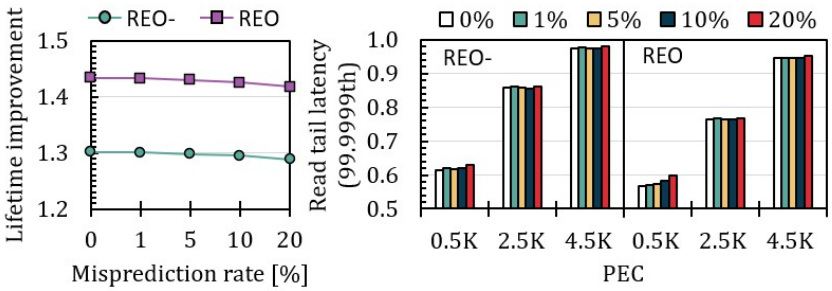


Figure 25. Impact of misprediction rate on REO's benefits.

We make two key observations. First, REO is highly effective at improving both SSD lifetime and I/O performance even when mispredictions happen. Even under a high misprediction rate of 20%, REO (REO-) provides 42% (29%) and 40% (37%) improvements over Baseline in SSD lifetime and read tail latency (at 0.5K PEC), respectively. This is because REO- and REO are still able to reduce erase latency when a misprediction happens, i.e., the amount of erase latency reductions (e.g., up to 3.5 ms) is higher than the misprediction penalty (e.g., 0.5 ms), in many cases. Second, the performance impact of misprediction becomes even smaller as PEC increases. Compared to when no misprediction occurs, 20% misprediction rate causes small increases (5.3% and 2.6% at 0.5K PEC in REO and REO-, respectively) in $\tau_{99.9999P}$, which significantly decreases (to 0.4% for both) at 4.5K PEC. This is because the total erase latency severely increases with PEC, thereby making the performance impact of misprediction much smaller.

Impact of Reliability Margin: We evaluate how REO's benefits change depending on the reliability margin that directly affects the effectiveness of aggressive `tEP` reduction. To this end, we evaluate the performance and lifetime benefits of REO while reducing the reliability requirement (i.e., the maximum raw bit errors per 1 KiB) to 40 and 50 (from 63), which can happen when using weaker ECC. Figure 26 shows SSD lifetime (left) and read tail latency (right) of REO- and REO under different reliability requirements, normalized to Baseline. Note that the lifetime of Baseline and REO- also degrades as the reliability requirement decreases (because they can tolerate fewer errors).

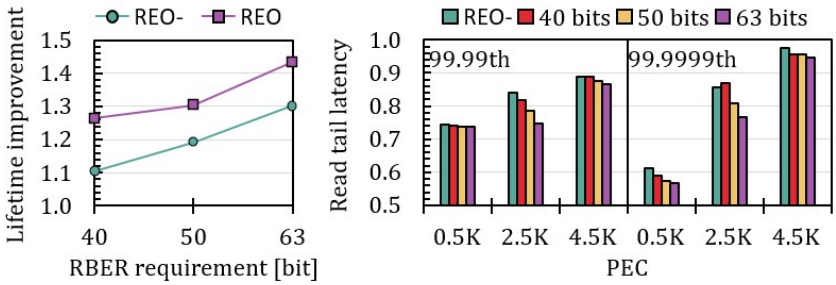


Figure 26. Impact of RBER requirement on REO's benefits.

We observe that REO can still improve SSD lifetime when the RBER requirement decreases considerably. Although the chance for aggressive `tEP` reduction significantly decreases when the RBER requirement is 40 bits (only if $N_{ISPE} = 2$ and $F(1) < \gamma$ as shown in Figure 17b), it still allows 14% lifetime enhancement over REO-. In particular, REO achieves the highest benefit at 2.5K PEC. This is because REO can completely erase most blocks with $N_{ISPE} \leq 3$, which allows it to aggressively reduce `tEP` for many blocks ($N_{ISPE} \leq 3$ and $F(N_{ISPE} - 1) < \delta$ in Figure 17b).

Based on the key observations in our evaluation, we conclude that REO is highly effective at improving both SSD lifetime and I/O performance for many real-world workloads under varying operating conditions. We believe that REO is a promising solution, considering its high lifetime and performance benefits that come with almost negligible overheads.

8. Related Work

To our knowledge, this work is the first to dynamically adjust erase latency by leveraging varying erase characteristics across blocks, providing significant lifetime and performance benefits for modern SSDs. We already discussed and compared to the state-of-the-art techniques [16,31] closely related to REO (Sections 3.4 and 7). We briefly describe other related work that aims to improve the lifetime and performance of SSDs.

Mitigating Negative Impact of Erase Operation: A large body of prior work has proposed various techniques to mitigate the negative impact of erase operations on SSD lifetime and I/O performance. Many studies have optimized the algorithms of internal SSD management tasks, e.g., garbage collection [17–25] and wear leveling [26–28], to reduce the number of erase operations invoked for servicing the same amount of user writes. To prevent an erase operation from delaying latency-sensitive reads for a long time, some studies propose to suspend an ongoing erase operation to service user reads (and resume the erase operation after completing the reads) [12,13]. Despite the significant lifetime and performance improvements made by the prior research, the existing techniques erase a block using the conventional ISPE scheme, thereby causing over-erasure of blocks frequently. REO introduces only small implementation overheads and thus can be easily integrated into the existing techniques to further improve the lifetime and performance of modern SSDs.

Process Variation: Many prior studies [11,15,32,34–37] leverage varying physical characteristics across flash cells to optimize modern SSDs. Hong et al. [11] propose a new erase scheme that applies a low voltage to error-prone WLs selectively (while keeping the same voltage for the other WLs), which makes only a small fraction of weak WLs (temporarily) unusable but eventually extends SSD lifetime. To fully utilize the potential lifetime of NAND flash blocks, Kim et al. [15] propose a new block wear index that can reflect significant endurance variation across blocks. Shim et al. [32] propose to skip some program-verify steps to improve I/O performance if the target WL has better reliability characteristics compared to other WLs. Out of many process-variation-aware optimizations, to our knowledge, our work is the first to identify a new optimization opportunity to improve both SSD lifetime and I/O performance by leveraging the significant variation in the minimum erase latency across blocks.

9. Conclusions

We propose REO, a new block erasure scheme that significantly improves both the lifetime and performance of modern NAND flash-based SSDs by dynamically adjusting erase latency and erase voltage. We identify new opportunities to optimize erase latency by leveraging varying characteristics across flash blocks and the large reliability margin in modern SSDs. We can also mitigate over-erased stress by optimizing effective erase voltage to each WL depending on its characteristics with little modification of SSDs. Throughout extensive characterization of 160 real 3D NAND flash chips, we demonstrate that it is possible to (i) accurately predict the minimum latency just long enough to completely erase a block based on in-execution information (i.e., fail-bit count), (ii) aggressively yet safely reduce erase latency by exploiting the reliability margin, and (iii) alleviate excessive erase stress by selectively modifying the WL gate voltage during the erase operation. Our results show that REO effectively improves SSD lifetime, I/O performance, and read tail latency with low overheads for diverse real-world enterprise and data center workloads under varying operating conditions. To strengthen the validity of our model, we plan to evaluate its effectiveness using advanced QLC NAND flash memory. In addition, we will confirm that our model can work under various operating conditions, such as varying temperature environments. For state-of-the-art applications like LLMs and in data centers, we plan to evaluate the impact of our model on power consumption.

Author Contributions: Investigation, B.K.; Writing—original draft, B.K.; Writing—review and editing, B.K. and M.K.; Visualization, B.K.; Supervision, M.K.; Project administration, M.K.; Funding acquisition, M.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Institute for Information and Communications Technology Promotion, grant number RS-2024-00347394; National Research Foundation of Korea, grant number RS-202400414964; National Research Foundation of Korea grant funded by the Korea government, grant number NRF-2022R1I1A3073170.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Terminology Summary

Table A1 summarizes new terminologies defined in this work.

Table A1. Summary of newly defined terminologies.

Terminology	Definition
N_{ISPE}	Number of erase loops for complete erasure
VR(i) / EP(i)	i-th Verify-Read / Erase-Pulse step
$F(i)$	Number of fail bits after EP(i)
F_{PASS}	Predefined erase pass threshold
F_{HIGH}	Full erase pulse threshold
`tEP` / `tVR`	Erase-Pulse / Verify-Read latency
V_{WG}	WL Gate Voltage
$m_{\texttt{tBERS}}$ / $m_{\texttt{tEP}}(i)$	Minimum `tBERS` / `tEP` (i)
M_{RBER}	Maximum raw bit errors

References

1. Samsung. Samsung Enterprise SSDs. 2023. Available online: https://semiconductor.samsung.com/ssd/enterprise-ssd (accessed on 9 February 2025).
2. SK Hynix. SK Hynix Enterprise SSDs. 2023. Available online: https://product.skhynix.com/products/ssd/essd.go (accessed on 9 February 2025).
3. Micron. Micron Enterprise SSDs. 2023. Available online: https://www.micron.com/products/ssd/product-lines/9400 (accessed on 9 February 2025).
4. Western Digital. Western Digital Data Center SSDs. 2023. Available online: https://github.com/axboe/fio (accessed on 9 February 2025).
5. Wong, H.S.P.; Raoux, S.; Kim, S.; Liang, J.; Reifenberg, J.P.; Rajendran, B.; Asheghi, M.; Goodson, K.E. Phase Change Memory. *Proc. IEEE* **2010**, *98*, 2201–2227. [CrossRef]
6. Zangeneh, M.; Joshi, A. Design and Optimization of Nonvolatile Multibit 1T1R Resistive RAM. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2013**, *22*, 1815–1828. [CrossRef]
7. Aggarwal, S. STT–MRAM: High Density Persistent Memory Solution. Available online: https://www.flashmemorysummit.com/Proceedings2019/08-07-Wednesday/20190807_NEWM-202B-1_Aggarwal.pdf (accessed on 9 February 2025).
8. Kawashima, S.; Cross, J.S. *FeRAM*; Springer: New York, NY, USA, 2009.
9. Cho, J.; Kang, D.C.; Park, J.; Nam, S.W.; Song, J.H.; Jung, B.K.; Lyu, J.; Lee, H.; Kim, W.-T.; Jeon, H.; et al. 30.3 A 512Gb 3b/Cell 7 th-Generation 3D-NAND Flash Memory with 184MB/s Write Throughput and 2.0 Gb/s Interface. In Proceedings of the 2021 IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, CA, USA, 13–22 February 2021.
10. Kim, M.; Yun, S.W.; Park, J.; Park, H.K.; Lee, J.; Kim, Y.S.; Na, D.; Choi, S.; Song, Y.; Lee, J.; et al. A 1Tb 3b/Cell 8th-Generation 3D-NAND Flash Memory with 164MB/s Write Throughput and a 2.4Gb/s Interface. In Proceedings of the 2022 IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, CA, USA, 20–26 February 2022.
11. Hong, D.; Kim, M.; Cho, G.; Lee, D.; Kim, J. GuardedErase: Extending SSD Lifetimes by Protecting Weak Wordlines. In Proceedings of the 20th USENIX Conference on File and Storage Technologies (FAST), Santa Clara, CA, USA, 22–24 February 2022.
12. Wu, G.; He, X. Reducing SSD Read Latency via NAND Flash Program and Erase Suspension. In Proceedings of the 10th USENIX Conference on File and Storage Technologies (FAST), San Jose, CA, USA, 15–17 February 2012.
13. Kim, S.; Bae, J.; Jang, H.; Jin, W.; Gong, J.; Lee, S.; Ham, T.J.; Lee, J.W. Practical Erase Suspension for Modern Low-latency SSDs. In Proceedings of the 2019 USENIX Annual Technical Conference (ATC), Renton, WA, USA, 10–12 July 2019.

14. Cai, Y.; Ghose, S.; Haratsch, E.F.; Luo, Y.; Mutlu, O. Error Characterization, Mitigation, and Recovery in Flash-Memory-Based Solid-State Drives. *Proc. IEEE* **2017**, *105*, 1666–1704. [CrossRef]
15. Kim, M.; Chun, M.; Hong, D.; Kim, Y.; Cho, G.; Lee, D.; Kim, J. RealWear: Improving performance and lifetime of SSDs using a NAND aging marker. *Perform. Eval.* **2021**, *48*, 120–121. [CrossRef]
16. Lee, D.W.; Cho, S.; Kang, B.W.; Park, S.; Park, B.; Cho, M.K.; Ahn, K.O.; Yang, Y.S.; Park, S.W. The Operation Algorithm for Improving the Reliability of TLC (Triple Level Cell) NAND Flash Characteristics. In Proceedings of the 2011 3rd IEEE International Memory Workshop (IMW), Monterey, CA, USA, 22–25 May 2011.
17. Lee, J.; Kim, Y.; Shipman, G.M.; Oral, S.; Wang, F.; Kim, J. A Semi-Preemptive Garbage Collector for Solid State Drives. In Proceedings of the IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), Austin, TX, USA, 10–12 April 2011.
18. Cui, J.; Zhang, Y.; Huang, J.; Wu, W.; Yang, J. ShadowGC: Cooperative Garbage Collection with Multi-level Buffer for Performance Improvement in NAND flash-based SSDs. In Proceedings of the 2018 Design, Automation & Test in Europe Conference & Exhibition (DATE), Dresden, Germany, 19–23 March 2018.
19. Kang, W.; Yoo, S. Dynamic Management of Key States for Reinforcement Learning-Assisted Garbage Collection to Reduce Long Tail Latency in SSD. In Proceedings of the 55th Annual Design Automation Conference (DAC), San Francisco, CA, USA, 24–29 June 2018.
20. Shahidi, N.; Kandemir, M.T.; Arjomand, M.; Das, C.R.; Jung, M.; Sivasubramaniam, A. Exploring the Potentials of Parallel Garbage Collection in SSDs for Enterprise Storage Systems. In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC), Salt Lake City, UT, USA, 13–18 November 2016.
21. Choi, W.; Jung, M.; Kandemir, M.; Das, C. Parallelizing Garbage Collection with I/O to Improve Flash Resource Utilization. In Proceedings of the 27th International Symposium on High-Performance Parallel and Distributed Computing (HPDC), Tempe, AZ, USA, 11–15 June 2018.
22. Guo, J.; Hu, Y.; Mao, B.; Wu, S. Parallelism and Garbage Collection Aware I/O Scheduler with Improved SSD Performance. In Proceedings of the 2017 IEEE International Parallel and Distributed Processing Symposium (IPDPS), Orlando, FL, USA, 19 May–2 June 2017.
23. Kang, W.; Shin, D.; Yoo, S. Reinforcement Learning-Assisted Garbage Collection to Mitigate Long-Tail Latency in SSD. *ACM Trans. Embed. Comput. Syst. (TECS)* **2017**, *16*, 1–20. [CrossRef]
24. Yang, P.; Xue, N.; Zhang, Y.; Zhou, Y.; Sun, L.; Chen, W.; Chen, Z.; Xia, W.; Li, J.; Kwon, K. Reducing Garbage Collection Overhead in SSD Based on Workload Prediction. In Proceedings of the 11th USENIX Workshop on Hot Topics in Storage and File Systems (HotStorage), Renton, WA, USA, 8–9 July 2019.
25. Lee, J.; Kim, Y.; Shipman, G.M.; Oral, S.; Kim, J. Preemptible I/O Scheduling of Garbage Collection for Solid State Drives. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. (TCAD)* **2013**, *32*, 247–260. [CrossRef]
26. Murugan, M.; Du, D. Rejuvenator: A Static Wear Leveling Algorithm for NAND Flash Memory with Minimized Overhead. In Proceedings of the 2011 IEEE 27th Symposium on Mass Storage Systems and Technologies (MSST), Denver, CO, USA, 23–27 May 2011.
27. Li, J.; Xu, X.; Peng, X.; Liao, J. Pattern-based Write Scheduling and Read Balance-oriented Wear-Leveling for Solid State Drivers. In Proceedings of the 2019 35th Symposium on Mass Storage Systems and Technologies (MSST), Santa Clara, CA, USA, 20–24 May 2019.
28. Dharamjeet.; Chen, Y.S.; Chen, T.Y.; Kuan, Y.H.; Chang, Y.H. LLSM: A Lifetime-Aware Wear-Leveling for LSM-Tree on NAND Flash Memory. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. (TCAD)* **2022**, *41*, 3946–3956. [CrossRef]
29. Jeong, J.; Hahn, S.S.; Lee, S.; Kim, J. Improving NAND Endurance by Dynamic Program and Erase Scaling. In Proceedings of the 5th USENIX Workshop on Hot Topics in Storage and File Systems (HotStorage), San Jose, CA, USA, 27–28 June 2013.
30. Jeong, J.; Hahn, S.S.; Lee, S.; Kim, J. Lifetime Improvement of NAND Flash-based Storage Systems Using Dynamic Program and Erase Scaling. In Proceedings of the 12th USENIX Conference on File and Storage Technologies (FAST), Santa Clara, CA, USA, 2014.
31. Jeong, J.; Youngsun, S.; Hahn, S.S.; Lee, S.; Kim, J. Dynamic Erase Voltage and Time Scaling for Extending Lifetime of NAND Flash-Based SSDs. *IEEE Trans. Comput. (TC)* **2017**, *66*, 616–630. [CrossRef]
32. Shim, Y.; Kim, M.; Chun, M.; Park, J.; Kim, Y.; Kim, J. Exploiting Process Similarity of 3D Flash Memory for High Performance SSDs. In Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), Columbus, OH, USA, 12–16 October 2019.
33. Luo, Y.; Ghose, S.; Cai, Y.; Haratsch, E.F.; Mutlu, O. Improving 3D NAND Flash Memory Lifetime by Tolerating Early Retention Loss and Process Variation. *Proc. ACM Meas. Anal. Comput. Syst. (POMACS)* **2018**, *2*, 1–48. [CrossRef]
34. Wang, Y.; Dong, L.; Mao, R. P-Alloc: Process-Variation Tolerant Reliability Management for 3D Charge-Trapping Flash Memory. *ACM Trans. Embed. Comput. Syst. (TECS)* **2017**, *16*, 1–19. [CrossRef]

35. Chen, S.H.; Chen, Y.T.; Wei, H.W.; Shih, W.K. Boosting the Performance of 3D Charge Trap NAND Flash with Asymmetric Feature Process Size Characteristic. In Proceedings of the 2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC), San Francisco, CA, USA, 18–22 June 2017.
36. Hung, C.H.; Chang, M.F.; Yang, Y.S.; Kuo, Y.J.; Lai, T.N.; Shen, S.J.; Hsu, J.Y.; Hung, S.N.; Lue, H.T.; Shih, Y.H.; et al. Layer-Aware Program-and-Read Schemes for 3D Stackable Vertical-Gate BE-SONOS NAND Flash Against Cross-Layer Process Variations. *IEEE J. Solid-State Circuits (JSSC)* **2015**, *50*, 1491–1501. [CrossRef]
37. Yen, J.N.; Hsieh, Y.C.; Chen, C.Y.; Chen, T.Y.; Yang, C.L.; Cheng, H.Y.; Luo, Y. Efficient Bad Block Management with Cluster Similarity. In Proceedings of the 2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA), Seoul, Republic of Korea, 2–6 April 2022.
38. Li, Q.; Ye, M.; Cui, Y.; Shi, L.; Li, X.; Kuo, T.W.; Xue, C.J. Shaving Retries with Sentinels for Fast Read over High-Density 3D Flash. In Proceedings of the 2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), Athens, Greece, 17–21 October 2020.
39. Cai, Y.; Haratsch, E.F.; Mutlu, O.; Mai, K. Threshold Voltage Distribution in MLC NAND Flash Memory: Characterization, Analysis, and Modeling. In Proceedings of the 2013 Design, Automation & Test in Europe Conference & Exhibition (DATE), Grenoble, France, 18–22 March 2013.
40. Cai, Y.; Luo, Y.; Ghose, S.; Mutlu, O. Read Disturb Errors in MLC NAND Flash Memory: Characterization, Mitigation, and Recovery. In Proceedings of the 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Rio de Janeiro, Brazil, 22–25 June 2015.
41. Park, J.; Kim, M.; Chun, M.; Orosa, L.; Kim, J.; Mutlu, O. Reducing Solid-State Drive Read Latency by Optimizing Read-Retry. In Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), Virtual, 19–23 April 2021.
42. Tavakkol, A.; Gómez-Luna, J.; Sadrosadati, M.; Ghose, S.; Mutlu, O. MQSim: A Framework for Enabling Realistic Studies of Modern Multi-Queue SSD Devices. In Proceedings of the 16th USENIX Conference on File and Storage Technologies (FAST), Oakland, CA, USA, 12–15 February 2018.
43. Maserjian, J.; Zamani, N. Behavior of the Si/SiO2 interface observed by Fowler-Nordheim tunneling. *J. Appl. Phys.* **1982**, *53*, 559–567. [CrossRef]
44. Park, J.; Azizi, R.; Oliveira, G.F.; Sadrosadati, M.; Nadig, R.; Novo, D.; Gómez-Luna, J.; Kim, M.; Mutlu, O. Flash-Cosmos: In-Flash Bulk Bitwise Operations Using Inherent Computation Capability of NAND Flash Memory. In Proceedings of the 2022 55th IEEE/ACM International Symposium on Microarchitecture (MICRO), Chicago, IL, USA, 1–10 October 2022.
45. Choi, J.; Seol, K.S. 3D approaches for non-volatile memory. In Proceedings of the Symposium on VLSI Technology-Digest of Technical Papers, Kyoto, Japan, 15–17 June 2011.
46. Park, Y.; Lee, J.; Cho, S.S.; Jin, G.; Jung, E. Scaling and reliability of NAND flash devices. In Proceedings of the IEEE International Reliability Physics Symposium (IRPS), Monterey, CA, USA, 14–18 April 2014.
47. Jang, J.; Kim, H.S.; Cho, W.; Cho, H.; Kim, J.; Shim, S.I.; Younggoan; Jeong, J.H.; Son, B.K.; Kim, D.W.; et al. Vertical cell array using TCAT(Terabit Cell Array Transistor) technology for ultra high density NAND flash memory. In Proceedings of the 2009 Symposium on VLSI Technology, Kyoto, Japan, 16–18 June 2009.
48. Ha, K.; Jeong, J.; Kim, J. An Integrated Approach for Managing Read Disturbs in High-Density NAND Flash Memory. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. (TCAD)* **2016**, *35*, 1079–1091. [CrossRef]
49. Liu, C.Y.; Lee, Y.; Jung, M.; Kandemir, M.T.; Choi, W. Prolonging 3D NAND SSD Lifetime via Read Latency Relaxation. In Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), 12–23 April 2021.
50. Cai, Y.; Mutlu, O.; Haratsch, E.F.; Mai, K. Program Interference in MLC NAND Flash Memory: Characterization, Modeling, and Mitigation. In Proceedings of the IEEE 31st International Conference on Computer Design (ICCD), Asheville, NC, USA, 6–9 October 2013.
51. Park, J.; Jeong, J.; Lee, S.; Song, Y.; Kim, J. Improving Performance and Lifetime of NAND Storage Systems Using Relaxed Program Sequence. In Proceedings of the 53nd ACM/EDAC/IEEE Design Automation Conference (DAC), Austin, TX, USA, 5–9 June 2016.
52. Kim, M.; Lee, J.; Lee, S.; Park, J.; Kim, J. Improving Performance and Lifetime of Large-page NAND Storages Using Erase-Free Subpage Programming. In Proceedings of the 54th ACM/EDAC/IEEE Design Automation Conference (DAC), San Francisco, USA, 18–22 June 2017.
53. Cai, Y.; Ghose, S.; Luo, Y.; Mai, K.; Mutlu, O.; Haratsch, E.F. Vulnerabilities in MLC NAND Flash Memory Programming: Experimental Analysis, Exploits, and Mitigation Techniques. In Proceedings of the IEEE International Symposium on High Performance Computer Architecture (HPCA), Austin, TX, USA, 4–8 February 2017.
54. Gallager, R. Low-Density Parity-Check Codes. *IRE Trans. Inf. Theory* **1962**, *8*, 21–28. [CrossRef]
55. Micheloni, R.; Marelli, A.; Eshghi, K. *Inside Solid State Drives (SSDs)*; Springer: Berlin, Germany, 2012.

56. Yan, S.; Li, H.; Hao, M.; Tong, M.H.; Sundararaman, S.; Chien, A.A.; Gunawi, H.S. Tiny-Tail Flash: Near-Perfect Elimination of Garbage Collection Tail Latencies in NAND SSDs. *ACM Trans. Storage* **2017**, *13*, 1–26. [CrossRef]
57. Schuegraf, K.; Hu, C. Effects of temperature and defects on breakdown lifetime of thin SiO/sub 2/at very low voltages. In Proceedings of the IEEE International Reliability Physics Symposium (IRPS), San Jose, CA, USA, 11–14 April 1994.
58. Yuh, J.; Li, J.; Li, H.; Oyama, Y.; Hsu, C.; Anantula, P.; Jeong, S.; Amarnath, A.; Darne, S.; Bhatia, S.; et al. A 1-Tb 4b/Cell 4-Plane 162-Layer 3D Flash Memory With a 2.4-Gb/s I/O Speed Interface. In Proceedings of the 2022 IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, CA, USA, 20–26 February 2022.
59. Kim, B.; Lee, S.; Hah, B.; Park, K.; Park, Y.; Jo, K.; Noh, Y.; Seol, H.; Lee, H.; Shin, J.; et al. 28.2 A High-Performance 1Tb 3b/Cell 3D-NAND Flash with a 194MB/s Write Throughput on over 300 Layers i. In Proceedings of the 2023 IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, CA, USA, 19–23 February 2023.
60. Micheloni, R.; Crippa, L.; Marelli, A. *Inside NAND Flash Memories*; Springer: Berlin, Germany, 2010.
61. Hollmer, S.C.; Hu, C.Y.; Le, B.Q.; Chen, P.L.; Su, J.; Gutala, R.; Bill, C. Erase Verify Scheme for NAND Flash. US Patent 6,009,014, 28 December 1999.
62. Aritome, S. *NAND Flash Memory Technologies*; Wiley & Sons: Hoboken, NJ, USA, 2015.
63. Ito, T.; Taito, Y. SONOS Split-Gate eFlash Memory. In *Embedded Flash Memory for Embedded Systems: Technology, Design for Sub-Systems, and Innovations*; Springer: Berlin, Germany, 2018.
64. Chen, S.H.; Chang, Y.H.; Liang, Y.P.; Wei, H.W.; Shih, W.K. An Erase Efficiency Boosting Strategy for 3D Charge Trap NAND Flash. *IEEE Trans. Comput. (TC)* **2018**, *67*, 1246–1258. [CrossRef]
65. Lue, H.T.; Hsu, T.H.; Wu, C.J.; Chen, W.C.; Yeh, T.H.; Chang, K.P.; Hsieh, C.C.; Du, P.Y.; Hsiao, Y.H.; Jiang, Y.W.; et al. A Novel Double-density, Single-Gate Vertical Channel (SGVC) 3D NAND Flash That Is Tolerant to Deep Vertical Etching CD Variation and Possesses Robust Read-disturb Immunity. In Proceedings of the 2015 IEEE International Electron Devices Meeting (IEDM), San Francisco, CA, USA, 7–9 December 2015.
66. Cha, J.; Kang, S. Data Randomization Scheme for Endurance Enhancement and Interference Mitigation of Multilevel Flash Memory Devices. *Etri J.* **2013**, *35*, 166–169. [CrossRef]
67. Favalli, M.; Zambelli, C.; Marelli, A.; Micheloni, R.; Olivo, P. A Scalable Bidimensional Randomization Scheme for TLC 3D NAND Flash Memories. *Micromachines* **2021**, *12*, 759. [CrossRef]
68. ONFI Workgroup. Open NAND Flash Interface Specification 4.1. Available online: https://onfi.org/files/onfi_4_1_gold.pdf (accessed on 9 February 2025).
69. Kang, D.; Jeong, W.; Kim, C.; Kim, D.H.; Cho, Y.S.; Kang, K.T.; Ryu, J.; Kang, K.M.; Lee, S.; Kim, W.; et al. 256Gb 3b/cell V-NAND Flash Memory with 48 Stacked WL Layers. In Proceedings of the 2016 IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, CA, USA, 31 January–4 February 2016.
70. JEDEC. JESD218B.02: Solid-State Drive (SSD) Requirements and Endurance Test Method. 2022. Available online: https://www.jedec.org/standards-documents/docs/jesd218b01 (accessed on 9 February 2025).
71. Arrhenius, S. Über die Dissociationswärme und den Einfluss der Temperatur auf den Dissociationsgrad der Elektrolyte. *Z. Phys. Chem.* **1889**, *4*, 96–116. [CrossRef]
72. Micheloni, R. *3D Flash Memories*; Springer: Berlin, Germany, 2016.
73. Vättö, K. Samsung SSD 840: Testing the Endurance of TLC NAND. 2012. Available online: https://www.anandtech.com/show/6459/samsung-ssd-840-testing-the-endurance-of-tlc-nand (accessed on 9 February 2025).
74. Tallis, B. The Samsung 960 Pro (2TB) SSD Review. 2016. Available online: https://www.anandtech.com/show/10754/samsung-960-pro-ssd-review (accessed on 9 February 2025).
75. Gupta, A.; Kim, Y.; Urgaonkar, B. DFTL: A Flash Translation Layer Employing Demand-Based Selective Caching of Page-Level Address Mappings. In Proceedings of the 14th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), Washington, DC, USA, 7–11 March 2009.
76. Weisberg, P.; Wiseman, Y. Using 4KB page size for Virtual Memory is obsolete. In Proceedings of the 2009 IEEE International Conference on Information Reuse & Integration, Las Vegas, NV, USA, 10–12 August 2009.
77. Chang, L.P.; Kuo, T.W. An Adaptive Striping Architecture for Flash Memory Storage Systems of Embedded Systems. In Proceedings of the Eighth IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), San Jose, CA, USA, 27 September 2002.
78. Li, J.; Wang, Q.; Lee, P.P.C.; Shi, C. An In-Depth Analysis of Cloud Block Storage Workloads in Large-Scale Production. In Proceedings of the 2020 IEEE International Symposium on Workload Characterization (IISWC), Beijing, China, 27–30 October 2020.
79. Narayanan, D.; Donnelly, A.; Rowstron, A. Write Off-Loading: Practical Power Management for Enterprise Storage. *ACM Trans. Storage (TOS)* **2008**, *4*, 1–23. [CrossRef]
80. Nadig, R.; Sadrosadati, M.; Mao, H.; Ghiasi, N.M.; Tavakkol, A.; Park, J.; Sarbazi-Azad, H.; Luna, J.G.; Mutlu, O. Venice: Improving Solid-State Drive Parallelism at Low Cost via Conflict-Free Accesses. In Proceedings of the 50th Annual International Symposium on Computer Architecture (ISCA), Orlando, FL, USA, 17–21 June 2023.

81. Liu, R.; Tan, Z.; Long, L.; Wu, Y.; Tan, Y.; Liu, D. Improving Fairness for SSD Devices through DRAM Over-Provisioning Cache Management. *IEEE Trans. Parallel Distrib. Syst. (TPDS)* **2022**, *33*, 2444–2454. [CrossRef]
82. Liu, R.; Liu, D.; Chen, X.; Tan, Y.; Zhang, R.; Liang, L. Self-Adapting Channel Allocation for Multiple Tenants Sharing SSD Devices. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. (TCAD)* **2022**, *41*, 294–305. [CrossRef]
83. Lv, Y.; Shi, L.; Song, Y.; Xue, C.J. Access Characteristic Guided Partition for NAND Flash based High-Density SSDs. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. (TCAD)* **2023**, *42*, 4643–4656. [CrossRef]
84. Wu, J.; Li, J.; Sha, Z.; Cai, Z.; Liao, J. Adaptive Switch on Wear Leveling for Enhancing I/O Latency and Lifetime of High-Density SSDs. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. (TCAD)* **2022**, *41*, 4040–4051. [CrossRef]
85. DeCandia, G.; Hastorun, D.; Jampani, M.; Kakulapati, G.; Lakshman, A.; Pilchin, A.; Sivasubramanian, S.; Vosshall, P.; Vogels, W. Dynamo: Amazon's Highly Available Key-Value Store. In Proceedings of the Twenty-First ACM SIGOPS Symposium on Operating Systems Principles (SOSP), Stevenson, WA, USA, 14–17 October 2007.
86. Gunawi, H.S.; Hao, M.; Suminto, R.O.; Laksono, A.; Satria, A.D.; Adityatama, J.; Eliazar, K.J. Why Does the Cloud Stop Computing? Lessons from Hundreds of Service Outages. In Proceedings of the Seventh ACM Symposium on Cloud Computing (SoCC), Santa Clara, CA, USA, 5–7 October 2016.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.