Increasing SSD Endurance for Enterprise Applications

Vidyabhushan Mohan[†] Sriram Sankar[‡] Sudhanva Gurumurthi[†]

[†]Department of Computer Science
University of Virginia
Charlottesville, VA 22904
{vm9u,gurumurthi}@virginia.edu

[‡] Microsoft Corporation
Redmond, WA
srsankar@microsoft.com

Technical Report CS-2011-04

May 2011

Abstract

NAND flash based Solid State Disks (SSDs) are fast becoming the choice of primary storage replacing the traditional Hard Disk Drive (HDD) based storage media. The power and performance benefits of SSDs over HDDs are especially attractive for use in data centers, whose workloads are I/O intensive. However, the limited lifetime of SSDs is often cited as an obstacle in adopting them for data centers. One aspect of NAND flash memory reliability that significantly hinders the adoption of SSDs in data centers is write endurance. In order to study this limitation and suggest solutions to overcome this limitation, we have built a reliability model framework called FENCE to study flash memory reliability. FENCE captures the time-dependent property of write endurance and data retention of NAND flash memory by taking into account both the stress and recovery effects on NAND flash memory cells. Using FENCE, we analyze the tradeoffs between write endurance and data retention for both SLC and MLC flash. We make a case for increasing the endurance of MLC based SSDs by trading off their data retention property. We illustrate some refresh policies that can be applied to ensure data integrity of SSDs and suggest changes to the design of flash memory controller to support these refresh operations.

1 Introduction

Flash memory has gained tremendous popularity in recent years. Although initially used only in mobile devices, such as cell phones and portable music players, the drop in the price of NAND flash memory has paved the way for its use in mass storage devices as well, in the form of Solid State Disks (SSDs). SSDs offer several advantages over Hard Disk Drives (HDDs) such as lower power, higher I/O performance (especially for random I/O), and greater ruggedness. While these advantages make SSDs an potential replacement for HDDs in laptops and desktops, the performance and power benefits are especially attractive for data centers, since many enterprise workloads are I/O intensive and perform a significant amount of random I/O.

	SLC	MLC	HDD	MLC+
Capacity of one drive	64	160	300	160
Number of drives for 1TB database (RAID10)	18	12	8	12
Estimated number of spares required for 5 years due to endurance issues	0	12 (estimate)	0	<=3(estimate)
Total drives	18	24	8	<=15
Total GB	1152	3840	2400	<=2400
$/GB	7	2	0.5	2
Total cost of storage	8064	7680	1200	<=4800
Write Performance	6000	3600	180	3600
Performance/Cost (higher the better)	0.744	0.4688	0.15	>=0.75

Table 1: Cost estimate for designing an enterprise server with different storage technologies

SSDs can be designed using Single-Level Cell (SLC) NAND Flash or Multi-Level Cell(MLC) NAND Flash chips. SLC based SSDs provide higher performance and have higher reliability but more expensive compared to MLC based SSDs. Based on current prices, an 8GB SLC chip is 3.5 times more expensive than an 8GB MLC chip [19]. In the past 4 years, MLC prices have reduced by a factor of 5 while, SLC prices have dropped by a factor of 2 [19]. Because the cost benefits of MLC based SSDs outweigh SLC based SSDs, companies have started building MLC based SSDs for use in enterprise applications [11, 33].

Despite these cost benefits and continuous drop in price of flash memory, one of the main impediments to the wide adoption of SSDs has been their limitation as a reliable storage media. Flash memory blocks can wear out after a certain number of write (program) and erase operations - a property referred to as limited write endurance. Chip manufacturer datasheets quote values that range from 10,000-100,000 program/erase (P/E) cycles for MLC and SLC flash endurance respectively. This endurance metric is typically coupled with another reliability metric called “data retention period” while rating the overall reliability of SSDs. Data retention period indicates the duration of time for which flash memory blocks can store data and can be read reliably. Chip manufacturers quote a data retention period of 5-10 years [13]. Based on these ratings, we estimated the cost of building a OLTP server that is typical in a scale out SQL deployment. We assumed that we have a 1TB database and calculated the costs for a SLC based SSD, an MLC based SSD and an HDD based solution. Because the write endurance of MLC based SSDs is an order of magnitude lesser than that of SLC based SSDs, we introduce additional spare SSDs to cover for the wear out, in addition to assuming a reliable storage solution like RAID. The results of this estimate are shown in table 1.

In Table 1, the last column (indicated as MLC+) represents an improved MLC solution which has higher endurance than the default MLC based solution, thereby reducing the need to have additional spares. Without the increased endurance, MLC based solutions are not optimal compared to the SLC counterparts. However, using expensive SLC based SSDs to build servers is cost prohibitive because of the significant capital expense involved. Table 1 represents the initial expenditure incurred in setting up a server, and if operating expenses are taken then all SSD configurations outweigh an HDD based solution because of their very low power budget. Table 1 motivates the need for high endurance among MLC based SSDs to design an optimal server solution in data centers. In this paper we have built an analytical model called FENCE to study flash reliability and have used this model to present techniques to increase the endurance of SSDs for enterprise applications.

FENCE is based on the premise that an accurate estimate of reliability of flash is possible only when we take into account the recovery process that flash memory cells undergo between successive P/E cycles. Using FENCE, we are able to show that the actual endurance of SSDs is much higher than the estimates quoted by datasheets. Using FENCE, we also analyze the impact of P/E cycles on the “data retention” of NAND flash. We show that, by trading off the data retention period of SSDs, the endurance of SSDs can be increased significantly. In order to ensure the integrity of data stored in the SSD, the SSD can be periodically refreshed without significant impact in their overall performance. In this report, we illustrate some refresh policies that can be applied to ensure data integrity while allowing us to increase the endurance of SSDs.

To summarize, the main contributions of this report are:

- We have motivated the need for higher endurance of MLC based SSDs to make them viable for use in data center environment.
- We have developed Flash EnduraNCE (FENCE), an analytical model that captures the time-dependent property of write endurance and data retention of NAND flash memory taking into account both the stress and recovery effects on NAND flash memory cells.
- Using FENCE, we quantify the impact of charge trapping and detrapping for both Single- Level Cell (SLC) and Multi-Level Cell (MLC) NAND flash memory.
- Using FENCE, we analyze the tradeoffs between write endurance and data retention for both SLC and MLC flash.
- We make a case for increasing the endurance of MLC based SSDs by trading off their data retention property. We illustrate some refresh policies that can be applied to ensure data integrity of SSDs and suggest changes to the design of flash memory controller to support these refresh operations.

The rest of the report is organized as follows: Section 2 provides a background of NAND Flash memory. Section 3 presents the analytical model that forms the crux of this work. Section 4 explains the list of workloads used, their properties and the properties of SSD used for this study. Section 5 illustrates the type of refresh policies that will be used to ensure data persistence while simultaneously increasing the endurance of SSDs. Section 6 explains the related work. Section 7 and Section 8 provide the future work and conclusion.

2 Background

This section provides an overview of how NAND flash memory operates and explains how these operations affect flash reliability. A detailed discussion on flash memory at the circuit level is given in [3]. Agrawal et al. describe the architecture of flash based SSDs [2]. Flash is a type of EEPROM (Electrically Erasable Programmable Read-Only Memory) which supports three basic operations: read, program (write), and erase. A flash memory chip consists of the flash memory array, a set of decoders and additional peripheral circuitry to perform operations. The flash memory array consists of Floating Gate Transistors (FGTs), which act as memory cells (in this proposed work, the terms memory cell and FGT refer to the same physical entity and are used interchangeably). Figure 1 represents the typical structure of a flash memory array. In addition to the FGT, which acts as the storage element, the memory array contains pass transistors to control the current through the array.

The FGT is similar to a regular NMOS transistor except for an additional floating gate between the channel and the control gate. This floating gate is isolated from the rest of the device by a dielectric (oxide).

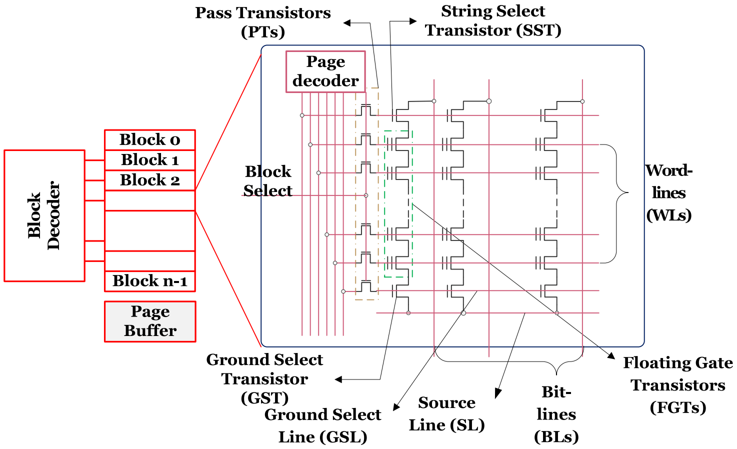


Figure 1: A NAND Flash Memory Array. Adapted from [3].

This helps retain charges on the floating gate for an extended period of time (on the order of years), hence providing non-volatility. Charges are added or removed to/from the floating gate through a process called Fowler-Nordheim tunneling (FN tunneling) which causes a shift in the threshold voltage of the FGT. This shift is sensed during the read operation to determine the logical bit corresponding to that threshold voltage. The operating voltage for FN tunneling is typically more than around 15V while that of the read operation are typically 4-5V [3].

Because high operating voltages are required for program and erase operations, they have a detrimental impact on the reliability of flash memory. Hence these operations are referred to as stress events. As a flash memory cell is repeatedly stressed, the oxide layer between the floating gate and channel gets damaged. Specifically, these stress events break the atomic bonds in the oxide layer, which increases the probability of charges getting trapped when they tunnel through the oxide layer. When charges are trapped in the tunnel oxide, it increases charge leakage from the floating gate due to a process called Trap Assisted Tunneling (TAT) [23]. This leakage current, which is exacerbated due to trap assisted tunneling under low electric fields, is referred to as Stress Induced Leakage Current (SILC). As charge trapping increases over a period of time, SILC also increases and as SILC increases, the time taken data retention period decreases.

On the other hand, endurance is a measure of the number of P/E cycles that a flash memory cell can tolerate while preserving the integrity of the stored data, and is a function of the charge trapping characteristics of the oxide [39, 40]. As every stress event increases the likelihood of charges getting trapped in the oxide, it can lead to an undesirable increase in the threshold voltage of the memory cell. If a sufficiently high number of charges get trapped in the oxide, it will no longer be possible to reliably read the cell.

Although a memory cell that undergoes a large number of stress events will have more charges trapped in its oxide, several transistor-level studies of NAND-flash memory have shown that it is possible to detrap (i.e., remove) some of the charges from the tunnel oxide under certain conditions [20, 37, 39]. Beneficial conditions for detrapping include higher external temperatures and quiescent periods between successive stress events. Furthermore, measurement studies indicate that introducing a quiescent period to allow detrapping can be applied at temperatures as low as 25°C, which is the typical external ambient temperature of a disk in a server [10]. Since the quiescent periods helps in detrapping charges from the tunnel oxide and improve retention, endurance and disturbs, they are referred to as recovery periods.

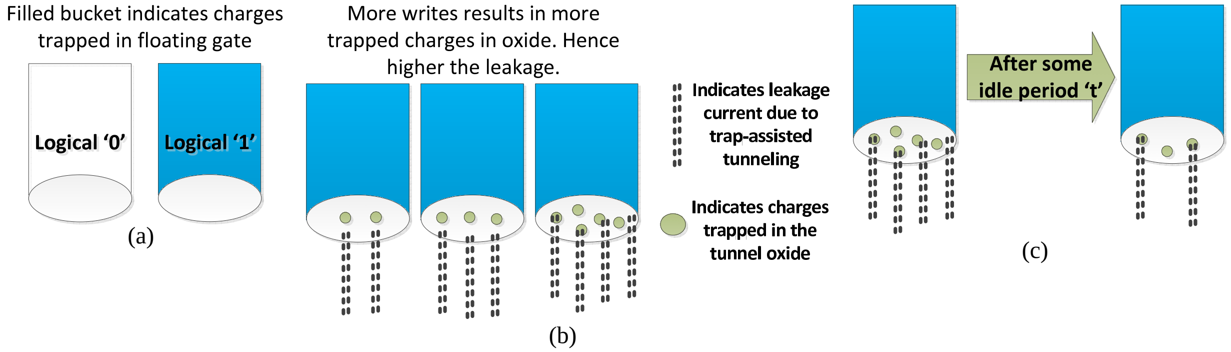

Filled bucket indicates charges trapped in floating gate
Logical '0'
Logical '1'
(a)
More writes results in more trapped charges in oxide. Hence higher the leakage.
(b)
Indicates leakage current due to trap-assisted tunneling
Indicates charges trapped in the tunnel oxide
After some idle period 't'
(c)

Figure 2: Illustration of Flash Endurance and Data Retention using Water and Bucket Analogy. (a)- Filled bucket indicates charges trapped in floating gate. (b)- Represents the wear and tear of the memory cell with increasing P/E cycles. (c) - Represents the effect of charge detrapping from the tunnel oxide after some idle period.

2.1 Bucket and Water Analogy

Figures 2(a) and 2(b) and 2(c) illustrate the impact of stress events on the endurance and data retention property of NAND flash using a bucket and water analogy. The bucket represents a the floating gate in a FGT. A filled bucket indicates the charges trapped in the floating gate (as shown in Figure 2(a)). The base of the bucket represents the tunnel oxide through which charges tunnel. In addition to tunneling of electrons into or out of the floating gate (filling or emptying the bucket), each stress event also damages the tunnel oxide (punctured holes in the bottom of the bucket). As the number of stress events increases, the number of holes in the bottom of the bucket also increases. As more charges are trapped in the tunnel oxide (more holes in the bottom of the bucket, as shown in Figure 2(b)), the rate of charge loss (leakage current) also increases, thereby reducing the retention period. When there is recovery period between successive stress events, some of the charges trapped in the oxide get detrapped (some of the holes in the bottom of the bucket get closed, as shown in Figure 2(c)), thereby improving the reliability of the memory cell.

3 FENCE - Modeling NAND Flash Memory Reliability

In order to analyze how stress events and recovery periods affect endurance and retention under various usage scenarios, I have developed an analytical model framework called FENCE for both these reliability issues. While endurance or retention failure can happen due to various distinct failure mechanisms like Time Dependent Dielectric Breakdown(TDDB), Hot Carrier Injection (HCI), Negative Bias Temperature Instability (NBTI) on both the peripherals and the the memory array, this project focuses on the failure induced on the FGTs in the memory array due to charge trapping and detrapping in the tunnel oxide after P/E cycling. Specifically, these models estimate data retention period by modeling the time to data loss due to SILC and estimate endurance by modeling the time to permanent read failure due to charge trapping. Organizations like Joint Electron Device Engineering Council (JEDEC) provide standards and documents that contain a detailed explanation of various failure modes and their effect on flash memory reliability [15]. A comprehensive modeling of flash memory reliability requires modeling all these failure modes.

The analytical models for retention and endurance are constructed by synthesizing information from device physics papers on NAND flash memory cells [14,20,21,23,39,40]. These papers provide information about how the various parameters affect endurance and retention, their relationship to each other, and values for some fitting constants used in the model. It is important to note that all these device physics papers are based on very similar transistor technology and hence are consistent with each other.

Section 3.1 explains the analytical model for endurance while Section 3.2 explains the model for data retention.

3.1 Model for Endurance

The endurance model consists of two parts: - one for stresses and the other for recovery. The first part of the model gives the relationship between the increase in threshold voltage due to charge trapping ($\delta V_{th,s}$) and the number of stress events on the tunnel oxide. The second part gives the relationship between the threshold voltage shift due to recovery ($\delta V_{th,r}$), the amount of trapped charges in the tunnel oxide due to stress calculated in the first part ($\delta V_{th,s}$), and the recovery period (t). Using these two parts, the effective increase in threshold voltage of a memory cell due to trapped charges (δV_{th}) after a stress event and a subsequent recovery period is calculated. These two components of the model are now explained in more detail.

3.1.1 The Stress Model

The threshold voltage of a memory cell increases due to charge trapping with the number of stress events (program or erase cycles) [40]. There are two types of traps that form in the tunnel oxide - interface traps and bulk traps - which contribute to the increase in the threshold voltage. It has been shown that both types of traps have a power-law relation to the number of P/E cycles on the memory cell as [40]:

$$\delta N_{it} = A * cycle^{0.62} \tag{1}$$

$$\delta N_{ot} = B * cycle^{0.30} \tag{2}$$

where A and B are constants, $cycle$ is the number of program or erase cycles on the cell, and the terms δN_{it} and δN_{ot} are the interface and bulk trap densities respectively. In addition to providing this power-law relationship, [40] also provides empirical data on how δN_{it} and δN_{ot} vary with $cycle$. The values of constants A and B are calculated to be 0.08 and 5, respectively, from this empirical data.

The total threshold voltage increase due to trapping is divided into interface trap voltage shift (δV_{it}) and bulk trap voltage shift (δV_{ot}). Park et al. [30] give the relationship between δV_{it} and δN_{it} and between δV_{ot} and δN_{ot} to be:

$$\delta V_{it} = \frac{\delta N_{it} * q}{C_{ox}} \tag{3}$$

$$\delta V_{ot} = \frac{\delta N_{ot} * q}{C_{ox}} \tag{4}$$

where q is electron charge (1.6×10^{-19} Coulombs) and C_{ox} is the capacitance of the tunnel oxide. The value of C_{ox} depends on the feature size of the NAND flash cell.

Hence the increase in threshold voltage of the memory cell due to trapped charges, $\delta V_{th,s}$, is given by:

$$\delta V_{th,s} = \delta V_{it} + \delta V_{ot} \tag{5}$$

3.1.2 The Recovery Model

According to Yamada et al. [39], the threshold voltage shift due to detrapping depends on the recovery period and the amount of charge trapped in the oxide. This relationship is given by the equation

$$\Delta V_{th,r} = c_{vt}.ln(t) \tag{6}$$

where t is the recovery period between successive stress events to the *same cell* (in seconds) and c_{vt} depends on the amount of trapped charge (Q) present in the oxide. The value of the recovery period, t, is assumed to be finite and greater than 1 second. We conservatively assume that no charge detrapping occurs for recovery periods less than one second. Yamada et al. [39] also show that c_{vt} has a logarithmic dependence on Q. Since Q is directly proportional to the stress voltage, $\Delta V_{th,s}$, c_{vt} also has a logarithmic dependence on $\Delta V_{th,s}$. Therefore, we get

$$c_{vt} \propto ln(\Delta V_{th,s}) \tag{7}$$

Equation (7) can be rewritten as

$$c_{vt} = K * ln(\Delta V_{th,s}) \tag{8}$$

where K is a constant that denotes the efficiency of the recovery process. [39] also provide plots of how c_{vt} varies with $\Delta V_{th,s}$.

Combining equations (6) and (8), the change in the threshold voltage shift due to recovery is given by:

$$\Delta V_{th,r} = K * ln(\Delta V_{th,s}).ln(t) \tag{9}$$

where $\Delta V_{th,s}$ is given by equation (5). We assume the value of K to be 60% based on discussions with industry [22]

The effective increase in the threshold voltage due to trapped charges after stress and recovery of the tunnel oxide, δV_{th}, is given by

$$\delta V_{th} = \Delta V_{th,s} - \Delta V_{th,r} \tag{10}$$

Equations (5) and (9) can be used to estimate the endurance of a NAND flash memory cell based on the number of stress events (P/E cycles) and the recovery periods that the cell experiences.

3.2 Model for Retention

The retention model estimates the duration of time taken by the memory cell to leak the charges stored in the floating gate. To determine this time duration, the model estimates SILC based on the number of charges trapped in oxide layer which is a function of the total number of stress events.

According to de Blauwe et al, SILC (J_{SILC}) is a sum of two components, (a) a time-dependent transient component ($J_{tr}(t)$) and (b) a time-independent steady state component (J_{ss}) [14]. We have,

$$J_{SILC} \quad = J_{tr}(t) + J_{ss} \tag{11}$$

Moazzami et al observed that for thinner tunnel oxide ($< 13nm$), the steady state component dominates the transient component [23]. Hence, to model J_{SILC}, it is sufficient to model the steady state component (J_{ss}). So, Equation (11) can be modified to

$$J_{SILC} \quad = J_{ss} \tag{12}$$

Because the tunnel oxide thickness is smaller than 13nm for many generations of NAND flash [13], Equation (12) provides a good estimate of SILC. Using the model derived by de Blauwe et al [14], J_{ss} can be written as

$$J_{ss} \propto \delta N_{ot} \cdot f_{FN}(\phi) \tag{13}$$

where δN_{ot} is the bulk trap density (as defined in Equation (2)) and f_{FN} symbolizes the FN-tunneling field dependence and is calculated using [18]. The value of ϕ is considered to be 0.9eV (from [14]). Replacing the proportionality sign in Equation (13) with a constant C and combining it with Equation (12), we get:

$$J_{SILC} = C \cdot N_{ot} \cdot f_{FN}(\phi = 0.9eV) \tag{14}$$

Equation (14) represents the SILC due to the presence of trapped charges in the tunnel oxide. It should be noted that in Equation (14), N_{ot} has a power relation to the number of cycles(based on Equation (2)) and hence, as the number of cycles increases, the SILC also increases. Assuming that $Q_{th,spread}$ to be the total charge stored in the floating gate corresponding a logical bit, δV_{th} to be total charge trapped in the tunnel oxide (calculated from Equation (10)), and J_{SILC} to be the leakage current, we can calculate the time taken for the charges to leak from the floating gate ($t_{retention}$) to be,

$$t_{retention} \quad = \frac{(Q_{th,spread} - \delta V_{th})}{J_{SILC}} \tag{15}$$

After $t_{retention}$ seconds, most of the charges from the floating gate would have leaked through the tunnel oxide and any read operation after this time will result in reading incorrect data. Since δV_{th} and J_{SILC} are functions of stress events and recovery period, Equation (15) provides an estimate of data retention period in NAND flash memory after taking stress and recovery into account.

3.3 Architecture-Level Simulations using FENCE

While FENCE captures the impact of stress and recovery on a single memory cell, the program and erase operations in NAND flash occur for a group of cells, such as within a page (for program) or within a block (for erase). Therefore, in architectural level simulations of these models, we track stress events due to program and erase operations at the granularity of a page and block respectively. Another point to note is that, the phenomenon of charge trapping and detrapping occurs in flash memory cells irrespective of whether they are used in the SLC or MLC mode. This is because the principle behind the stress events (FN tunneling) is the same for both SLC and MLC flash and the main difference between them is the maximum allowed threshold voltage shift. In either case, one can use the methodology given above to derive models from their physical transistor-level characterizations to estimate their endurance and retention.

3.4 Analyzing NAND Flash Memory Reliability using FENCE

So far, we have derived two models that calculate the threshold voltage shift due to trapped charges after stress and recovery and estimate the data retention period after a given number of stresses. Using FENCE, we now analyze the impact of charge detrapping on flash memory cells over different timescales. The goal of this analysis is to ascertain the extent to which charge detrapping can improve the reliability of flash memory cells by delaying endurance or retention related failure and understand how the duration of the quiescent period affects the extent of the recovery.

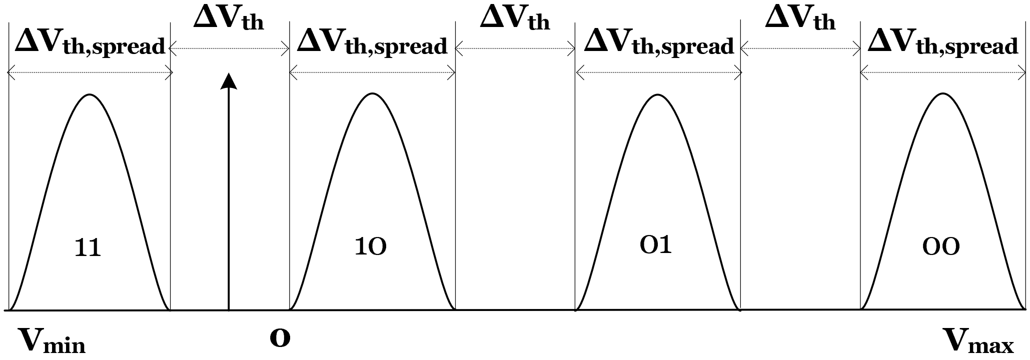


Figure 3: Threshold voltage distribution for a 2-bit MLC

Before we begin the analysis, we first need to precisely define what “failure” means with respect to endurance and retention. The data stored in a flash memory cell is identified by a specific voltage level. An n-bit MLC has 2^n *distinct* voltage levels, each of which corresponds to an n-bit value (an SLC flash cell is merely the case where n=1, which corresponds to two voltage levels - one for a digital “0” and the other for a “1”). Let $\Delta V_{th,spread}$ be the threshold voltage range for a single voltage level in a memory cell and ΔV_{th} be the difference in voltage between adjacent levels. Then, the entire operating voltage range of a memory cell varies from V_{min} to V_{max}, where $V_{max} = V_{min} + (\Delta V_{th,spread} * 2^n) + \Delta V_{th} * (2^n - 1)$. This is illustrated in Figure 3 for $n = 2$ (2-bit MLC).

When the charges trapped in the oxide result in a threshold voltage increase of ΔV_{th} or higher, it will no longer be possible to clearly distinguish between different voltage levels. As a consequence, it will not be possible to reliably read from or write to the memory cell. We define this situation where the increase in threshold voltage due to trapped charges, δV_{th}, is greater than or equal to ΔV_{th} as an *endurance failure*.

Similarly, as charges accumulate in the tunnel oxide, δV_{th} increases and as δV_{th} increases, the SILC also increases (because of trap assisted tunneling) and when δV_{th} becomes equal to ΔV_{th}, the SILC is high enough that the floating gate is no longer able to retain charges. We define this situation as a *retention failure* and when retention failure is encountered, the state of the cell can no longer be sensed reliably leading to an unrecoverable data loss.

However, products that use NAND flash as the underlying storage medium typically specify a data retention period - a *minimum* duration of time for which the data written in a flash memory cell should be retained without data loss. Most manufacturer datasheets quote a 10 year period for data retention. Guaranteeing the specified data retention period also limits the amount of cycling because beyond a certain amount of cycling, the data retention period of the cell falls below the specified period. We define this limit as the *endurance limit*, the maximum number of P/E cycles that can a memory cell can tolerate after which the retention period of the memory cell drops below the data retention requirement. Beyond the endurance limit and but until the endurance failure is reached, the memory cell still retains data, but for a period lesser than the *rated* data retention period.

The higher the ΔV_{th}, the larger the number of P/E cycles required for δV_{th} to reach this value. Similarly, the longer the recovery period between successive P/E cycles, the higher the detrapping and therefore a larger number of P/E cycles will be allowed before δV_{th} reaches ΔV_{th}. It should be noted that while it appears that choosing a large ΔV_{th} can provide high endurance, a high threshold voltage directly translates to a higher write latency [3], which can significantly degrade performance and actually increase the duration of stress.

3.4.1 Impact of detrapping on Endurance

Manufacturer datasheets specify an endurance rating of 10K and 100K P/E cycles for MLC and SLC chips respectively. However, these values specify the *minimum* number of P/E cycles that the chip is expected to tolerate before failure, tested under high stress conditions where the flash cells are continuously erased and rewritten with little or no recovery time between successive stress events [37]. There is anecdotal evidence in recently published papers on measurements of NAND flash chips, that, in the common case, when there are recovery periods between the stress events, the endurance of flash is higher than the values specified in datasheets [5, 8].

In Figure 4, we plot the change in δV_{th} with the number of P/E cycles, over a number of timescales for the recovery period, for the 80nm process technology [12] for which published memory cell characterization data is available [20, 39, 40]. We consider the case where there is no recovery between successive stress events, which is how the datasheet values are computed, and also cases where the recovery time is varied from 10 seconds to over 2 days.

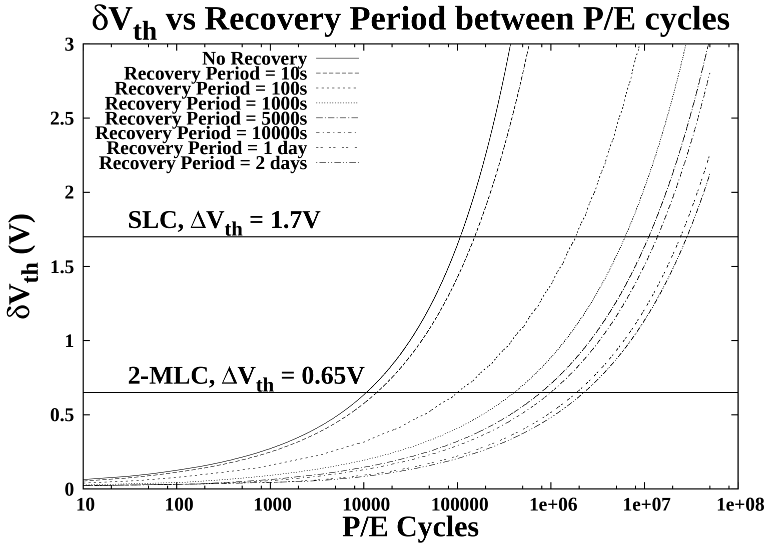


Figure 4: Increase in δV_{th} with P/E cycles for different recovery periods.

To illustrate how these curves translate to endurance, we plot the ΔV_{th} for SLC and 2-bit MLC flash. These values are shown as horizontal lines in the graph and are obtained from threshold voltage distributions of prototype NAND flash memory chips published in the literature. 2-bit MLC devices have ΔV_{th} values that are approximately equal to $\Delta V_{th,spread}$ and have been shown to vary from 0.6V to 0.7V [4]. We assume ΔV_{th} to be equal to $\Delta V_{th,spread}$ for SLC devices as well. The $\Delta V_{th,spread}$ of SLC has been reported to vary from 1.4V to 2.0V [17, 24]. Based on this data, we assume the ΔV_{th} of SLC to be 1.7V and 2-bit MLC to be 0.65V. The portions of the curves below the horizontal lines correspond to failure-free operation of the cell. The number of P/E cycles attainable for each recovery period and the improvement in endurance over the case where there is no detrapping between successive stresses is given in Table 2 (for clarity, in the figure we omit a few of the data points given in the table).

We can see that when there are no recovery periods, the P/E cycles for the SLC and MLC data points approximately match the values given in datasheets (100K and 10K P/E cycles respectively), which concurs with the expected behavior. We can also see that a recovery period between successive P/E cycles can significantly boost endurance, which concurs with recent flash chip measurement studies [5, 8]. Even a recovery

Recovery Period	SLC, $\Delta V_{th} = 1.7V$		2-bit MLC, $\Delta V_{th} = 0.65V$	
	P/E Cycles	**Endurance Increase**	**P/E Cycles**	**Endurance Increase**
No recovery	107535	1x	10652	1x
10 seconds	153186	1.4x	13749	1.3x
50 seconds	1028724	9.6x	52444	4.9x
100 seconds	1837530	17.7x	99913	9.3x
1000 seconds	6214983	57.8x	403082	37.8x
5000 seconds	11093823	103x	780723	73.3x
10000 seconds	13753999	127x	990014	92.9x
15000 seconds	15497892	144x	1129379	106x
1 day	24274492	225x	1879352	176x
2 days	28487539	264x	2247910	211x

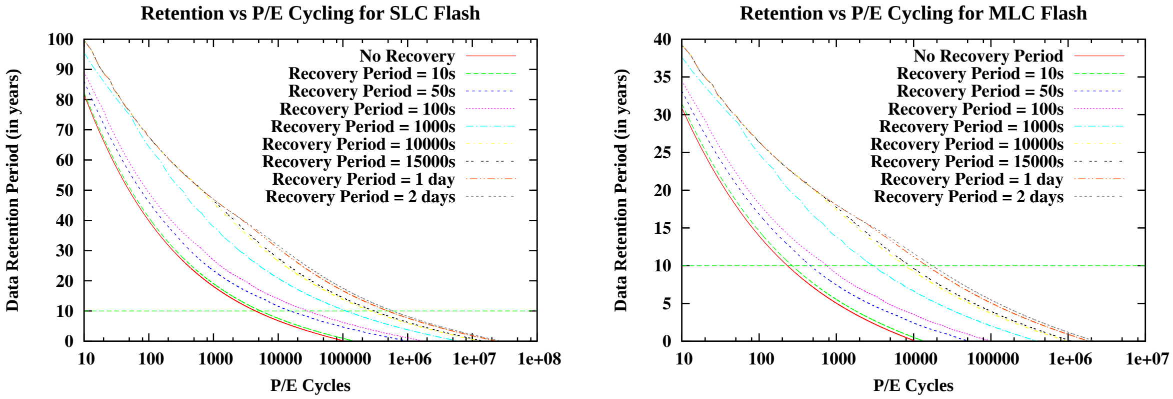
Table 2: Endurance limits with charge detrapping.

period of approximately a minute between stress events can provide a large improvement in endurance. Note that although the I/O request inter-arrival times to an SSD tend to be much shorter in an enterprise system and a significant fraction of the requests can be writes [29], the time between successive stress events to a *specific physical flash page* on the SSD is much longer due to the fact that NAND flash does not support in-place writes, due to the decisions made by the wear-leveling and cleaning policies of the FTL, and the logical block addresses in the I/O request stream arriving at the SSD. Further increase in the duration of the recovery periods provides increased endurance, and a recovery period of a few hours provides two orders of magnitude improvement. However, as the recovery periods increase beyond a day, we start getting diminishing endurance benefits.

3.4.2 Impact of detrapping on Data Retention

Having examined the impact of detrapping on endurance, we now analyze the effect of detrapping on retention. Typically, manufacturer datasheets specify a retention period of at least 10 years for NAND flash. This rating is usually very conservative and many supplementary documents provided by manufacturers show that the typical retention period is close to 100 years for NAND flash [25–28]. Using this information and conservatively assuming that the retention period of NAND flash to be 100 years for a new SLC device, we analyze the impact of flash memory cycling with various recovery periods in between the cycles on its retention period. The results of this analysis is shown in Figure 5(a) for SLC and Figure 5(b) for 2-bit MLC flash.

From Figure 5(a) and Figure 5(b), it can be observed that when there is no recovery between successive cycles, the memory cell encounters retention failure when the total number of P/E cycles is around 100K for SLC flash and 10K for MLC flash, which concurs with the expected behavior. However, as the recovery period between successive stress events increases, the number of times the cell can be cycled before retention failure occurs increases exponentially. Both SLC and 2-bit MLC flash experience a steep drop in their retention period when the flash memory cell is relatively new(few hundreds to thousands of cycles). For SLC flash, the retention period drops from nearly 100 years to about 20 years in a few thousand cycles, while in case of MLC flash, the retention period drops from nearly 40 years to 5-10 years in about a thousand cycles. However, after this steep drop, the rate of decrease in the retention period slows down. As the memory cell is cycled, its retention period keeps dropping and when the memory cell is cycled up to its *endurance failure* point, the flash memory cell experiences retention failure. From Figure 5(a) and Figure 5(b), we can note that even though SLC flash and 2-bit MLC flash exhibit the same trend, the initial retention period of a new MLC flash memory is nearly 40 years, while that of the SLC flash is nearly 100 years. This is because, the



(a) SLC Flash

(b) 2-bit MLC Flash

Figure 5: Impact of different recovery periods on Data Retention for SLC and 2-bit MLC flash

ΔV_{th} for MLC flash is about 2.6 times lesser than the ΔV_{th} for SLC flash and hence the lower data retention period.

3.4.3 Trading off Data Retention for Increased Endurance

In Figure 5(a) and Figure 5(b), we can see that there is clear tradeoff between the data retention and the number of cycles possible. If the data retention for NAND flash can be relaxed linearly, the total number of P/E cycles possible can be increased exponentially. For example, in the case of 2-bit MLC flash having an average recovery period of about 10,000 seconds, the P/E cycle count can be increased from 8000 to 52,800 when the data retention period requirement is be reduced from 10 years to 5 years. In a real world scenario, this translates to a high number of rewrites for NAND flash based storage media for a reduced data retention period. In architectures which typically use NAND flash as a front-end of a tiered storage system (like enterprise storage), NAND flash memory can be cycled beyond its endurance limit provided that a periodic backup of the data is performed to avoid data loss. Another option, which we explore extensively in this report, to handle reduced retention is to periodically refresh the SSD (say once every month), to make sure that data integrity is preserved. The main purpose of refresh operation is to identify flash memory pages whose retention period is below the rated retention period and refresh the data present in them, to ensure data longevity. The refresh operation for SSDs will involve a combination of read operation (on the source block to be refreshed) and a write operation (on a target block) to ensure data persistence. Since, each refresh operation also involves a write operation, the damage to the tunnel oxide increases, which reduces the retention period even further. Hence, as NAND flash is cycled, the refresh frequency for SSDs need to be increased to handle the reduced retention, thus providing an opportunity to increase the endurance of NAND flash without losing the data written to them. In the rest of this report, we examine refresh policies that can be used to increase the endurance of NAND flash based SSDs while preserving their data integrity.

3.5 Design Changes to Flash Memory Controller to Support Refresh Operation

In order to periodically refresh the data in the SSDs, the flash memory controllers present in the SSDs need to be aware of the degree of wear out of each flash memory block. Current FTL algorithms that have been proposed in the literature typically store the cycle count of every SSD block to keep track of the wear out of the device [2, 6, 9]. As shown in Section 3.4.3, just maintaining the cycle count is not sufficient to

Command (arguments)	Description
refresh(source page, target page)	Read the data from the source page and write it to the target page. The source and target page can be in same or different chips.
refresh(source block,target block)	Read all valid pages from source block and write them to the target block. The target block should have sufficient space to accommodate the valid pages in source block. The source and target block can be in same or different chips.
refresh(source plane, target plane)	Read all valid pages from all blocks in the source plane and write them to the target plane. If the source and target plane are equal, then the plane should contain at least one free block to move data within the plane.

Table 3: Possible commands from the flash controller to the flash chips to perform the refresh operation.

estimate the wear out of the device and metrics like average recovery periods and the δV_{th} information are required to accurately estimate the wear-out of SSDs blocks. For a page based FTl, storing the average recovery period and δV_{th} for every page along with the existing metadata information requires as much 8 additional bytes. Since existing SLC and MLC flash chips have spare area associated with flash memory page (which varies from 128 to 256 bytes) to store metadata information of every page [32], storing 8 additional bytes incurs an space overhead of 3.1% to 6.2%. For a block based FTL, the average recovery period and δV_{th} for every flash block is stored. Assuming that there are 128 pages in a block and each page has a spare area of 128 to 256 bytes, the space overhead for storing the metadata information is about 0.02% to 0.05% depending on the availability of spare area.

With this additional metadata information, the FTL decides which memory blocks[1] require refresh. The policies that the FTL will use to determine those memory blocks that require refresh can be classified as time and/or space dependent policies and can be dependent or independent of the workload under consideration. The main objectives of such policies is to identify the source memory block(s) that needs to be refreshed and target memory block(s) to where the data should be migrated, the timing of this data migration and the commands used to perform such data migration while having minimum impact on the performance of SSD. While, we present a detailed discussion of the first two objectives in Section 5, we now discuss how the controller can command the flash chips to perform these refresh operation.

Once the source and target memory blocks are identified, the flash memory controller can command the flash chips to perform the data migration. To perform such a migration, the controller can send a refresh command to the chip with source and target memory blocks as parameters. Depending on the granularity of the memory blocks, one of the commands from the Table 3 can be used to perform the refresh operation. We would like to clarify that the list of commands provided in Table 3 is not comprehensive but is presented to provide an idea of the type of commands that the flash controller can issue to the flash memory chips to perform the refresh operation. Depending on the degree of parallelism available, read or write operations to some or all parts of the chip will be queued until the refresh operation is complete, which increases the response time of such requests and slowing down the disk.

Existing flash chips can also respond to commands like *two-plane read for copy back, two-plane copy back program, two-plane page program, two plane block erase* to parallelize data movement within a chip [32]. Instead of using a new refresh command, such commands can be reused by the controller to perform the refresh operation efficiently.

Having examined the design changes necessary to the controller to support different refresh policies,

[1] The term memory block in this discussion refers to any contiguous section of flash memory like a page, block, plane, chip or the entire SSD.

Parameter	M-SSD
Page Size	4KB
Pages/Block	128
Blocks/Plane	1024
Planes/Chip	8
Chips	25
Usable capacity	90%
Page Read	$50\mu s$
Page Program	$900\mu s$
Block Erase	$3.5ms$
Serial Data Transfer	$25\mu s$

Table 4: Architecture-level Parameters for a MLC(M-SSD) based SSD.

Section 5 discusses some refresh policies that can be performed to ensure data persistence and increase the endurance of SSDs.

4 Experimental Methodology

Using the FENCE model, we now analyze the reliability of NAND flash based SSDs running enterprise workloads. We describe the simulation infrastructure and the workload details in Section 4.1. In section 4.2, the evaluation methodology is explained.

4.1 Simulator and Workloads:

We use Disksim [7], a widely used trace driven simulator for evaluating storage systems. In order to simulate a SSD, we use the Disksim SSD extension [2] that facilitates studying a variety of SSD designs. For evaluation, we simulate an enterprise class 100GB MLC based SSD(M-SSD) similar to [11] .The configuration of these M-SSD is summarized in Table 4.

Our workloads consist of block-level I/O traces collected from various production systems within Microsoft [31, 36]. The details of the enterprise workloads evaluated are specified in Table 5. Each workload consists of several sub-traces, each of which correspond to the I/O activity during a specific interval of time (e.g., an hour) on a typical day, and the collection of these sub-traces span anywhere from 6 hours to one full day. We use all the sub-traces of each workload in the simulation to characterize the variations in the I/O behavior and their impact on the SSD reliability.

4.2 Evaluation Methodology:

Because reliability issues in storage systems take a long duration of time to manifest, we evaluate the impact of the workloads on the SSDs over a 5 year service life. The main reasons behind choosing such a large simulation period are that reliability problems in SSDs due to wear out typically take such time periods to manifest themselves and also the fact that the typical replacement cycle for disks is around 5-10 years [34]. We report the average retention period and the average number of P/E cycles experienced by the SSDs over this time scale. In order to get an idea of the impact of recovery periods on the SSD reliability, we also report the maximum change in threshold voltage of flash memory blocks over this time scale. While

Workload	Duration (hours)	Total I/Os (in millions)	Read-Write ratio	Average inter-arrival time(ms)
Live Maps Backend (LM)	24	44.7	3.73:1	1.9
MSN File Server (MSNFS)	6	29.4	2.05:1	0.75
MSN Meta-data Server (MSN-CFS)	6	4.5	2.82:1	4.8
Exchange Server (EXCH)	24	54.2	0.59:1	1.59
Radius Authentication Server (RAD)	18	2.0	0.11:1	24.8
Radius Backend Authentication Server (RAD-BE)	18	4.2	0.21:1	11.8
Display Ads Platform Payload Server (DAPPS)	24	1.09	1.27:1	79.3

Table 5: Characteristics of Enterprise Workloads used for Evaluation

the service life spans multiple years, the traces record at most a single day of activity. We need a way of estimating the activity on the SSD over this long time period. As each trace represents the I/O activity over the course of a typical day, one approach could be to repeatedly replay the trace in Disksim and simulate 5 years worth of activity.

However, this approach would require excessively long simulation times. We instead use a statistical approach to estimate the I/O activity on the SSD.

In order to estimate endurance and retention, we need to capture two aspects of stress behavior: (1) the distribution of stress events across various pages and blocks in the SSD (spatial behavior), and (2) the distribution of the recovery periods to individual pages and blocks (temporal behavior). To determine these distributions, we collect an output trace over the course of a Disksim simulation that records when a particular page or block within a certain flash chip is programmed or erased. We do not record reads to a page since read operations have a negligible impact on endurance. We collect one such output trace for each sub-trace,which allows us to capture any phase behavior within a workload. as well as the behavior of the wear-leveling and cleaning algorithms within each sub-trace period. From this output trace, we characterize the spatial behavior of the workload by creating a histogram of the stresses to the different flash chips in the SSD to determine the frequency at which pages/blocks within a particular chip are stressed. Since the FTL performs wear-leveling operations within each flash chip in an SSD [2], we use a *uniform distribution* to model the pattern of stresses *within a single chip*. However, across the multiple chips, the original spatial distribution of the workload is still maintained. We characterize the temporal behavior of the workload by creating a histogram of the recovery periods of all the pages within the SSD. Using these statistical distributions of the spatial and temporal characteristics of a workload's stress behavior on the SSD, we extrapolate the stress behavior over the service life of 10 years.

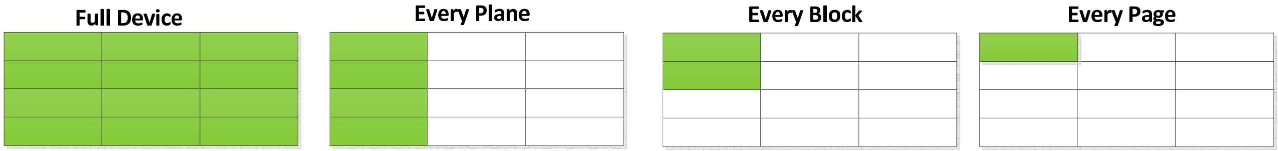


Figure 6: Illustration of various spatial refresh policies

5 Refresh Policies to Ensure Data Persistence and Increase Endurance of Enterprise SSDs

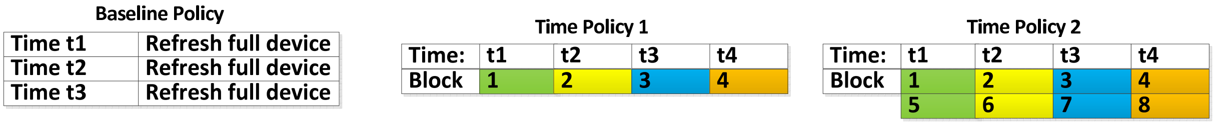
The policy space for performing a refresh is of significant interest, since this determines the limit to which we can push the endurance of the flash device. In this section, we explore some spatial and temporal policies that are both workload dependent and independent. We provide illustrations of how these policies can be implemented and leave the detailed analysis of these policies to future work.

5.1 Workload Independent Spatial Policies

Refresh can be done at multiple spatial granularities. For instance, we can refresh the entire SSD device as a whole, or we can refresh one page at a time or narrow in to the block level for more granular control. Figure 6 illustrates the different possible spatial granularities in which the refresh operation can be performed.

5.2 Workload Independent Temporal Policies

Temporal variation in refresh policies would allow refresh to proceed at the same time as normal operations. Figure 7 illustrates the various temporal granularities in which the refresh operation can be performed. At the baseline is a policy where we refresh the entire device at the same time, however the amount of refresh would prohibit any other operation to occur at the same time. On the other end is a policy where we can refresh one spatial unit at a time. This spatial unit can be a plane, block or a page (referred to as Time policy 1 in Figure 7). This would allow other operations to continue without having to wait for the refresh operation to complete as long as they are performed on a different spatial unit. We can also group multiple pages or blocks (belonging to different planes) together for refresh depending on the refresh bandwidth availability. This policy would allow for quicker refreshes and at the same time not impact performance significantly (indicated as Time policy 2 in Figure 7).



Baseline Policy

Time t1	Refresh full device
Time t2	Refresh full device
Time t3	Refresh full device

Time Policy 1

Time:	t1	t2	t3	t4
Block	1	2	3	4

Time Policy 2

Time:	t1	t2	t3	t4
Block	1	2	3	4
	5	6	7	8

Figure 7: Illustration of various temporal refresh policies

Another approach similar to time policy 2 is to have groupings of different sizes. The sensitivity of refresh operations to different sized grouping can be explored and the grouping that works for the best for each application will be reported.

5.3 Workload Dependent Policies

Based on our understanding of data center workloads, we can design workload specific policies that can behavior of individual workloads. One example of such policy is a predictive policy which determines when the write traffic would be higher and to which LBN ranges. Depending on this prediction, we should be able to identify the time granularity and space granularity to select and the refresh operation can be performed accordingly. Both the time and space granularity can be changed dynamically and the performance of such a refresh policy can be determined. Since we perform trace based evaluation, we can have an oracle for every trace that does exactly the needed refreshes. The performance of the oracle based policy will be analyzed and compared with the other policies that are mentioned above.

We will evaluate the above mentioned policies using Disksim and evaluate metrics like performance, the number of refresh operations and the increase in total endurance of enterprise SSDs due to the various refresh policies.

6 Related Work

Charge Trapping/Detrapping in CMOS Transistors: The generation of interface traps at the Si/SiO_2 interface causes a reliability problem for CMOS transistors as well and has been studied in the context of Negative Bias Temperature Instability (NBTI) and Positive Bias Temperature Instability (PBTI). Similar to recovery in FGTs, the interface traps can be removed (detrapped) by applying a suitable logic value as input to the gate of the device and a number of techniques have been proposed to achieve this at runtime [1, 16, 35, 38].

Wear-Leveling Techniques for Flash: A number of wear-leveling techniques have been proposed to balance the wear on flash memory blocks within an SSD to improve endurance and they are discussed in [6]. The proposed techniques include threshold based schemes using per-block erase-counters, techniques that use randomization to choose erase blocks, and those that separate hot and cold blocks when making wear leveling decisions [6]. While these wear-leveling techniques implicitly take into account the impact of charge trapping by counting the number of P/E cycles on a page/block to effect a policy, they do not consider the impact of recovery.

NAND Flash Endurance Measurements: Grupp et al. [8] study the performance, power, reliability of several SLC and MLC NAND flash chips and show that the endurance of these chips tend to be much higher than their datasheet values. Desnoyers [5] conducted a similar study of the performance and endurance characteristics of several NAND flash memory chips and found the endurance trends to be similar to those reported in [8]. These papers show that the number of P/E cycles that the pages and blocks can sustain is much higher than those given in datasheets. However, these papers do not explain the underlying cause for this trend.

7 Future Work

While this work has focused on the endurance and data retention property of NAND flash, we are also trying to answer other questions related to NAND flash reliability. Our immediate focus is to validate our model with real chip measurements. So far, our analytical model framework does not capture the impact of temperature on the flash memory reliability. Since temperature is a first order constraint in flash memory reliability, we plan to extend our model to consider this parameter in future.

8 Conclusion

Flash Memory reliability has long been a concern for deploying SSDs in data centers. In order to study this limitation and suggest solutions to overcome this limitation, we have built a reliability model framework called FENCE to study flash memory reliability. Using FENCE, we analyze the tradeoffs between write endurance and data retention for both SLC and MLC flash. We make a case for increasing the endurance of MLC based SSDs by trading off their data retention property. We illustrate some refresh policies that can be applied to ensure data integrity of SSDs and suggest changes to the design of flash memory controller to support these refresh operations.

References

[1] J. Abella, X. Vera, and A. Gonzalez. Penelope: The NBTI-Aware Processor. In *Proceedings of the 40th IEEE/ACM International Symposium on Microarchitecture*, 2007.

[2] N. Agrawal and et al. Design Tradeoffs for SSD Performance. In *Proceedings the USENIX Technical Conference (USENIX)*, June 2008.

[3] J. Brewer and M. Gill, editors. *Nonvolatile Memory Technologies with Emphasis on Flash*. IEEE Press, 2008.

[4] T. Cho and et al. A dual-mode NAND flash memory: 1-Gb multilevel and high-performance 512-Mb single-level modes. *Solid-State Circuits, IEEE Journal of*, 36(11):1700–1706, Nov 2001.

[5] P. Desnoyers. Empirical Evaluation of NAND Flash Memory Performance. In *Proceedings of the Workshop on Hot Topics in Storage and File Systems (HotStorage)*, October 2009.

[6] E. Gal and S. Toledo. Algorithms and Data Structures for Flash Memories. *ACM Computing Surveys*, 37(2):138–163, June 2005.

[7] G. Ganger, B. Worthington, and Y. Patt. *The DiskSim Simulation Environment Version 4.0 Reference Manual*. http://www.pdl.cmu.edu/DiskSim/.

[8] L. Grupp and et al. Characterizing flash memory: Anomalies, observations, and applications. In *Microarchitecture, 2009. MICRO-42. 2009 42nd IEEE/ACM International Symposium on*, Nov. 2009.

[9] A. Gupta, Y. Kim, and B. Urgaonkar. DFTL: a flash translation layer employing demand-based selective caching of page-level address mappings. In *ASPLOS '09: Proceeding of the 14th international conference on Architectural support for programming languages and operating systems*, pages 229–240, 2009.

[10] S. Gurumurthi, A. Sivasubramaniam, and V. Natarajan. Disk Drive Roadmap from the Thermal Perspective: A Case for Dynamic Thermal Management. In *Proceedings of the International Symposium on Computer Architecture (ISCA)*, pages 38–49, June 2005.

[11] Intel X25-M Extreme SATA Solid-State Drive. `http://www.intel.com/design/flash/nand/mainstream/index.htm`.

[12] Process Integration and Device Structures, ITRS 2007 Edition. `http://www.itrs.net/Links/2007ITRS/2007_Chapters/2007_PIDS.pdf`.

[13] Process Integration and Device Structures, ITRS 2009 Edition. `http://www.itrs.net/links/2009ITRS/2009Chapters_2009Tables/2009Tables_FOCUS_C_ITRS.xls`.

[14] J. de Blauwe, J. van Heudt, D. Wellekens, G. Groeseneken, and H.E. Maes. SILC-related effects in flash E^2PROM's-Part I: A quantitative model for steady-state SILC. *Electron Devices, IEEE Transactions on*, 45(8):1745 –1750, Aug. 1998.

[15] JEDEC - Standards and Documents. `http://www.jedec.org/standards-documents`.

[16] S. Kumar, C. Kim, and S. Sapatnekar. Impact of NBTI on SRAM Read Stability and Design for Reliability. In *Proceedings of the International Symposium on Quality Electronic Design*, 2006.

[17] J. Lee and et al. A 1.8V NAND Flash Memory for Mass Storage Applications. In *Proceedings of the IEEE International Solid-State Circuits Conference (ISSCC)*, pages 290–494, February 2003.

[18] M. Lenzlinger and E. Snow. Fowler-Nordheim tunneling into thermally grown SiO_2. *Electron Devices, IEEE Transactions on*, 15(9):686–686, Sep 1968.

[19] Memory Exchange. `http://memoryexchange.com/Price/NationalFlashDetail.aspx`.

[20] N. Mielke and et al. Recovery effects in the distributed cycling of flash memories. In *Reliability Physics Symposium Proceedings, 2006. 44th Annual., IEEE International*, pages 29–35, March 2006.

[21] N. Mielke, T. Marquart, N. Wu, J. Kessenich, H. Belgal, E. Schares, F. Trivedi, E. Goodness, and L. Nevill. Bit error rate in nand flash memories. In *Reliability Physics Symposium, 2008. IRPS 2008. IEEE International*, pages 9 –19, May 2008.

[22] N. R. Mielke. Intel Corporation, October 2009. Private Correspondence.

[23] Moazzami, R. and Chenming Hu. Stress-induced current in thin silicon dioxide films. In *Electron Devices, IEEE Transactions on*, pages 139 –142, Dec. 1992.

[24] H. Nakamura and et al. A 125mm^2 1Gb NAND Flash Memory with 10 MB/s Program Throughput. In *Proceedings of the IEEE International Solid-State Circuits Conference (ISSCC)*, pages 106–107, February 2002.

[25] Application Report: Understanding MSP430 Flash Data Retention. `http://focus.ti.com/lit/an/slaa392/slaa392.pdf`.

[26] Data Remanence in Flash Memory Devices. `http://www.cl.cam.ac.uk/~sps32/DataRem_CHES2005.pdf`.

[27] Typical Data Retention for Nonvolatile Memory. `http://www.freescale.com/files/microcontrollers/doc/eng_bulletin/EB618.pdf`.

[28] Typical Data Retention for Nonvolatile Memory. `http://www.spansion.com/Support/AppNotes/EnduranceRetention_AN.pdf`.

[29] D. Narayanan, A. Donnelly, and A. Rowstron. Write Off-Loading: Practical Power Management for Enterprise Storage. In *Proceedings of the USENIX Conference on File and Storage Technologies (FAST)*, February 2008.

[30] Y. Park and D. Schroder. Degradation of thin tunnel gate oxide under constant fowlernordheim current stress for a flash eeprom. In *IEEE Transactions on Electron Devices*, 1998.

[31] S. Kavalanekar and et al. Characterization of storage workload traces from production Windows servers. In *Proceedings of the IEEE International Symposium on Workload Characterization (IISWC)*, pages 119–128, October 2008.

[32] Samsung corporation. K9XXG08XXM flash memory specification. K9XXG08XXM Datasheet.

[33] Samsung releases MLC-based SSDs for data centers. `http://www.computerworld.com/s/article/9201959/Samsung_releases_MLC_based_SSDs_for_data_centers`.

[34] B. Schroeder and G. Gibson. Disk Failures in the Real World: What Does an MTTF of 1,000,000 Hours Mean to You? In *Proceedings of the USENIX Conference on File and Storage Technonologies (FAST)*, February 2007.

[35] J. Shin, V. Zyuban, P. Bose, and T. Pinkston. A Proactive Wearout Recovery Approach of Exploiting Microarchitectural Redundancy to Extend Cache SRAM Lifetime. In *Proceedings of the International Symposium on Computer Architecture (ISCA)*, pages 353–362, June 2008.

[36] SNIA IOTTA Trace Repository. http://iotta.snia.org/.

[37] Application Report: MSP430 Flash Memory Characteristics. http://focus.ti.com/lit/an/slaa334a/slaa334a.pdf.

[38] A. Tiwari and J. Torrellas. Facelift: Hiding and Slowing Down Aging in Multicores. In *Proceedings of the International Symposium on Microarchitecture (MICRO)*, November 2008.

[39] R. Yamada and et al. A novel analysis method of threshold voltage shift due to detrap in a multi-level flash memory. In *Symposium on VLSI Technology, Digest of Technical Papers*, 2001.

[40] H. Yang and et al. Reliability issues and models of sub-90nm NAND flash memory cells. In *International Conference on Solid-State and Integrated Circuit Technology*, 2006.