Understanding the Impact of Threshold Voltage on MLC Flash Memory Performance and Reliability

Wei Wang
Computational Science
Research Center
San Diego State University
wang@rohan.sdsu.edu

Tao Xie
Computer Science
Department
San Diego State University
txie@mail.sdsu.edu

Deng Zhou
Computational Science
Research Center
San Diego State University
zhoud@rohan.sdsu.edu

ABSTRACT

MLC (multi-level cell) NAND flash memory based solid state drives (SSDs) have been increasingly used in supercomputing centers because of their merits in cost, performance, and energy-efficiency. However, as each cell starts to store two or more bits, a threshold voltage range employed to represent a state has to be continuously shrunk, and a narrowed threshold voltage range causes more bit errors. An ad-hoc solution to this problem is to apply an enhanced ECC (error correction code) scheme. Still, a comprehensive understanding of the impact of threshold voltage on MLC flash performance and reliability is an open question. In this paper, we first empirically measure the correlations between threshold voltage and program/erase (P/E) performance as well as reliability. After analyzing experimental results, we make several interesting observations: 1) a memory cell programmed to a lower threshold voltage has a faster programming speed (up to 31%) as well as a fewer number of bit errors; 2) the programming time of an MSB page is about 2 to 3 times shorter than that of an LSB page; 3) erase performance is highly correlated to threshold voltage. These new findings provide system implications for the development of a better SSD. Further, to demonstrate how these findings can be leveraged to enhance MLC flash, we propose an approach called threshold voltage reduction (TVR), which increases programming speed and longevity by 50% and 7.1%, respectively. Finally, we conduct a study on TVR-powered SSDs. Simulation results show that overall mean response time can be reduced by up to 35%.

Categories and Subject Descriptors

B.3.3 [**Memory Structure**]: Performance Analysis and Design Aids; C.4 [**Performance of Systems**]: Design studies; D.4.2 [**Operating Systems**]: Storage Management

Keywords

MLC flash; threshold voltage; P/E performance; reliability; solid state disk

ICS'14, June 10–13 2014, Munich, Germany.
Copyright 2014 ACM 978-1-4503-2642-1/14/06 ...$15.00.
http://dx.doi.org/10.1145/2597652.2597681.

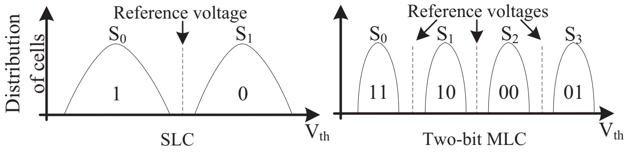


Figure 1: Threshold voltage levels in SLC and MLC flash.

1. INTRODUCTION

SLC (single-level cell) flash memory has been the main choice for SSDs in data centers due to its high performance and endurance [13, 22]. Recently, less-expensive MLC has started to enter into the storage market of data centers thanks to the advanced controller technology that can provide a larger ECC capability to compensate MLC's shorter life span and higher error correction overhead [12, 13, 18]. However, as manufacturers are aggressively pushing flash memory into smaller geometries and each memory cell has to store more bits [10], the improved controller and processing technology might not be able to keep pace with the shrinking NAND lithography [12], which makes the future of MLC flash SSD in data centers unclear. One reason is that the rate of increasing bit errors exceeds the capacity of enhanced ECC schemes [10, 12]. When a memory cell is pushed to store more bits, it is prone to more bit errors due to a much narrowed threshold voltage range [10].

Threshold voltages are used to represent different states in flash memory [3]. As illustrated in Figure 1, an SLC flash cell has two threshold voltage states (i.e., S_0, S_1), which indicate data '1' and '0', respectively. A 2-bit MLC flash cell provides four states (i.e., '11', '10', '00', '01') defined by four threshold voltage ranges (see Figure 1). Each of them is roughly half of that of an SLC flash cell. A programming operation is to charge a memory cell to a particular threshold voltage, whereas an erase operation is a reversal procedure [3]. If a cell's threshold voltage shifts across the reference voltage, the stored data will be misinterpreted, and thus, a bit error occurs [3, 21]. Obviously, MLC trends to generate more bit errors as a narrowed threshold voltage range could not tolerate even a slight threshold voltage shift. Between two adjacent threshold voltage ranges, a wide margin is reserved to combat retention errors. It is because a memory cell loses charge over time, which causes a left shift of threshold voltage [21]. As a threshold voltage range is continuously shrunk, its negative impact on endurance and program/erase (P/E) performance becomes more noticeable

[7, 28]. For example, while a typical SLC flash can tolerate ∼100k P/E cycles with a 200 μs programming delay, a 2-bit MLC can only survive ∼10k P/E cycles with a 600∼900 μs delay [9]. Apparently, threshold voltage greatly influences MLC's performance, endurance and reliability.

Unfortunately, little investigation on the impact of threshold voltage on P/E performance and reliability of flash memory has been reported in the literature. To understand the role of threshold voltage in flash memory's performance and reliability, in this paper we empirically study the correlations between threshold voltage and MLC flash memory[1] P/E performance as well as reliability. All experiments are conducted on a hardware platform including an FPGA evaluation board [27] and a flash daughter board [6], which can issue chip-level commands to raw flash chips without ECC. To the best of our knowledge, this is the first work that empirically evaluates the impact of threshold voltage on MLC flash memory performance and reliability. Our experimental results demonstrate that the level of threshold voltage significantly impacts MLC flash performance and reliability. Since there exists a one-one correspondence between a threshold voltage and a particular state or data [3] (see Figure 1), we can indirectly adjust a memory cell's threshold voltage by programming it to different data. Our new findings, in turn, provide SSD designers with insights to developing a better SSD.

Our new findings and key contributions include:

- *Reliability and Endurance*

  Flash memory reliability in terms of number of bit errors is highly correlated with threshold voltage. A memory cell that is programmed to a higher threshold voltage is likely to generate more bit errors. Further, a memory cell programmed in a high threshold voltage ages faster as the number of bit errors increases more rapidly. Based on the experimental results, a reliability model is derived to explore the relationship between threshold voltage and a cell's reliability in terms of bit error number. These observations provide a new venue for further reliability enhancement of MLC flash.

- *P/E Performance*

  Programming a page to a lower threshold voltage is much faster than programming a page to a higher threshold voltage. For example, in a 2-bit MLC flash the speed of programming a memory cell to state S_0 (i.e., cell programmed as '11') is 15.5%, 23%, and 31% faster than that of programming it to state S_1, S_2, and S_3, respectively. Furthermore, irrelevant to its threshold voltage, a memory cell's programming speed always increases when its number of P/E cycles enlarges. On average, a memory cell's programming speed could improve 11.4% at the end of its lifetime. Besides, the time to program an MSB (most significant bit) page is around 2 to 3 times shorter than that of programming an LSB (least significant bit) page. In addition, under different threshold voltages, the erasing speed of a cell slightly changes. For instance, while erasing a block programmed as '11' costs 3.3 ms, erasing a block programmed as '01' only takes 3.1 ms.

[1] in this study we only investigate 2-bit MLC flash, which is currently the dominant type of MLC flash.

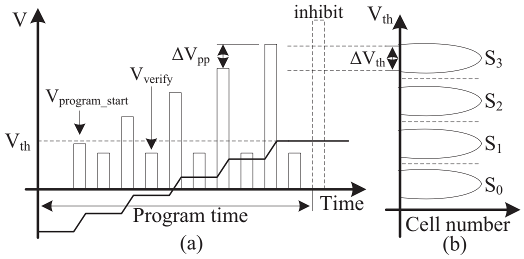


Figure 2: (a) An illustrative example of ISPP; the increased bars show the programming voltage, and the stair-step line shows the change of threshold voltage during the whole ISPP process; (b) distribution of cells in four levels for MLC flash.

- *Threshold Voltage Reduction*

  Inspired by our findings, we propose a new approach called threshold voltage reduction (TVR) that can trade an over-provisioned retention time for an increased write speed and longevity. Its basic idea is to shrink the margin between two neighbor states by reducing their threshold voltages. It can increase write speed by up to 50% while prolonging flash memory lifetime by 7.1%. A simulation study on TVR-powered SSDs shows that overall mean response time can be reduced by up to 35%. TVR serves as a case study to show how the system implications provided by our new findings can be applied to enhance MLC flash SSDs.

The remainder of the paper is organized as follows. Section 2 analyzes threshold voltage in related flash models. An empirical threshold voltage reliability model derived from experimental results is introduced in Section 3. Section 4 investigates the correlations between threshold voltage and P/E performance. The TVR technique is presented in Section 5. Section 6 briefly summarizes the related work. The paper is concluded in Section 7.

2. BACKGROUND AND MODEL ANALYSIS

2.1 ISPP Model

The basic storage unit in flash memory is a floating gate transistor, which is also called as memory cell [3]. An array of memory cells can form one or several pages depending on the number of bits that a cell can store. Typically, 64 or 128 pages are grouped as one block. While page is the smallest granularity for read and program operation, erase operation can only be performed on a block level. Because of the variation of characteristics of memory cells, the threshold voltage on each cell under the same programmed state is not uniform among cells, which results in a bell-shape threshold voltage distribution.

An incremental step pulse programming (ISPP) with a bit-by-bit verifying method [3, 25], therefore, is used to control the precision of final threshold voltage (see Figure 2). ISPP gradually increases the programming voltage from the starting voltage to the maximum voltage step by step. Between two consecutive steps, a bit-by-bit verifying operation is performed [21]. Within this process, the cells whose threshold voltages already reach their state voltage levels will be inhibited to program in next step. The step length between two adjacent steps, ΔV_{pp}, is the most important parameter for programming performance and threshold voltage distribution. First of all, the width of threshold voltage ΔV_{th} is in direct proportion to ΔV_{pp} [3]. The smaller a step length is, the more precise threshold voltage is. Also, ΔV_{pp} is proportional to programming speed [14, 23] for programming time can be shortened if ΔV_{pp} is increased. However, the programming performance improvement is at the cost of an increased raw bit error rate (RBER) [23]. From the study of Jung *et al.* [14], we can approximately model the threshold voltage programmed by ISPP as:

$$V_{th} = V_{start} + \beta \Delta V_{pp} N_s \ . \tag{1}$$

N_s represents the number of steps that a programming process needs. V_{start} is the initial voltage level of a programmed cell and β is a material related coefficient. Clearly, programming speed can be improved by reducing the number of programming steps N_s [17, 23]. Alternatively, we may also reduce the value of V_{th} while keeping ΔV_{pp} unchanged to obtain a smaller N_s based on equation (1).

2.2 Threshold Voltage Distribution Model

The threshold voltages of cells that are programmed to the same state are not identical. Therefore, a probability density function is used to describe the threshold voltage distribution of threshold voltage for each state. Even though there are asymmetries in a threshold voltage distribution, it still can be approximated by a sum of Gaussian distributions [16, 20]. Thus, the model of M-bit cell threshold voltage distribution is given by:

$$f(x) = \sum_{s=0}^{2^M-1} P(S_s) \frac{1}{\sqrt{2\pi}\delta_s} \exp\{\frac{-(x-\mu_s)^2}{2\delta_s^2}\} \ . \tag{2}$$

$P(S_s)$ is the probability of state S_s. μ_s and δ_s are the associated mean and variance in the Gaussian probability density function. Ideally, $P(S_s)$ for every state is approximately equal to $\frac{1}{2^M}$ as there are a large amount of memory cells in one flash chip.

Figure 2b illustrates the threshold voltage distribution of a 2-bit flash memory. Without losing generality, 2-bit MLC flash is used as an example in the remainder of this paper. A threshold voltage distribution for more-than-2-bit cell flash memory can be readily derived from the 2-bit cell case. While S_0 is the erase state that represents data '11', S_1, S_2, and S_3 are the other three program states, which represent '10', '00', and '01', respectively. This mapping method is known as Gray-mapping [26]. Two bits data in one cell are separated into an LSB page and an MSB page in the LSB/MSB programming scheme [7], which is widely adopted in current MLC flash memories. A margin with approximately equal width of ΔV_{th} is implemented between two adjacent states in modern flash memory. The wide margin is designed to provide a reliability mechanism for flash memory to combat the voltage drift due to memory cell defects and long retention time [3]. However, the retention problem is almost negligible when the lifetime of data is much shorter than the JEDEC standard[17, 24].

2.3 Cell-to-Cell Interference

Cell-to-cell interference, a major noise also referred to as floating gate coupling, can significantly widen the threshold voltage distribution curve for each state [8]. It degrades the overall reliability of flash memory. The reason behind is that the threshold voltage of a cell is largely affected by its surrounding cells' threshold voltages. To minimize its impact, the page programming order is restricted to an ascending order in a block [15]. In the worst case, a cell is affected by its five neighbor cells [8]. For simplicity, the floating gate coupling influence caused by the upper right cell and upper left cell is ignored because their coupling effect is relatively small compared with other direct neighbors [8]. The change of threshold voltage due to its neighbors' interference can be modeled as [8]:

$$\Delta V_{th}^{(p,q)} = \gamma_{fg1} \Delta V_{th}^{(p,q+1)} + \gamma_{fg2} (\Delta V_{th}^{(p-1,q)} + \Delta V_{th}^{(p+1,q)}) \ . \tag{3}$$

p and q denote the pth bitline and the qth wordline in a memory block. γ_{fg1} and γ_{fg2} are the floating gate coupling ratios in a bitline and in a wordline, respectively. The values of the two coupling ratios are determined by the materials and structure [8]. In the worst scenario, the ΔV_{th} between two cells is equal to the difference between V_{th} of state S_0 and V_{th} of state S_3. We can simply calculate the voltage change by:

$$\Delta V_{th}^{(p,q)} = (\gamma_{fg1} + 2\gamma_{fg2}) \Delta V_{th}^{max} \ . \tag{4}$$

Cho *et al.* discovered that the worst floating gate coupling effect is even larger than the incremental step voltage ΔV_{pp} [8]. Intuitively, without increasing the complexity of manufacturing and degrading the programming performance, we can directly reduce ΔV_{th}^{max} to tighten the threshold voltage distribution curve. For example, in case that $\gamma_{fg1} = 0.02$, $\gamma_{fg2} = 0.006$, and $\Delta V_{th}^{max} = 5.4$ [8], a $2V$ reduction of ΔV_{th}^{max} can decrease the floating gate coupling effect by 37% based on equation (4).

2.4 Read Disturb

Whenever a flash memory cell is read, a voltage V_{pass} is applied to all deselected wordlines in that block [21]. V_{pass} must be higher than the highest threshold voltage so that the deselected cells on the same wordline can serve as transfer gates, which let the read current from the cell being read to be measured [21]. The V_{pass} unintentionally injects electrons into the floating gate through either stress-induced leakage current (SILC) or tunnel oxide traps filling [21]. Consequently, it introduces bit errors if the increased V_{th} caused by injected electrons exceeds the value of read reference voltage. Mielke *et al.* found that mistakenly reading state S_0 as state S_1 is the dominant read disturb error [21]. The reason behind this is that the gap between V_{pass} and V_{th} of state S_0 is the most significant, which causes the highest field stress in the tunnel oxide under read bias. The leakage current, I, generated by field stress grows exponentially with the voltage across the tunnel oxide [2]:

$$I = I_0 \cdot e^{b_0 v_{ox}} \ . \tag{5}$$

I_0 and b_0 are two constants, and v_{ox} is the voltage applied on the tunnel oxide. Although read disturb is negligible compared with write error [21], a reduced V_{pass} determined by the highest V_{th} could eventually decrease RBER caused by read disturb.

3. IMPACT ON RELIABILITY

Several studies [7, 21, 28] characterized flash error patterns and trends to understand the correlation between flash memory reliability and P/E cycles on page level. However, since one memory cell is being pushed to store more bits, two or more pages are logically divided from one physical memory cell page. Existing page level reliability investigations [7, 21, 28] cannot reveal the relationship between the threshold voltage of an individual flash memory cell and its reliability. In this section, we empirically evaluate the reliability of flash memory cells under various threshold voltages on a hardware platform [6, 27].

3.1 Testing Methodology

The number of bit errors per page is currently used as an indicator of flash memory reliability [3]. When the number of bit errors per page exceeds the ECC capability, the original data on that page can no longer be recovered, which results in a bad block problem [3]. Because of MLC flash memory mapping scheme, a threshold voltage change (e.g., state S_0 to state S_1) on a cell can result in two consequences. Either an error occurs in both associated LSB page and MSB page or an error happens only in one associated page (i.e., either an LSB page or an MSB page) [7, 28]. The mapping scheme also leads to a phenomenon that the LSB pages generally have a higher RBER than that of MSB pages [28]. We think that the number of cell errors is more suitable than the number of bit errors per page for measuring MLC flash reliability because the influence from logical layer mapping scheme can be avoided. In our experiments, the bit errors are collected and counted in a cell level. Any bit flip in a 2-bit cell, no matter which page it belongs to, is recorded as a cell error.

Figure 3 illustrates the program/erase scheme that is used to collect cell errors in our experiments. In this scheme, P/E procedures are performed on selected blocks cycle-by-cycle. In each P/E cycle, the entire block is first erased. Next, data are programmed into each page within the block. Once all the pages have been programmed, data are immediately read back and then are compared with their original values. Finally, the number of bit flips is recorded for future analysis. The error collecting procedure is repeated for thousands of cycles until the flash memory comes to the end of its lifetime. Since current flash memory does not provide a hardware mechanism for users to dynamically adjust its threshold voltage, programming different data patterns to indirectly change threshold voltage becomes the only feasible solution. This is because MLC flash employs distinct threshold voltages to represent different data patterns [26]. In other words, programming different data patterns (i.e., ’11’, ’10’, ’00’, or ’10’) onto a cell results in distinct threshold voltages. An MSB page and its associated LSB page are grouped as a cell page. Several scattered cell pages within one block are selected in a way that cell-to-cell interference among them can be ignored. During each P/E cycle, different cell pages are programmed with different data patterns. Within each cell page, however, all cells are programmed with the same data pattern, which represents a particular threshold voltage (i.e., V_{th0}, V_{th1}, V_{th2}, or V_{th3}). In this way, different cell pages are programmed to different threshold voltages in every P/E cycle. After N cycles, the accumulated threshold voltage for each page is NV_{th0}, NV_{th1}, NV_{th2}, and NV_{th3}, respectively.

The program/erase testing scheme explained above eliminates the cell-to-cell interference within one cell page because the same content is programmed across the entire page. In order to mimic a practical flash memory usage pattern, the cell-to-cell interference effect needs to be taken into consideration. Therefore, in our testing scheme, the available P/E cycles of a flash memory are divided into multiple segments, with each consisting of 1,000 P/E cycles. During the first 900 P/E cycles of each segment, the same content is repeatedly programmed into each cell of a selected cell page to simply increase the accumulated threshold voltage without counting the cell errors. Periodically programming the same content to a particular cell page makes the discrepancies of accumulated threshold voltage among cell pages more obvious. Next, in the last 100 P/E cycles a batch of pseudo-random data are programmed and cell errors are collected after immediate reading back so that the testing scheme can simulate a real application environment, whereas the impact of different threshold voltages can still be measured. Figure 3 shows an illustrative example of this procedure. An example of selected pages are shown in Table 1. Page 10 and 16 belong to the same cell page that resides on wordline 4. While page 10 is an MSB page, page 16 is an LSB page. This cell page will be programmed by content ’11’, which can be performed by programming ’1’ to page 10 at first, and then writing one-page ’1’ data to page 16. Page 26 and 32 form a cell page and will be programmed as ’10’. Cell page that contains page 42 and 48 will be programmed as ’00’. The cell page on wordline 16 will be written as ’01’.

A hardware platform, which consists of a Xilinx XUPV5-Lx110t evaluation board [27] and a Ming II flash daughter board [6], is built so that commands can be issued directly to raw flash chips without ECC. The flash memory used in our experiments is 2-bit MLC NAND flash, which is specified to

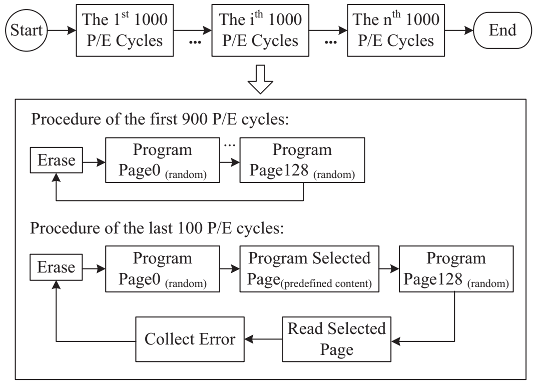
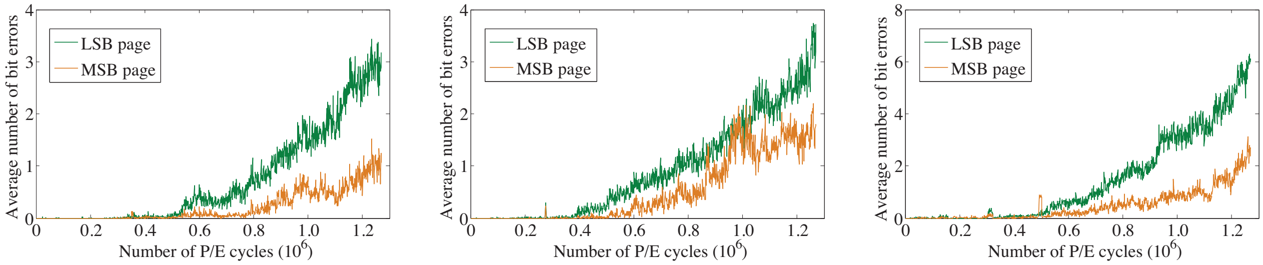


Figure 3: Flash memory error collecting scheme.

Table 1: An example of selected pages.

Page No.	*Wordline*	*Page Type*	*Program Pattern*
10	4	MSB page	1
16	4	LSB page	1
26	8	MSB page	1
32	8	LSB page	0
42	12	MSB page	0
48	12	LSB page	0
58	16	MSB page	0
64	16	LSB page	1



(a) Cell page programmed as 10

(b) Cell page programmed as 00

(c) Cell page programmed as 01

Figure 4: Programming errors in LSB and MSB page within one cell page.

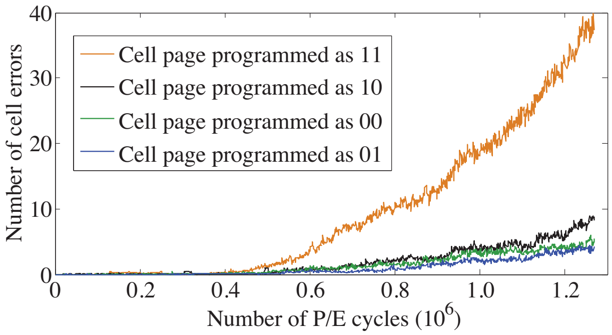


Figure 5: Average number of cell errors versus P/E cycles.

survive 10,000 P/E cycles under 10 year data retention time by using a minimum of 4-bit ECC per 528 bytes of data and a bad-block replacement algorithm [19]. All experiments are conducted in a controlled laboratory environment with a room temperature. The error data are collected from a limited number of blocks on a few flash chips from a particular manufacturer.

3.2 Experimental Results

Figure 5 shows the number of cell errors during the entire flash memory's lifetime. The cell errors are collected in the last 100 P/E cycles within each 1,000 P/E segment. All the lines are serrated because the number of cell errors between two consecutive P/E cycle segments fluctuates significantly. Clearly, the cell errors increase as the number of P/E cycles becomes larger. Further, pages programmed to different threshold voltages generate different number of cell errors. The number of cell errors in each cell page grows exponentially when the number of P/E cycles enlarges.

The cell page programmed as '11' exhibits the most unreliable characteristic as it has much more errors than that of the other three pages. In addition, the number of cell errors on this cell page grows much faster than the other three cell pages as the number of P/E cycles increases. In fact, errors in this cell page are erase errors, whereas the other three cell pages have programming errors. Since '11' represents the erase state, no programming process is carried out when '11' is written to a memory cell. This result suggests that a flash memory cell is more likely to be unreliable if either it is continuously erased without any data programmed or it is always programmed as '11' during a long period of time. The reason behind this is that a negative voltage is applied on the floating-gate during an erase operation, which causes damage to the tunnel oxide [20]. A programming process, on the other hand, can neutralize and mitigate such damage because of a positive voltage applied on that gate. Therefore, memory cells without programming processes cannot alleviate the erase damage, and thus, generate much more errors compared with the cells on which erase and program operation are alternately performed.

Experimental results obtained from the other three cell pages are consistent with real applications. Figure 5 illustrates that among those three cell pages the cell page programmed as '01' has the largest number of cell errors. On the contrary, the cell page programmed as '10' is more reliable because of the smallest number of cell errors. The number of cell errors from the cell page programmed as '00' lies between that of the '01' cell page and the '10' cell page. According to the mapping method used, it is clear that the cell page always programmed to the lowest threshold voltage (i.e., '10') is more reliable than the cell pages often programmed to a higher threshold voltage (i.e., '00' and '01'). A detailed mathematical analysis will be provided to establish a model, which defines the relationship between threshold voltage and flash memory reliability.

In a 2-bit MLC flash memory, one cell page contains two logical pages referred to as MSB page and LSB page, respectively. Figure 4 shows the bit errors in an MSB page and an LSB page. From the figure, we can see that LSB pages generally have a larger number of bit errors than that of MSB pages under all programming cases. This is due to the fact that the assignment of 2-bit data to threshold voltages within a cell makes an LSB page more susceptible to error than an MSB page [28, 16]. Therefore, the conclusion that LSB pages are more prone to errors than MSB pages holds under various threshold voltages. The errors of '11' cell page are erase errors, while the errors in the other three pages are programming errors. Therefore, the error data of '11' is not illustrated here.

System implications: Continuously programming '11' to a cell page or erasing a block without any data programmed should be avoided, otherwise the reliability of the certain cell page degrades rapidly. Besides, writing data patterns that are represented by a lower threshold voltage could prolong flash's lifetime.

3.3 Model Establishment

In this section, we build an empirical threshold voltage reliability model for MLC flash memory. Cell pages programmed as '11' do not reflect real application situations

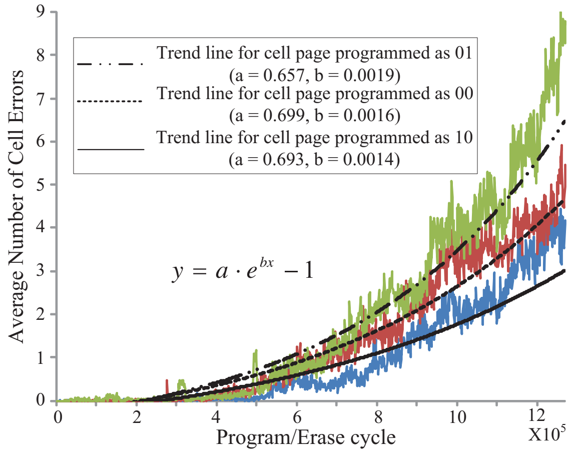


Figure 6: Fitting to exponential trends.

because they only generate erase errors. Thus, we only consider the data collected from the other three cell pages.

Figure 6 clearly demonstrates that the cell errors increase exponentially as the P/E cycle becomes larger. We adopt the exponential law model to describe reliability behavior of the three cell pages in Figure 6 (nonlinear least squares fitting method). Two new parameters a and b are defined in this model. We determine them through parameter fitting. It is clear that each cell page has its own value of a and b as shown in Figure 6. This model, however, only gives us the relationship between the number of cell errors (i.e., reliability) and the P/E cycles for a cell page. It is obvious that threshold voltages of cell pages vary because they are programmed to distinct data patterns. Hence, we can then apply the parameter fitting method again for the three cell pages to obtain the relationship between a threshold voltage and a combination of a and b.

Using the mean distribution voltage values listed in Figure 11a, we can approximate the threshold voltage for the three cell pages. Applying the parameter fitting we get the two equations (6) and (7) to describe the relationship between threshold voltage and parameters a and b.

$$a = -5E^{-4}ln(V_{th}) + 0.0019 \ , \qquad (6)$$

$$b = 0.036ln(V_{th}) + 0.6616 \ . \qquad (7)$$

We substitute a and b in the exponential law model and get an empirical reliability model with respect to threshold voltage V_{th} shown as below:

$$Err = (-5E^{-4}ln(V_{th}) + 0.0019)e^{(0.036ln(V_{th})+0.6616)N} - 1 \ , \qquad (8)$$

where V_{th} is the threshold voltage of flash memory cells. N is the number of P/E cycles, and Err is the number of cell errors. The value of cell errors is the indicator of flash reliability. Equation (8) can be generalized as:

$$Err = (\alpha_1 ln(V_{th}) + \beta_1)e^{(\alpha_2 ln(V_{th})+\beta_2)N} - 1 \ . \qquad (9)$$

The parameters $\alpha_1, \beta_1, \alpha_2,$ and β_2 are determined by the characteristics of a flash memory. The empirical reliability model presented by equation (9) discloses the relationship between threshold voltage and flash reliability.

The empirical threshold voltage reliability model in its current format has one limitation. The error data is collected by reading the programmed data back immediately. Therefore, the time-dependent behavior of RBER is not considered in our model. However, Belgal *et al.* discovered that in the first few weeks of flash retention period almost no retention error is detected [2]. A study on a wide range of real-world traces found that 49-99% of writes require less than 1-week retention time [17]. Therefore, time-dependent characteristics can be safely ignored by the model for workloads that require short retention time.

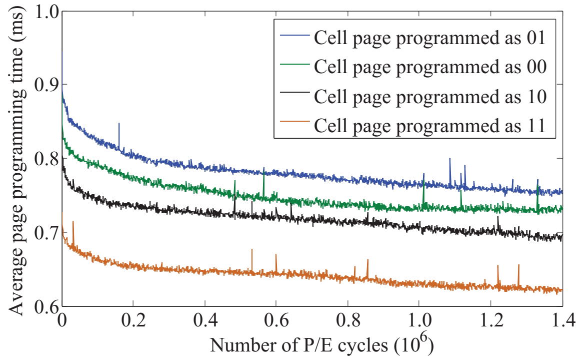


Figure 7: Average page programming performance.

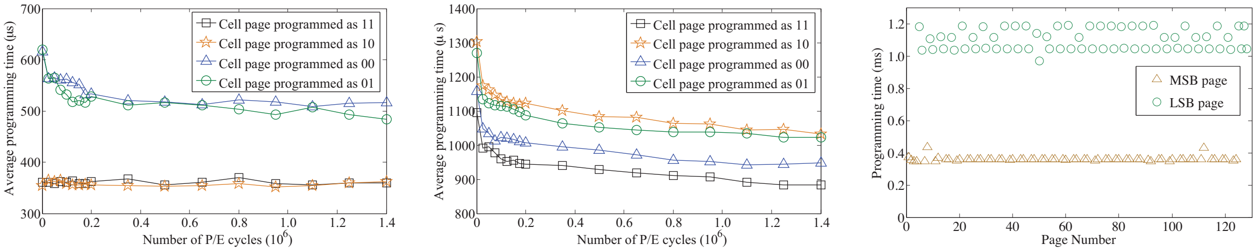
4. IMPACT ON P/E PERFORMANCE

The P/E performance of MLC flash under different threshold voltages are examined in this section. Similar to the testing methodology used in Section 3, a P/E procedure is performed on flash cycle-by-cycle until flash reaches the end of its lifetime. In each P/E cycle, all cell pages in a block are programmed one-by-one to a particular threshold voltage and then the entire block is erased. By measuring the programming and erase time we can evaluate the P/E performance under a particular threshold voltage.

4.1 Page Programming

Page programming time under different threshold voltages is shown in Figure 7. It is obtained by averaging the programming time of all pages within a block whose memory cells are programmed to a particular threshold voltage. Two interesting observations are made. Firstly, pages programmed to a lower threshold voltage have a better programming performance. For example, programming a page to state S_0 (i.e., cell page programmed as '11') typically costs 672 μs, whereas pages programmed to state S_3 need roughly 880 μs. On average, the speed of programming a page to state S_0 is 15.5%, 23%, and 31% faster than that of programming a page to state S_1, S_2, and S_3, respectively. Secondly, under all threshold voltage situations the programming time decreases as the number of P/E cycles increases. The rationale behind it is that as the tunnel oxide of a memory cell wears out (i.e., the number of P/E cycles increases) electrons are more easily to be injected into a cell's floating gate [20]. Further, the programming time decreases rapidly in the early lifetime of MLC flash. In the range of 1 to 2×10^5 P/E cycles, programming time reduces 10.3% on average. In the rest of flash memory's lifetime, programming time only decreases 4.8%.

Figure 8 illustrates the programming performance of different page types. Figure 8a shows average programming



(a) MSB page programming time (b) LSB page programming time (c) Page programming time in a block

Figure 8: Programming time of LSB and MSB page in a block.

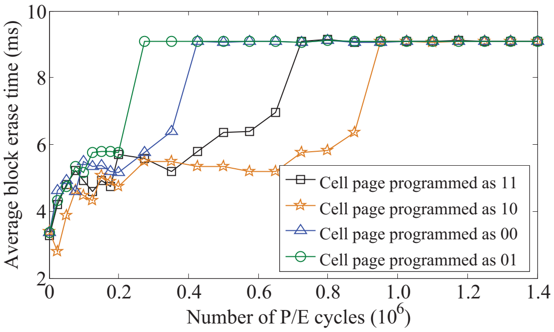


Figure 9: Average block erase time.

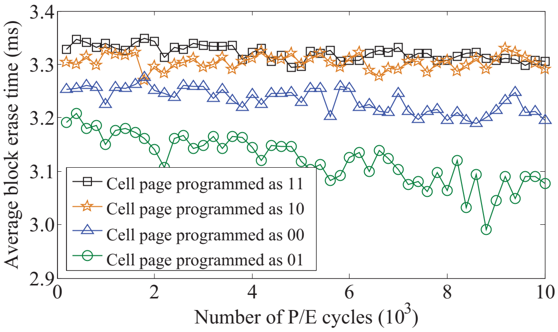


Figure 10: Average block erase time in small P/E cycles.

time of MSB pages versus P/E cycles and Figure 8b gives that of LSB pages. It is clear that programming speed of an MSB page is much faster than that of an LSB page. For example, MSB pages programmed as ’11’ (i.e., the lowest threshold voltage) only take around 380 μs for a programming operation. Writing the same content onto an LSB page, surprisingly, needs around 1,100 μs. Besides, programming time of MSB pages that are programmed as ’00’ and ’01’ (i.e., state S_2 and S_3) is 1.42 times longer than that of MSB pages programmed as ’11’ and ’10’ (i.e., state S_1 and S_1) on average. The programming time of LSB pages, on the contrary, does not show the same trend as that of MSB pages does. LSB pages programmed as ’11’ have the lowest programming time (on average 919 μs), whereas programming an LSB page to ’10’ state takes the longest programming time (1,082 μs). Figure 8c illustrates the programming speed of each page within a block, which is consistently programmed as ’11’ when the number of P/E cycles equals to 2,000. It is obvious that the programming time of each page varies. While MSB pages have almost the same programming time, the programming time of LSB pages varies substantially. Further, the programming time of MSB pages is much shorter than that of LSB pages.

4.2 Block Erase

Figure 9 shows the block erase time under different threshold voltages in a flash memory’s entire lifetime. It is clear that the erase time increases and varies largely as the number of P/E cycles enlarges. The erase time in all threshold voltage situations increases to 9 ms when flash comes to the end of its lifetime. When the number of P/E cycles is greater than 2×10^5, the erase time of differently programmed blocks exhibits noticeable differences. for instance, the erase time of blocks programmed as ’01’ rapidly goes to 9 ms, whereas erase time of blocks programmed as ’10’ increases much slower. The reason why erase time in all situations finally becomes 9 ms is uncertain.

Figure 10 illustrates the erase time in flash’s early lifetime (P/E cycles $< 1 \times 10^4$). Blocks programmed to a higher threshold voltage have a better erase performance. On average, blocks programmed as ’01’ have the fastest erase speed, which is 3.1 ms. Blocks programmed as ’11’ on average need 3.3 ms to finish an erase operation.

System implications: 1) programming content that is represented by a lower threshold voltage can have a higher P/E performance; 2) judiciously rearranging the programming order could improve SSD’s overall performance as the programming speeds on an MSB page and an LSB page are noticeably different; 3) intentionally choosing blocks that have a faster erase speed can reduce garbage collection cost.

5. TVR: A CASE STUDY

This section demonstrates an example of how to apply the system implications provided by our new findings on SSD design. We first present the TVR approach based on an approximate V_{th} distribution [23]. Next, a method of calculating the amount of reduced V_{th} is illustrated. Finally, after quantitatively analyzing programming speed improvements caused by TVR, a simulation study on its impact on SSDs is briefly presented.

5.1 Threshold Voltage Reduction

A standard 2-bit MLC threshold voltage distribution model [16] is illustrated in Figure 11a. Three margins (D_0, D_1, and D_2) between every two adjacent states are configured to increase reliability so that after a long period of time data can still

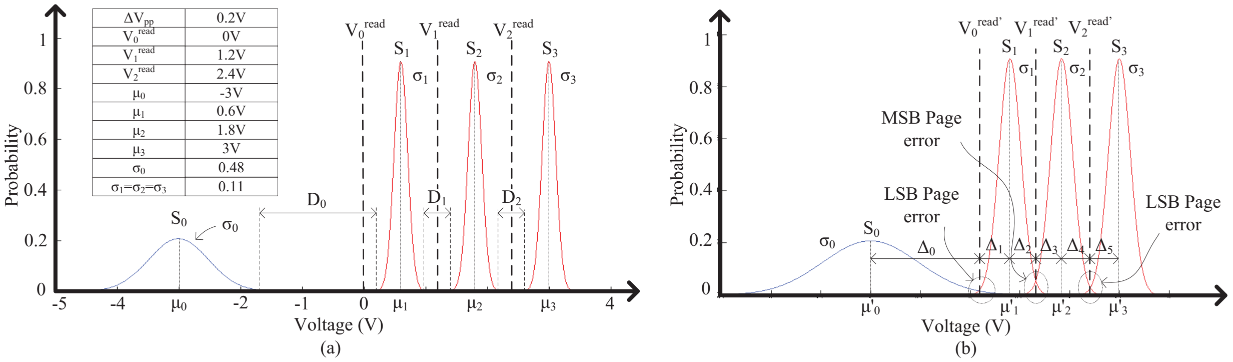


Figure 11: 2-bit MLC flash memory threshold voltage distribution; (a) the threshold voltage distribution simulation result of the flash memory with standard threshold voltage levels; the value of parameters used in the simulation is shown in the table; (b) an example of a retention free threshold voltage distribution; $\mu_0', \delta_0, \delta_1, \delta_2,$ and δ_3 are the same as the value in (a).

be read correctly with changed threshold voltage distributions. A recent investigation [17] on a wide range of real-world workloads, however, discoveries that data retention capability offered by the worst-case oriented design is always under-utilized, especially in the early lifetime of a flash memory when data retention problem even does not exist. For example, it finds that 49-99% of writes require less than 1-week retention time [17]. In typical data center workloads like proxy and MapReduce, most data are overwritten frequently, which suggests a short data retention in only days or even hours [17]. Therefore, in a retention relaxation situation such as flash memory used to store log data, the width of those margins can be largely reduced without affecting reliability. Based on the observation of over-provisioned data retention time from [17] and our new findings, we propose the TVR approach. It improves programming performance and reliability by reducing threshold voltages of all states. As a result, margins between two adjacent states can be shrunk. The amounts of reduced threshold voltages lie in a range between zero to a standard margin width. An extreme scenario of TVR is to shrink the width of all the three margins to zero (see Figure 11b), which is called a retention free case. In the remainder of this section, we use the retention free case to quantitatively analyze the maximum improvement in performance and reliability for TVR-powered flash.

In our analysis, the tails of a Gaussian distribution curve are used to calculate RBER because raw bit errors only occur in the overlap area between the tails of two neighbor threshold voltage distribution curves [16]. The two Gaussian tails are asymmetric because V_{th} is more prone to enlarge in a programming process, while retention time diminishing the value of V_{th} [21]. However, since the retention free case does not consider retention errors, the retention free threshold voltage distribution simply takes the two Gaussian tails as symmetric, which is also suggested by [16, 23].

Similar to the simplification used in [23], we set the variances of the three non-erased states as the same ($\delta_1 = \delta_2 = \delta_3$). According to [4], parameters like μ_s, δ_s, V_s^{read}, and ΔV_{pp} are set as the values shown in Figure 11a. Based on the simulation using the defined parameters, it is clear that the erase state S_0 has a flattened bell-shape while the three non-erased states S_1, S_2, and S_3 are much taller and tighter due to the small value of step length in a programming process. Three read reference voltages (i.e., V_0^{read}, V_1^{read}, and V_2^{read}) are set for read operation. V_1^{read} is used to identify the content in an MSB page. If a cell's threshold voltage is higher than V_1^{read}, it is read as '0'. Otherwise, it is read as '1'. For an LSB page, two voltage comparisons are performed. If a cell's threshold voltage lies between V_0^{read} and V_2^{read}, it is taken as '0'. Otherwise, the content is identified as '1'. Based on this strategy, the bit errors in an MSB page always happen in an area around V_1^{read} while bit errors in an LSB page come from the areas around V_0^{read} and V_2^{read}.

5.2 Quantitative Analysis of TVR

Figure 11b illustrates a threshold voltage distribution after applying TVR. When TVR is applied, RBER increases because the voltages of cells at the tail of the distribution will be misread when they pass the read reference voltage. The probability that the voltages of cells cross the read reference voltage can be calculated by the tail probability function [16, 29]. This value is the same as RBER. However, the calculation methods for MSB pages and LSB pages are different for their reading/programming schemes are distinct [16].

Since the number of errors caused by misinterpreting a state as a non-immediate-neighbor state is very small [7, 16], only the errors generated by state changes in two adjacent states are considered in our computation. For an MSB page, misreading errors are around $V_1^{read'}$. RBER is computed by the equation below:

$$\mathrm{RBER}_{\mathrm{MSBpage}} = \frac{1}{4}Q_1(\frac{|\Delta_2|}{\delta_1}) + \frac{1}{4}Q_2(\frac{|\Delta_3|}{\delta_2}) \; . \tag{10}$$

For an LSB page, however, misreading errors are around $V_0^{read'}$ and $V_2^{read'}$. Therefore, the calculation of RBER is revised as:

$$\begin{aligned} \mathrm{RBER}_{\mathrm{LSBpage}} = {} & \frac{1}{4}Q_0(\frac{|\Delta_0|}{\delta_0}) + \frac{1}{4}Q_1(\frac{|\Delta_1|}{\delta_1}) \\ & + \frac{1}{4}Q_2(\frac{|\Delta_4|}{\delta_2}) + \frac{1}{4}Q_3(\frac{|\Delta_5|}{\delta_3}) \; . \end{aligned} \tag{11}$$

Δ_i ($0 \le i \le 5$) is the distance from μ_s' to its adjacent read reference voltage $V_s^{read'}$ as shown in the Figure 11b. $Q_s(x)$ ($s = 0, ..., 3$) is the tail probability function of each state. As explained in Section 4.1, the threshold voltage distribution is asymmetric as the errors caused by a state change from a lower voltage state to a higher voltage state (i.e., $Q_0(\frac{|\Delta_0|}{\delta_0})$, $Q_1(\frac{|\Delta_2|}{\delta_1})$, and $Q_2(\frac{|\Delta_4|}{\delta_2})$) are mainly generated by programming processes. Majority errors due to a reverse change (i.e., $Q_1(\frac{|\Delta_1|}{\delta_1})$, $Q_2(\frac{|\Delta_3|}{\delta_2})$, and $Q_3(\frac{|\Delta_5|}{\delta_3})$) are caused by retention time [21]. To simplify the computation, we assume that the errors made by a forward state change (i.e., from a lower voltage state to a higher voltage state) and a backward state change (i.e., from a higher voltage state to a lower voltage state) have the same probability. Hence, we have:

$$Q_1(\frac{|\Delta_2|}{\delta_1}) = Q_2(\frac{|\Delta_3|}{\delta_2}) \ , \tag{12}$$

$$Q_0(\frac{|\Delta_0|}{\delta_0}) = Q_1(\frac{|\Delta_1|}{\delta_1}) = Q_2(\frac{|\Delta_4|}{\delta_2}) = Q_3(\frac{|\Delta_5|}{\delta_3}) \ . \tag{13}$$

The maximum RBER tolerated by flash memory is determined by the adopted ECC capability and the size of redundant area fabricated by manufacturers [3, 21]. For modern flash memory, RBER must be lower than 4.5×10^{-4} if the uncorrectable bit error rate requirement is set to be 10^{-16} [17] . By solving equations (10) and (11) using parameters shown in Figure 11a, we obtain:

$$\Delta_0 \approx 1.40; \ \Delta_1 = \Delta_4 = \Delta_5 \approx 0.36; \ \Delta_2 = \Delta_3 \approx 0.34.$$

Δ_i $(0 \le i \le 5)$ gives the distance between mean threshold voltage of each state and its neighbor read reference voltage in the maximum threshold voltage reduction situation. Finally, the reduced threshold voltages can be calculated based on μ_0' and Δ_i $(0 \le i \le 5)$. We find that Δ_i $(i = 2, 3)$ is smaller than Δ_j $(j = 1, 4, 5)$.

5.3 Programming Speed Improvement

Compared with read operation, the programming speed in flash memory is one order of magnitude slower. Thus, the programming speed is a main performance bottleneck for flash memory. Flash memory programming speed depends on the slowest memory cell in an ISPP programming process [25]. Equation (1) gives the relationship between a threshold voltage and the number of programming steps under certain step length ΔV_{pp}. It can be rearranged as:

$$\lceil N_s \rceil = \frac{\Delta V_{th}}{\beta \Delta V_{pp}} \ . \tag{14}$$

$\Delta V_{th} = V_{th} - V_{start}$ and $\lceil N_s \rceil$ stands for the smallest integer bigger than N_s. Parameter β is determined by the ISPP model based on the curve-fitting values shown in [14]. When $\Delta V_{pp} = 0.2V$ (see Figure 11a) the model fits empirical data [14] very well with β equal to 1.14.

According to reading and programming strategy in MLC flash memory, the largest ΔV_{th} in an MSB page is given by $\mu_2 - \mu_0$ (i.e., from state S_0 to S_2), whereas $\mu_1 - \mu_0$ (i.e., from state S_0 to S_1) yields the largest ΔV_{th} in an LSB page. Therefore, before TVR is applied, the programming time for an MSB page is $22 \cdot t_{step}$ according to equation (14), where t_{step} is the time for one ISPP step. For an LSB page, the programming time is approximately equal to $16 \cdot t_{step}$. On average, the programming time can be evaluated by $19 \cdot t_{step}$.

After applying TVR, the largest ΔV_{th} in an MSB page programming process is $\sum_{i=0,1,2,3} \Delta_i = 2.44$. Hence, programming time can be reduced to $11 \cdot t_{step}$. Similarly, in an LSB page, the largest ΔV_{th} is $\sum_{i=0,1} \Delta_i = 1.76$ and programming time is $8 \cdot t_{step}$. The average programming time after applying TVR is $9.5 \cdot t_{step}$. Compared with a non-TVR scenario, programming speed is improved by 50%.

5.4 Reliability Improvement

By using the empirical reliability model established in Section 3, it is easy for us to evaluate the reliability improvement via threshold voltage reduction. If data size written to each cell page is sufficiently large and random, we can assume that the four different threshold voltage states shown in Figure 11 have an equal chance to be written into a cell page. Thus, we can use the mean voltage throughout flash memory's lifetime to get the number of cell errors at a given P/E cycle. Using the results got from Section 5.2, if RBER is lower than 4.5×10^{-4} [17] the maximum number of P/E cycles is 12.469×10^5 according to equation (8). In the TVR approach, however, the maximum number of P/E cycles that a flash memory can reach is 13.35×10^5. Therefore, the TVR can improve the flash memory reliability by 7.1%.

5.5 Impact on SSDs

To understand the impact of TVR on the overall performance of SSDs, we carry an experimental study on TVR-powered SSDs based on a validated simulation environment (i.e., DiskSim 4.0 [5] and the Microsoft SSD module [1]) and six real-world traces including enterprise data center applications (e.g., Financial1, Financial2) as well as file system benchmarks on workstations (e.g., Iozone, Postmark). Simulation results demonstrate that TVR can reduce SSD's overall mean response time including read and write by 11% to 35%. Besides, TVR consistently increases an SSD's overall performance as the number of packages enlarges. Due to space limit, detailed simulation results are not included.

6. RELATED WORK

Several recent studies [11, 17, 29] have discussed about how to tactfully control the threshold voltage to improve the performance and endurance of flash memory and SSD. The threshold voltage distribution will drift as the number of P/E cycles and retention time increase [29]. Hence, the predetermined fixed read reference voltage often results in significant asymmetric errors in flash memory's lifetime. To overcome this problem, Zhou *et al.* introduced a dynamic reading thresholds scheme, which is applied to single-level cells to reduce raw bit errors caused by voltage drift [29]. Further, Sala *et al.* extended their dynamic reading thresholds scheme to MLC memories [11]. The enhanced dynamic threshold voltage scheme combined with new ECC can significantly improve the reliability of a flash memory especially when it is aged. In contrast, in this research TVR dynamically changes both read reference voltages and state voltages in an MLC flash to improve the write speed.

The write speed of flash memory is in direct proportion to the number of programming steps in ISPP [25]. By examining such process, Pan *et al.* first proposed a device-aware design strategy exploiting the under-utilized ECC redundancy to improve flash memory performance [23]. Its basic idea is to reduce the number of programming steps by increasing the step length in ISPP and use the under-utilized ECC redundancy to correct the errors caused by step length increasing. Liu *et al.*, on the other hand, found that the redundant ECC capability and data retention are usually under-utilized after studying a wide range of real-world traces [17]. A retention-aware flash translation layer is proposed to optimize the overall performance of flash SSDs [17]. TVR employs a totally different approach to exploiting flash SSDs' over-provisioned data retention capability to improve performance and reliability. Instead of shrinking step length, TVR reduces threshold voltages to trade the over-provisioned data retention capability for an improved performance and longevity.

7. CONCLUSIONS

Thanks to recent advances in flash controller technology, cost-effective MLC flash SSDs are gradually becoming popular storage devices in data centers [12, 13, 18]. To further decrease its cost, manufacturers are aggressively pushing flash into smaller geometries and to store more bits per cell. As a result, the reliability of MLC flash continuously decreases, which demands an increasing ECC capacity and controller capability [10]. However, the speed of controller technology development might lag behind the increasing error rate [12]. A main reason for the increasing error rate is that threshold voltage ranges have been largely reduced [10, 28]. Since threshold voltage plays a central role in flash reliability and performance, in this paper we investigate the impact of threshold voltage on the performance and reliability of MLC flash memory. We find that the P/E performance and reliability of MLC flash are highly correlated to threshold voltages. Several important observations have been made. Next, a flash reliability model is established based on experimental results to reveal the relationship between the threshold voltage and the number of bit errors. Finally, we conduct a case study (i.e., the TVR approach) on how to apply the insights derived from our new findings on enhancing MLC flash SSDs. TVR can transform over-provisioned flash memory data retention capability into an increased write speed and longevity. A mathematical analysis demonstrates that TVR improves write speed by up to 50% and prolong flash memory's lifetime by 7.1%. A simulation study shows that TVR reduces mean response time by up to 35%.

In future work, we will conduct a comprehensive study on the system implications provided by this research and their interplay on the overall performance and reliability of MLC flash SSDs. For example, MLC flash P/E performance and reliability can be improved by judiciously rearranging page programming order and reassembling page content.

Acknowledgments

We would like to thank Steven Swanson and his Non-volatile Systems Laboratory for their support on the hardware platform. Also, we would like to thank the anonymous reviewers for their constructive comments that improve this paper. This work was supported in part by the U.S. National Science Foundation under grant CNS (CAREER)-0845105.

8. REFERENCES

[1] N. Agrawal, et al. Design tradeoffs for ssd performance. In *USENIX ATC*, pages 57–70, 2008.

[2] H. P. Belgal, et al. A new reliability model for post-cycling charge retention of flash memories. In *Reliability Physics Symposium*, pages 7–20, 2002.

[3] R. Bez, et al. Introduction to flash memory. *Proceedings of the IEEE*, 91(4):489–502, 2003.

[4] J. Brewer and M. Gill. *Nonvolatile Memory Technologies with Emphasis on Flash: A Comprehensive Guide to Understanding and Using Flash Memory Devices*. Wiley-IEEE Press, 2011.

[5] J. S. Bucy, et al. The disksim simulation environment version 4.0 reference manual. *CMU*, 2008.

[6] T. Bunker, M. Wei, and S. J. Swanson. Ming ii: A flexible platform for nand flash-based research. Technical report, UCSD, 2012.

[7] Y. Cai, et al. Error patterns in mlc nand flash memory: Measurement, characterization, and analysis. In *DATE*, pages 521–526, 2012.

[8] T. Cho, 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*, 2001.

[9] J. Cooke. The inconvenient truths about nand flash memory. *Micron MEMCON*, 7, 2007.

[10] E. Deal. Trends of nand flash memory error correction. *Cyclic Design, June*, 2009.

[11] S. Frederic, G. Ryan, and D. Lara. Dynamic threshold schemes for multi-level nonvolatile memories. Non-volatile Memory Workshop, 2013.

[12] C. George. Will MLC SSD Replace SLC?, 2011.

[13] J. He, et al. Dash: a recipe for a flash-based data intensive supercomputer. In *SC*, pages 1–11, 2010.

[14] T.-S. Jung, et al. A 117-mm^2 3.3-v only 128-mb multilevel nand flash memory for mass storage applications. *Solid-State Circuits, IEEE Journal of*, 31(11):1575–1583, 1996.

[15] T.-S. Jung, et al. A 3.3 V 128 Mb multi-level nand flash memory for mass storage applications. In *42nd IEEE ISSCC*, pages 32–33, 1996.

[16] Y. Kim, et al. Verify level control criteria for multi-level cell flash memories and their applications. *Journal on Advances in Signal Processing*, 2012.

[17] R.-S. Liu, et al. Optimizing nand flash-based ssds via retention relaxation. *In UESNIX FAST*, 2012.

[18] H. Marks. SSDs in the data center: SLC out, MLC in, Dec. 2012.

[19] Micron. MT29F8G08MAAWC datasheet.

[20] N. Mielke, et al. Flash eeprom threshold instabilities due to charge trapping during program/erase cycling. *IEEE TDMR*, 4(3):335–344, 2004.

[21] N. Mielke, et al. Bit error rate in nand flash memories. In *IEEE IRPS*, pages 9–19, 2008.

[22] M. Moshayedi and P. Wilkison. Enterprise ssds. *Queue*, 6(4):32–39, 2008.

[23] Y. Pan, G. Dong, and T. Zhang. Exploiting memory device wear-out dynamics to improve nand flash memory system performance. In *USENIX FAST*, 2011.

[24] J. Standard. Stress-test-driven qualification of integrated circuits. *JEDEC*, pages 1–26, 2010.

[25] K.-D. Suh, et al. A 3.3 v 32 mb nand flash memory with incremental step pulse programming scheme. *Journal of Solid-State Circuits*, 30(11), 1995.

[26] Z. Wang, et al. Reliable mlc nand flash memories based on nonlinear t-error-correcting codes. In *IEEE DSN*, pages 41–50, 2010.

[27] Xilinx. Xilinx university program xupv5-lx110t development system.

[28] E. Yaakobi, et al. Error characterization and coding schemes for flash memories. In *GLOBECOM Workshops*, pages 1856–1860. IEEE, 2010.

[29] H. Zhou, et al. Error-correcting schemes with dynamic thresholds in nonvolatile memories. In *ISIT*, 2011.