Error Patterns in MLC NAND Flash Memory: Measurement, Characterization, and Analysis

Yu Cai[1], Erich F. Haratsch[2], Onur Mutlu[1] and Ken Mai[1]
[1]Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA
[2]LSI Corporation, 1110 American Parkway NE, Allentown, PA
[1]{yucai, onur, kenmai}@andrew.cmu.edu, [2]erich.haratsch@lsi.com

Abstract— **As NAND flash memory manufacturers scale down to smaller process technology nodes and store more bits per cell, reliability and endurance of flash memory reduce. Wear-leveling and error correction coding can improve both reliability and endurance, but finding effective algorithms requires a strong understanding of flash memory error patterns. To enable such understanding, we have designed and implemented a framework for fast and accurate characterization of flash memory throughout its lifetime. This paper examines the complex flash errors that occur at 30-40nm flash technologies. We demonstrate distinct error patterns, such as cycle-dependency, location-dependency and value-dependency, for various types of flash operations. We analyze the discovered error patterns and explain why they exist from a circuit and device standpoint. Our hope is that the understanding developed from this characterization serves as a building block for new error tolerance algorithms for flash memory.**

Keywords-NAND flash; error patterns; endurance; reliability; error correction

I. Introduction

NAND flash memory [1] has been widely used as a storage medium for many systems such as laptops, PDAs, and mobile phones because of its high performance, large storage density, non-volatility and low power consumption. The per-bit cost of NAND flash memory continues to fall dramatically every year due to aggressive technology scaling and the introduction of multi-level flash cells. This allows NAND flash to be applicable for even more applications, such as solid-state disks (SSDs) for personal computers and enterprise servers.

However, the widespread adoption of flash-based storage in performance-intensive applications has led to concerns regarding the reliability and endurance of the underlying flash memories. A flash memory cell has limited endurance, i.e. data cannot be reprogrammed into the cell more than a limited number of times. A single-level flash cell (SLC) can tolerate ~10k program/erase (P/E) cycles while a 2-bit multi-level cell (MLC) can only survive for ~3k P/E cycles for 30-40nm (i.e., 3x-nm) technology generations [2]. The available P/E cycles are expected to decrease even more in the near future as flash cells continue to scale down in size and more than 2 bits are programmed per cell. Generally, storage systems have strict requirements on reliability. For example, the uncorrectable bit error rate during usage should be less than 10^{-15} and stored data should be available for 5-10 years [3]. Enterprise-class SSDs are expected to support at least 10 full disk writes per day for at least five years under fully random data patterns. Assuming typical write amplification of 2 times (due to additional writes caused by garbage collection and wear leveling [4]) and ideal wear-leveling, current MLC flash based storage will use up all its reliable P/E cycles (e.g., 3000) within 5 months. It is therefore clear that flash memories cannot satisfy the lifetime requirements for enterprise SSDs, which require much longer than 5 months of lifetime.

Various endurance tolerance solutions have been proposed and implemented on the flash controller to improve flash memory lifetime reliability. Wear-leveling algorithms and error correction codes (ECC) [5] are the most widely used techniques. Wear leveling primarily attempts to prolong the service life of flash memory by arranging data so that program/erase events are evenly distributed across the entire flash memory: the goal is to have all storage blocks deteriorate at the same speed and come to the end of lifetime at the same time. ECC primarily leverages additional parity bits to protect stored data against errors introduced by the flash mediums. The effectiveness of both wear-leveling and ECC designs are highly dependent on the specific error properties of flash memory. To design effective and low-cost algorithms that can take advantage of underlying error properties, system engineers must understand the error patterns of flash memories.

Our goal in this paper is to enable a strong understanding of the error patterns observed in contemporary flash memory, with the goal of aiding system designers in developing more effective and low-cost error tolerance mechanisms. To accomplish this goal, we have designed and implemented an FPGA-based framework for fast and accurate characterization of flash memory errors throughout its lifetime. Using this platform, we have measured and characterized error patterns for state-of-the-art 3x-nm NAND flash memories. We present the results of our measurements and characterization, and provide insight into why the observed error patterns happen in flash memory. Developing error tolerance techniques that leverage our characterization and understanding is out-of-scope of this paper, yet we expect this is an important area of future work.

To our knowledge, this, along with concurrent work [16], is the first paper that empirically analyzes error patterns in 3x-nm flash memory. The major contributions and new observations of this paper beyond previous work, including [16], are as follows:

1) We characterize and analyze errors in modern flash memory from flash controller's point of view, categorizing them into four types: erase errors, program interference errors, retention errors and read errors.

2) We show the relationship between various types of errors and demonstrate empirically using real 3x-nm flash chips that retention errors are the most dominant error type.

3) We demonstrate that different flash errors have distinct patterns: retention errors and program interference errors are P/E-cycle-dependent, memory-location-dependent, and data-value-dependent. To our knowledge, this is the first paper to empirically demonstrate the location dependency of retention errors and program interference errors.

4) We describe the underlying circuit and device-level mechanisms that result in the observed error patterns. Since the observed error patterns are due to fundamental circuit and device behavior inherent in flash memory we expect our observations and error patterns to also hold in flash memories beyond 30-nm technology.

II. Flash Memory

NAND flash memory can be of two types: single level cell (SLC) flash and multi-level cell (MLC) flash. Only one bit of information can be stored in an SLC flash cell, while multiple bits (e.g. 2-4 bits) can be stored in an MLC flash cell [6,7,8]. MLC flash represents n bits by using 2^n non-overlapping threshold voltage (V_{th}) windows. The threshold voltage of a given cell is mainly affected by the

978-3-9810801-8-6/DATE12/©2012 EDAA

number of electrons trapped on the floating gate. Figure 1 shows the bit mapping to V_{th} and the relative proportion of electrons on the floating gates of a 2-bit MLC flash.

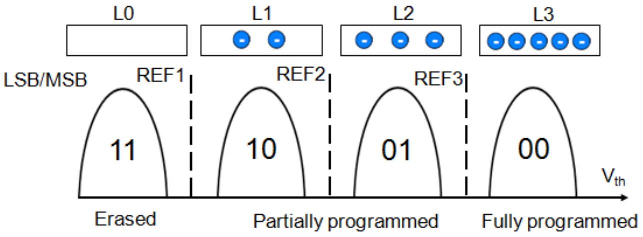

Figure 1. Threshold voltage distribution of 2-bit MLC flash

A NAND flash memory chip is composed of thousands of blocks. Each block is a storage array of floating gate transistors. A flash block usually has 32 to 64 wordlines. The cells on the same wordline can be divided into two groups: even and odd, depending on the physical location. For SLC flash, each group corresponds to just one logical page, i.e. even pages and odd pages. As MLC flash cell stores multiple bits, the bits corresponding to the same logical location of a cell in a group form one logical page. For example, all the most significant bits (MSB) of the cells of an even group form one MSB-even page. Similarly, other types of pages are MSB-odd page, LSB-even page and LSB-odd page. The page number assignments for each bit of the flash memory are shown in Figure 2(a), ranging from 0 to 127 for the selected flash in this paper. The size of each page is generally between 2kB and 8kB (i.e. 16k and 64k bitlines). The stack of flash cells in the bitline direction forms one string. The string is connected to a bit line through SGD (the select gate at the drain end) and connect to the common source diffusion through SGS (the select gate at the source end) as shown in Figure 2(b). Flash memories generally support three fundamental operations as follows:

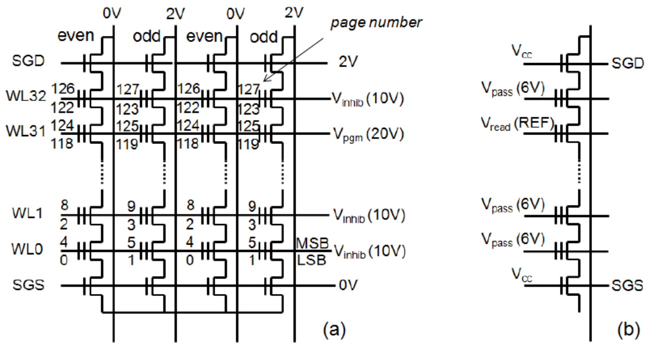

Figure 2. NAND flash organization and operations: (a) Partial block organization and program operation on page 118; (b) Read operation

Erase: During *erase* operation, a high positive erase voltage (e.g. 20V) is applied to the substrate of all the cells of the selected block and the electrons stored on the floating gate are tunnelled out through Fowler-Nordheim (FN) mechanisms [7]. After a successful erase operation, all charge on the floating gates is removed and all the cells are configured to L0 (11) state. Erase operation is at the granularity of one block.

Program: During *program* operation, a high positive voltage is applied to the wordline, where the page to be programmed is located. The other pages sharing the same wordline are inhibited (from being programmed) by applying 2V to their corresponding bitlines to close SGD and boost the potential of corresponding string channel. The voltage bias for programming page 118 is shown in Figure 2(a) as an example. The programming process is typically realized by incremental step pulse program (ISPP) algorithm [9]. ISPP first injects electrons into floating gates to boost the V_{th} of programmed cells through FN mechanisms and then performs a verification to check whether the V_{th} has reached the desired level. If V_{th} is still lower than the desired voltage, the program-and-verify iteration will continue until the cell's V_{th} has reached the target level. Note that the NAND flash program operation can only add electrons into the floating gate and cannot remove them from the gate. As a result, the threshold voltage can only shift toward the right in Figure 1 during programming. The program operation is executed at page granularity.

Read: The read operation is also at the page granularity and the voltage bias is shown in Figure 2(b). The SGD, SGS and all deselected wordlines are turned on. The wordline of selected read page is biased to a series of predefined reference voltages and the cell's threshold voltage can be determined to be between the most recent two read reference voltages when the cell conducts current.

III. NAND Flash Error Behavior Model

We test the NAND flash memory using the cycle-by-cycle programming model as shown in Figure 3. During each P/E cycle, the selected flash block is first erased. Then data are programmed into the block on a page granularity. Once a page has been programmed, it cannot be re-programmed again unless the whole block is erased for the next P/E cycle. The stored data will be alive in the block until it becomes invalid. Before the stored data becomes invalid, it can be accessed multiple times. The time interval between two accesses is variable depending on the access patterns of the applications. As a result, whether or not the data is retained correctly between two accesses depends on the time distance of two consecutive accesses. We repeat the above per-P/E-cycle procedure for thousands of cycles until the flash memory block becomes unreliable and comes to the end of its lifetime. Errors could happen in any stage of this testing process. We classify the observed errors into four different types from the controller's point of view:

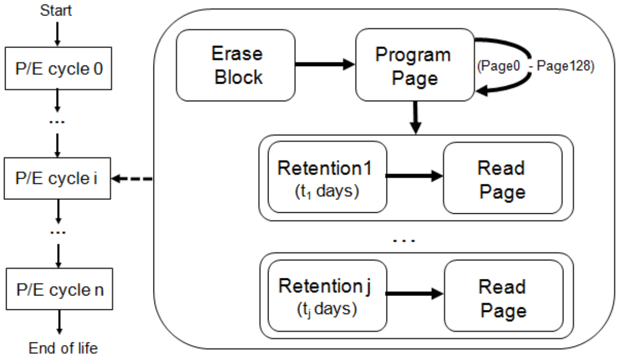

Figure 3. NAND flash behavior error modeling

- Erase error – happens when an erase operation fails to reset the cells to the erased state. This is mainly due to manufacturing process variations or defects caused by trapped electrons in the tunnel oxide after stress due to repeated P/E cycles.
- Program interference error – happens when the data stored in a page changes (unintentionally) while a neighbouring page is being programmed due to parasitic capacitance-coupling.
- Retention error – happens when the data stored in a cell changes over time. The main reason is that the charge programmed in the floating gate may dissipate gradually through the leakage current.
- Read error – happens when the data stored in a cell changes as a neighboring cell on the same string is read over and over.

IV. Testing Methodology

A. Experimental Hardware

To characterize the error patterns, we built a hardware test platform that allows us to issue commands to raw flash chips without ECC. The test platform mainly consists of three components: HAPS-52 board with Xilinx Virtex-5 FPGAs used as NAND flash controller, a USB daughter board used to connect to the host machine, and a custom flash daughter board. The flash memory under test is 2-bit MLC NAND flash device manufactured in 3x-nm technology. The device is specified to survive 3000 P/E cycles stress under 10-year data retention time if ECC with 4-bit error correction per 512 bits is applied. Details of the experimental flash test platform we use to collect our data are provided in [15].

B. Flash Error Testing Procedure

To test the P/E-cycle-dependence of errors, we stress-cycle flash memory blocks up to a certain number of erase cycles and check if the data is retained. This is achieved by iteratively erasing a block and programming pseudo-random data to it at room temperature.

We test whether the data is retained after T amount of time, to characterize retention errors. T is called the *retention test time* and is varied in the range of 1 day, 3 days, 3 weeks, 3 months, 1 year, and 3 years. We consider T = {1 day, 3 days} to be short-term retention tests, while the remaining values of T as long-term retention tests. Short-term retention errors are characterized under room temperature. Long-term retention errors are characterized by baking the flash memory in the oven under 125℃. According to the classic temperature-activated Arrhenius law [10], the baking time at 125℃ corresponds to about 450 times of the lifetime at room temperature (e.g. 25℃).

To characterize (inter-page) program interference errors, data was first written into a given block page by page and read out immediately after each page is programmed. Each page that is read back is logged as *data before program interference*. While a page is being programmed, the data in the previously programmed pages in the same block could be interfered with (or, disturbed) by the programming of the current page. When the programming of the whole block finishes, the data stored in each page is read out and logged as *data after program interference*. The two recorded data checkpoints are then compared and, if they differ from each other, a program interference error is recorded.

Read errors are tested by continuously reading a given block and comparing the read data with the originally-stored data. Erase errors are tested by counting the number of 0 bits in a block after each erase operation.

V. Experimental Results

We provide our experimental measurements of the errors in the state-of-the-art 3x-nm MLC NAND flash memory we have tested using our infrastructure. NAND flash errors show strong correlation with the number of P/E cycles, location of the physical cells, and the data values programmed into the cells. The following subsections analyze detailed error properties and describe the causes of the observed phenomena.

A. Overall Error Comparison

The various types of NAND flash errors are shown in Figure 4. The x-axis shows the number of P/E cycles and the y-axis depicts the raw bit error rate. The flash error results are characterized from the beginning of flash's life until the region of >100x times of its specified lifetime (3000 P/E cycles for the chips we tested). We make several observations about error properties.

First, all types of errors are highly correlated with P/E cycles. At the beginning of the flash's lifetime, the error rate is relatively low and the raw bit error rate is below 10^{-4}, within the specified lifetime (3k cycles). As the P/E cycles increase, the error rate increases exponentially. The P/E cycle-dependence of errors can be explained by the deterioration of the tunnel oxide under cycling stress. During erase and program operations, the electric field strength across the tunnel oxide is very high (e.g., several million volts per centimeter). Such high electric field strength can lead to structural defects that trap electrons in the oxide layer. Over time, more and more defects accumulate and the insulation strength of the tunnel oxide degrades. As a result, charge can leak through the tunnel oxide and the threshold voltage of the cells can change more easily. This leads to more errors for all types of flash operations.

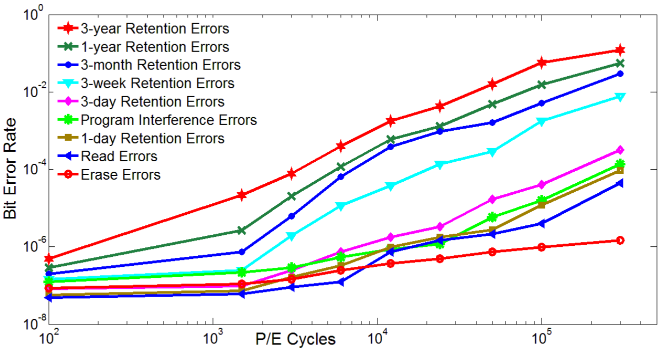

Figure 4. Rates of various types of errors as P/E cycles increase

Second, there is a significant error rate difference between various types of errors. The long-term retention errors are the most dominant; their rate is highest. The program interference error rate ranks the second and is usually between error rates of 1-day and 3-day retention errors. The read error rate is slightly less than 1-day retention error rate, while the erase error rate is only around 7% of the read error rate.

Third, retention error rates are highly dependent on retention test time. If the time before we test for retention errors is longer, the floating gate of flash memory is more likely to lose more electrons through leakage current. This eventually leads to V_{th} shift across V_{th} windows and causes errors. From our experimental data, we can see that the retention error rate increases linearly with the retention test time. For example, the 3-year retention error rate is almost three orders of magnitude higher than one-day retention.

B. Retention Error Analysis

Value dependence of retention errors: We find that the retention errors are value dependent; their frequency is asymmetric with respect to the value stored in the flash cell. Figure 5 demonstrates this asymmetric nature of retention errors by showing how often each possible value transition was observed due to an error. We characterized all possible error transitions, in the format AB→CD, where AB are the two bits stored in the cell before retention test, while CD are the two bits recorded in the cell after retention test. If the errors are not value dependent, the fraction of erroneous changes between each of the different value pairs should be equal. But, we find that this is not the case. The most common retention errors are 00→01, 01→10, 01→11 and 10→ 11, with their relative percentage over all retention errors being 46%, 44%, 5% and 2%, respectively. The relative percentages among various error transitions are almost constant for different P/E cycles.

To understand the reasons for value dependence, we need to observe Figure 1 in conjunction with the value transition observed in the most common retention errors. We find that the most common retention errors (00→01, 01→10, 01→11 and 10→ 11) are all cases in which V_{th} shifts towards the left (see Figure 1). This can be explained by an understanding of the retention error mechanisms. During retention test, the electrons stored on the floating gate gradually leak away under stress induced leakage current (SILC). When the floating gate loses electrons, its V_{th} shifts left from the state with more electrons to the state with fewer programmed electrons (as seen in

Figure 1, states to the left have fewer electrons trapped on the gate than states to the right). It is significantly less likely for the cells to shift right in the opposite direction because this requires the addition of more electrons. As the states of 00 and 01 hold the largest number of electrons on the floating gates, SILC is higher in these states and therefore it is more likely for the V_{th} of the cells in these two states to shift left, which leads to the observation that most common errors are due to shifting from these states (00→01, 01→10, 01→11).

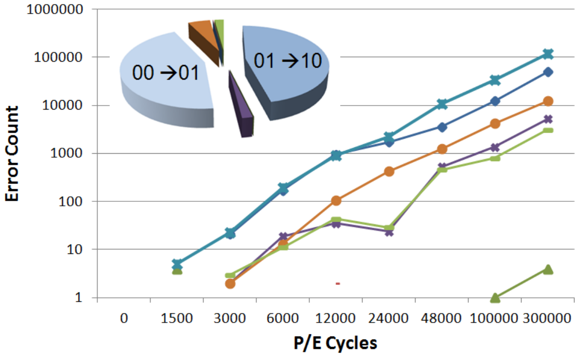

Figure 5. Value dependence of retention errors

We also make two other observations based on our evaluations. First, note that a cell in state 11 cannot shift to another state by losing electrons because there is no other state to the left of it. Consistent with this observation, our experimental results show that there is no retention error for cells in 11 state. Second, our test results in Figure 5 show that the probability of the V_{th} of a cell shifting left by more than one state is rather low, as it would take a long time for the cell to lose enough electrons to jump two states. The only exception is the 01→11 error transition. We hypothesize that this is because a relatively narrow voltage threshold window for intermediate states 01 and 10 in the flash memory we tested.

Location dependence of retention errors: We also characterized the relation between retention errors and their physical locations. The experimental results are shown in Figure 6. The x-axis shows the wordline number of a block and the y-axis shows the bit error rates of pages on the corresponding wordline (observed after 50k P/E cycles). Each wordline contains 4 pages, including LSB-even, LSB-odd, MSB-even and MSB-odd. The bit error rates of these four types of pages are shown in Figure 6. Several major observations are in order.

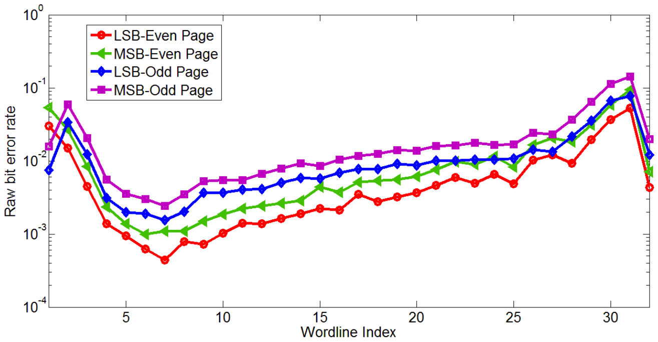

Figure 6. Retention error rate vs. physical location

First, the error rate of the MSB page is higher than that of the corresponding LSB page. In our experimental data, the MSB-even page error rate is 1.88 times higher than the LSB-even page error rate and the MSB-odd page error rate is 1.67 times higher than the LSB-odd page error rate on average. This phenomenon can be explained by understanding the bit mapping within the flash memory. Dominant retention errors are mainly due to the shifting of V_{th} between two adjacent threshold voltage levels, that is shifting of V_{th} from the i-th level to the (i-1)-th level. From the bit mapping in Figure 1, we can see that such a V_{th} shift can cause an LSB error only at the border REF2 between state L2 (01) and state L1 (10) because these are the only two adjacent threshold voltage levels where LSB differs. On the other hand, such a V_{th} shift can cause an MSB error on any border (REF1, REF2, REF3) between any two adjacent states because MSB differs between all possible adjacent threshold voltage levels. Hence, since the likelihood of a change in MSB when a V_{th} shift happens between adjacent states is higher than the likelihood of a change in LSB, it is more common to see retention errors in MSB rather than LSB.

Second, the retention error rate of odd pages is always higher than that of the corresponding even pages. For example, error rate of MSB-odd pages is 2.4 times higher than that of MSB-even pages, and the error rate of LSB-odd pages is 1.61 times higher than that of LSB-even pages, on average. This result can be explained by the over-programming introduced by inter-page interference. Generally, the pages inside a flash block are programmed sequentially, and a block is programmed in the order from page 0 to page 127. For the same wordline, even pages are programmed first followed by odd pages. When odd pages are programmed, a high positive program voltage is applied to the control gates of all the cells on the wordline, including the cells of the even page, which has already been programmed. Thus, the even page comes under programming current disturbance and some additional electrons could be attracted into the floating gates of the even page. As a result of this, the V_{th} of cells of the even pages will slightly shift to the right. Consequently, the cells of the even pages will hold more electrons than the cells of the odd pages, even if they are programmed to the same logic value and are in the same threshold voltage window (in some sense, the cells of the even pages are thus more resistant to leakage because they hold more electrons). When electrons leak away over time during the retention test, as a result, it is more likely for the cells of even pages to still keep their original threshold voltage window and hold the correct value. In contrast, since the cells of the odd pages hold fewer electrons, they are more likely to transition to a different threshold voltage window and hence acquire an incorrect value as electrons leak over time.

Third, the bit error rates of all the four types of pages have the same trend related to physical wordlines. For example, the error rates of the four types of pages are all high on wordline #31 and are all low on wordline #7. We conclude that error rates are correlated with wordline locations. This could possibly be due to process variation effects, which could be similar across the same wordline.

The major takeaway from our results is that, the rate of retention errors, which are the most common form of flash errors, is asymmetric in both original cell value and the location of the cell in flash bit organization. This observation can potentially be used to devise error protection or correction mechanisms that have varying strength based on cell value and location.

C. *Program Interference Error Analysis*

Value dependence of program interference errors: Figure 7 shows the frequency of occurrence and P/E cycle dependence of different program interference errors. The most dominant programming interference errors are 11→10 and 10→01. Their relative percentages are 70% and 24% respectively. Less common errors are 10→00, 11→01, and 01→ 00. Their relative percentages are 2.2%, 1.5% and 0.4% respectively. Similar to retention errors, program interference errors also show strong asymmetry with respect to the cell value, but they occur in the opposite direction with regard to the V_{th} shift. The cell states mainly shift from the states with fewer programmed electrons to the states with more electrons (i.e., from left to right in Figure 1).

This phenomenon can be explained by examining how resistant a cell that holds a particular value is to the programming voltage applied to it. When a page is being programmed, a high positive programming voltage is applied to *all* the control gates on the selected wordline, including those of the cells of the other pages that share the same word-line but that are not supposed to be programmed. This high positive voltage could attract additional electrons into the floating gates of these other pages through tunneling even though such pages may have already been programmed. If there are too many electrons attracted to these gates, the V_{th} of disturbed cells will shift towards the right into the higher threshold window. While programming voltage generates an intrinsic electric field in the direction from the control gate down to the channel (thereby causing potential disturbance in the value stored in the cell), the electrons on the programmed floating gates generate an electric field in the opposite direction from the channel up to the floating gates. The electric field generated by the electrons can partly counteract the electric field generated by the disturbing programming voltage to reduce the effective electric field across the tunnel oxide. If the cells are already programmed with more electrons, the effective electric field across the tunnel oxide is further reduced and it is less possible for the high programming voltage to tunnel the electrons into the floating gates. As a result, it is less likely for the programming voltage to disturb a cell that already holds many electrons but easier for it to disturb a cell that holds few electrons. This explanation is supported by our experimental data. We can see that the error rate of 11→10 is higher than that of 10→01, and the error rate of 10→01 is higher than that of 01→ 00 as the cells in state 11 hold fewer electrons than cells in 10 (and hence it is easier to disturb the state of the cells in state 11 than that of cells in state 10), and cells in state 10 hold fewer electrons than cells in 01 state (ditto for state 10 and state 01).

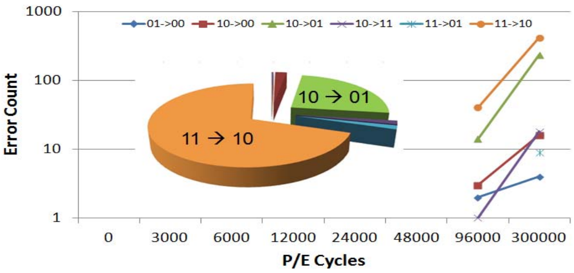

Figure 7. Value dependence of program interference errors

Note that we did not observe any program interference errors in cells that are in state 00. Even if additional electrons are injected into the cells in state 00 and the V_{th} may shift right under programming voltage interference, the cell will hold its value because its voltage threshold window will still stay the same (i.e., there is no other state to the right of 00; any voltage value that is greater than the reference voltage REF2 in Figure 1 still causes the state of the cell to be 00). Finally, the program error rates introduced by more than one V_{th} level shift to the right are relatively low as it is more difficult to attract enough electrons for V_{th} to shift right multiple threshold windows.

Location dependence of program interference errors: The program interference error distribution inside one flash block is shown in Figure 8. The x-axis is the logical page number inside a block and the y-axis shows the error rate of each page that shows errors. We find that the program interference error rate is dependent on the physical location of the cell in two major ways.

First, the error rate of even pages is higher than that of the odd pages. This measurement is the opposite of what we observed for retention errors (recall that the retention error rate of an even page was much lower than that of an odd page). This is due to the programming order of pages inside a block in flash memory: even pages are programmed earlier than odd pages. The previously programmed even page can be disturbed by later programmed odd pages, while the odd pages are free of interference from programming even pages when they share the same wordline. As a result, program interference error rate is higher in even pages than in odd pages.

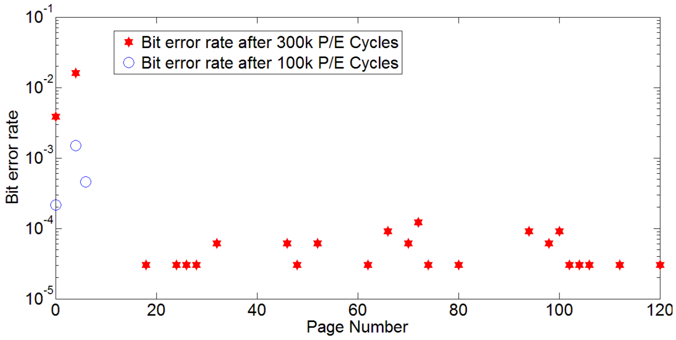

Figure 8. Program interference error rate distribution inside one block

Second, the error rates of *bottom pages*, the lower-numbered pages located on wordlines close to the bottom of a flash block, e.g. WL0 in Figure 2, are much higher than that of the other pages. As we can see in Figure 8, the error rates of bottom pages are around 10^{-2}, while the error rates of the other pages are below 10^{-4}: error rates of bottom pages are almost two orders of magnitude higher. The high error rate of bottom pages can be explained by boosting hot-carrier injection (HCI) noise [11, 12]. During program operation, 0V and Vdd are applied to control gates of SGS and SGD respectively, as shown in Figure 2. The high program voltage V_{pgm} (e.g., 20V) is applied to the selected wordline and V_{pass} (e.g., 10V) is applied to the remaining wordlines. The cells on the selected wordline but not on the selected page are inhibited from programming by biasing the corresponding bitline to V_{dd}. Let's focus on the effect of programming interference on one of these cells that are inhibited from programming. The (vertical) string that contains one of the inhibited cells (WLn) is shown in Figure 9. Both SGS and SGD are closed, and the channel voltage of the inhibited string is boosted under the coupling of high Vpgm and Vpass on the control gates. Thus, a sufficient voltage difference appears between the SGS and WL0 and a high transverse electric field is generated. Such a high electric field creates electron-hole pairs. As a result, electrons are accelerated between SGS and WL0 and the likelihood of them being injected into the floating gates of WL0 increases. This injection of electrons into the floating gates of WL0 can change the threshold voltage of the cells on WL0 as the accelerated electrons have relatively high energy. Consequently, the bottom page (WL0) will incur errors. In fact, we found that programming errors that happen by jumping two threshold voltage windows are mainly found in WL0, indicating that the amount of hot electrons injected into the floating gate of cells on WL0 is very high.

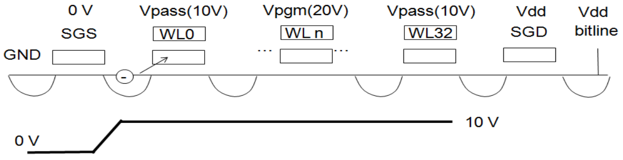

Figure 9. Hot-carrier injection noise mechanism

Note that this HCI noise based error mechanism affects the bottom page (WL0) regardless of which other page is being programmed as long as programming causes bottom pages on WL0 to be in inhibited state. The fundamental reason for the error is the high voltage difference between SGS and WL0 that appears when WL0 is inhibited.

D. *Read Error Analysis*

We test all possible error transitions for read errors and show their relative percentages over total read errors in Figure 10. The 2-bit flash cells can be programmed into four possible values (00, 01, 10 and 11), and the total read error rates are approximately close among the cells with these four programmed values. We can also see that the read errors happen mainly due to threshold voltage shifting to the adjacent threshold voltage window. For example, the cells programmed with value 10 tend to be misread as 11 and 01, while they are less likely to be misread as 00. In the edge states 11 and 00, read errors can only happen by V_{th} shifting right and left, respectively, so we see some error transitions that jump two V_{th} windows, e.g. 11→01. For the middle states 10 and 01, V_{th} tends to shift right with slightly higher probability, that is in the direction from the states with fewer electrons to the states with more electrons (e.g., 10→01 error rate is slightly higher compared to 10→11). We conclude that read errors also demonstrate value dependence, but their overall rate is relatively low compared to retention and program interference errors.

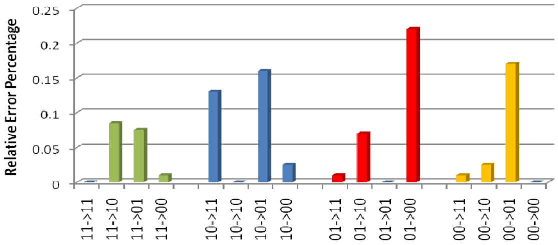

Figure 10. Value dependence of read errors

E. *Erase Error Analysis*

Erase errors are the least significant among NAND flash errors and they occur often only after millions of P/E cycles, which is more than >100x times the specified lifetime of the flash memory we test. Figure 11 shows one interesting finding on erase errors: the erase error rate is correlated to the number of consecutive erases before programming. This figure displays the erase error rate (observed after three million P/E cycles) for two cases: 1) if we erase the block only once before programming, 2) if we erase the block N times before programming, where N is the value in the x-axis of Figure 11. If we erase the block only once before each programming, the tested erase error rate is relatively constant. However, if we continue to erase a certain block without programming, the erase error rate decreases exponentially. We can see that even 10 consecutive erases before programming can decrease the erase error rate by more than 95%. The reason for this is simple: erase errors are mainly due to failure to remove electrons from the floating gate; performing multiple erase operations consecutively is equivalent to applying a negative voltage to the floating gate for a longer time, which increases the likelihood that the electrons on the floating gates are removed.

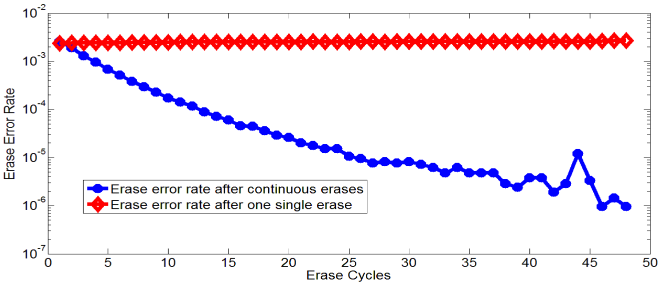

Figure 11. Erase error rate vs. erase count (once vs. multiple consecutive)

VI. RELATED WORK

The closest work to ours is very recent work that published limited flash error characterization results for 72nm [13], 50-70nm [14] and 30-50nm [16] flash technologies. [13] characterizes the error results for 72nm flash memory. [14] and [16] show raw error data without providing detailed error type breakdowns and analyses. Our work advances the state of the art beyond these three works by 1) demonstrating the dominance of retention errors, 2) demonstrating error location dependency in flash memory, and 3) providing detailed circuit-level and device-level explanations for the cause of errors.

VII. CONCLUSIONS & FUTURE WORK

We have studied and provided a detailed characterization of errors in state-of-the-art 3x-nm NAND flash memory, with the goal of understanding and explaining prevalent error patterns. We make several major observations and conclusions based on our empirical analyses, which can hopefully aid the design of efficient error tolerance mechanisms in flash memory controllers. First, we find that retention errors are the most dominant errors and they should therefore be given the first priority in the design of error tolerance/correction mechanisms. Second, flash error rates increase super-linearly with program/erase cycles for all types of bit errors, so stronger error correction codes will likely have diminishing benefits in improving flash memory lifetime. Third, retention errors and program interference errors are asymmetric in nature: error rates are strongly dependent on the value and the location of the cells. We analyze the causes of the errors' value and location dependence and find that they are due to fundamental device and circuit mechanisms in flash memory. Hence, we expect the observed error patterns to affect future sub-30-nm flash memories.

We hope that the characterization and understanding of modern flash memory error patterns developed in this paper will serve as an enabler for new, more effective, and more efficient error tolerance mechanisms for flash memory. Two initial ideas we are currently exploring are value-asymmetry aware coding techniques and cell-location-aware wear leveling mechanisms that respectively exploit the value-dependence and location-dependence of flash memory errors.

ACKNOWLEDGEMENTS

We gratefully acknowledge the support of LSI Corporation and the Data Storage Systems Center at Carnegie Mellon.

REFERENCES

[1] R. Bez, et al, "Introduction to Flash memory", Proceedings of the IEEE, 2006.

[2] Y. Koh, "NAND Flash Scaling Beyond 20nm", IMW, pp. 1-3, 2009.

[3] S. Tanakamaru, et al, "Post-manufacturing, 17-times acceptable raw bit error rate enhancement, dynamic codeword transition ECC scheme for highly reliable solid-state drives, SSDs", IMW, pp. 1-4, 2010.

[4] E. Gal, et al, "Algorithms and data structures for flash memories", ACM Comput. Surv., vol. 37, pp. 138-163, 2005.

[5] S. Gregori, et al, "On-chip error correcting techniques for new-generation flash memories", Proceedings of the IEEE, vol. 91, pp. 602-616, 2003.

[6] T. Hara, et al, "A 146-mm2 8-Gb multi-level NAND flash memory with 70-nm CMOS technology", JSSC, vol. 41, pp. 161-169, 2006.

[7] Y. Li, et al, "A 16Gb 3-Bit Per Cell(X3) NAND Flash Memory on 56nm Technology With 8MB/s Write Rate", JSSC, vol. 44, pp. 195-207, 2009.

[8] N. Shibata, et al, "A 70nm 16Gb 16-Level-Cell NAND flash Memory", JSSC, vol. 43, pp. 929-937, 2008.

[9] K. Suh, et al, "A 3.3V 32Mb NAND Flash Memory with Incremental Step Pulse Program Scheme", JSSC, vol. 30, no.11, pp. 1149-1156, 1995.

[10] M. Xu, et al, "Extended Arrhenius law of time-to-breakdown of ultrathin gate oxides", Applied Physics Letters, vol. 82, pp. 2482-2484, 2003.

[11] M. Park, et al, "NAND Flash reliability degradation induced by HCI in boosted channel potential", IEEE IRPS, pp. 975-976, 2010.

[12] H. Wang, et al, "A New Read-Disturb Failure Mechanism Caused by Boosting Hot-Carrier Injection Effect in MLC NAND Flash Memory", IMW, pp. 1-2, 2009.

[13] N. Mielke, et al, "Bit Error Rate in NAND Flash Memories", IRPS, pp. 9-19, 2008.

[14] L.M. Grupp, et al, "Characterizing flash memory: Anomalies, observations, and applications", MICRO, pp. 24-33, 2009.

[15] Y. Cai, et al, "FPGA-Based Solid-State Drive Prototyping Platform", FCCM, pp. 101-104, 2011.

[16] H. Sun, et al, "Quantifying Reliability of Solid-State Storage from Multiple Aspects", SNAPI, 2011