Improving NAND Flash Read Performance Through Learning

Haleh Tabrizi*, Borja Peleato†, Rajiv Agarwal‡, Jeff Ferreira*
*EMC / DSSD, Inc
{haleh, jpf}@dssd.com
†Electrical and Computer Engineering, Purdue University
peleato@purdue.edu
‡Electrical Engineering, Stanford University
rajivag@stanford.edu

***Abstract*—Two important performance metrics for a storage system are the latency associated with retrieving data from its storage medium and the effective lifetime of its storage medium. Both metrics are directly affected by the number of raw read errors (i.e. errors prior to exploiting error-correction mechanisms). In this paper, we focus on NAND flash memories, where a read is performed by comparing stored voltages with a threshold voltage. The unwanted variation of stored voltages causes read errors. We identify the number of flash program-erase (PE) cycles, time elapsed between writing and reading, and the page number (physical location) as the main sources of voltage variations. We propose a method for learning how read thresholds should vary with these parameters such that the storage controller can dynamically vary thresholds and minimize read errors. Lab experiments show that at the flash end-of-life, the proposed method lowers the raw bit-error-rate by an order of magnitude, as compared to manufacturer's default read settings.**

I. Introduction

Read time or the latency associated with retrieving data is an important performance metric for a storage system. The read latency for a storage system can be decreased if it is able to reliably retrieve error-free data from the storage medium without exploiting error-correction mechanisms. For example, the storage system may use mechanisms such as error correcting codes (ECC) and/or RAID to remove read errors, which results in an increase in read latency and decrease in performance. Furthermore, a storage medium is functional only until its error-correction mechanisms are fully capable of retrieving all user data correctly, which sets a limit on the maximum number of raw read errors. Hence lowering the number of raw read errors translates into a longer storage medium lifetime, too.

For several decades, CPUs have doubled their speed every two years in what is commonly known as Moore's law. Unfortunately, traditional spinning disk storage has not been able to keep up with this trend. Current computers are not limited by the speed at which information can be processed, but rather by the speed at which information can be read and written. This situation, as well as the recent information explosion, has created a huge market for storage applications.

NAND flash memories are faster and more power efficient than hard drives but their higher cost and limited endurance has traditionally been an obstacle for their widespread use. Manufacturers have spent the last decade, if not more, addressing this problem by aggressively scaling the technology and packing more bits in each flash cell. Their efforts have succeeded in reducing the cost per Gigabyte, but have brought other problems related to the memories' endurance and the reliability of the stored information. Most of the memories which are available today use 20 nm cells, with two bits stored in each cell, but can typically only sustain around 10,000 PE-cycles before the bit error rate (BER) becomes too large for the error correcting code (ECC) to correct.

A flash cell is a floating gate transistor whose threshold voltage can be adjusted by injecting charges into its floating gate. Information is stored by setting this voltage threshold to specific values. In its simplest form, one bit is stored in each cell, depending on whether it is charged or discharged. Memories of this type are known as single level cells (SLC). In order to increase the capacity (and reduce their cost accordingly) most applications now use multi-level cells (MLC) memories, which can be programmed to four different voltage levels and store two bits in each cell. Some manufacturers have gone even further, producing memories which store three (TLC) or even four bits in each cell [1].

Cells are grouped into pages, and pages are grouped into blocks. Blocks are the smallest unit of erase, while pages are the smallest unit for reads and writes. Pages are read by comparing the voltage threshold of all the cells within the page with a common reference voltage. This reference voltage is known as read threshold. An MLC flash memory, where cells can take four different voltage levels to store two bits, employs three different thresholds to separate the levels. This is further discussed in Section III.

Traditionally, these reference thresholds were pre-fixed by the manufacturer and kept hidden from the flash controller, but the need for improved noise canceling techniques has pushed some manufacturers to make available an interface for changing the thresholds. Most flash controllers are now able to shift the read thresholds up and down to account for write noise and leakage, respectively. Furthermore, it is possible to apply very powerful signal processing techniques to recover the information reliably [1]. However, such methods can degrade the fast read advantage of flash due to excessive post-read signal processing.

When the memory is reused to store new data, old data needs to be erased before new data is programmed, which contributes to the memory PE-cycle count. Time elapsed between program and read, also known as retention, number of program/erase (PE) cycles and memory location such as page number, (among many other factors) affect the optimum read threshold voltage. The objective of this paper is to present a method for learning how optimal read-thresholds vary with system parameters. Based on the learned values, the threshold voltages can be selected optimally from a pre-computed table of read threshold values at read-time to minimize read errors. Through multiple lab experiments on NAND flash, we obtain the optimum threshold values for several combinations of PE-cycle, retention time and page number through a brute-force method. The end result is a look-up table (or a linear math function) that allows the storage controller to identify the optimum read threshold voltage in a fast manner.

The paper is organized as follows: Section II describes some related work in the literature. Section III introduces our system setup and presents the notation used throughout the rest of the paper. The method used to collect data and obtain a table of read thresholds is discussed in Section IV. The look-up table is further compressed into a mathematical model in Section V. The BER performance gain resulting from the look-up table and the mathematical model is evaluated in Section VI. Section VII summarizes the paper.

II. Prior work

Since NAND flash manufacturers started giving the academic community access to the read threshold shift commands, there have been several papers analyzing how this new tool could be used to improve the performance of the memories. The concept of dynamic read thresholds appeared in the literature before the commands were accessible by most researchers. For example, Zhou, Jiang, and Bruck introduced in [2] error correcting codes specifically tailored to dynamic thresholds and we proposed a method for finding the optimal read thresholds in [3]. Kim et al [4] studied the data patterns that are most prone to create errors and proposed employing modulation codes to avoid those patterns once the memory was worn out. However, none of those works presented results for real data.

Once the commands were available to selected researchers, some publications started presenting detailed analysis of how the different sources of noise affected the voltages stored in the cells and consequently the read voltages that should be used to read them. Cai, Mutlu, Haratsch, and Mai performed some very detailed analysis in [5] and[6]. They showed detailed plots with the threshold distributions, as well as their evolution with the number of PE cycles and the voltages programmed in adjacent cells. They also tried fitting multiple statistical models to those thresholds, to finally conclude that the Gaussian model provided the best fit. They did not, however, address the problem of finding the read threshold that minimizes BER. They did not look at retention or page-to-page variation either.

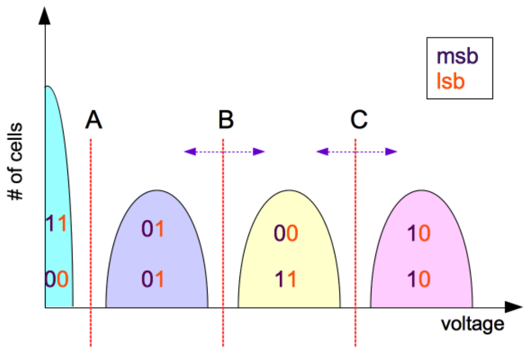

Fig. 1. MLC NAND read thresholds.

Authors in [7] characterizes bit-error behavior and proposed an error-correction mechanism that outperforms BCH codes. In this paper, we restrict our study to raw BER minimization that aides any and all error correction methods.

III. System Model

This paper assumes a (2-bit) MLC flash, where each cell stores one of 4 distinct predefined voltage levels. However, all the methods and results can be readily extended to SLC and any multi-level cell (TLC, QLC) case. In an MLC flash memory, each cell can take any one of four different voltage levels, hence requiring three thresholds to separate them. These thresholds are denoted by A, B, and C, as depicted in Figure 1. Two bits are mapped to those levels using Gray coding, which reduces BER by ensuring that errors between adjacent levels only affect one bit. Two such bit-mappings is represented in Figure 1. Furthermore, each of the two bits is assigned to a different page. This reduces the number of comparisons required to read a page. For example, consider the top bit-mapping in Figure 1: $11, 01, 00, 10$. LSB pages are read by comparing the cell voltages with the B threshold. MSB pages are read by comparing the cell voltages with the A and C thresholds simultaneously. Experiments have shown that the relative performance gain obtained from changing A threshold is small. Hence to reduce complexity, only thresholds B and C are optimized in this paper.

The endurance of a flash memory device is generally measured by the number of times the flash is programmed and erased. Each program and erase cycle causes a small physical damage to the flash medium. This damage accumulates over time and usually represents itself through cell voltage leakage until eventually Flash memory cells become incapable of storing any voltage or data. The number of PE cycles that a given device can sustain before problems become prohibitive or the bit error rate is higher than the underlying error-correction mechanisms can resolve, varies with the type of technology.

Retention time corresponds to the time that has elapsed between the writing of the data to a physical location on a storage medium and the time that the data is being read from that physical location. When a significant amount of time passes between the writing and reading of a page, cells suffer voltage leakage. This effect is more pronounced for the higher levels. For example, cells storing '10' as depicted in Figure 1.

Cells in a NAND flash are organized in terms of pages and programming generally occurs in some order according to the page number, which depends on the memory manufacturer. The order of page programming is chosen by the manufacturer to minimize write noise effects. During page programming there is some perturbance in cell voltages commonly known as write noise and there exists inter-cell interference (ICI) [8]. As the technology scales and more cells are packed in the same area, the interaction between them increases. Parasitic capacitances can therefore cause some cells to increase their voltage level. The page number can then be a good indicator of the amount of cell voltage variations.

Reading pages in an MLC flash is performed by comparing each cell's stored voltage with threshold voltages A, B, and C. Comparison of the stored voltage with the B threshold returns a 0 for corresponding LSB page if the stored voltage is less than B, and returns 1 if the voltage is greater than B. Similarly, if the stored voltage is less than C and greater than A threshold, the read operation returns a 0 for the corresponding MSB page and vice-versa. However, the aforementioned sources of voltage disturbance can cause the cells to be misclassified, making the reads noisy. The choice of a read threshold therefore becomes important to minimize the BER in reads.

The default read threshold value for A, B, and C is specified by the manufacturer of the solid-state memory modules. Further, the granularity of the shift values can be specified by the voltage shift from a corresponding default read threshold value or it could be an absolute voltage value and depends on the manufacturer. We discretize each of the above-mentioned sources of cell voltage variation into a small number of intervals and empirically compute the threshold that provides the minimum bit error rate (BER) in each case. These optimal thresholds can then be stored in a table that the controller can use when reading each page.

IV. Generating Read Threshold Table

The operating conditions of flash memory in a real scenario can be significantly different from those simulated in the lab. The speed and temperature at which the blocks are cycled, the method used to compensate for charge trapping, and the different delays between each step in the experiment can influence the end results. Hence, the temperature and cycling rate that corresponds to a customer-use scenario must be identified and using the Arrhenius model the required temperature and cycling rate in the lab environment can be obtained [9].

Given the four flash system parameters, retention $t \in \mathbb{T}$, PE-cycle count $p \in \mathbb{P}$, page number $n \in \mathbb{N}$, and page type $s \in \{LSB, MSB\}$, the objective is to obtain optimum read threshold voltages. To compensate for uncontrolled lab artifacts, multiple experiments with the objective of obtaining BER values for the same combinations of system parameters are performed. Each experiment consists of performing erase, write, and read operations on $B = 1000$'s of blocks. To obtain results for $P = |\mathbb{P}|$ different PE-cycle points, we divide the blocks into P groups and repeat the process of writing pseudo-random data and erasing them to obtain P different set of PE-cycled blocks. Given the operating NAND temperature and using the Arrhenius model, we obtain the time duration and NAND bake temperature that model the effect of $T = |\mathbb{T}|$ discrete retention points. The NAND flash is then baked for a duration that accelerates the time corresponding to each t point. Each block consists of $N \in |\mathbb{N}|$ LSB pages, and N MSB pages.

To obtain optimum read threshold values, each page is read K times using the flash module's available K different read voltage settings for B and C thresholds. For each LSB and MSB page, respectively, the B and C thresholds are shifted for every $k \in \{1, ..., K\}$ values. Other than the K shifted reads, each page is read using the NAND manufacturer's default settings. The default setting is one of the K values chosen by the manufacturer, but unknown to the user. The raw BER (RBER) is then calculated by comparing the read data with the data written on the page and counting the number of bits in error.

Each experiment e operates over a set of blocks $b \in \mathbb{B}_e$, such that $\bigcup_{e\in\{1,2,...E\}} \mathbb{B}_e = \mathbb{B}$. With E sets of such experiments, for every combination of (t, p, n, s) BER values r are generated. The BER data collected can be grouped into three sets:

- shifted read: BERs indexed by thresholds 1 to K corresponding to reads that are used for generating the read-threshold table, $r_{\text{shift}}(t, p, n, s, b, k)$.
- default read: BER corresponding to default controller reads, $r_{\text{def}}(t, p, n, s, b)$
- optimum read: the minimum BER among all shifted (1 to K) reads per block b, $r_{\text{opt}}(t, p, n, s, b) = r_{\text{shift}}(t, p, n, s, b, k_{\text{opt}})$, where

$$k_{\text{opt}}(t, p, n, s, b) = \arg\min_{k} r_{\text{shift}}(t, p, n, s, b, k) \qquad (1)$$

Once all the BER data has been collected, it is split into two equal sets: one for training the model (i.e. generating the table) and the other for later validation of the results. We use half the data (blocks) from each of the E experiments as training data and the other half as validation data ($\mathbb{B}_{tr}$ and $\mathbb{B}_v$, respectively, such that $\mathbb{B}_{tr} \cup \mathbb{B}_v = \mathbb{B}$). This method dilutes the effect of independent experiment artifacts. The objective is to show that the validation set performs similar to training set when using the table generated by the training set. The training and validation processes are discussed in the following subsections.

A. Training

Given the 5-dimensional system-based data (t, p, n, s, b, k) and the corresponding BER values $r_{\text{shift}}(t, p, n, s, b, k)$, the threshold value k^* that minimizes BER over all training set of blocks can be obtained as follows:

$$k^*(t,p,n,s) = \arg\min_k \sum_{b \in \mathbb{B}_{tr}} r_{\text{shift}}(t,p,n,s,b,k). \quad (2)$$

For each 5-tuple (t,p,n,s,b), using training data, we find the threshold value k^* that minimizes the sum of BER over all training blocks. A four-dimensional table of read thresholds as a function of (t,p,n,s) is then generated. We evaluate per block BER values based on the generated table and compare to default BER r_{def} and optimum read BERs r_{opt} on the training sets. Note that k_{opt} corresponds to the threshold value that minimizes BER for each block, while k^* corresponds to the threshold value that minimizes *sum* of BER over all training blocks.

At this point some experimental anomalies might be encountered, which need to be accounted for. For example, we might find that in some cases, the average table-based BER is higher than average default BER. The table-based read, by definition, should perform better than default read on average. This scenario could be representative of hitting process corners (discussed in more detail in Section VI-C) or inconsistent experimental data, which implies acquiring more data to average out the independent experimental artifacts.

B. Verification

Using the table generated in training, $k^*(t,p,n,s)$, the corresponding BER values of the verification dataset are obtained:

$$r_{\text{verif}} = r_{\text{shift}}(t,p,n,s,b,k^*(t,p,n,s)), \ \forall b \in \mathbb{B}_v. \quad (3)$$

and the resulting BERs are compared with the verification set's default and optimum read BERs.

Again, if the table-based average BER values is higher than average default read BER values, we can either: (a) run more experiments and collect more data, (b) modify the amount of data in training and verification sets, or (c) use combination of default and table-based read (as will be discussed in Section VI-C).

V. Mathematical Model

The table obtained in Section IV requires a large number of entries, which is equal to the product of the number of retention points (T) , PE-cycle points (P), and page numbers for LSB and MSB $(2N)$. If there is not enough memory at the flash controller to store such table, a more compressed version in the form of a mathematical model can be obtained. Such mathematical model also allows us to better visualize how threshold voltages vary with system parameters (t,p,n,s). This method although less memory-intensive than a look-up table, suffers from some computational complexity to generate the threshold values.

Note that all axes values in the following Figures 2, 3, and 4 are relative. Actual values have been obscured for reasons of agreements with the NAND manufacturers. In any case the purpose of this paper is to present a novel table-based method, rather than discuss specific performance results for a given NAND manufacturer.

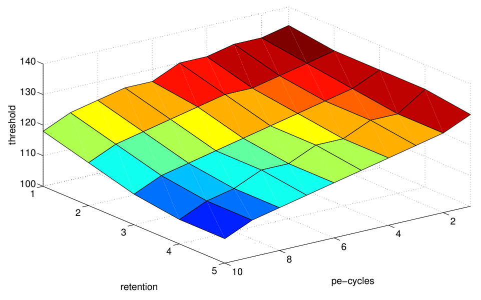

Fig. 2. threshold variation across retention and PE-cycle for an LSB page.

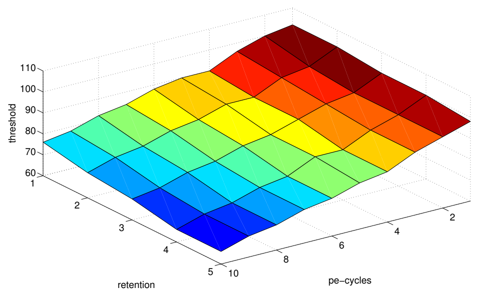

Fig. 3. threshold variation across retention and PE-cycle for an MSB page.

To obtain such mathematical model, we first investigate how thresholds vary as a function of (t,p,n,s). Figure 2 represents the table-based threshold values (k^*) as a function of PE-cycle and retention time for a specific LSB page. As expected, the k^* decreases with retention time and PE-cycle, which is a result of cell voltage leakage in time (retention) and leakage due to cell damage (increasing PE-cycles). The table-based threshold voltages decrease almost linearly with both PE-cycle and retention time and hence, we can use a linear model to represent threshold variations as a function of p and t. A similar threshold trend is observed for s = MSB pages, as shown in Figure 3. The main difference among the two is a shift in the range of threshold values.

Figure 4 represents table-based threshold values (k^*) as a function of page number n and PE-cycle p for LSB pages. Again, we observe the decrease in threshold voltage with increasing PE-cycles. Further, for a given PE-cycle value, the threshold voltage increases with decreasing page number. This increase in threshold voltage value can be explained according to the order of page programming. When pages are programmed in order from 0 to 127, the inter-cell interference causes the lower numbered pages to accumulate more charge and hence require a larger threshold voltage as compared to higher numbered pages. Similar threshold trend is also observed across MSB pages. The threshold variation across page numbers can also be modeled as a linear function, but with a relatively smaller slope as compared to retention and PE-cycle dependencies.

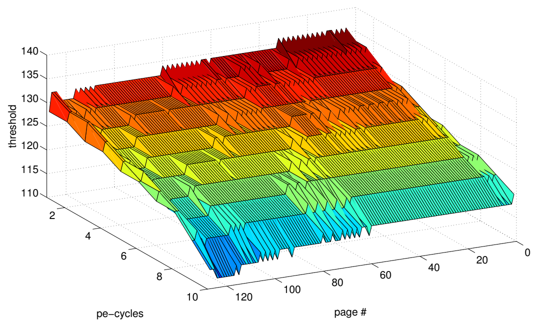

Fig. 4. threshold variation across page number and PE-cycle.

Given a set of 4-dimensional data points $X = (t, p, n, s)$ and the corresponding table-based threshold values $k^*(X)$, we can generate a linear function $f_\theta : \mathbb{R}^4 \to \mathbb{R}$ that best fits the table mapping $X \to K^*(X)$. The function f_θ is defined as:

$$f_\theta(t, p, n, s) = \theta_0 + \theta_1 t + \theta_2 p + \theta_3 n + \theta_4 s. \tag{4}$$

The parameter $\theta \in \mathbb{R}^5$ is obtained by minimizing the linear least squares error between k^* and f_θ:

$$\theta = \arg\min ||f_\theta(X) - k^*(X)||^2, \tag{5}$$

where $||.||$ denotes the Euclidean norm. Solving for θ given X and $K^*(X)$, we obtain the θ values shown in first row of Table I.

TABLE I
θ VALUES

s	θ_0	θ_1	θ_2	θ_3	θ_4
0 or 1	144.50	−2.07	−2.64	−0.04	−38.17
0	141.66	−2.15	−2.08	−0.04	–
1	109.18	−2.00	−3.19	−0.04	–

Row one of Table I shows that at every PE-cycle granularity used, the relative decrease in table-based threshold value is 2.07, and for each retention granularity, the relative decrease is about 2.64. In order to obtain a more accurate model or to analyze the rate of change of thresholds, we also obtain separate models for MSB and LSB pages. The second and third rows of Table I, present the θ values when only LSB ($s = 0$) and MSB ($s = 1$) pages are used to obtain a linear model, respectively. These values show that the rate of change of relative threshold values with retention time is higher for MSB pages (-3.19) as compared to LSB pages (-2.08). However the rate of change with PE-cycle and page number is similar.

VI. BER Results

This section presents simulation results evaluating the table-based read scheme and the linear model of read thresholds proposed in Sections IV and V in terms of raw read BER.

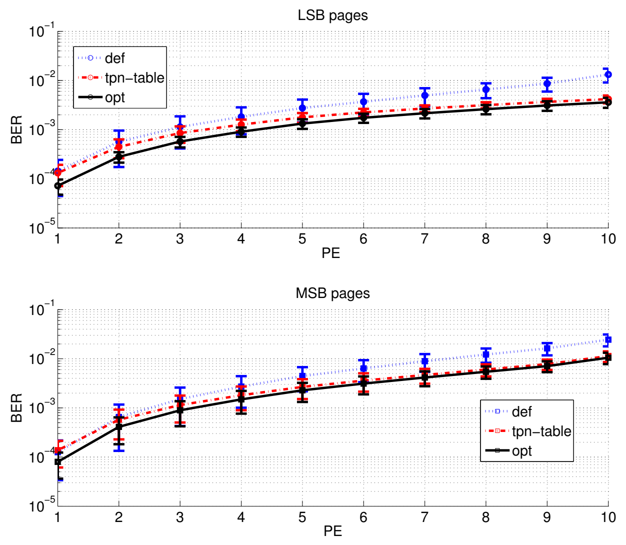

Fig. 5. RBER at 1 month retention.

A. Lab Experiments

The range and/or the relative units of the the system parameters evaluated are as follows:

- retention $t \in \mathbb{T} = \{1, 2, ..., 5\}$
- PE-cycle $p \in \mathbb{P} = \{1, 2, ..., 10\}$
- page number $n \in \mathbb{N} = \{0, 1, ..., 127\}$
- page type $s \in \{0, 1\}$ (LSB (0) and MSB (1))
- threshold values $k \in \mathbb{K} = \{0, 1, ..., K\}$

We select $T = 5$ retention points, $P = 10$ PE-cycle values and the regular $N = 128$ pages for LSB and MSB pages. Each experiment consists of approximately 13000 blocks. We select 10 blocks from several MLC chip and repeat the process of writing pseudo-random data and erasing them to obtain $\{1, 2, ..., 10\}$ relative PE-cycle valued blocks per NAND chip. As a result, each of the 10 blocks within each NAND chip corresponds to a different PE-cycle value. The flash modules are cycled at a temperature and speed that is equivalent to a general user's cycling speed according to the Arrhenius model. Then all blocks are baked at a temperature and time interval that corresponds to a user read retention time as explained in Section IV.

B. Discussion

The BER values shown in Figure 5 are obtained by averaging the bit error values across all blocks and corresponding LSB or MSB pages (top and bottom plots, respectively). The horizontal axis represents the PE-cycle count, which varies from and the vertical axis represents the BER values obtained using each read method. The figure represents the verification set BER values (r_{verif}) for $t = 1$. Similar plots are obtained for other retention values.

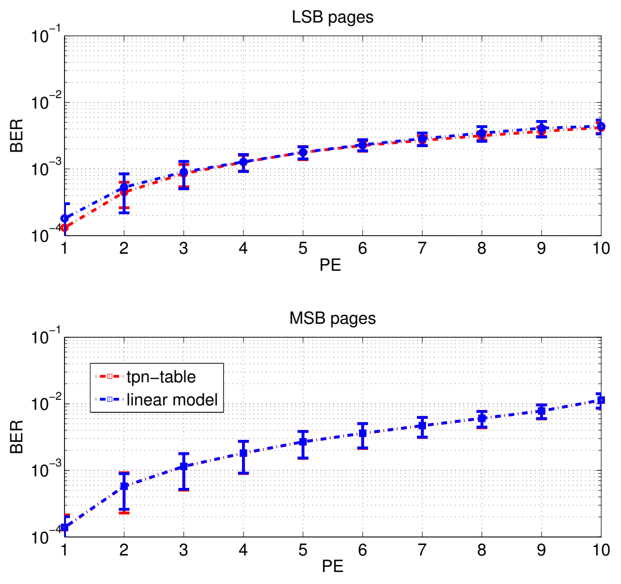

Fig. 6. Table-based vs. linear model comparison RBER.

Figure 5 shows that the table-based read BER is always between the default read BER and the optimum read BER. Table-based performance approaches that of the optimal (r_{opt}) performance as PE-cycle values increase. This figure shows a factor of 5 decrease in BER using the table-based method as opposed to manufacturer's obtained BER (def) at high PE-cycle values.

Comparison of the linear model versus the table-based read is presented in Figure 6 for the same retention point evaluated in Figure 5. The blue dashed line represent the read BER performance when equation (4) with Table I, row one, θ values are used. The solid red plot represents the tpn-table rber. The performance degradation in using a linear function for obtaining threshold values as opposed to a table at $t = 1$ for MSB pages is $4.98e - 10$ and for LSB pages is $3.6595e - 08$. At higher retention values, $t > 1$, this degradation is even smaller.

C. Default-Shift Hybrid Model

There are at least 5 main sources of read voltage variation, which we have considered only four (t, p, n, s) in this paper. Optimum read threshold also depends on where the block within a die sits. In other words, the process corner may have an effect on the read threshold voltage. The NAND manufacturer default read voltage is generally optimized based on the process corner or block number. The table-based read proposed here optimizes over (t, p, n, s), while averaging over b. The default read optimizes over blocks (b) while averaging the effect of (t, p, n, s). If there are (t, p, n, s) cases where the default read outperforms table-based read in terms of BER values, the Hybrid model suggests using default read operation.

VII. Summary

NAND flash controllers typically use a default read threshold voltage to read data. However, such thresholds are not generally optimum in the presence of noise and leakage and result in high bit-error values that activate error-correction mechanisms. This paper discusses increasing the utilization of solid-state storage by dynamically modifying read threshold values over the lifetime of the flash storage. This paper proposes a learning scheme for obtaining near-optimum read thresholds during its lifetime and under different leakage conditions. The ability to select the best read threshold values on a per read request basis allows for lower raw bit-error. With lower raw bit-error, the latency due to error-correction mechanisms decreases and hence the storage system performance increases. Based on the storage system constraints, the optimum thresholds can either be stored in a look-up table or obtained using a linear function of system parameters. This paper shows that both methods result in an order of magnitude lower RBER than the manufacturer's default read settings.

References

[1] B. Shin, C. Seol, J.-S. Chung, and J. J. Kong, "Error control coding and signal processing for flash memories," in Circuits and Systems (ISCAS), 2012 IEEE International Symposium on. IEEE, 2012, pp. 409–412.

[2] H. Zhou, A. Jiang, and J. Bruck, "Error-correcting schemes with dynamic thresholds in nonvolatile memories," in Information Theory Proceedings (ISIT), 2011 IEEE International Symposium on. IEEE, 2011, pp. 2143–2147.

[3] B. Peleato, R. Agarwal, J. Cioffi, M. Qin, and P. H. Siegel, "Towards minimizing read time for nand flash," in Global Communications Conference (GLOBECOM), 2012 IEEE. IEEE, 2012, pp. 3219–3224.

[4] Y. Kim, B. Vijaya Kumar, K. L. Cho, H. Son, J. Kim, J. J. Kong, and J. Lee, "Modulation coding for flash memories," in Computing, Networking and Communications (ICNC), 2013 International Conference on. IEEE, 2013, pp. 961–967.

[5] Y. Cai, O. Mutlu, E. F. Haratsch, and K. Mai, "Program interference in mlc nand flash memory: Characterization, modeling, and mitigation," in Computer Design (ICCD), 2013 IEEE 31st International Conference on. IEEE, 2013, pp. 123–130.

[6] Y. Cai, E. F. Haratsch, O. Mutlu, and K. Mai, "Threshold voltage distribution in mlc nand flash memory: Characterization, analysis, and modeling," in Proceedings of the Conference on Design, Automation and Test in Europe. EDA Consortium, 2013, pp. 1285–1290.

[7] E. Yaakobi, J. Ma, L. Grupp, P. Siegel, S. Swanson, and J. Wolf, "Error characterization and coding schemes for flash memories," in GLOBECOM Workshops (GC Wkshps), 2010 IEEE. IEEE, Dec 2010, pp. 1856–1860.

[8] B. Peleato and R. Agarwal, "Maximizing MLC NAND lifetime and reliability in the presence of write noise," in Proc. of IEEE International Conf. Comm., 2012.

[9] N. Mielke, H. Belgal, A. Fazio, Q. Meng, and N. Righos, "Recovery effects in the distributed cycling of flash memories," in Reliability Physics Symposium Proceedings, 2006. 44th Annual., IEEE International, March 2006, pp. 29–35.