Dynamic Voltage Allocation with Quantized Voltage Levels and Simplified Channel Modeling

Haobo Wang
whb12@ucla.edu

Nathan Wong
nsc.wong@ucla.edu

Richard D. Wesel
wesel@ucla.edu

***Abstract*—After numerous program and erase (P/E) operations the Flash memory read channel experiences significant degradation, especially after certain retention time. As the volume of data programmed and erased from the device increases, successful recovery of the stored data becomes more difficult. Dynamic voltage allocation (DVA) allocates the write threshold voltages of each level as a function of the current degree of channel degradation to increase the lifetime of Flash memory. Several real-world constraints can limit the performance of DVA in practical Flash systems. This paper proposes specific solutions to address two major constraints: imperfect channel modeling and quantized voltage levels. The resulting implementation provides performance close to that of idealized settings.**

***Index Terms*—Flash memory, dynamic voltage allocation, channel modeling, quantization, adaptive signaling**

I. Introduction

Modern Flash memory provides an energy efficient, high-throughput and compact storage solution. In the process of reducing the cost and increasing the capacity of the memory, the read channel degradation problem has become a major concern. The read channel experiences significant degradation over time and eventually is not able to reliably support the complete recovery of the stored data.

In our research, the effect of channel degradation is characterized by lifetime. The lifetime of Flash memory is defined as the number of program and erase (P/E) cycles after which newly stored data will lose its integrity after a certain fixed retention time. There are two related factors that determine the lifetime: the degradation that occurs as a function of the number of P/E cycles and the degradation that occurs during the retention time. The P/E cycle count essentially represents the cumulative amount of new data written to the device. Retention time is the amount of off-line time the memory can endure before the stored data becomes unrecoverable.

Many different solutions have been proposed to extend the lifetime of Flash memory. Commonly, channel codes [1]–[5] are used to provide additional redundancy for the stored data, and guarantee data integrity for a range of degraded channels. Recent work shows that three dimensional layout of the storage cells can dramatically improve the durability of the channel [6], [7]. Another approach is to reduce the number of P/E cycles used to write certain volume of data. Write-once memory (WOM) codes [8]–[10] and rank modulation [11]–[13] are two such examples.

In [14], [15], dynamic threshold assignment (DTA) adjusts the *read* threshold voltages (similar to [5]) to match the changing channel characteristics. Dynamic voltage allocation (DVA) [16], [17] directly adapts *write* levels to extend Flash memory lifetime. DVA optimizes the write threshold voltage of each potential level a memory cell could store. The optimization result is a channel distribution that provides sufficient mutual information to guarantee successful data recovery through error correction coding while reducing the channel degradation caused by each P/E cycle by writing the least amount of charge possible.

This paper focuses on analyzing two major constraints when applying DVA to practical memory systems, quantized voltage levels and imperfect channel modeling, and shows that DVA can still perform well despite these constraints. Throughout the paper Multi-level Cell (MLC) Flash (with four possible levels) is assumed for all the models and simulations, and the retention time is fixed to be one year.

The remainder of this paper is organized as follows: Sec. II presents two models for the Flash read channel. Sec. III introduces the DVA algorithm and a DVA framework for practical systems. Sec. IV examines the performance impact when channel estimation and DVA rely on a simple Gaussian channel model instead of the detailed channel models presented in Sec. II . Sec. V analyses the performance impact of a limited number of levels being available for the read and write threshold voltage placements. Sec. VI concludes the paper.

II. Channel Model & Parameters

Our precursor conference papers [16] and [17] present a Gaussian-exponential parameterized channel model characterizing the read channel as having three additive noise components: programming noise, wear-out noise, and retention noise. In this paper, this model is called *Model 1*. While Model 1 characterizes the major Flash read channel distribution properties, this paper also presents *Model 2* as an improved model that adds two noise components: cell-to-cell interference and programming error. Model 2 provides a more precise characterization of the read channel in practical systems.

This work is supported by Micron through the UCLA Center on Development of Emerging Storage Systems. This material is based upon work supported partially by the National Science Foundation under Grant Number 1162501. Any opinions findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

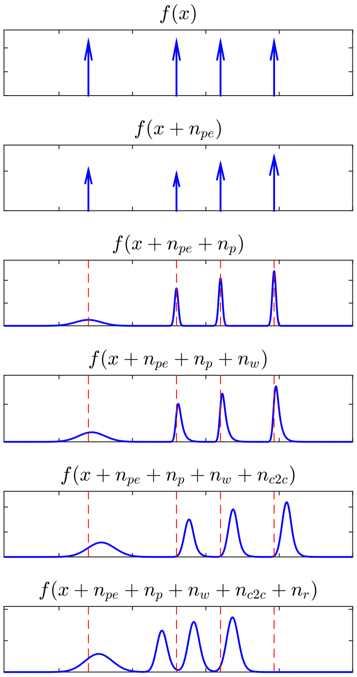

Fig. 1. Flash read channel PDFs illustrating noise components.

A. *Channel Model with Additive Components*

Thus, our two Flash memory read channel models are formulated as follows [16]–[19]:

- Model 1:

$$y = x + n_p + n_w + n_r. \tag{1}$$

- Model 2:

$$y = x + n_{pe} + n_p + n_w + n_{c2c} + n_r, \tag{2}$$

where x is the write threshold voltage, and y is the measured threshold voltage. Noise n_p represents the programming noise, n_w represents the wear-out noise, n_r represents the retention noise, n_{c2c} represents the cell-to-cell interference, and n_{pe} represents the programming error. Fig. 1 shows an example voltage distribution of the five noise components. The arrows denote delta functions.

1) Programming Error n_{pe}: The Programming Error noise is modeled with a probability mass function (PMF) $P(Y = y|X = x)$, which is the conditional probability of actually writing level y when the intended level is x. Note that programming error, which results from misreading the least significant bit before writing the most significant bit, always results in a valid write level.

2) Programming Noise n_p: Programming noise is modeled with a Gaussian distribution for each level. The probability density function (PDF) is

$$f_{n_p}(n_p|x = l) = \begin{cases} \mathcal{N}(0, \sigma_p^2) & \text{if } l = 0 \\ \mathcal{N}(0, \sigma_e^2) & \text{if } l > 0 \end{cases}, \tag{3}$$

where $\sigma_e > \sigma_p$. Index l represents the level corresponding to a write threshold voltage. For MLC Flash, $l \in \{0, 1, 2, 3\}$ where $l = 0$ indicates the erased level.

3) Wear-out Noise n_w: Wear-out noise is described by a positive-side exponential[1] noise for each level. The PDF is

$$f_{n_w}(n_w) = \begin{cases} \frac{1}{\lambda} e^{-\frac{n_w}{\lambda}} & \text{if } n_w \geq 0, \\ 0 & \text{if } n_w < 0. \end{cases} \tag{4}$$

4) Cell-to-cell Interference n_{c2c}: Cell-to-cell interference to a certain cell is modeled by a weighted sum of neighboring cells' voltage increase due to write operations. For Flash memories employing the common even-odd structure for writing and reading operations assuming even cells are written first in each wordline, the interference can be represented as

$$\begin{aligned} V_{n_{c2c,odd}} &= \gamma_{x,right} \times V_{x,left} + \gamma_{x,left} \times V_{x,right} \\ &\quad + \gamma_y \times V_y, \qquad \text{(5a)} \\ V_{n_{c2c,even}} &= V_{n_{c2c,odd}} + \gamma_{xy,upper-left} \times V_{xy,upper-left} \\ &\quad + \gamma_{xy,upper-right} \times V_{xy,upper-right}. \qquad \text{(5b)} \end{aligned}$$

Voltage $V_{n_{c2c,odd}}$ is the interference experienced by the odd cells in each wordline, and $V_{n_{c2c,even}}$ is the interference experiences by the even cells. The voltage increases V_x, V_y and V_{xy} represent the voltage difference written to the adjacent cells of the cell of interest. Subscript x indicates adjacent cells on the same wordline, y indicates the adjacent cell on the subsequent wordline, but the same bit line, and xy indicates diagonally adjacent cells on the subsequent wordline and an adjacent bitline.

5) Retention Noise n_r: The retention noise is modeled as a Gaussian random variable with PDF

$$f_{n_r}(n_r) = \frac{1}{\sigma_r \sqrt{2\pi}} e^{-\frac{(n_r - \mu_r)^2}{2\sigma_r^2}}. \tag{6}$$

B. *Channel Parameters*

The channel parameters in noise components in Sec. II-A determine both the static and dynamic characteristics of the channel. The static channel parameters are: σ_p and σ_e in programming noise. These parameters remain constant over the lifetime of the memory. The dynamic channel parameters are: the various $P(Y = y|X = x)$ values in programming noise; the λ values in wear-out noise; the γ values in cell-to-cell interference; and μ_r and σ_r in retention noise. These parameter values change with the number of P/E cycles and retention time, representing the channel degradation process. The channel parameter degradation models used to calculate dynamic channel paramaters are described in detail in [16], [18], [19].

III. Dynamic Voltage Allocation

Dynamic Voltage Allocation (DVA) [16], [17] optimizes Flash memory read channel to provide the necessary amount of mutual information (above the minimum required for reliable decoding) after a certain number of P/E cycles and for a specified retention time. The algorithm uses a single factor

[1] Wear-out noise can also be a negative-side exponential or a Laplace (double-sided exponential) distribution depending on actual memory implementation.

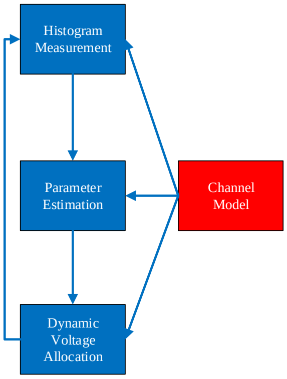

Fig. 2. DVA framework for practical systems.

to scale the write threshold voltage of each level. The effect of the scaling is a controlled increase of the distance between the write threshold voltages of adjacent levels. In this paper, the minimum mutual information limit of the system is 1.9 bits and the target of DVA is set to be 1.94 bits, providing 0.04 bits of margin.

The ideal DVA algorithm relies on perfect knowledge of channel distribution. In reality, this information is not readily available. We propose a DVA framework to provide the channel information by estimating the channel parameters in the channel model. The framework is depicted in Fig. 2. In the first step, a histogram is read from the memory using read threshold placements optimized for channel estimation. In the second step, a least squares algorithm estimates the channel parameters based on the measured histogram. In the third step, DVA calculates the scaling factor based on the channel model induced by the estimated parameters. The process is repeated regularly to provide the channel with sufficient mutual information. In all the simulations in this paper, the repetition period is set to be 100 P/E cycles.

From [17], a nine-read equal-probability bin-placement scheme is the optimal choice for the first step. This bin-placement scheme places reads used to measure the histogram such that each bin has approximately the same height. For the second step, the Levenberg-Marquardt algorithm is shown in [17] to have the optimal parameter estimation performance.

For Flash memory systems which do not need to consider cell-to-cell interference and programming error, the framework shown in Fig. 2 can be used directly to implement DVA. The channel model in this case is Model 1. For more common systems characterized by Model 2, the implementation of DVA needs to take into account the performance difference between even cells and odd cells in each wordline. We propose a DVA system for this type of memory based on Fig. 2 but with two modifications. The first additional procedure for the DVA system is the switching of write order between even and odd cells. The order will be switched in each wordline every 100 P/E cycles in sync with the period of the DVA framework to equalize the channel degradation of the even and odd cells.

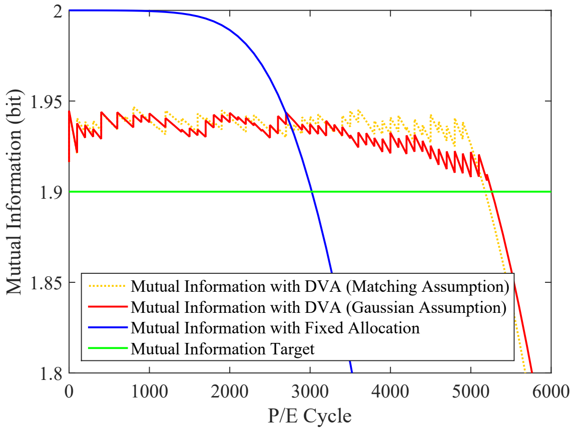

Fig. 3. DVA's performance with Gaussian model. (Ground truth model is Model 1.)

The second modification is that there are two DVA instances and correspondingly two scaling factors, one for even cells and one for odd cells.

IV. DVA with Simplified Channel Modeling

In this section, DVA is implemented using a Gaussian distribution as a simplified channel model for each level. Thus, the channel parameters need to be estimated are the means and variances of the Gaussian distributions. Note that Models 1 and 2 are still used as the ground truth read distribution in both our analysis and our simulations. Thus there is a mismatch between the simple Gaussian model (GM) used for channel estimation and DVA and the actual channel.

Perfect matching between the estimated channel and the actual channel can only be achieved under two conditions:

1) The channel model exactly matches the actual channel.
2) The estimated channel parameters are precise.

These two conditions are hard to satisfy in reality because many factors are involved in shaping the channel characteristics. Even if the exact channel model is known, the complexity of the model may make its application in practical DVA systems impossible because of the computational complexity. A simple but capable channel model which can significantly increase the computational efficiency of the DVA system is desired in practical implementations. The performance loss caused by the channel model mismatch can be controlled when the optimization target of the DVA algorithm, which is the mutual information target of the process, are set properly.

A. Model 1

Fig. 3 shows a comparison of the performance difference between using the actual channel of Model 1 for channel estimation and GM-DVA. The performance of the simplified GM-DVA system is comparable to the performance of the model-matching system. This is expected as Model 1 is similar to the multi-modal Gaussian model. The lifetime of the device is extended by 74.2% from 3020 P/E cycles to 5260 P/E cycles using GM-DVA.

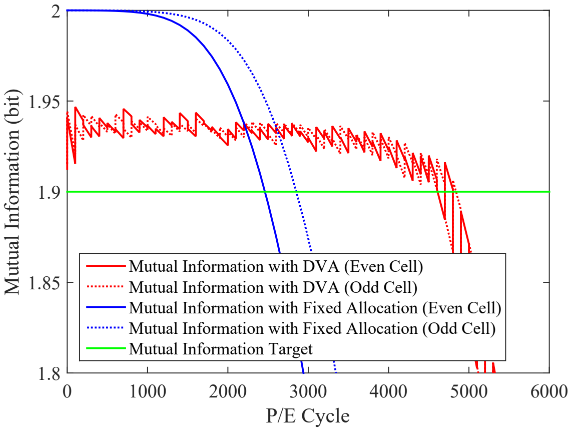

Fig. 4. DVA performance with Gaussian model. Ground truth model is Model 2.

Note that in Fig. 3, the performance of the model-mismatching system is slightly better than that of the model-matching system. Also, the deviation of the two curves representing the two systems grows in the second half of the device's lifetime. The deviation represents that the mismatch between Gaussian distribution and Model 1 increases with P/E cycles. Furthermore, the deviation shows Gaussian distribution underestimates the degree of channel degradation as the channel worsens. This tendency of underestimation leads the GM-DVA algorithm to provide voltage allocation results which utilize the built-in 0.04 bits margin more aggressively.

B. Model 2

Fig. 4 shows the Monte Carlo simulation performance of GM-DVA with Model 2 as the ground truth distribution. The result indicates that the periodic switching of the write order in each wordline effectively reduces the performance difference between the even-cell channel and the odd-cell channel. Similar to the case in Sec. IV-A, the downward bend of the curves suggests that GM-DVA underestimates the channel degradation. The overall lifetime extension is about 87.4% from 2460 P/E cycles to 4609 P/E cycles in this case.

V. DVA with Quantized Voltage Levels

In the analysis and simulations presented above, both write and read threshold voltages have floating point precision. In practice, hardware limitations only allow the thresholds to be placed at certain voltage values, and the originally calculated values need to be quantized. This constraint adversely impacts DVA performance.

In this paper, the performance of DVA with quantized voltage levels is analyzed under three practical quantization schemes: 256, 128 and 64-level uniform quantization. The mutual information v.s. P/E cycle curves under these quantization schemes are compared with Fig. 4 to see if they cause a premature dip below the minimum 1.9 bits mutual information target. Uniform quantization means adjacent voltage placements are separated by a constant difference. In the following simulations, the quantization range is set to be from -1 Volt to 8 Volts. The desired system should have a quantization scheme with the smallest possible total number of possible voltage placements.

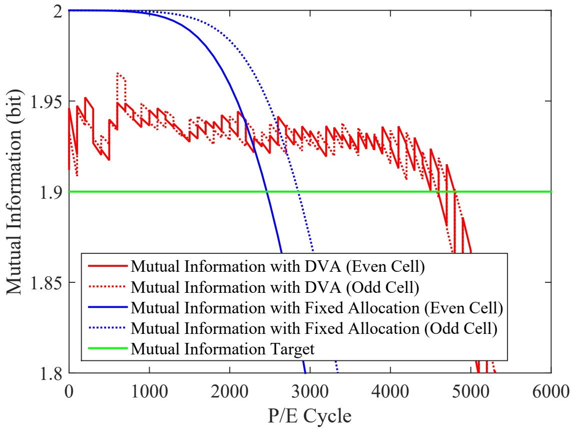

Fig. 5. DVA's performance with quantized placements. (128-level uniform quantization. Ground truth model is Model 2. The overall lifetime is extended by 87.0% from 2460 P/E cycles to 4600 P/E cycles.)

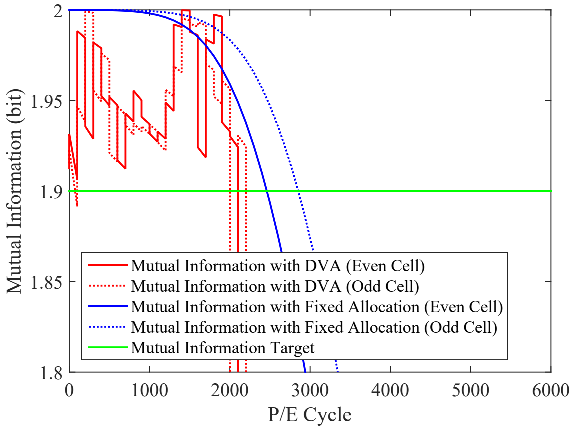

Fig. 6. DVA performance with quantized placements. (64-level uniform quantization. Ground truth model is Model 2.)

The simulations suggest that quantization with 128 possible values strikes a nice balance between DVA performance and the number of potential voltage values. When using 128-level quantization, the overall lifetime is extended by 86.1% from 2460 P/E cycles to 4578 P/E cycles as shown in Fig. 5. The quantization interval is 0.0714 Volts. Figs. 6 suggests 64-level quantization with an interval of 0.1452 Volts will prohibit the system from functioning properly. In this case, we observed that two or more read threshold voltages would be in the same place. This reduces the effective resolution of the measured histogram, as a result, DVA performance suffers from less reliable channel estimations.

The read and write quantization processes have very different properties. Write threshold quantization affects the system's performance directly. Read threshold quantization affect the system's performance through the reliability of channel estimations. We conducted simulations to determine which

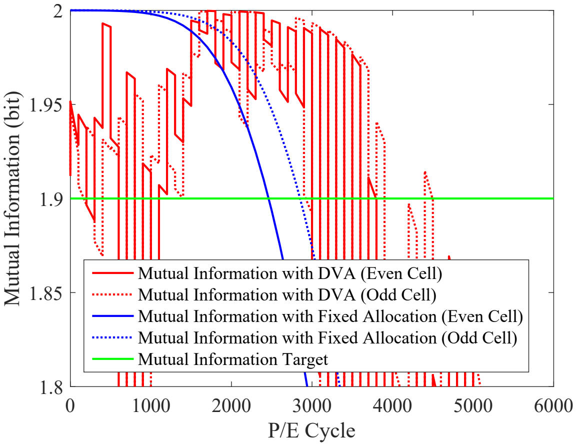

Fig. 7. DVA performance with quantized bin placements only. (64-level uniform quantization. Ground truth model is Model 2.)

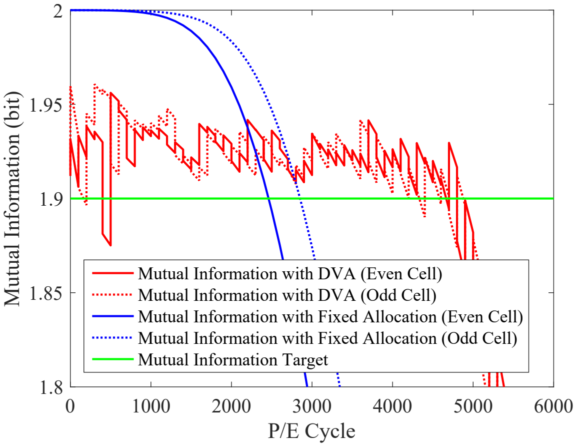

Fig. 8. DVA performance with quantized DVA placements only. (64-level uniform quantization. Ground truth model is Model 2.)

quantization is the major factor of the performance loss when quantizing to 64 levels. Fig. 7 shows GM-DVA performance with no quantization of write levels and 64-level uniform *read threshold quantization*. The curves show many excursions below the target mutual information and performance differs significantly from Fig. 4. In contrast, Fig. 8 suggests that the GM-DVA system functions well in most P/E cycle conditions (and generally tracks Fig. 4) with only 64-level uniform *write threshold quantization* when the read levels are unquantized. We conclude that read threshold quantization has a more critical impact on the performance of DVA. Practical DVA implementation needs to provide sufficiently fine-grain quantization, especially for read thresholds.

VI. Conclusion

This paper studies two important practical implementation constraints of Dynamic Voltage Allocation (DVA), imperfect channel modeling and quantized voltage levels for reading and writing thresholds. Analysis and simulation results demonstrate that the a Gaussian channel model can be used to estimate complex Flash channels for DVA without significantly degrading performance. Similarly, we found that quantizing to 128 voltage levels for reading and writing does not significantly impact the performance of DVA. Thus, DVA can function properly under these two practical constraints and extend Flash memory lifetime significantly.

References

[1] B. Chen, X. Zhang, and Z. Wang, “Error correction for multi-level NAND flash memory using Reed-Solomon codes,” in *Signal Processing Systems, 2008. SiPS 2008. IEEE Workshop on*, Oct. 2008, pp. 94–99.

[2] Y. Maeda and H. Kaneko, “Error Control Coding for Multilevel Cell Flash Memories Using Nonbinary Low-Density Parity-Check Codes,” in *Defect and Fault Tolerance in VLSI Systems, 2009. DFT ’09. 24th IEEE International Symposium on*, Oct. 2009, pp. 367–375.

[3] T. Klove, B. Bose, and N. Elarief, “Systematic, Single Limited Magnitude Error Correcting Codes for Flash Memories,” *Information Theory, IEEE Transactions on*, vol. 57, no. 7, pp. 4477–4487, Jul. 2011.

[4] F. Zhang, H. Pfister, and A. Jiang, “LDPC codes for rank modulation in flash memories,” in *Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on*, Jun. 2010, pp. 859–863.

[5] J. Wang, K. Vakilinia, T.-Y. Chen, T. Courtade, G. Dong, T. Zhang, H. Shankar, and R. Wesel, “Enhanced Precision Through Multiple Reads for LDPC Decoding in Flash Memories,” *Selected Areas in Comm., IEEE Journal on*, vol. 32, no. 5, pp. 880–891, May 2014.

[6] J. Choi and K. S. Seol, “3d approaches for non-volatile memory,” in *VLSI Technology (VLSIT), 2011 Symposium on*, Jun. 2011, pp. 178–179.

[7] K.-T. Park, D.-S. Byeon, and D.-H. Kim, “A world’s first product of three-dimensional vertical NAND Flash memory and beyond,” in *Non-Volatile Memory Technology Symposium (NVMTS), 2014 14th Annual*, Oct. 2014, pp. 1–5.

[8] A. Jiang, “On The Generalization of Error-Correcting WOM Codes,” in *Information Theory, 2007. ISIT 2007. IEEE International Symposium on*, Jun. 2007, pp. 1391–1395.

[9] E. Yaakobi, S. Kayser, P. Siegel, A. Vardy, and J. Wolf, “Codes for Write-Once Memories,” *Information Theory, IEEE Transactions on*, vol. 58, no. 9, pp. 5985–5999, Sep. 2012.

[10] R. Gabrys and L. Dolecek, “Constructions of Nonbinary WOM Codes for Multilevel Flash Memories,” *Information Theory, IEEE Transactions on*, vol. 61, no. 4, pp. 1905–1919, Apr. 2015.

[11] A. Jiang, M. Schwartz, and J. Bruck, “Error-correcting codes for rank modulation,” in *Information Theory, 2008. ISIT 2008. IEEE International Symposium on*, Jul. 2008, pp. 1736–1740.

[12] A. Mazumdar, A. Barg, and G. Zemor, “Constructions of Rank Modulation Codes,” *Information Theory, IEEE Transactions on*, vol. 59, no. 2, pp. 1018–1029, Feb. 2013.

[13] M. Qin, A. Jiang, and P. Siegel, “Parallel programming of rank modulation,” in *Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on*, Jul. 2013, pp. 719–723.

[14] 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*, Jul. 2011, pp. 2143–2147.

[15] F. Sala, R. Gabrys, and L. Dolecek, “Dynamic Threshold Schemes for Multi-Level Non-Volatile Memories,” *IEEE Transactions on Communications*, vol. 61, no. 7, pp. 2624–2634, Jul. 2013. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6528074

[16] T.-Y. Chen, A. R. Williamson, and R. D. Wesel, “Increasing flash memory lifetime by dynamic voltage allocation for constant mutual information,” in *Information Theory and Applications Workshop (ITA), 2014*, Feb. 2014, pp. 1–5.

[17] H. Wang, T.-Y. Chen, and R. Wesel, “Histogram-based Flash channel estimation,” in *Communications (ICC), 2015 IEEE International Conference on*, Jun. 2015, pp. 283–288.

[18] G. Dong, S. Li, and T. Zhang, “Using Data Postcompensation and Predistortion to Tolerate Cell-to-Cell Interference in MLC nand Flash Memory,” *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 57, no. 10, pp. 2718–2728, Oct. 2010.

[19] T. Parnell, N. Papandreou, T. Mittelholzer, and H. Pozidis, “Modelling of the threshold voltage distributions of sub-20nm NAND flash memory,” in *Global Communications Conference (GLOBECOM), 2014 IEEE*, Dec. 2014, pp. 2351–2356.