<table>
<thead>
<tr>
<th>Title</th>
<th>Read and Write Voltage Signal Optimization for Multi-Level-Cell (MLC) NAND Flash Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author(s)</td>
<td>Aslam, Chaudhry Adnan; Guan, Yong Liang; Cai, Kui</td>
</tr>
<tr>
<td>Date</td>
<td>2016</td>
</tr>
<tr>
<td>URL</td>
<td><a href="http://hdl.handle.net/10220/40548">http://hdl.handle.net/10220/40548</a></td>
</tr>
<tr>
<td>Rights</td>
<td>© 2016 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. The published version is available at: [<a href="http://dx.doi.org/10.1109/TCOMM.2016.2533498">http://dx.doi.org/10.1109/TCOMM.2016.2533498</a>].</td>
</tr>
</tbody>
</table>
Read and Write Voltage Signal Optimization for Multi-Level-Cell (MLC) NAND Flash Memory

Chaudhry Adnan Aslam, Student Member, IEEE, Yong Liang Guan, Member, IEEE,
Kui Cai, Senior Member, IEEE

Abstract—The multi-level-cell (MLC) NAND flash channel exhibits non-stationary behavior over increasing program and erase (PE) cycles and data retention time. In this paper, an optimization scheme for adjusting the read (quantized) and write (verify) voltage levels to adapt to the non-stationary flash channel is presented. Using a model-based approach to represent the flash channel, incorporating the programming noise, random telegraph noise (RTN), data retention noise and cell-to-cell interference as major signal degradation components, the write-voltage levels are optimized by minimizing the channel error probability. Moreover, for selecting the quantization levels for the read-voltage to facilitate soft LDPC decoding, an entropy-based function is introduced by which the voltage erasure regions (error dominating regions) are controlled to produce the lowest bit/frame error probability. The proposed write and read voltage optimization schemes not only minimize the error probability throughout the operational lifetime of flash memory, but also improve the decoding convergence speed. Finally, to minimize the number of read-voltage quantization levels while ensuring LDPC decoder convergence, the extrinsic information transfer (EXIT) analysis is performed over the MLC flash channel.

Index Terms—MLC NAND flash memory, read-voltage, write-voltage, LDPC code, error performance.

I. INTRODUCTION

THE NAND flash memory is the ubiquitous storage medium in many consumer electronic products. With multi-level-cell (MLC) technology, flash memory can store multiple bits over a single memory cell, leading to significant growth in its storage capacity. Using advanced chip manufacturing processes, it has become viable to replace magnetic storage disks with NAND flash memory based solid-state drives (SSD) for large enterprise data applications.

In flash memory, a cell can either be represented with the erased state (no electrons stored over the floating-gate) or with the programmed state (electrons stored over the floating-gate). Initially, all memory cells are in the erased state. The programming operation shifts the threshold voltage (voltage required to turn-on the transistor) of a memory cell to a particular write-voltage level. Considering an example of 2-bit per cell flash memory, a memory cell can be configured to four distinct voltage levels, say $V_{\text{min}}, V_1, V_2, V_{\text{max}}$, for representing data symbols ‘11’, ‘10’, ‘00’ and ‘01’, respectively, where $V_{\text{min}}$ is the mean threshold voltage of the erased cells and $V_{\text{max}}$ is the mean threshold voltage of programmed cells configured with the highest write-voltage level. It is observed that the probability distribution functions for these distinct voltage levels are not identical due to the presence of some non-stationary and asymmetric channel noise. In this scenario, $V_{\text{min}}$, $V_1$, $V_2$ and $V_{\text{max}}$ should not be equally-spaced but determined based on some optimization criteria.

To read a flash memory cell, the amount of stored electrical charge is measured by applying the read-voltage in discrete steps. For a memory cell configured to threshold voltage $V_0$, a read-voltage greater than $V_0$ is required to measure the stored electric charge. For 2-bit per cell flash memory, we need at least 3 read-voltage levels $R_1$, $R_2$ and $R_3$, where $R_1$ can be set between $V_{\text{min}}$ and $V_1$ to distinguish between symbols ‘11’ and ‘10’, $R_2$ can be set between $V_1$ and $V_2$ to distinguish between symbols ‘10’ and ‘00’ and $R_3$ can be set between $V_2$ and $V_{\text{max}}$ to distinguish between symbols ‘00’ and ‘01’, as shown in Fig. 1. Since the write-voltage levels are not equally-spaced, the read-voltage levels should also be appropriately designed so that the error-rate is minimized.

For a state-of-the-art flash memory, only 3-level memory sensing scheme may not be sufficient to achieve the satisfactory error performance. This is because the modern flash memory chips experience severe voltage level distortions due to some circuit-level noise and interference effects. As a consequence, flash data reliability is degraded [1]–[3]. To overcome the reliability issues, strong error correcting codes (ECCs), such as low-density parity-check (LDPC) codes, are seriously considered [4], [5]. To reap full benefits of LDPC code, it is desirable to have LDPC decoder’s input values quantized as finely as possible. This requires high-precision memory sensing, and consequently more reading latency, which may not be acceptable for time-critical applications. Against these background, we analyze two research problems associated with flash memory read-voltage signal design: identifying the minimum number of read-voltage levels required under the given flash channel condition, and then finding the ideal values for these read-voltage levels.

A. Related Work

To the best of our knowledge, the write-voltage optimization over a non-stationary MLC flash channel has not been reported in the open literature. In [6], the write-voltage levels are optimized based on a simple flash channel by considering random telegraph noise (RTN) as the only source of voltage signal degradation. Another relevant work is reported in [7] where the write-voltage levels are optimized assuming an AWGN
channel for flash memory. Our paper presents the optimization of write-voltage signals using a more comprehensive flash channel model that includes the effects of programming noise, cell-to-cell interference as well as non-stationary RTN and data retention noise.

In [8], an ECC scheme is jointly designed with write-voltage levels, relating the code’s error correction capability with the voltage signal magnitude. A constrained coding scheme is presented in [9] for mitigating the effect of cell-to-cell interference in which certain voltage levels of neighboring cells are forbidden so that a wider gap between adjacent voltage distributions is assured.

In addition to MLC technology, where discrete write-voltage levels are used to store information, relative value (ordering) of charge levels can also be used to represent flash data, as discussed in rank modulation scheme [10]. In this scheme, non-overlapping voltage regions are required to represent data symbols. In [11], these optimal continuous voltage regions are designed. The rank modulation scheme can increase the cell storage capacity and overcome the programming overshoot errors. However, to implement this scheme, a large number of read operations are required to learn the voltage-level ordering. Thus, in this paper we focus on MLC technology and strive to optimize its discrete write-voltage levels.

In the conventional MLC flash, to avoid the effect of voltage overshoot errors, memory cells are configured on desired write-voltage levels in multiple rounds of programming. In [11]–[13], for both MLC and rank modulation schemes, the optimal programming step size has been investigated, keeping the number of programming rounds constant and assuming fixed write-voltage levels. In this paper, we address the problem of finding the optimized write-voltage levels to ensure that the flash channel error probability is minimized.

With regards to the read-voltage quantization for soft-decision decoding, multiple memory read operations are typically performed. In this direction, a simplistic approach is to apply equally spaced read-voltage levels (uniform memory sensing). However, this is not suitable for flash channel and yields poor error performance [14]. For this reason, a non-uniform quantization scheme is adopted in [14], where the quantization levels are obtained at the intersecting region between two adjacent distribution functions by using constant ratio method. Alternative to enhanced precision, hard-decision based dynamic quantization schemes are also reported in [15], [16] where the read-voltage levels are adjusted according to the non-stationary behavior of flash channel.

Another important work related to quantization design is presented in [17] in which the quantization levels are obtained by maximizing the mutual-information (MMI) between flash channel’s input and output voltage signals. In this paper, we propose a read-signal quantization scheme based on a novel voltage entropy function that is able to give better decoding error performance than the MMI scheme.

B. Contributions

In this paper, we present a novel analytical approach to optimize the write-voltage signals for 2-bit per cell flash memory. The proposed principles can be readily extended to 3-bit or 4-bit per cell flash technology. In view of the varying flash memory channel, which causes the threshold voltage distribution function to change over the number of PE cycles and data retention time (the duration of time since the memory cell was last programmed), we propose to adapt the write-voltage signals on-the-fly such that the channel error probability is minimized. Furthermore, we present a voltage entropy-based quantization scheme for reading the flash memory cells, so that the dominating error regions (erasure) are enclosed within the designed quantization levels. Finally, based on the given flash channel condition, we recommend to use the EXIT curves to identify the minimum number of quantization levels required for the convergence of soft LDPC decoding.

The rest of the paper is organized as follows. In Section II, we present the flash channel model. In Section III, we formulate the probability of error expression for flash channel and find optimum write-voltage levels. In Section IV, we discuss the read-voltage signal design scheme using voltage entropy function. In Section V, we analyze the error performance of the proposed read-voltage scheme. In Section VI, we discuss the improvements in decoding convergence speed as a consequence of optimized write-voltage levels. In Section VII, we compare the impact of write-voltage optimization between the flash memory’s early and end-of-retention times. In Section VIII, we describe the usage of EXIT curves for the selection of memory sensing precision for LDPC decoder convergence. In Section IX, we draw conclusions.

II. MLC Flash Memory Channel Model

We model the MLC flash memory channel by incorporating the effects of programming noise, random telegraph noise (RTN), data retention noise and cell-to-cell interference (CCI). Channel models similar to ours are extensively reported in the open literature [18]–[24].

A. Initial Threshold Voltage Distribution

In flash memory array, the threshold voltage distribution for erased cells, \( p_{11} \), can be modeled with Gaussian distribution [18]–[25], given by

\[
p_{11}(x) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x-v_{th})^2}{2\sigma^2}}
\]

(1)
where $V_{\text{min}}$ and $\sigma_e$ are the mean and standard deviation of cell’s threshold voltage. Using a 2-bit per cell flash memory, we represent the data symbol 11 with voltage $V_{\text{min}}$, and 10, 00, and 01 with three programmed voltage levels $V_1$, $V_2$, and $V_{\text{max}}$, respectively. To configure the memory cells on one of these programmed voltage levels, an iterative incremental step pulse programming (ISPP) [26] technique is used. This causes the voltage distribution for programmed cells to follow uniform distribution [18]–[20], [24], given by

$$p_{\nu_p}(x) = \begin{cases} \frac{1}{\Delta V_{pp}}, & \text{for } V_p \leq x \leq V_p + \Delta V_{pp} \\ 0, & \text{otherwise} \end{cases}$$

(2)

where $V_p$ is the desired write-voltage level and $\Delta V_{pp}$ is the programming voltage step size. In this paper, we fix the value of $\Delta V_{pp}$ to 0.3, however, it can be optimized to increase the cell storage capacity [11]–[13]. Furthermore, the programmed cells are also affected by the programming noise, $p_t$, which can be modelled using Gaussian distribution [20]–[23] with zero mean and $\sigma_p$ standard deviation. The overall voltage distribution for programmed cells can then be represented as the convolution integral (*) of uniform and Gaussian distribution functions, given by

$$p_{\nu_p}(x) = p_{\nu_p}(x) \ast p_t(x)$$

(3)

where $p_{\nu_p} \in \{p_{\nu_{110}}, p_{\nu_{010}}, p_{\nu_{001}}\}$ for $V_p \in \{V_1, V_2, V_{\text{max}}\}$.

B. Cell-to-Cell Interference (CCI)

The cell-to-cell interference (CCI) is induced as a result of parasitic capacitive-coupling between the adjacent memory cells. According to [14], [18]–[20], [22], [24], [27], the threshold voltage shift due to CCI, $\Delta V_{CCI}$, can be given as

$$\Delta V_{CCI} = \sum_{k} \Delta V_k \gamma_k$$

(4)

where $\Delta V_k$ is the change in the threshold voltage of interfering (neighboring) cells due to their programming and $\gamma_k$ is the capacitive coupling ratio. A victim cell can be interfered by three or five neighboring cells. For an even-odd bit-line architecture, where the even bit-lines are programmed before the odd bit-lines, there are five interfering cells for each even bit-line cell and three interfering cells for each odd bit-line cell, as shown in Fig. 2. Alternatively, in an all bit-line architecture, where all word-line cells are programmed simultaneously, a victim cell is interfered by three neighboring cells located on the next word-line.

According to [28], the strength of CCI can be estimated before the cell programming operation and can be removed from the desired write-voltage level using cell pre-coding technique. However, this technique cannot mitigate the interference effect from the erased state cells. Thus, we can still approximate the voltage distribution of programmed cells using (3). However, for the erased state cells, we model their threshold voltage using Gaussian distribution function with shifted mean $V_{\text{min}}$, given as

$$V_{\text{min}}^{\text{even}} = V_{\text{min}} + \Delta V_{\text{ave}}(2\mu_{\gamma_e} + \mu_{\gamma} + 2\mu_{\gamma_{xy}})$$

(5)

$$V_{\text{min}}^{\text{odd}} = V_{\text{min}} + \Delta V_{\text{ave}}(\mu_{\gamma} + 2\mu_{\gamma_{xy}})$$

(6)

where $\Delta V_{\text{ave}} = (V_{\text{min}} + V_{\text{max}})/2 - V_{\text{min}}$. Here (5) and (6) are used for even and odd bit-line cells, respectively.

C. Random Telegraph Noise (RTN)

In flash memory, the RTN is a non-stationary noise component whose effect is related with memory PE cycles. According to [18]–[28], [31], it can be modeled using a symmetric exponential distribution function. However, for mathematical tractability, we approximate the RTN distribution by a zero-mean Gaussian distribution function, given by

$$p_{\nu_t}(x) = \frac{1}{\sigma_{\nu_t}\sqrt{2\pi}} e^{-\frac{(x-\mu_{\nu_t})^2}{2\sigma_{\nu_t}^2}}$$

(7)

where the RTN variance, $\sigma_{\nu_t}^2$, is a non-stationary parameter which varies with respect to the PE cycles in a power-law fashion. In this paper, we set $\sigma_{\nu_t} = 0.00025(PE)^{0.62}$.

D. Data Retention Noise

Data retention noise is also recognized as a non-stationary and data-dependent effect related to the memory PE cycles and data retention time. According to [18]–[24], the retention noise can be approximated with Gaussian distribution, given as

$$p_t(x) = \frac{1}{\sigma_t\sqrt{2\pi}} e^{-\frac{(x-\mu_t)^2}{2\sigma_t^2}}$$

(8)

We set the data-dependent mean $\mu_t$, and variance $\sigma_t^2$, parameters according to [20], as given by

$$\mu_t = (V_s - x_0) \cdot [A_t(PE)^{\alpha_t} + B_t(PE)^{\alpha_t}] \cdot \log(1 + T)$$

(9)

$$\sigma_t = 0.4 \cdot \mu_t$$

(10)

where $\mu_t \in \{\mu_{t_{s11}}, \mu_{t_{s10}}, \mu_{t_{s01}}, \mu_{t_{s00}}\}$ and $\sigma_t \in \{\sigma_{t_{s11}}, \sigma_{t_{s10}}, \sigma_{t_{s01}}, \sigma_{t_{s00}}\}$ for $V_s \in \{V_{\text{min}}, V_1, V_2, V_{\text{max}}\}$. Here, $T$ is the data retention time. For all our simulations, we set the following flash parameters: $V_{\text{min}} = 1.4$, $\sigma_e = 0.35$, $V_1 = 2.6$, $V_2 = 3.2$, $V_{\text{max}} = 3.93$, $\sigma_p = 0.05$, $\mu_{\gamma_e} = 0.08$, $\mu_{\gamma} = 0.006$, $x_0 = 1.4$, $A_t = 0.00055$, $B_t = 0.000235$, $\alpha_t = 0.62$ and $\alpha_0 = 0.32$.

$^1$The effect of random telegraph noise (RTN) on threshold voltage signal is less significant as compared to other noise components present in the flash channel [29], [30].
The final threshold voltage distribution can be computed as the convolution integral of initial voltage distribution function with RTN and data retention noise. Thus, we have
\[
p_{s_{11}}(v) = \frac{1}{\sigma_{s_{11}} \sqrt{2\pi}} e^{-\left(\frac{(v_{\text{min}} - \mu_{s_{11}}))^2}{2\sigma_{s_{11}}^2}\right)} \quad (11)
\]
where \(\sigma_{s_{11}}^2 = \sigma_e^2 + \sigma_n^2 + \sigma_{r_{11}}^2\)
\[
p_{s_{10}}(v) = \frac{1}{\Delta V_{pp}} \left( \text{erf} \left( \frac{V_1 + \Delta V_{pp} - v - \mu_{s_{10}}}{\sqrt{2}\sigma_{s_{10}}} \right) \right) - \frac{1}{\Delta V_{pp}} \left( \text{erf} \left( \frac{V_1 - v - \mu_{s_{10}}}{\sqrt{2}\sigma_{s_{10}}} \right) \right) \quad (12)
\]
where \(\sigma_{s_{10}}^2 = \sigma_e^2 + \sigma_n^2 + \sigma_{r_{10}}^2\)
\[
p_{s_{00}}(v) = \frac{1}{\Delta V_{pp}} \left( \text{erf} \left( \frac{V_2 + \Delta V_{pp} - v - \mu_{s_{00}}}{\sqrt{2}\sigma_{s_{00}}} \right) \right) - \frac{1}{\Delta V_{pp}} \left( \text{erf} \left( \frac{V_2 - v - \mu_{s_{00}}}{\sqrt{2}\sigma_{s_{00}}} \right) \right) \quad (13)
\]
where \(\sigma_{s_{00}}^2 = \sigma_e^2 + \sigma_n^2 + \sigma_{r_{00}}^2\)
\[
\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-u^2} du
\]

III. WRITE-VOLTAGE SIGNAL DESIGN

For a non-stationary flash memory channel, varying w.r.t PE cycles and data retention time \(T\), it is important to optimize and update the write-voltage levels for the optimal error-rate performance. For write-voltage optimization, it is assumed that the flash memory controller knows about the current PE count of individual memory blocks. The PE count history is mostly recorded inside the memory controller for performing the flash wear-leveling operation [32]. However, as the data retention time is generally difficult to predict at the time of cell programming, the retention time can either be set to zero (\(T = 0\), representing the early retention time) or to some large value (e.g. \(T = 1\) year, representing flash’s end-of-retention). Given these two flash channel parameters, the four distribution functions (11)-(14) can be evaluated, paving the way for finding the optimal write-voltage levels. In other words, given \(V_{\text{min}}\) and \(V_{\text{max}}\) (based on the device physics), and the PE count, we next want to assign the intermediate write-voltage levels \((V_1\) and \(V_2\)). To answer this question, we formulate the probability of error expression \(P_e\) for the flash channel. This expression can be defined using the error probability of individual symbols, \(P(e|s_p)\), given as
\[
P_e = \frac{1}{4} \left( P(e|s_{11}) + P(e|s_{10}) + P(e|s_{00}) + P(e|s_{01}) \right) \quad (15)
\]
where \(1/4\) models the equi-probable input data symbols. To express \(P(e|s_p)\), we need the decision boundaries \(R_1\) and \(R_3\) between adjacent distribution functions as shown in Fig. 1. Following this figure, we can define
\[
P(e|s_{11}) = p_{s_{11}} (v > R_1) \quad (16)
\]
\[
P(e|s_{10}) = p_{s_{10}} (v < R_1) + p_{s_{10}} (v > R_2) \quad (17)
\]
\[
P(e|s_{00}) = p_{s_{00}} (v < R_2) + p_{s_{00}} (v > R_3) \quad (18)
\]
\[
P(e|s_{01}) = p_{s_{01}} (v < R_3) \quad (19)
\]
These decision boundaries can also be considered as hard-decision levels on individual symbols. To get these decision boundaries, we equate the adjacent distribution functions and solve for the common intersecting point. Thus, for \(R_1\), \(R_2\) and \(R_3\), we solve the following three equations
\[
p_{s_{11}} (v = R_1) = p_{s_{10}} (v = R_1) \quad (21)
\]
\[
p_{s_{10}} (v = R_2) = p_{s_{00}} (v = R_2) \quad (22)
\]
\[
p_{s_{00}} (v = R_3) = p_{s_{01}} (v = R_3) \quad (23)
\]
By simplifying these expressions we get \(R_1 \in (V_{\text{min}}, V_1)\), \(R_2 \in (V_1, V_2)\) and \(R_3 \in (V_2, V_{\text{max}})\), respectively. Given the individual symbol error probabilities (16)-(19), we can re-write (15) in-terms of \(V_1\), \(V_2\), \(PE\) and \(T\) and optimize the desired parameters \(V_1\) and \(V_2\). In this section, we perform the voltage optimization for \(T = 0\). The end-of-retention optimization will be discussed in the following section. Thus, instead of keeping the write-voltage levels fixed throughout the operational lifetime of flash memory, we propose to adjust them according to the current channel condition such that the error probability is minimized. From the practical implementation perspective, the write-voltage levels can be pre-computed as a function of PE cycles and stored into a look-up table to preclude the runtime optimization complexity. To formulate the optimization problem, we define the objective function using (15) as
\[
(V_1^*, V_2^*) = \min_{(V_1, V_2)} P_e(V_1, V_2, PE, T = 0) \quad (20)
\]
This optimization function behaves as a convex function over \(V_1\) and \(V_2\) and can be solved by using any convex optimization technique. Since the write-voltage levels are optimized offline, the complexity of the chosen optimization algorithm is inconsequential. In this paper, we apply the gradient-descent (GD) [33] method to find the optimal \(V_1^*\) and \(V_2^*\) that yield the minimum \(P_e\). The GD method minimizes the objective function \(P_e\) by iteratively solving the following equation
\[
\begin{bmatrix} V_1^{(k+1)} \\ V_2^{(k+1)} \end{bmatrix} = \begin{bmatrix} V_1^{(k)} \\ V_2^{(k)} \end{bmatrix} - \eta \frac{\partial P_e}{\partial V_1} + \frac{\partial P_e}{\partial V_2} \begin{bmatrix} V_1^{(k)} \\ V_2^{(k)} \end{bmatrix} \quad (21)
\]
where \(\eta\) denotes the partial derivative, and \(k\) and \(\eta\) are the GD iteration count and step size, respectively.

The optimized write-voltage levels along with the corresponding minimum \(P_e\) values over different PE cycles are enumerated in Table I. To make comparison between the proposed optimal write-voltage scheme and the conventional
channel invariant (fixed) write-voltage scheme, we plot the $P_e$ function as shown in Fig. 3. Here, the solid curve represents the minimized $P_e$ values computed against channel invariant write-voltage levels fixed at $V_1 = 2.6$ and $V_2 = 3.2$. As expected, the proposed scheme yields lower error probability, particularly at low-to-medium PE count. In addition to improved error performance, the optimal write-voltage levels also help to reduce the ECC decoding latency, as discussed in the subsequent section.

IV. QUANTIZATION DESIGN FOR THE READ-VOLTAGE

In this section, we present a novel quantization scheme, using the voltage entropy function, to read flash memory cells. For 2-bit per cell flash memory, we require at least three quantization levels to detect four possible stored data symbols. However, when LDPC code is used as ECC for flash channel, we may need to perform high precision memory sensing for more accurate computation of log-likelihood-ratios (LLRs) for LDPC decoding. To achieve this, one straightforward approach is to set the quantization levels between $V_{\text{min}}$ and $V_{\text{max}}$ with equi-spaced separation $D_v$. Fig. 4 shows a pictorial representation of 6-level uniform quantization scheme, where each vertical dashed-line ($R_1$ to $R_6$) represent a particular quantization level. Since each flash symbol represents 2 bits, we compute the LLR values correspond to the most significant bit, $L_{\text{mb}}$, and the least significant bit, $L_{\text{sb}}$, positions.

Given the threshold voltage $v$, when $R_{n-1} \leq v < R_n$ for $n = 1, 2, 3, 4, 5, 6, 7$, where $R_0 = -\infty$ and $R_7 = +\infty$, we have

$$L_{\text{mb}} = \log \frac{\int_{R_{n-1}}^{R_n} \{p_{00}(v) + p_{01}(v)\} dv}{\int_{R_{n-1}}^{R_n} \{p_{10}(v) + p_{11}(v)\} dv}$$

(22)

$$L_{\text{sb}} = \log \frac{\int_{R_{n-1}}^{R_n} \{p_{00}(v) + p_{10}(v)\} dv}{\int_{R_{n-1}}^{R_n} \{p_{01}(v) + p_{11}(v)\} dv}$$

(23)

However, for a given memory sensing precision (e.g. 6-level, 9-level, 12-level), the uniform quantization scheme is shown to be ineffective as compared to the non-uniform quantization schemes [14]. The objective of a non-uniform quantization scheme is to set the memory sensing levels closer to the erasure (intersection) regions where adjacent distribution functions are overlapped. Since the symbol error probability is dominant in this region, it requires more accurate memory sensing. Furthermore, for a non-uniform quantization scheme, it is crucial to identify the optimum width of erasure region so that the resultant soft-information (LLR values) achieves the best decoder error performance. For this purpose, we first define the entropy of cell’s threshold voltage, $H(v)$, given by

$$H(v) = \sum_i \left[ \frac{p_i(v)}{\sum_i p_i(v)} \log_2 \left( \frac{\sum_i p_i(v)}{p_i(v)} \right) \right]$$

(24)

where $i \in \{11, 10, 00, 01\}$. An example of voltage entropy function is illustrated in Fig. 5. It can be observed that the voltage entropy is only dominant within the erasure region, denoted as high-entropy-region. This should be the targeted memory sensing region as it incurs high error probability. Therefore, we set the quantization levels closer to the erasure region, instead of setting them uniformly from $V_{\text{min}}$ to $V_{\text{max}}$. With 3 erasure regions in-place, we require at least 6 quantization levels in order to enclose them within the quantization boundary. To this end, we define a parameter $\theta \in [0, 1]$ to select the width of each erasure region. To be precise, we set each quantization level such that

$$H(R_n) = \theta$$

(25)

for $n = 1, 2, 3, 4, 5, 6$. In other words, the memory sensing levels are designed where the voltage entropy is equal to $\theta$. By varying the entropy parameter $\theta$, we can obtain the desired width for erasure regions. We refer to this memory sensing scheme as entropy-based quantization scheme. Besides, another well-known non-uniform quantization scheme for MLC flash is reported in [17], where the quantization levels are obtained by maximizing the mutual-information (MMI) between the input and output of flash channel. This scheme involves solving a multi-variable optimization problem to get the quantization levels. However, with MMI quantization, it is not possible to select the width of erasure regions. This constraint affects the resultant error-rate performance of LDPC.

TABLE I: Optimized write-voltage levels over PE cycles.

<table>
<thead>
<tr>
<th>PE</th>
<th>$V_1^*$</th>
<th>$V_2^*$</th>
<th>$P_e$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>2.77</td>
<td>3.35</td>
<td>7.15 x 10^{-4}</td>
</tr>
<tr>
<td>2000</td>
<td>2.75</td>
<td>3.34</td>
<td>0.0010</td>
</tr>
<tr>
<td>5000</td>
<td>2.69</td>
<td>3.31</td>
<td>0.0023</td>
</tr>
<tr>
<td>10000</td>
<td>2.61</td>
<td>3.27</td>
<td>0.0072</td>
</tr>
<tr>
<td>15000</td>
<td>2.55</td>
<td>3.24</td>
<td>0.0115</td>
</tr>
</tbody>
</table>

Fig. 3: Comparison of channel error probability, $P_e$, between the proposed write-voltage optimization scheme and the conventional channel invariant write-voltage scheme: solid curve $(V_1^*, V_2^*$ from Table I), dotted curve $(V_1 = 2.6, V_2 = 3.2)$. 
Mutual Information (bits/cell) 

θ
level) quantization schemes. The entropy parameter is set to

θ
decoder [36], with maximum iteration count, set to 25. We apply the proposed entropy-based quantization scheme (6-level) and compare with MMI (6-level) [17] and uniform (12-level) quantization schemes. The entropy parameter is set to θ = 0.35 as it leads to optimum error-rate performance.

We first notice that, contrary to MMI quantization, the entropy-based scheme does not maximize the mutual information as shown in Fig. 6. Next, we plot the frame-error-rate (FER) curves for LDPC 4K-code and 8K-code with retention time set to zero (T = 0) as shown in Fig. 7. In this figure, we show the error performance of MMI and uniform quantization schemes by using both channel invariant (conventional) and channel optimized (proposed) write-voltage levels. We observe that the proposed entropy-based quantization outperforms the former schemes. Furthermore, as stated earlier, the uniform quantization scheme is not very effective as the 12-level quantization is, surprisingly, worse than the 6-level non-uniform quantization schemes.

In order to simulate the flash memory under data retention scenario, we need to estimate the voltage distribution functions prior to memory sensing operation. This is required because the retention noise shifts the voltage distribution functions over increasing retention time T. In this work, we rely upon the prior-art schemes [37], [38] to estimate the shifted distribution functions. Based on that, we compute new set of entropy-based quantization levels and derive the corresponding LLRs using (22) and (23). In Fig. 8, we illustrate the frame-error-rate (FER) performance of LDPC 4K-code, where the retention time is set to 6 months, 12 months and 24 months, respectively. It is evident that the proposed entropy-based quantization scheme has superior error-rate performance as compared to the MMI scheme, substantiating the effectiveness of the proposed quantization under data retention scenario.

It should be noted that, in this paper, we strive to optimize the 6-level quantization scheme. This is because the channel capacity with 6-level quantization registers a big jump from 3-level quantization (hard decision), while further increase in the quantization levels registers diminishing returns towards the infinite-precision limit, as shown in Fig. 9.
observe that the threshold voltage range is split into quantization levels, as depicted in Fig. 10. In this figure, we posed entropy-based quantization scheme over the MMI quantization scheme [17]. We first draw attention to the fact that the improvement in error performance achieved by using the proposed entropy (proposed) and MMI [17] quantization schemes, setting retention time ($T$) at 6, 12 and 24 months.

V. CONTROLLED ERASURE DECODING USING ENTROPY BASED QUANTIZATION

In this section, we investigate the reason behind the improvement in error performance achieved by using the proposed entropy-based quantization scheme over the MMI quantization scheme [17]. We first draw attention to the fact that the entropy-based quantization levels (given by the optimal value of $\theta$) tend to produce wider erasure regions than the MMI quantization levels, as depicted in Fig. 10. In this figure, we observe that the threshold voltage range is split into 3 erasure regions $E_1$, $E_2$ and $E_3$, and 4 data regions $D_1$, $D_2$, $D_3$ and $D_4$, respectively. Here, it must be realized that adjusting the width of erasure regions affects the error performance of the code, as shown in Fig. 11. This plot shows the error performance of LDPC 4K-code over a range of $\theta$ values for PE = 21K. The width of erasure regions corresponding to different $\theta$ values are shown in Table II. Observing Fig. 11 and Table II reveals that if the erasure region is too narrow, the decoder will not be able to determine sufficient erasures; while if the erasure region is too wide, it will get too many erased symbols which exceed the code's error correcting capability; and the entropy parameter $\theta$ allows the erasure region width to be optimized. With MMI quantization, it is not possible to alter the width of erasure region as it is optimized (fixed) over a given channel (for a fixed channel noise), irrespective of the type of code and decoding algorithm used. Fig. 11 also marks the entropy parameter value corresponding to the MMI quantization levels, which clearly does not coincide with the minimum error-rate point. This is because the MMI quantization optimization attempts to achieve zero error probability while assuming infinite channel code length, but in practical systems this assumption is violated as only finite block-length codes can be adopted. On the contrary, in entropy-based quantization, changing the value of $\theta$ influences the symbol error probability, $P_{\text{error}}$, and the symbol erasure probability, $P_{\text{erasure}}$, of the flash channel, and consequently alters the magnitude of input LLRs
used for BP decoding. Here, $P_{\text{error}}$ refers to the uncoded error probability computed in 4 data regions and $P_{\text{erasure}}$ refers to the erasure probability computed in 3 erasure regions. These two quantities are mathematically written as

$$P_{\text{error}} = \int_{v \in \mathcal{E}_1} (p_{s_{10}}(v) + p_{s_{00}}(v) + p_{s_{01}}(v)) \, dv$$

$$+ \int_{v \in \mathcal{E}_2} (p_{s_{11}}(v) + p_{s_{00}}(v) + p_{s_{01}}(v)) \, dv$$

$$+ \int_{v \in \mathcal{E}_3} (p_{s_{11}}(v) + p_{s_{10}}(v) + p_{s_{01}}(v)) \, dv$$

$$+ \int_{v \in \mathcal{E}_4} (p_{s_{11}}(v) + p_{s_{10}}(v) + p_{s_{00}}(v) + p_{s_{01}}(v)) \, dv$$

(26)

$$P_{\text{erasure}} = \int_{v \in \mathcal{E}_1} (p_{s_{11}}(v) + p_{s_{10}}(v) + p_{s_{00}}(v) + p_{s_{01}}(v)) \, dv$$

$$+ \int_{v \in \mathcal{E}_2} (p_{s_{11}}(v) + p_{s_{10}}(v) + p_{s_{00}}(v) + p_{s_{01}}(v)) \, dv$$

$$+ \int_{v \in \mathcal{E}_3} (p_{s_{11}}(v) + p_{s_{10}}(v) + p_{s_{00}}(v) + p_{s_{01}}(v)) \, dv$$

(27)

We evaluate and plot these two expressions for different PE cycles in Fig. 12. In this figure, compared to MMI, the proposed entropy-based quantization reduces the symbol error probability by having a higher erasure probability. It should be noted that the LLR values associated with erasure regions are the least reliable LLRs and therefore reset to zero (erased). To be precise, the LSB LLRs correspond to $\mathcal{E}_1$ and $\mathcal{E}_3$, and the MSB LLRs correspond to $\mathcal{E}_2$ are erased. Thus, with entropy-based quantization scheme, we perform a controlled erasure decoding by means of optimizing the parameter $\theta$, in such a way that some LLR values are erased to ensure that the symbol error probability is reduced. This adjustment between the symbol error and symbol erasure probability helps the entropy scheme to perform better than the MMI scheme.

![Fig. 11: BER/FER performance of LDPC 4K-code over varying entropy parameter $\theta$ at PE = 21K: values of $\theta$ corresponding to optimal error-rate performance and MMI quantization levels are marked with vertical arrows.](image)

![Fig. 12: Comparison of symbol error probability, $P_{\text{error}}$, and symbol erasure probability, $P_{\text{erasure}}$, between the entropy (proposed) and MMI [17] quantization schemes: entropy quantization produces lower $P_{\text{error}}$ at the cost of higher $P_{\text{erasure}}$.](image)

As a future work, it is of great interest to analytically evaluate the optimum value of $\theta$ for which the decoded error probability is minimized. Apart from entropy function, it is compelling to investigate other non-linear functions to see if the error performance can be further improved.

VI. WRITE-VOLTAGE OPTIMIZATION FOR FASTER DECODING CONVERGENCE

In the previous section, we mainly analyze the improvement in error-rate performance that comes as a result of using optimal write-voltage signals, as shown in Fig. 3 and Fig. 7. It should be noted that the need for write-voltage optimization is not critical in the initial state of flash memory as compared to its end-of-life (EOL). This is because the error-rate in the initial state typically does not exceed the error correction capability of ECC decoder. However, as the LDPC code is becoming the mainstream ECC in flash memory controller, its long decoding latency starts to deteriorate the system performance. In this situation, the proposed optimization method to reduce the error-rate, and consequently to reduce the decoding latency, becomes important. It other words, we can improve the decoding convergence in the initial state of flash memory.

<table>
<thead>
<tr>
<th>$\theta$</th>
<th>$\mathcal{E}_1$</th>
<th>$\mathcal{E}_2$</th>
<th>$\mathcal{E}_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.20</td>
<td>0.336</td>
<td>0.221</td>
<td>0.220</td>
</tr>
<tr>
<td>0.25</td>
<td>0.303</td>
<td>0.202</td>
<td>0.202</td>
</tr>
<tr>
<td>0.30</td>
<td>0.277</td>
<td>0.185</td>
<td>0.184</td>
</tr>
<tr>
<td>0.35</td>
<td>0.253</td>
<td>0.171</td>
<td>0.170</td>
</tr>
<tr>
<td>0.40</td>
<td>0.233</td>
<td>0.158</td>
<td>0.156</td>
</tr>
<tr>
<td>0.45</td>
<td>0.214</td>
<td>0.146</td>
<td>0.144</td>
</tr>
<tr>
<td>0.50</td>
<td>0.197</td>
<td>0.135</td>
<td>0.133</td>
</tr>
<tr>
<td>0.55</td>
<td>0.181</td>
<td>0.124</td>
<td>0.122</td>
</tr>
<tr>
<td>0.60</td>
<td>0.166</td>
<td>0.114</td>
<td>0.112</td>
</tr>
</tbody>
</table>
by optimizing the write-voltage levels. It should be noted that the total read latency of NAND flash memory based devices is composed of firmware processing, memory sensing, and DMA (data transfer from controller to NAND flash memory) times, and the most time consuming process is the memory sensing operation. The LDPC decoding latency in included in the DMA time.

In Fig. 13, we plot the BER curves versus the maximum iteration count, $I_{\text{max}}$, for different PE cycles. It can be observed that the decoding latency is reduced by up to 16% in the initial state, even though the BER at these early states are low and typically have enough margins from ECC failures, and up to 22% in the EOL. Therefore, in comparison with conventional flash memory system where the write-voltage levels are fixed, our approach is more effective in-terms of error performance and decoding latency for both early and late stages of flash memory usage.

VII. WRITE-VOLTAGE OPTIMIZATION FOR FLASH END-OF-RETENTION (EOR) TIME

Previously, we performed the write-voltage optimization for the early retention time of flash memory, by setting the retention time to zero. However, in this section, we address the write-voltage scheme optimized for the flash memory’s end-of-retention (EOR) time, and compare between the two voltage design schemes. In this direction, we set the retention time to some large value and then minimize the objective function (20). As a case study, we set the retention time to 12 months and optimize the write-voltage levels over varying PE cycles. The optimization for EOR is recommended if the flash data is expected to be stored for a long period of time without reading/re-writing. We use $T_{\text{write}} = 0$ and $T_{\text{write}} = 12$-months to distinguish between optimizing the write-voltage schemes for early retention and end-of-retention times, respectively.

In Fig. 14, we plot the flash memory endurance (PE cycles) versus the data retention time ($T$) for the two voltage design schemes, $T_{\text{write}} = 0$ and $T_{\text{write}} = 12$-months, keeping the FER fixed at $10^{-3}$ and $10^{-5}$. We observe that if the data retention time is relatively short, under 4 months, the write-voltage scheme optimized for early retention time outperforms the EOR optimization scheme by providing longer flash memory endurance. Conversely, as the data retention time increases, the write-voltage scheme optimized for EOR time yields better endurance performance. In summary, the problem of write-voltage optimization for MLC flash is a design trade-off between the achievable endurance performance and the expected retention time of stored flash data.

VIII. MINIMUM MEMORY SENSING PRECISION FOR LDPC DECODER CONVERGENCE

Until now, we perform simulations using 6-level memory sensing precision. Since the flash memory channel is non-stationary, even the number of quantization levels should not be fixed but should instead be optimized in conjunction with soft LDPC decoding. For instance, it is intuitive to expect that when flash memory is relatively fresh and has gone through small number of PE cycles, we can use fewer quantization levels, say 3-level. However, as the channel noise variance increases due to larger number of PE cycle, we must switch to higher quantization levels, e.g. 6-level. This adaptive memory precision approach has been reported in [20] in which the author has used the minimum quantization levels (1-level for 1-bit per cell, 3-levels for 2-bit per cell flash memory, and so on) to initiate the LDPC decoder and progressively increased the memory precision if the lower-precision sensing fails to produce successful LDPC code-word. However, this method is not optimal in terms of decoding latency since the flash read-out data is decoded for different quantization levels until the code-word is successfully produced.
we should not always start the decoding process from the minimum quantization levels, but should select the memory sensing precision based on the knowledge of the current flash channel status (PE count).

In this direction, we propose to select the minimum required number of quantization levels based on the extrinsic information transfer (EXIT) curves [39]. For a given LDPC code, we plot the EXIT diagram using Monte-Carlo simulations for different quantization levels over varying PE cycles and select the smallest number of read levels which successfully transfer the mutual-information curve. This EXIT curve based quantization selection approach seems to be more reliable as the simulations take into account the effect of LDPC finite block-length, providing more accurate prediction of the error-rate performance. Alternatively, relying on the channel capacity curve to choose the number of quantization levels may not be accurate enough because of its infinite block-length assumption. In Fig. 16, using the same LDPC 4K-code, we plot the EXIT curves for 3-level ($\theta = 1$) and 6-level ($\theta = 0.35$) quantization schemes for different PE cycles, keeping the retention time fixed at $T = 0$. For 3-level quantization, we may notice that beyond 14K PE cycles, the mutual-information curve is not successfully transferred. At this point, we observe in Fig. 15 that the BER is around $10^{-6}$. Therefore, if we want to operate the flash memory beyond 14K PE cycles, we must switch to higher precision memory sensing. This can be verified from Fig. 16, which shows that 6-level quantization can be employed between 14K to 20K PE cycles, beyond which we again require higher quantization levels. Thus, EXIT analysis can be used as a reliable tool to choose the required number of read-voltage levels.

**IX. Conclusion**

This paper investigates the optimization of read and write voltage signals for MLC NAND flash memory. Based on a non-stationary flash channel model, considering programming noise, random telegraph noise, cell-to-cell interference and data retention noise, the write-voltage levels are optimized by minimizing the channel error probability. The trade-off between write-voltage optimization based on early retention time versus end-of-life retention time of the stored data is discussed. The proposed write-voltage optimization scheme is validated to have superior error-rate performance over the conventional channel-invariant write-voltage scheme.

Besides, a novel read-voltage quantization scheme based on the voltage entropy function is proposed to find the right balance between the flash channel symbol error and symbol erasure probability through controlled erasure decoding. This quantization scheme enables the LDPC decoder to achieve soft BP decoding performance improvement over prior-art quantization schemes. By virtue of both write and read voltage optimization, the overall error-rate performance and decoding convergence are improved. Finally, a method to minimize the number of quantization levels based on EXIT curve is proposed to enable the flash memory controller to reduce the memory sensing latency while guaranteeing the LDPC decoder convergence.

**References**


Chaudhry Adnan Aslam received his B.E. degree from Mehran University of Engineering and Technology Pakistan and M.S. degree from the University of Southern California (USC) Los Angeles, CA, in 2005 and 2009, respectively. He is currently pursuing the Ph.D. degree from the School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore. His research interests include coding theory and signal processing for data storage and wireless communications systems.

Yong Liang Guan obtained his Ph.D. from the Imperial College of London, UK, and Bachelor of Engineering with first class honors from the National University of Singapore. He is now an Associate Professor at the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. His research interests broadly include modulation, coding and signal processing for communication systems, and information security systems. His homepage is at http://www3.ntu.edu.sg/home/eylguan/index.htm.
Kui Cai received B.E. degree in information and control engineering from Shanghai Jiao Tong University, Shanghai, China, M.Eng degree in electrical engineering from National University of Singapore, and joint Ph.D. degree in electrical engineering from Technical University of Eindhoven, The Netherlands, and National University of Singapore. Currently she is an Associate Professor with Singapore University of Technology and Design (SUTD). Before joining SUTD, she had been with Data Storage Institute (DSI), Singapore, since 1999, where she was the Program Leader of non-volatile memory (NVM) coding and signal processing. Cai Kui is a senior member of IEEE and the Vice-Chair (Academia) of IEEE Communications Society, Data Storage Technical Committee (DSTC). She is the recipient of 2008 IEEE Communications Society Best Paper Award in Coding and Signal Processing for Data Storage. Her research interests include coding theory, communication theory, and signal processing for various data storage systems and digital communications.