Decoding Flash Memory with Progressive Reads and Independent vs. Joint Encoding of Bits in a Cell

Nathan Wong*, Ethan Liang†, Haobo Wang†, Sudarsan V. S. Ranganathan‡, and Richard D. Wesel†
*Physical Optics Corporation, 1845 W 205th St, Torrance, CA 90501
†Department of Electrical and Computer Engineering, University of California, Los Angeles, Los Angeles, CA 90095
‡Institute for Data, Systems, and Society, Massachusetts Institute of Technology, Cambridge, MA 02142
Emails: nwong@poc.com, emliang@ucla.edu, whb12@ucla.edu, sudarvsr@mit.edu, wesel@ucla.edu

***Abstract*—This paper develops a paradigm for optimizing progressive reads for flash memory cells that maximize the conditional mutual information (MI) given previous reads and shows that some progressive reads provide substantially more MI than others. We study two flash storage techniques: 1) the common practice of independently encoding each bit of a cell into a separate codeword and 2) jointly encoding all the bits in the cell into the same codeword. We quantify the MI gap between joint and independent encoding and show that this gap becomes negligible when progressive reads are available. The paper provides LDPC simulations that confirm the MI analysis.**

I. Introduction and Motivation

The process of writing to the cells in a page, reading the page possibly multiple times, and then erasing the page is called the Program/Erase (P/E) cycle. Each P/E cycle degrades the flash read channel so that read distortions become more severe over time. To improve performance as the channel degrades, progressive reads add enhanced precision only when the decoding based on initial read(s) fails [1].

Wang *et al.* [2], [3] present a framework to optimize the values of read thresholds for enhanced precision beyond hard decoding to maximize the mutual information (MI). However, these papers do not address the progressive nature of enhanced precision. Instead, they assume that all the thresholds will be used to read every cell. As discussed above, practical flash systems perform the enhanced-precision reads progressively. Below, we describe how to efficiently determine the best thresholds in the context of progressive reads.

As a separate matter, practical flash systems minimize the number of read processes required to access a page by having pages store each bit of user information in a different cell. In this way, the bits of the page can all be read from their corresponding cells simultaneously [4]. In flash systems where cells hold more than one bit of information, this paper defines "independent encoding" to refer to systems where the bits in any particular cell are encoded independently of each other, i.e. in different codewords. Independent encoding is used by practical flash devices because it allows each bit in a page to reside in a different cell.

We define "joint encoding" to refer to an approach considered in the academic literature, where all the bits in the cell are jointly encoded in the same codeword [5]. Joint encoding provides an information-theoretic benefit. Only the scenario of joint encoding is considered in the enhanced precision technique described in [2], [3], leaving unexplored the practically important case of independent encoding.

A. Contributions

This paper extends the paradigm of [2], [3] to progressive reads and to the scenario of independent encoding. The progressive nature of the enhanced precision actually simplifies the optimization process, which can prove useful for either online computation or pre-computing threshold changes based on offline modeling. The paper shows that some reads of progressive enhanced precision provide substantially more MI than others. We optimize the ordering of progressive reads to maximize the MI provided at each stage.

This paper also addresses the additional benefit made possible by joint encoding. We examine joint encoding both for traditional hard decoding and for enhanced precision decoding with progressive reads. We show that the benefit provided by joint encoding is significantly diminished when progressive reads are available. This paper provides simulations of low-density parity-check (LDPC) codes confirming that frame error rate (FER) behavior corresponds to the information-theoretic analysis, i.e. more MI indicates a lower FER.

B. Organization

The rest of this paper proceeds as follows: Section II presents a simplified model for noise and degradation on the flash channel. Section III presents an information-theoretic approach to optimize progressive reads by maximizing the conditional MI for a system where the bits in a cell are encoded independently. Section IV considers the benefit of a system where all the bits in a cell are jointly encoded as compared to systems where each bit in the cell is encoded independently. This benefit is significantly diminished when progressive reads are available. Section V presents LDPC code FER simulations that confirm the information-theoretic analysis. Section VI presents our conclusions.

This research is supported by National Science Foundation (NSF) grant CCF-1618272 and Physical Optics Corporation (POC). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect views of the NSF or POC.

II. Flash Channel Noise and Degradation Model

Let X be a discrete random variable describing the voltage level that is written to the cell. Let Y be the resulting voltage level after degradation by the flash channel. The voltage Y is not directly accessible to the decoder. Rather, a sense-amp comparator provides one or more single-bit measurements, each of which answers a question of the form "Is $Y > T$?", where T is a voltage threshold.

For multi-level cell (MLC), X has four possible write values for the voltage level. Each value has an associated two-bit label, which represents the stored information. Our analysis generalizes to triple-level cells (TLCs) with eight levels and quad-level cells (QLCs) with 16 levels, but this paper analyzes only MLC cells to simplify the exposition.

X_{LSB} denotes the least significant bit (LSB) of the two-bit MLC label. X_{MSB} denotes the most significant bit (MSB). In MLC, LSB bits and MSB bits are used to store separate pages of memory. This independent encoding allows the decoder to perform only one read for hard decoding of the LSB page and two reads for the MSB page. Section IV explores joint encoding of the LSB and MSB bits, which requires three reads for hard decoding of the jointly encoded page.

The flash read channel includes a variety of impairments including Programming Noise, Inter-Cell Interference, Wear-out, and Retention Loss [6]. The noise is input-dependent, and the flash read channel degrades over time as the electrons moving in and out of the floating gate degrade its ability to retain charge and resist inter-cell interference. This paper uses a simplified model of the MLC flash read channel that facilitates theoretical analysis and produces simulation results that are easily replicated. This model still captures the essential behaviors studied in [6] including the input-dependent nature of flash noise and degradation over time.

We use an input-dependent time-varying Gaussian model that captures the three key aspects MLC flash noise: 1) the erased state has a variance that is larger than the programmed states because the feedback control loop associated with programming reduces voltage variation of the programmed states, 2) the noise variance increases with the number of P/E cycles, and 3) the voltage levels decrease due to retention loss, and this decrease becomes larger as the number of P/E cycles increases.

Our model is $Y = X + N(t, X)$ where N is a Gaussian random variable whose mean $\mu(X, t)$ and whose variance $\sigma^2(X, t)$ depends on both the number t of P/E cycles and the input X. The specific time-varying functions for the means and variances are computed using Model 1 described in [6], are shown in Fig. 1, and the values can be found in [7].

This characterization is meant to provide an example of the behavior in general rather than accurately model a specific flash device. Each flash device would have its own characterization according to this model. The variance $\sigma^2(X, t)$ depends on X only in that $\sigma^2(X, t)$ for the erased voltage v_e is larger and increases more slowly than $\sigma^2(X, t)$ for a programmed voltage, i.e.,

$$\sigma(X,t) = \begin{cases} \sigma_e(t) & \text{if } X = v_e \\ \sigma_p(t) & \text{if } X \neq v_e \end{cases} . \tag{1}$$

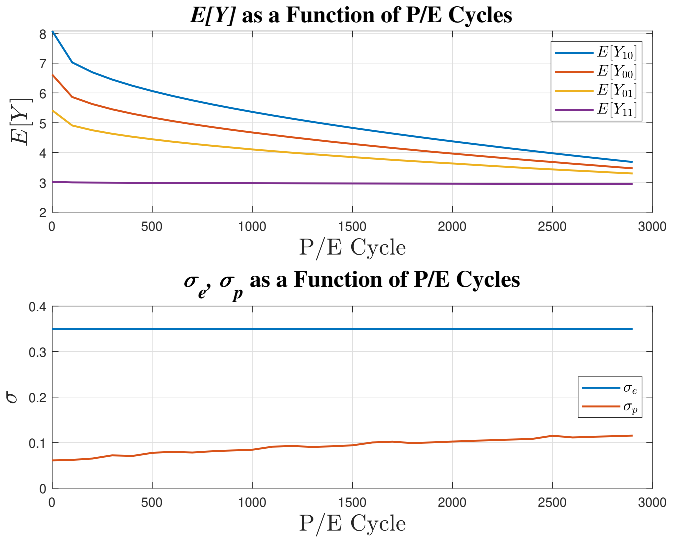

Fig. 1. $E[Y]$ (top) and σ_e and σ_p (bottom) as a function of the number of P/E cycles. $E[Y]$ is the mean voltage after distortion by the flash channel for each of the four MLC levels. Gaussian noise standard deviation σ_e is for the erased voltage level and σ_p is for the programmed voltage levels.

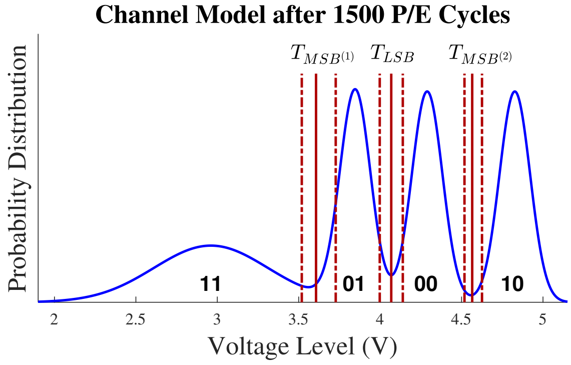

Fig. 2. The probability distribution of voltage level for a four-level flash device after 1500 P/E cycles using the noise and degradation model of this paper. Also shown are read thresholds optimized as described in Sec. III for the three initial reads (solid) and six progressive reads (dashed).

Figure 2 shows the probability distribution of voltage level for our model after 1500 (P/E) cycles. Our input-dependent time-varying Gaussian model can be applied to flash cells with any number of levels and to any specific noise model as long as the distribution can be estimated using, for example, the histogram techniques described in [6], [8], [9] and modeled with input-dependent Gaussians.

For each group of three thresholds in Figure 2, the solid red line is an initial threshold that is used to make the initial hard decision on either the LSB or MSB as indicated by the subscript (e.g. $T_{\text{LSB}}^{\text{initial}}$). The dotted red lines to the left and right of each solid line (e.g. $T_{\text{LSB}}^{\text{left}}$ and $T_{\text{LSB}}^{\text{right}}$) indicate the progressive read thresholds that provide enhanced precision.

III. Optimizing Progressive Reads

MLC flash as described in [1], [4] and [10] encodes the MSB bits and LSB bits on separate pages and uses progressive reads for enhanced precision. Initially, only the hard-decoding reads are performed. If those reads are not sufficient to decode the channel code, e.g. an LDPC code, then additional reads are performed to provide enhanced precision.

In [3], information-theoretic optimization of thresholds for enhanced precision was developed assuming that the LSB and MSB bits of a cell are jointly encoded in the same codeword and assuming all the enhanced precision bits are read for every cell. This section develops a paradigm for optimizing progressive reads for standard flash systems in which the LSB and MSB bits are stored on separate pages (and thus encoded in separate codewords) and progressive reads are performed only when the initial reads did not result in successful decoding.

A. Hard Decoding of MLC bits

Gray coding labels the values of each MLC voltage level [4], [10], [11] as shown in Fig. 2. The solid lines in the figure represent the initial read thresholds, and the dashed lines represent the left and right progressive reads that provide enhanced precision.

For the LSB page, the initial read at the threshold $T_{\text{LSB}}^{\text{initial}}$ determines the binary value of $Y_{\text{LSB}}^{\text{initial}}$ as follows:

$$Y_{\text{LSB}}^{\text{initial}} = \begin{cases} 1 & \text{when } Y \leq T_{\text{LSB}}^{\text{initial}} \\ 0 & \text{when } Y > T_{\text{LSB}}^{\text{initial}} \end{cases} . \tag{2}$$

For the case of independent encoding, we select $T_{\text{LSB}}^{\text{initial}}$ to maximize the MI $I(X_{\text{LSB}}; Y_{\text{LSB}}^{\text{initial}})$. The associated hard-decoding estimate of X_{LSB} is $\hat{X}_{\text{LSB}} = Y_{\text{LSB}}^{\text{initial}}$. This one-parameter optimization can be accomplished with Newton's method as in [3] and can be applied to more complicated models than our input-dependent time-varying Gaussian model such as those derived from the histogram techniques described in [6], [9].

For the MSB page, two initial reads at the thresholds $T_{\text{MSB}^{(1)}}^{\text{initial}}$ and $T_{\text{MSB}^{(2)}}^{\text{initial}}$ determine the binary values of $Y_{\text{MSB}^{(1)}}^{\text{initial}}$ and $Y_{\text{MSB}^{(2)}}^{\text{initial}}$ as follows:

$$Y_{\text{MSB}^{(1)}}^{\text{initial}} = \begin{cases} 1 & \text{when } Y \leq T_{\text{MSB}^{(1)}}^{\text{initial}} \\ 0 & \text{when } Y > T_{\text{MSB}^{(1)}}^{\text{initial}} \end{cases} , \tag{3}$$

$$Y_{\text{MSB}^{(2)}}^{\text{initial}} = \begin{cases} 0 & \text{when } Y \leq T_{\text{MSB}^{(2)}}^{\text{initial}} \\ 1 & \text{when } Y > T_{\text{MSB}^{(2)}}^{\text{initial}} \end{cases} . \tag{4}$$

For the case of independent encoding we select $T_{\text{MSB}^{(1)}}^{\text{initial}}$ and $T_{\text{MSB}^{(2)}}^{\text{initial}}$ to maximize the MI $I\left(X_{\text{MSB}}; Y_{\text{MSB}^{(1)}}^{\text{initial}}, Y_{\text{MSB}^{(2)}}^{\text{initial}}\right)$. The hard-decoding estimate of X_{MSB} is

$$\hat{X}_{\text{MSB}} = \begin{cases} 1 & \text{when } Y_{\text{MSB}^{(1)}}^{\text{initial}} = 1 \text{ or } Y_{\text{MSB}^{(2)}}^{\text{initial}} = 1 \\ 0 & \text{otherwise} \end{cases} . \tag{5}$$

A practical approach to jointly optimize the two thresholds is to begin with $T_{\text{MSB}^{(1)}}^{\text{initial}}$ and $T_{\text{MSB}^{(2)}}^{\text{initial}}$ at the cross-over points of the two conditional distributions and perform Newton's method to adjust $T_{\text{MSB}^{(1)}}^{\text{initial}}$ while holding $T_{\text{MSB}^{(2)}}^{\text{initial}}$ constant and vice versa until the process (rapidly) converges.

In practice, implementation of the hard decision thresholds aims to minimize hard decision error probability [12], [13]. Such thresholds minimize the sum of the crossover probabilities from 0 to 1 and 1 to 0. In contrast, our paradigm determines thresholds by maximizing MI. For hard decoding, these two paradigms yield essentially identical thresholds.

B. Optimizing Progressive Thresholds for the LSB Page

Each progressive read provides an additional bit indicating whether the voltage Y is above or below the new threshold. (e.g. the bit $Y_{\text{LSB}}^{\text{left}}$ indicates whether V is above or below $T_{\text{LSB}}^{\text{left}}$.) Now we consider optimizing the position of T^{left} and T^{right} given that T^{initial} has already been set to optimize the initial read process for hard decoding. The threshold positions for both LSB and MSB pages are illustrated in Figure 2.

For the LSB page we seek $T_{\text{LSB}}^{\text{left}}$ and $T_{\text{LSB}}^{\text{right}}$ to maximize $I\left(X_{\text{LSB}}; Y_{\text{LSB}}^{\text{initial}}, Y_{\text{LSB}}^{\text{left}}, Y_{\text{LSB}}^{\text{right}}\right)$ where $Y_{\text{LSB}}^{\text{initial}}$ has already been read using $T_{\text{LSB}}^{\text{initial}}$. The chain rule for MI decomposes this into the MI from the initial read and the additional MI provided by the progressive reads:

$$\begin{aligned} & I\left(X_{\text{LSB}}; Y_{\text{LSB}}^{\text{initial}}, Y_{\text{LSB}}^{\text{left}}, Y_{\text{LSB}}^{\text{right}}\right) && (6) \\ & \quad = I(X_{\text{LSB}}; Y_{\text{LSB}}^{\text{initial}}) + I(X_{\text{LSB}}; Y_{\text{LSB}}^{\text{left}}, Y_{\text{LSB}}^{\text{right}} | Y_{\text{LSB}}^{\text{initial}}) . && (7) \end{aligned}$$

The progressive-read term $I(X_{\text{LSB}}; Y_{\text{LSB}}^{\text{left}}, Y_{\text{LSB}}^{\text{right}} | Y_{\text{LSB}}^{\text{initial}})$ can be further decomposed as follows:

$$\begin{aligned} & P(Y_{\text{LSB}}^{\text{initial}} = 1) \cdot I(X_{\text{LSB}}; Y_{\text{LSB}}^{\text{left}} | Y_{\text{LSB}}^{\text{initial}} = 1) && (8) \\ & \quad + P(Y_{\text{LSB}}^{\text{initial}} = 0) \cdot I(X_{\text{LSB}}; Y_{\text{LSB}}^{\text{right}} | Y_{\text{LSB}}^{\text{initial}} = 0) , && (9) \end{aligned}$$

because once $Y_{\text{LSB}}^{\text{initial}}$ is established by the initial read, one of the two progressive reads becomes deterministic. The threshold optimization thus becomes two decoupled optimization problems in which $T_{\text{LSB}}^{\text{left}}$ is selected to optimize $I(X_{\text{LSB}}; Y_{\text{LSB}}^{\text{left}} | Y_{\text{LSB}}^{\text{initial}} = 0)$ and $T_{\text{LSB}}^{\text{right}}$ is selected to optimize $I(X_{\text{LSB}}; Y_{\text{LSB}}^{\text{right}} | Y_{\text{LSB}}^{\text{initial}} = 1)$. As in [3], a Newton's method can quickly identify the optimal thresholds. The contribution of the two progressive reads is very similar after about 500 P/E cycles and progressive reads are not necessary below 500 P/E cycles so these two progressive reads are equally helpful.

C. Optimizing Progressive Thresholds for the MSB Page

For the MSB page, we seek $T_{\text{MSB}^{(1)}}^{\text{left}}$, $T_{\text{MSB}^{(1)}}^{\text{right}}$, $T_{\text{MSB}^{(2)}}^{\text{left}}$, and $T_{\text{MSB}^{(2)}}^{\text{right}}$ to maximize the overall mutual information. The chain rule for MI decomposes the overall mutual information into the MI from the two initial reads and the additional MI provided by the progressive reads:

$$\begin{aligned} & I(X_{\text{MSB}}; Y_{\text{MSB}^{(1,2)}}^{\text{initial}}, Y_{\text{MSB}^{(1,2)}}^{\text{left}}, Y_{\text{MSB}^{(1,2)}}^{\text{right}}) = && (10) \\ & I(X_{\text{MSB}}; Y_{\text{MSB}^{(1,2)}}^{\text{initial}}) + I(X_{\text{MSB}}; Y_{\text{MSB}^{(1,2)}}^{\text{left}}, Y_{\text{MSB}^{(1,2)}}^{\text{right}} | Y_{\text{MSB}^{(1,2)}}^{\text{initial}}), \end{aligned}$$

where $Y_{\text{MSB}^{(1,2)}}^{\text{initial}}$ is the pair $\{Y_{\text{MSB}^{(1)}}^{\text{initial}}, Y_{\text{MSB}^{(2)}}^{\text{initial}}\}$ for concise notation, and likewise for the left and right Y_{MSB} values.

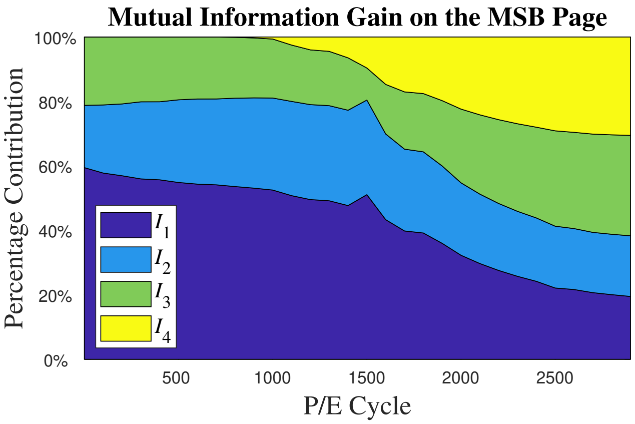

Fig. 3. This figure shows the percent of the overall gain expressed by (10) provided by each of the four progressive reads. During the first 1500 P/E cycles, the most gain is provided by $I_1 = P(\mathcal{E}_{00}) \cdot I(X_{\text{MSB}}; Y^{\text{right}}_{\text{MSB}^{(1)}} | \mathcal{E}_{00})$. I_2 is (12), $I_1 + I_3$ is (14), and I_4 is (13). Progressive reads are recommended to be chosen in this order.

The second term in the right hand side of (10) which can be decomposed as

$$I(X_{\text{MSB}}; Y^{\text{left}}_{\text{MSB}^{(1,2)}}, Y^{\text{right}}_{\text{MSB}^{(1,2)}} | Y^{\text{initial}}_{\text{MSB}^{(1,2)}}) \tag{11}$$
$$= P(Y^{\text{initial}}_{\text{MSB}^{(1)}} = 1) \cdot I(X_{\text{MSB}}; Y^{\text{left}}_{\text{MSB}^{(1)}} | Y^{\text{initial}}_{\text{MSB}^{(1)}} = 1) \tag{12}$$
$$+ P(Y^{\text{initial}}_{\text{MSB}^{(2)}} = 1) \cdot I(X_{\text{MSB}}; Y^{\text{right}}_{\text{MSB}^{(2)}} | Y^{\text{initial}}_{\text{MSB}^{(2)}} = 1) \tag{13}$$
$$+ P(\mathcal{E}_{00}) \cdot I(X_{\text{MSB}}; Y^{\text{right}}_{\text{MSB}^{(1)}}, Y^{\text{left}}_{\text{MSB}^{(2)}} | \mathcal{E}_{00}) \,, \tag{14}$$

where $\mathcal{E}_{00}$ is the event that $Y^{\text{initial}}_{\text{MSB}^{(1)}} = 0$ and $Y^{\text{initial}}_{\text{MSB}^{(2)}} = 0$.

This decouples the optimization problem into three independent optimization problems so that $T^{\text{left}}_{\text{MSB}^{(1)}}$, $T^{\text{right}}_{\text{MSB}^{(2)}}$, and the pair $(T^{\text{right}}_{\text{MSB}^{(1)}}, T^{\text{left}}_{\text{MSB}^{(2)}})$ can be independently optimized by maximizing the conditional MI expressions in (12), (13), (14), respectively. Only the pair $(T^{\text{right}}_{\text{MSB}^{(1)}}, T^{\text{left}}_{\text{MSB}^{(2)}})$, need to be jointly optimized. Extending the analysis to TLC and QLC reveals that no more than two thresholds need to be jointly optimized at a time. In contrast, the joint optimization of numerous thresholds was a main obstacle in [3].

The time to perform each read is longer than the time required to perform LDPC decoding in the controller, so only the number of progressive reads required for successful decoding will be performed. As shown in Figure 3, the MI contribution differs significantly among the various MSB progressive reads.

The goal is to identify an ordering of the read thresholds that provides the maximum cumulative MI after each read over the operating range of P/E cycles, which (as we will see in Sec. V) is approximately 1600 P/E cycles for this example. Figure 3 indicates that the ordering of read thresholds to achieve that goal begins with $T^{\text{right}}_{\text{MSB}^{(1)}}$, which provides the MI denoted as I_1 in Fig. 3. Note that for the first 1600 P/E cycles, I_1 provides more mutual information than any other threshold. Next, $T^{\text{left}}_{\text{MSB}^{(1)}}$ should be read, so that the cumulative MI is $I_1 + I_2$ in Fig. 3. Third, $T^{\text{left}}_{\text{MSB}^{(2)}}$ should be read, providing $I_1 + I_2 + I_3$. Finally $T^{\text{right}}_{\text{MSB}^{(2)}}$ is read, which provides negligible benefit until after 1000 P/E cycles.

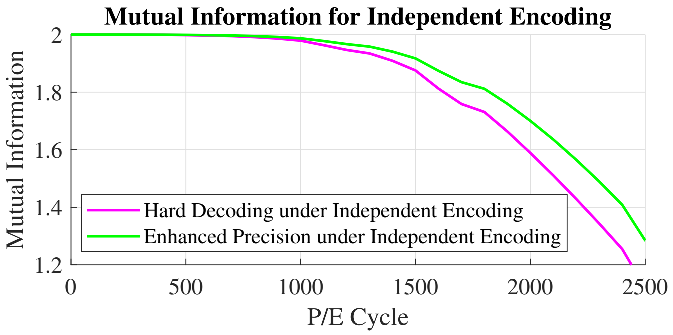

Fig. 4. For independent encoding this figure shows MI as a function of P/E cycles for the MI with initial hard-decoding reads, which is $I(X_{\text{LSB}}; Y^{\text{initial}}_{\text{LSB}}) + I\left(X_{\text{MSB}}; Y^{\text{initial}}_{\text{MSB}^{(1)}}, Y^{\text{initial}}_{\text{MSB}^{(2)}}\right)$, and the MI with enhanced precision which is given by the sum of (6) and (10).

Figure 4 shows the MI benefit of enhanced precision through progressive reads for independent encoding.

IV. Benefit of Jointly Encoding MSB and LSB

This section explores the benefit of jointly encoding the LSB and MSB bits. We consider both hard decoding and enhanced precision obtained by progressive reads. For joint encoding of LSB and MSB, the total information that can be recovered from the flash cell is

$$I_{\text{joint}} = I(X_{\text{LSB}}, X_{\text{MSB}}; \mathbf{Y}_{\text{LSB}}, \mathbf{Y}_{\text{MSB}^{(\mathbf{1,2})}}), \tag{15}$$

where for hard decoding $\mathbf{Y}_{\text{subscript}}$ is simply $Y^{\text{initial}}_{\text{subscript}}$ and "subscript" is LSB or MSB$^{(1,2)}$. For enhanced precision,

$$\mathbf{Y}_{\text{subscript}} = \left[Y^{\text{initial}}_{\text{subscript}} \quad Y^{\text{left}}_{\text{subscript}} \quad Y^{\text{right}}_{\text{subscript}} \right] . \tag{16}$$

For independently encoding using separate MSB and LSB pages, the total information that can be recovered is

$$I_{\text{indep.}} = I(X_{\text{LSB}}; \mathbf{Y}_{\text{LSB}}) + I(X_{\text{MSB}}; \mathbf{Y}_{\text{MSB}^{(\mathbf{1,2})}}) \,. \tag{17}$$

Jointly encoding all the bits in a cell provides an MI benefit over independently encoding each bit in the cell. To see this, compute the difference between I_{joint} and $I_{\text{indep.}}$ as

$$\begin{aligned} I_{\text{joint}} - I_{\text{indep.}} =& I(X_{\text{LSB}}, \mathbf{Y}_{\text{LSB}}; X_{\text{MSB}}, \mathbf{Y}_{\text{MSB}^{(\mathbf{1,2})}}) \\ &- I(X_{\text{LSB}}; X_{\text{MSB}}) - I(\mathbf{Y}_{\text{LSB}}; \mathbf{Y}_{\text{MSB}^{(\mathbf{1,2})}}). \end{aligned} \tag{18}$$

Since X_{LSB} and X_{MSB} may be assumed to be independent, $I(X_{\text{LSB}}; X_{\text{MSB}}) = 0$ so that

$$\begin{aligned} I_{\text{joint}} - I_{\text{indep.}} =& I(\mathbf{Y}_{\text{LSB}}; X_{\text{MSB}} | \mathbf{Y}_{\text{MSB}^{(\mathbf{1})}}, \mathbf{Y}_{\text{MSB}^{(\mathbf{2})}}) \\ &+ I(X_{\text{LSB}}; X_{\text{MSB}}, \mathbf{Y}_{\text{MSB}^{(\mathbf{1})}}, \mathbf{Y}_{\text{MSB}^{(\mathbf{2})}} | \mathbf{Y}_{\text{LSB}}) \,, \end{aligned}$$

which is always positive. Thus MI is lost in independently encoding the bits associated with a cell as compared to jointly encoding of all bits stored in the cell. Some schemes such as [5] take advantage of this MI benefit.

We now consider *how much* information is lost from independently encoding (and decoding) as compared to jointly encoding (and decoding). We consider both the hard decoding and enhanced precision cases. For the case of hard decoding,

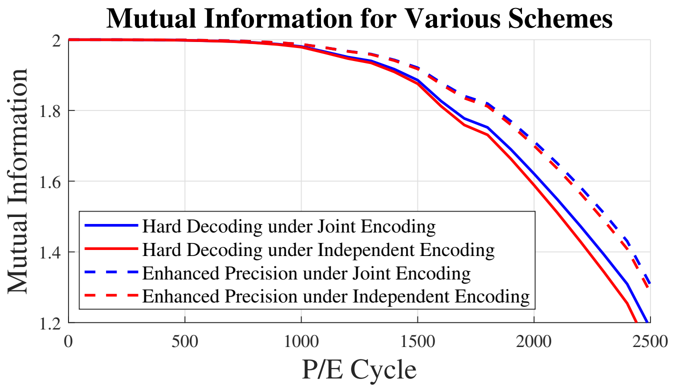

Fig. 5. MI between the inputs and output of a flash cell as a function of the number of P/E cycles for joint encoding and for independent encoding, considering both hard decoding and enhanced precision.

Figure 5 shows a small benefit for jointly encoding over independently encoding. For the case of enhanced precision, that small MI benefit is significantly reduced. While jointly encoding has a theoretical MI benefit, the small actual benefit available for MLC with enhanced precision may not be worth the incurred complexity and latency costs.

V. Simulation Results

This section provides LDPC [14]–[16] simulations that show the correspondence between the additional MI provided by progressive reads and FER performance improvement. Code rates around 0.9 are commonly used for the flash read channel. Consistent with this, our simulations used the binary LDPC code defined in [17] with rate 8/9 that encodes 14,400 input information bits to produce a 16,200-bit codeword.

This LDPC code was constructed using a protograph [18] of size 4×36. The check nodes of the protograph have a degree of 35 and the variable nodes of the protograph have degrees 3 and 4. A two-step lifting procedure using circulant progressive edge growth (CPEG) and approximate cycle extrinsic message degree (ACE) algorithms [19], [20], [14] was used to lift the protograph by factors of 3 and 150.

Each FER point was obtained by gathering at least 100 frame errors and the LDPC decoder used a maximum of 20 iterations of standard belief propagation. While actual flash systems require FERs lower than simulated in this paper, the primary purpose of these simulations is to show the correspondence between the MI provided by the hard-decoding and enhanced precision reads and FER performance in the independently encoded and jointly encoded flash models.

Figure 6 shows that the FER performance vs. number of P/E cycles tracks the MI performance for LDPC coding of the LSB page using hard decoding, one progressive read (either left or right), and two progressive reads. Note in particular that the MI benefit is identical for either of the two progressive reads by itself and that the FER performances are also identical when either of these progressive reads is used by itself.

Figure 7 shows that the FER performance vs. number of P/E cycles tracks the MI performance for LDPC coding of the

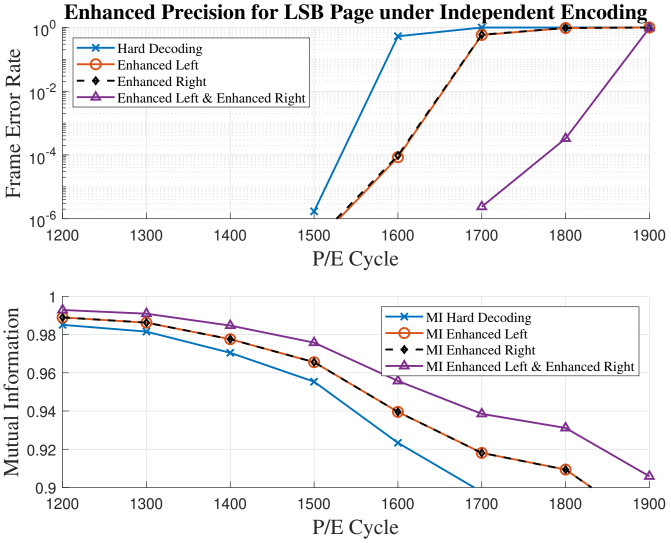

Fig. 6. FER (top) and MI (bottom) vs. number of P/E cycles for LDPC coding of the LSB page using hard decoding, one progressive read (either left or right) and two progressive reads. Lower FER values correspond to higher values of MI.

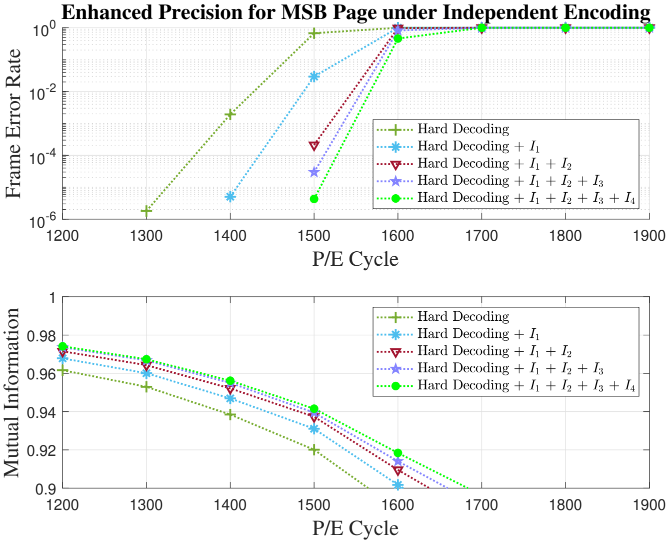

Fig. 7. FER (top) and MI (bottom) vs. number of P/E cycles for LDPC coding of the MSB page using hard decoding, and one, two, three or four progressive reads. Each progressive read improves performance but the progressive reads that add minimal mutual information provide a minimal FER improvement.

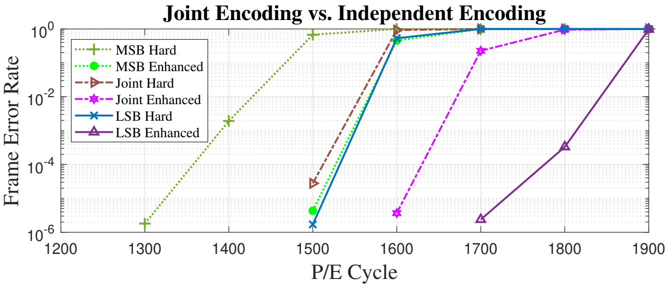

Fig. 8. FER vs. number of P/E cycles for LDPC coding of MSB page, LSB page, and a page that jointly encodes both MSB and LSB. Curves are shown for hard decoding and for fully enhanced precision with all progressive reads.

TABLE I
NUMBER OF P/E CYCLES AT FER= 10^{-5} AND AVERAGE PAGE LIFETIME

	MSB	**LSB**	**Indep. APL**	**Joint APL**
Hard	1325	1514	1419.5	1490
Enhanced	1507	1729	1618	1609

MSB page using hard decoding and one, two, three, or four progressive reads. The most additional MI is provided by I_1, which is reflected in the largest FER improvement occurring after the first progressive read.

Figure 8 shows the FER performance of independently encoding the MSB and LSB pages as compared to jointly encoding the MSB and LSB bits in a single page. This comparison is made both for hard decoding and for fully enhanced precision with all progressive reads.

To consider the benefit of joint encoding over independent encoding, we introduce a metric called the Average Page Lifetime (APL). APL is the average number of P/E cycles before a page exceeds a specified target FER. Here we consider a target FER of 10^{-5} for purposes of discussion in the context of our available simulation results, although for a real flash device, a lower value would be selected.

We define the page lifetime as the number of P/E cycles at which the page FER first exceeds the target FER. For a standard flash memory that independently encodes LSB and MSB pages, the APL is the average of the LSB page lifetime and the MSB page lifetime. For a flash device that jointly encodes the LSB and MSB bits on a single page, the APL is simply the page lifetime of the jointly encoded page.

Table I shows page lifetimes for MSB, LSB, and jointly encoded pages for hard decoding and for enhanced precision with all available progressive reads, i.e. two progressive reads for the LSB page, four progressive reads for the MSB page, and six progressive reads for the jointly encoded page.

Simulation results in Table I show that under hard decoding, the flash device using jointly encoded pages has a higher APL (70.5 more P/E cycles) than a standard flash memory that independently encodes LSB and MSB pages. This is consistent with Figure 5 which shows a small but noticeable MI benefit for joint encoding over independent encoding.

With all progressive reads available, Table I actually indicates that the APL of jointly encoding is *less* than for independent encoding by 9 P/E cycles. Figure 5 would have predicted a negligible benefit for jointly encoding, and the small benefit we measured for independent encoding may result from variation in our simulation results for the LSB with all progressive reads which seems to have a slightly lower than expected FER at 1800 P/E cycles.

VI. CONCLUSION

This paper shows that the information-theoretic objective of maximizing MI leads to a straightforward optimization procedure for determining the best placement of thresholds for progressive reads that provided enhanced precision when initial LDPC decoding fails. The information-theoretic approach not only identifies the threshold placement but also indicates which progressive reads should be performed first because they will provide the most benefit. Our information-theoretic analysis also shows that the additional MI that can be obtained through jointly encoding the LSB and MSB bits is small even for hard decoding and negligible when enhanced precision is available through progressive reads. LPDC simulations confirmed the information-theoretic analysis.

REFERENCES

[1] Y. Cai, S. Ghose, E. F. Haratsch, Y. Luo, and O. Mutlu, "Error characterization, mitigation, and recovery in flash-memory-based solid-state drives," *Proceedings of the IEEE*, vol. 105, no. 9, pp. 1666–1704, Sep 2017.

[2] J. Wang, T. Courtade, H. Shankar, and R. D. Wesel, "Soft information for LDPC decoding in flash: mutual-information optimized quantization," in *2011 IEEE Global TeleCommun. Conf. - GLOBECOM 2011*, Dec 2011, pp. 1–6.

[3] J. Wang, K. Vakilinia, T.-Y. Chen, T. Courtade, G. Dong, T. Zhang, S. H., and R. D. Wesel, "Enhanced precision through multiple reads for LDPC decoding in flash memories," *IEEE J. Select. Areas Commun.*, vol. 32, no. 5, pp. 880 – 891, May 2014.

[4] T. Parnell, C. Dünner, T. Mittelholzer, and N. Papandreou, "Capacity of the MLC NAND Flash Channel," *IEEE J. Select. Areas Commun.*, vol. 34, no. 9, pp. 2354–2365, Sept 2016.

[5] S. Lee, D. Kim, and J. Ha, "A paired-page reading scheme for NAND flash memory," in *2016 Int. Conf. on Inf. and Communication Tech. Convergence (ICTC)*, Oct 2016, pp. 1065–1067.

[6] H. Wang, N. Wong, T.-Y. Chen, and R. D. Wesel, "Using dynamic allocation of write voltage to extend flash memory lifetime," *IEEE Trans. Commun.*, vol. 64, no. 11, pp. 4474 – 4486, November 2016.

[7] H. Wang, "Flash model: mean and standard deviation." [Online]. Available: http://www.seas.ucla.edu/csl/codes/enhnPrecisionMuSigma.txt

[8] H. Wang, N. Wong, and R. D. Wesel, "Dynamic voltage allocation with quantized voltage levels and simplified channel modeling," in *49th Asilomar Conf. on Signals, Syst. and Comput.*, Nov 2015, pp. 834–838.

[9] D. h. Lee and W. Sung, "Estimation of NAND flash memory threshold voltage distribution for optimum soft-decision error correction," *IEEE Trans. Signal Process.*, vol. 61, no. 2, pp. 440–449, Jan 2013.

[10] L. Dolecek and F. Sala, *Channel Coding Methods for Non-Volatile Memories*. Now Foundations and Trends, 2016. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=8187358

[11] S. Liu and X. Zou, "QLC NAND study and enhanced gray coding methods for sixteen-level-based program algorithms," *Microelectronics Journal*, vol. 66, no. Supplement C, pp. 58 – 66, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0026269216307169

[12] B. Peleato and R. Agarwal, "Maximizing MLC NAND lifetime and reliability in the presence of write noise," in *2012 IEEE Int. Conf. on Commun. (ICC)*, June 2012, pp. 3752–3756.

[13] C. A. Aslam, Y. L. Guan, and K. Cai, "Read and write voltage signal optimization for multi-level-cell (MLC) NAND flash memory," *IEEE Trans. Commun.*, vol. 64, no. 4, pp. 1613–1623, Apr 2016.

[14] T. Y. Chen, K. Vakilinia, D. Divsalar, and R. D. Wesel, "Protograph-based raptor-like ldpc codes," *IEEE Trans. Commun.*, vol. 63, no. 5, pp. 1522–1532, May 2015.

[15] J. Kim, D. h. Lee, and W. Sung, "Performance of rate 0.96 (68254, 65536) eg-ldpc code for nand flash memory error correction," in *2012 IEEE Int. Conf. on Comm. (ICC)*, June 2012, pp. 7029–7033.

[16] K. Vakilinia, D. Divsalar, and R. D. Wesel, "Optimized degree distributions for binary and non-binary ldpc codes in flash memory," in *Int. Symp. on Inf. Theory and its Appl.*, Oct 2014, pp. 6–10.

[17] S. V. S. Ranganathan, "Flash LDPC code: rate 8/9." [Online]. Available: http://www.seas.ucla.edu/csl/codes/Flash_Rate_8_9_LDPC.txt

[18] J. Thorpe, "Low-density parity-check (LDPC) codes constructed from protographs," JPL, IPN-PR 42-154, Aug. 2003.

[19] X.-Y. Hu, E. Eleftheriou, and D.-M. Arnold, "Regular and irregular progressive edge-growth tanner graphs," *IEEE Trans. Inform. Theory*, vol. 51, no. 1, pp. 386–398, Jan. 2005.

[20] T. Tian, C. R. Jones, J. D. Villasenor, and R. D. Wesel, "Selective avoidance of cycles in irregular LDPC code construction," *IEEE Trans. Commun.*, vol. 52, no. 8, pp. 1242–1247, Aug. 2004.