Index Programming for Flash Memory

Hachem Yassine, Justin P. Coon, *Senior Member, IEEE*, and David E. Simmons

***Abstract*— We present a novel data programming scheme for flash memory. In each word-line, exactly k out of n memory cells are programmed while the rest are kept in the erased state. Information is then conveyed by the index set of the k programmed cells, of which there are $\binom{n}{k}$ possible choices (also called activation patterns). In the case of multi-level flash, additional information is conveyed by the threshold-voltage levels of the k programmed cells (similar to traditional programming). We derive the storage efficiency of the new scheme as a function of the fraction of programmed cells and determine the fraction that maximizes it. Then, we analyze the effect of this scheme on cell-to-cell interference and derive the conditions that ensure its reduction compared with the traditional programming. Following this, we analyze the performance of our new scheme using two detection methods: fixed reference detection and dynamic reference detection, and conclude that using dynamic reference detection will result in page error performance improvements that can reach orders of magnitude compared with that attainable by the fixed reference approach. We then discuss how logical pages can be constructed in the index programming similarly to traditional programming. Finally, we discuss the results and tradeoffs between storage efficiency and error resilience proposed by the scheme along with some future directions.**

***Index Terms*— Flash memory, index modulation, rank modulation, CCI reduction, dynamic detection.**

I. Introduction

OVER the last two decades, flash memory has become a major pillar of non-volatile storage. It is employed in applications ranging from massive enterprise storage (data centres) using solid state drives, to personal devices (smartphones) [1]. The unprecedented levels of data being generated have led to an increasing capacity demand, to which the manufacturers have responded with an aggressive bit-cost reduction strategy [2]. These bit-cost reduction techniques include the shrinking of cell size and the use of triple and multiple-level cell (TLC, MLC) technologies to allow more bits to be stored in the same cell as opposed to the original single level cell (SLC) technology where only a single bit was stored in each cell [1]. However, these strategies come at the expense of a deteriorating reliability, e.g., the first generation of SLC devices endured 10^5 program/erase (P/E) cycles, while recent TLC devices are limited to less than 1000 P/E cycles [3]. In the last decade, the subject of reliability of flash memory has gained a lot of interest. A multitude of ideas were proposed to overcome the drawbacks of device densification and the use of multi-level cells. Densification amplifies cell-to-cell interference (CCI), which widens the threshold voltage distribution of each state (level), while the use of more levels divides the limited voltage range into a greater number of windows, reducing the read error margins [2].

To mitigate these drawbacks, many techniques have been proposed, such as the introduction of 3D flash which consists of a new layering approach where the number of cells can be increased vertically with little additional cost and without shrinking the size of a cell [4]. Besides the physical approach, many system level approaches have been proposed too, such as the use of advanced error correcting codes [5] (e.g., LDPC codes with soft decoding) to increase the error correction capability. However, this approach causes an increase in read times due to the extraction of soft information, and also requires precise knowledge of the flash channel characteristics [6].

In order to minimize the number of errors while keeping the read times low, it is crucial that the read references are appropriately configured. Since the characteristics of such memory deteriorate with usage and time, it is clear that the optimal read references should be adapted with usage/time. Two main approaches have been proposed to adaptively estimate the optimal references as the device deteriorates. The first approach is to assume a theoretical model of the noise distribution (e.g., Gaussian), and then estimate its parameters by building histograms and minimizing a certain metric [7]–[9], or by the use of progressive reading [10]. The second approach is to fix the number of cells programmed to each state, which allows dynamic change of the read references until the correct number of cells in each state is detected. We refer to this type of read reference as a *dynamic read reference*. The approach was introduced first for single level cells (SLC) in [11] and generalized for multiple level cells (MLC) in [12], where it was shown that this detection is sub-optimal but has a performance that is superior to that of *fixed read references*. In this paper, fixed and dynamic read references will play pivotal roles within our study. The idea of fixing the number of programmed cells was also employed in [13] to design a rewriting scheme for flash memory.

Besides suffering from deterioration with time/usage, flash memory is also inherently asymmetric: the smallest programmable unit is a word-line, while the smallest erase unit is the whole block. Consequently, it is crucial when programming multi-level cells that the voltages have a low variance distribution to avoid expensive programming errors.

Manuscript received August 24, 2016; revised December 12, 2016; accepted February 4, 2017. Date of publication February 14, 2017; date of current version May 13, 2017. This work was supported by the EPSRC [grant number EP/M507520/1]. The associate editor coordinating the review of this paper and approving it for publication was L. Dolecek.

The authors are with Department of Engineering Science, University of Oxford, Oxford OX1 3PJ, U.K. (e-mail: hachem.yassine@eng.ox.ac.uk; justin.coon@eng.ox.ac.uk; david.simmons@eng.ox.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCOMM.2017.2669028

To alleviate this problem, rank modulation [14] introduced a new paradigm of how data can be stored; in particular, which variable conveys the information. This technique conveys information by the permutation induced through the ranking of voltages of the cells instead of their absolute voltages. Besides mitigating the asymmetry in writing, rank modulation can also increase the storage efficiency. Another asymmetry in flash memory is that of errors, because some errors are more likely to occur than others. In [15], an empirical error analysis of MLC devices from two different manufacturers showed that approximately 80% of errors correspond to programmed cells detected in other programmed states. Similar results were obtained in [16]. Among the sources of these errors, CCI is becoming the major issue as cells become closer to each other. Cells programmed to higher levels interfere more than those in low levels, and some specific data patterns result in cells suffering very high levels of CCI (e.g. a cell in the erased state stuck between cells programmed to the highest state). To reduce this undesirable effect, constrained codes were proposed to eliminate such patterns [17]–[19]. In [20], the authors employed known codes for symmetric errors to construct new codes to protect data from asymmetric errors. Finally, in [21] an endurance coding scheme using constrained coding was employed to extend the endurance of flash memory.

In this paper, we introduce a new method of storing information by constraining each word-line to have exactly k programmed cells out of n, while the rest of the cells are left erased. This will give rise to $\binom{n}{k}$ unique writeable patterns into the n cells. The information is then stored by one of the $\binom{n}{k}$ possible index sets, which we also refer to as *activation pattern*. Our proposed scheme exploits the asymmetry of errors in flash memory; intuitively, since the majority of errors correspond to cells in a programmed state shifting into another programmed state (i.e., non-erased), it is less likely that the activation pattern changes, resulting in an enhanced robustness. We refer to this method as *index programming* (IP), inspired by the recent works on spatial modulation techniques for wireless systems [22] where additional information is transmitted by the set of activate transmit antennas. Similar idea was employed at the sub-carrier level OFDM systems with index modulation [23]–[25], where instead of modulating all the sub-carriers in an OFDM system, only k sub-carriers are modulated (i.e., activated) and additional information is conveyed by the set of indices of active sub-carriers.

The main conclusion of our paper is that IP with dynamic reference detection outperforms traditional programming from an error rate perspective, whilst also being able to (asymptotically in n) achieve identical storage efficiency. The price for this performance advantage (which can be as high as four orders of magnitude compared to traditional programming) is additional complexity due to dynamic detection (which is equivalent to a sort operation), and the reduction in storage efficiency in practical set-ups (non asymptotic). The following is a comprehensive list of our manuscript's contributions:

- we introduce IP for flash memory devices and derive its storage efficiency;
- we then derive the conditions under which the CCI is reduced compared to traditional amplitude programming (AP) for two memory channel models;
- following this, we propose a new paging structure under IP that consists of two logical pages on each word-line;
- we then propose two IP detection methods: dynamic and fixed detection and show that the proposed IP scheme outperforms the AP scheme in page error rates for both of them;
- finally, we discuss our results and the trade-off provided by IP and propose a more general paging structure that resembles AP.

The paper is organized as follows. Section II will review the flash memory basics. Index programming and its storage efficiency are introduced in section III. Section IV provides a detailed analysis of the CCI reduction under the new scheme. Section V discusses the detection of the proposed scheme. In section VI, paging structure of IP and the page error rate are analytically evaluated and numerically simulated. Section VII provides a detailed discussion of the advantages of IP. section VIII introduces a more general paging structure for IP that resembles AP. The paper concludes in section IX with a summary of the contributions as well as future directions.

II. Flash Memory Basics

Flash memory is organized into blocks. A block is a two-dimensional grid of cells each consisting of a floating gate transistor where charge can be retained for long periods. Cells on the same row are said to be on the same word-line, while those on the same column are said to be on the same bit-line. The smallest erase unit is the whole block while the smallest write/read unit is a word-line [26]. Each cell can be set into one of q logical states in $\mathcal{Q} = \{s_0, \ldots, s_{q-1}\}$, where s_0 corresponds to the erased state (no charge) while the other states represent the programmed states with different charge for each. The cases with $q = 2, 4$ and 8 are also known as single, multi and triple-level cell (SLC, MLC and TLC) technologies respectively. The logical state of a cell is mapped into a voltage level reflecting the charge stored in its floating gate, i.e., s_0 reflects the lowest voltage while s_{q-1} reflects the highest voltage. Without loss of generality, it can be assumed that the states themselves are the voltages [12].

The current industry practice is to treat each cell as an individual storage medium. Since it can be in one of q states, each group of $\log_2 q$ bits can be stored into a cell by setting its voltage to the corresponding value. For example, in the case of $q = 4$, we get $11 \rightarrow s_0, 10 \rightarrow s_1, 00 \rightarrow s_2$ and $01 \rightarrow s_3$ when employing gray coding. When a cell stores several bits, each bit is considered to be on a separate unit called a page, i.e., a word-line is divided into several pages. For example, for $q = 4$, one possible setting is that each word-line is divided into two pages, one for the most significant bits (MSB) and the other for the least significant bits (LSB). These pages store independent data and are read independently, which allows for faster read operation and fewer bit errors on the individual pages. Fig. 1 describes the current standard

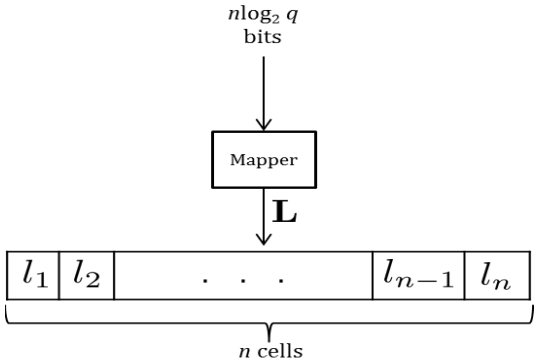


Fig. 1. Diagram showing the traditional amplitude programming process.

in general: first, $n \log_2 q$ bits are mapped into a q-ary states vector $\mathbf{L} = [l_1, \ldots, l_n] \in Q^n$. This vector is then mapped into a vector of voltages (called a write pattern) that maps the state of each cell into its corresponding voltage, resulting in the vector $\mathbf{X} = [x_1, \ldots, x_n]$. Finally the cells' voltages are programmed according to $\mathbf{X}$. Since the states are assumed (without loss of generality) to be the voltages, we can write $\mathbf{X} = \mathbf{L}$. Throughout this paper we will refer to this method of conveying information as traditional amplitude programming, or simply amplitude programming (AP).

The voltage of a cell can be altered by different sources, such as noise and CCI. The major sources of noise are random telegraph noise and charge leakage. They are the result of the deterioration of the floating gate insulation due to being repeatedly subject to high electric stress [27]. In addition to noise, cells close to each other will exhibit a parasitic capacitive coupling that makes the voltage shift of one cell induce a voltage shift in its neighboring cells. This effect gives rise to CCI.

When reading a word-line, the actual voltage of a cell is not readily available. Rather, the voltage range is divided into q regions, each corresponding to one state. The boundaries of the regions are described by a read vector $\mathbf{t} = [t_1, \ldots, t_{q-1}] \in \mathbb{R}^{q-1}$, where $t_i > t_{i-1}$ $\forall i$ [12]. If the voltages of a word-line of size n are described by $\mathbf{X} = [x_1, \ldots, x_n] \in Q^n$ and the voltages of a word-line after suffering from noise and interference are represented by $\mathbf{Y} = [y_1, \ldots, y_n] \in \mathbb{R}^n$ then we assume the word-line is read as the vector $\hat{\mathbf{X}}(\mathbf{Y}, \mathbf{t}) = [\hat{x}_1, \ldots, \hat{x}_n]$, where $\hat{x}_i = s_j$ if $t_j < y_i \leq t_{j+1}$ with $t_0 = -\infty$ and $t_q = +\infty$. By increasing q, more bits per cell can be stored at a relatively small cost. However, the size of the voltage window corresponding to each state shrinks, increasing the probability that a cell is detected in an erroneous state because of noise and CCI [2]. Fig. 2 illustrates the different states of an MLC device and the read references when hard decision detection is considered (i.e., q states $\Rightarrow$ q detection regions).

III. Index Programming

In this section, we introduce the IP scheme and determine its storage efficiency. Let us begin with the working principles of the IP scheme.

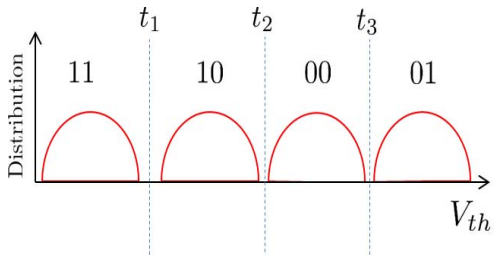


Fig. 2. Diagram showing the distributions and read references for MLC.

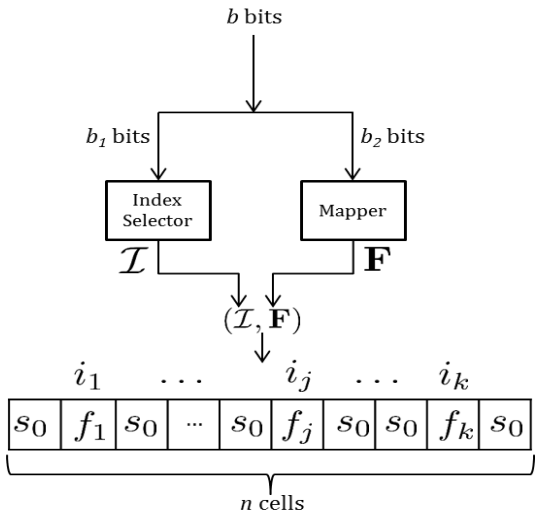


Fig. 3. Diagram showing the index programming process.

A. Working Principles

Index programming (IP) differs from the AP method of storing information in its fundamental working principles. Consider a word-line with n cells supporting q states each. The cells are initially in the erased state. Assume that exactly k cells are chosen out of the n cells to be programmed (active) while the other $n - k$ cells are left in the erased state s_0 (inactive). There are $\binom{n}{k}$ distinct possible ways to choose such k cells. Let $\mathcal{C}(n, k) = \{\mathcal{I}_1, \ldots, \mathcal{I}_{\binom{n}{k}}\}$ be the set of all k-subsets of $\{1, \ldots, n\}$ (i.e., subsets in $\{1, \ldots, n\}$ of size k). An element $\mathcal{I}_m$ of this set is defined as $\mathcal{I}_m = \{i_{m,1}, .., i_{m,k}\}$, with $i_{m,j} \in \{1, \ldots, n\}$ and $i_{m,j} < i_{m,r}$ for $j < r$. Each element represents a unique set of indices of the k active cells, and we will refer to each element of $\mathcal{C}(n, k)$ as an activation pattern. Each programmed cell can be in one of $q - 1$ states. Consider $\mathbf{F}_m = \left[f_{m,1}, \ldots, f_{m,k}\right]$ to be the vector representing the states to which the k programmed cells are set, i.e., $f_{m,j} \in Q \backslash \{s_0\}$. Note that for each activation pattern there are $(q-1)^k$ possible vectors $\mathbf{F}_m$ that could be programmed into the chosen k cells. In total, there are $(q-1)^k \binom{n}{k}$ possible pairs $(\mathcal{I}_m, \mathbf{F}_m) \in \mathcal{L} = \mathcal{C}(n, k) \times (Q \backslash \{s_0\})^k$, each representing a unique write pattern reflected by the voltages of the n cells. This is the essence of the IP method.

Fig. 3 explains the IP process, where an incoming stream B of b bits is divided into two sub-streams, $B_{\mathcal{I}}$ and $B_{\mathbf{F}}$, of sizes $b_1 = \left\lfloor \log_2 \binom{n}{k} \right\rfloor$ bits and $b_2 = \left\lfloor k \log_2 (q-1) \right\rfloor$ bits, respectively, such that $b = b_1 + b_2$. $B_{\mathcal{I}}$ is then mapped by the

index selector block into an activation pattern $\mathcal{I}$ from $\mathcal{C}(n,k)$. For small n and/or k, the index selector can be a simple lookup table. However, the lookup table becomes very large when n is large, so that in this scenario another technique is required (see section VII). The second sub-stream $B_{\mathbf{F}}$ is mapped into a $(q-1)$-ary vector $\mathbf{F}$ of size k, representing the states to which each of the cells indexed by $\mathcal{I}$ is to be programmed. Finally, a write pattern $\mathbf{X}$ is constructed and written onto the n cells. The total information b that can be stored in n cells under IP can be bounded by removing the floors in b_1 and b_2 as follows

$$\begin{aligned} k\log_2(q-1) + k\log_2\binom{n}{k} - 2 \le b \\ \le k\log_2(q-1) + k\log_2\binom{n}{k}. \end{aligned} \tag{1}$$

Using Stirling's approximation ($\sqrt{2\pi}n^{n+\frac{1}{2}}e^{-n} \le n! \le en^{n+\frac{1}{2}}e^{-n}$) of the factorial function, these bounds can be re-written as follows

$$\begin{aligned} \alpha n\log_2(q-1) + nH_2(\alpha) - \log_2\frac{e^2}{\sqrt{2\pi}}\sqrt{\alpha n(1-\alpha)} - 2 \\ \le b \le \alpha n\log_2(q-1) + nH_2(\alpha) - \log_2\frac{2\pi}{e}\sqrt{\alpha n(1-\alpha)}, \end{aligned} \tag{2}$$

where $\alpha = k/n \in (0,1)$ represents the ratio of the number of programmed cells to all available cells. Let us proceed with some illustrative examples.

B. Illustrative Examples

1) $q = 2$: Assume a memory with $n = 4, q = 2$ and $k = n/2$. Each cell can either be in the erased state (s_0) or the programmed state (s_1). There are $\binom{4}{2} = 6$ ways to program only two out of the four cells (activation patterns). As a result, the possible write patterns that can be written into the four cells are $[s_1, s_1, s_0, s_0]$, $[s_1, s_0, s_1, s_0]$, $[s_1, s_0, s_0, s_1]$, $[s_0, s_1, s_0, s_1]$, $[s_0, s_0, s_1, s_1]$, and $[s_0, s_1, s_1, s_0]$. Hence a maximum of $\log_2(6)$ bits can be stored in these 4 cells.

2) $q = 3$: Now consider the same parameters but with $q = 3$, i.e., a cell can be either in the erased state (s_0) or in one of two programmed states (s_1 and s_2). There are still six possible activation patterns, but each programmed cell now has several possible states to which it can be programmed. This gives an additional degree of freedom. For example, if the first and third cells are active, there are four possible write patterns when $q = 3$: $[s_1, s_0, s_1, s_0]$, $[s_1, s_0, s_2, s_0]$, $[s_2, s_0, s_1, s_0]$, $[s_2, s_0, s_2, s_0]$. For each of the other five activation patterns, there are four write patterns and we are able to store up to $\log_2(6 \times 4) \simeq 4.58$ bits in four cells.

C. Storage Efficiency

Storage efficiency is defined as the average number of bits that can be stored in a cell according to a specific scheme when the number of cells goes to infinity. In the traditional AP scheme, $\log_2 q$ bits can be stored in each cell (even when the number of cells is small). It is easy to see that by dividing the bounds in (2) by n and letting $n \to \infty$ the bounds meet and the storage efficiency of IP can be written as follows

$$\eta = \lim_{n\to\infty}\frac{b}{n} = H_2(\alpha) + \alpha\log_2(q-1), \tag{3}$$

where $H_2(x)$ is the binary entropy function. The final expression has two parts. Intuitively, it can be seen that the first term relates to the information conveyed by the activation pattern since a cell is active with probability α. The second term relates to the information stored in $\mathbf{F}$. In order to maximize the information conveyed by the activation pattern, it is clear that one should program exactly half the cells (i.e. $k = n/2$) to maximize the number of possible activation patterns (and to maximize $H_2(\alpha)$). However, this does not maximize η. Maximizing η can be done by differentiating η with respect to α and solving for α the equation $\frac{d\eta}{d\alpha} = 0$, the result is then

$$\alpha_0 = \frac{q-1}{q}, \tag{4}$$

giving a maximum of

$$\eta_0 = \log_2 q \text{ bits/cell}. \tag{5}$$

This maximum η_0 is equal to the storage efficiency of traditional AP. Fig. 4 shows the plot of η (eq. (3)) as a function of α for different number of levels. It is important to note that the storage efficiencies plotted in Fig. 4 assume $n \to \infty$ as opposed to AP where the efficiency is $\log_2 q$ no matter how many cells are considered. For the example given in the previous section ($q = 3$ and $\alpha = 0.5$), the efficiency was only 1.145 bits/cell while from the Fig. 4 we see that the efficiency can reach 1.5 bits/cell for $n \to \infty$.

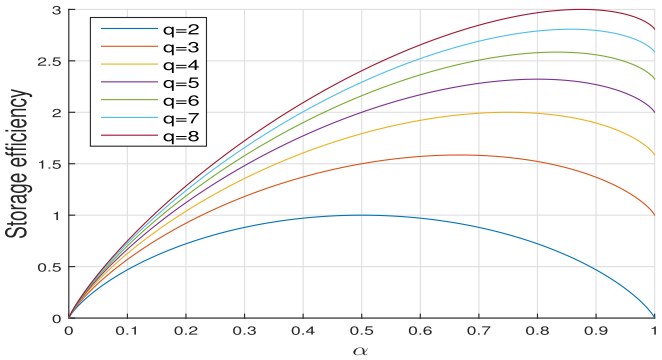


Fig. 4. Plot of the asymptotic storage efficiency of IP (eq.(3)) as a function of α for different number of levels.

The perceived reduction in storage efficiency relative to AP in the IP scheme comes with the advantage of reduced CCI and increased resiliency as we will show in sections IV and VI.

IV. Analysis of Cell to Cell Interference

In this section we consider two CCI models. The first one is a general model where each voltage shift in a neighbouring cell creates interference in the victim cell. The second is more simplified and is frequently used in the context of interference mitigating constrained coding. For the first, we analyse the reduction in the average CCI while for the second, we analyse the reduction in the probability of occurrence of a specific unwanted pattern.

A. General Interference Model

Normally, a neighbour cell in the erased state (s_0) does not interfere with the victim cell, while a cell in the highest state (s_{q-1}) interferes most aggressively. Consider a victim cell with N neighbours. The total interference T suffered by this cell is the sum of the interferences from each neighbour:

$$T = \sum_{k=1}^{N} \Delta_k, \tag{6}$$

where Δ_k is the amount of CCI induced by the k-th neighbour cell, which depends on its state. The expected total interference is then given by

$$\overline{T} = \sum_{k=1}^{N} \mathbb{E}[\Delta_k]. \tag{7}$$

Assume $x_k \in \{s_0, \ldots, s_{q-1}\}$ is the state of the kth neighbour. By the law of total expectation, the term $\mathbb{E}[\Delta_k]$ from (7) is given by

$$\begin{aligned}\mathbb{E}[\Delta_k] &= \mathbb{E}[\mathbb{E}[\Delta_k|x_k]] \\ &= \sum_{j\in\{0,\ldots,q-1\}} p_{k,j}\mathbb{E}\left[\Delta_k|x_k = s_j\right],\end{aligned} \tag{8}$$

where $p_{k,j}$ is the probability that the kth interfering cell is programmed to state $s_j \in \{s_0, \ldots, s_{q-1}\}$. Furthermore, since a cell in the erased state does not interfere, we obtain

$$\mathbb{E}[\Delta_k] = \sum_{j\in\{1,\ldots,q-1\}} p_{k,j}\mathbb{E}\left[\Delta_k|x_k = s_j\right]. \tag{9}$$

Typically, the probability of a cell being programmed to an arbitrary state is independent of its location, unless a cell position is known to be corrupt (e.g., stuck at defect). Under this assumption, $p_{k,j} = p_j$, i.e., the probability of x_k equalling state s_j is independent of k. However, p_j depends on how the input is distributed. If an equiprobable input scheme is used, then $p_j = 1/q$ for all $j \in \{0, \ldots, q-1\}$. In our proposed scheme, only k cells out of every n are programmed, which means that $p_0 = 1 - k/n$ and the other $q-1$ states are equiprobable. Hence, we get that $p_j = k/(n(q-1))$ for all $j \in \{1, \ldots, q-1\}$. The equiprobable input – which maximizes the entropy of the input – is the current practice used in the AP scheme. Under this scheme the expected total interference for AP becomes

$$\overline{T}_{AP} = \frac{1}{q}\sum_{k=1}^{N}\sum_{j=1}^{q-1}\mathbb{E}\left[\Delta_k|x_k = s_j\right]. \tag{10}$$

In contrast, the proposed IP scheme admits the following expected total interference expression

$$\overline{T}_{IP} = \frac{k}{n(q-1)}\sum_{k=1}^{N}\sum_{j=1}^{q-1}\mathbb{E}\left[\Delta_k|x_k = s_j\right]. \tag{11}$$

To compare these two schemes, we observe the ratio of total expected interferences

$$\frac{\overline{T}_{IP}}{\overline{T}_{AP}} = \alpha\frac{q}{(q-1)}. \tag{12}$$

In order to observe a reduction in the average CCI under the IP scheme, the ratio given by (12) should be less than 1. This condition can be rewritten as

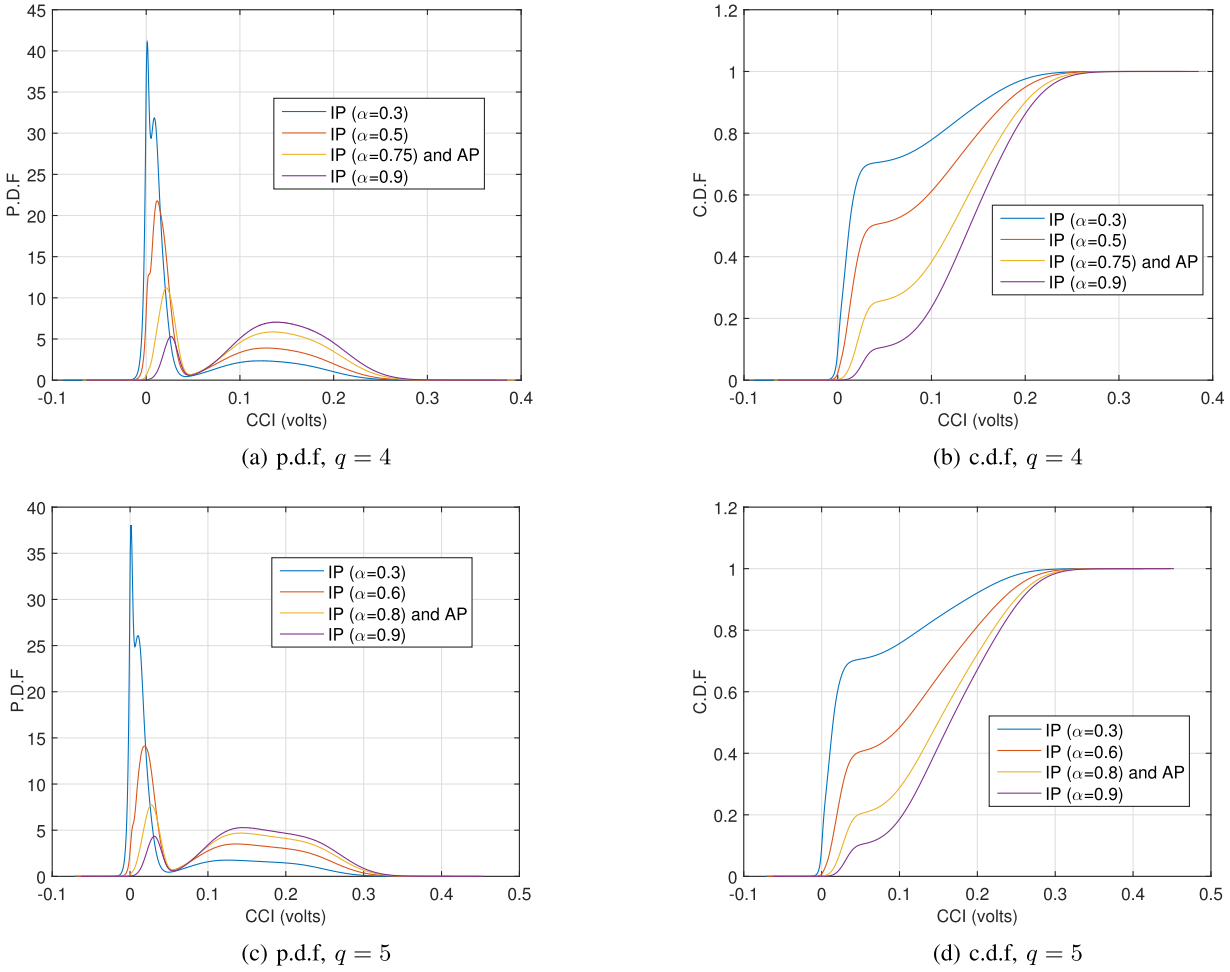
$$\alpha \leq \frac{q-1}{q}. \tag{13}$$

When (13) is met with equality, the average CCI is not reduced. However, as we have seen in (5), this value of α maximizes the storage efficiency. Hence, there is a trade-off between maximizing storage efficiency and reducing the average CCI. Chip designers should be able to exploit this trade-off to achieve a specified storage efficiency/robustness goal in practice. Fig. 5 shows both the p.d.f and the c.d.f of CCI for AP and IP. These distributions were simulated using the model in [28] for $q = 4$ (Figs. 5a and 5b) and $q = 5$ (Figs. 5c and 5d). The following parameters were employed $\bar{\gamma}_y = 0.08$, $\bar{\gamma}_{xy} = 0.0048$, $\mathbb{V}\left[\gamma_y\right] = 2.1 \times 10^{-5}$, $\mathbb{V}\left[\gamma_{xy}\right] = 7.6 \times 10^{-8}$ and 15000 P/E cycles. For both values of q, the interval in which the majority of the distribution is confined becomes larger as α increases. This indicates the increase in the severity of CCI (uncertainty about it). For $\alpha = (q-1)/q$ (yellow) the distributions of CCI are the same for IP and AP. For $\alpha > (q-1)/q$, CCI becomes more severe in IP than in AP since now more cells on average are programmed and hence contribute to the interference. In general, when IP is employed, with (13) satisfied, there is a larger probability that the system exhibits a lower CCI. As mentioned in the introduction, narrowing the CCI distribution will help improve the performance from an error rate perspective, which is exactly what increasing the fraction of erased cells achieves.

B. Bit-Line Interference Model

Consider the following simplistic model: the only source of errors is bit-line interference, i.e., only adjacent cells on the same bit-line interfere with each other. Furthermore, we assume that an error occurs only when an erased cell (E) falls between two cells that are on the highest level (H). This model is rather simplistic as CCI is not the only source of errors, and the mentioned pattern is not the only pattern to create CCI errors, but this model is often used in the literature of constrained coding for memories [19]. In the case of AP, the probability that the mentioned pattern (HEH) occurs is simply $1/q^3$, i.e., three consecutive word-lines having H, E, and H respectively on the same bit-line. In the case of IP (assuming the same number of levels q), the probability of occurrence of the pattern is $\alpha^2(1-\alpha)/(q-1)^2$. To reduce this probability to 0 in the case of AP, the number of levels has to be increased to infinity, which is impractical. However, in the case of IP, for any fixed number of levels, this probability can go to 0 by making α go to either 0 or 1.[1] This is possible in practice because of the very large number of cells in a word-line. From Fig. 4, it is clear that letting α go to 0 results in a zero storage efficiency for all q, while α going to 1 preserves a non-zero storage efficiency for $q > 2$. For example, consider a word-line with $n = 1000$ cells, $\alpha = 0.999$ and $q = 4$.

[1] if $\alpha = 1$, the system transforms into AP with $q-1$ states, which is useless. Hence, α should always be less than 1.



(a) p.d.f, $q = 4$ (b) c.d.f, $q = 4$

(c) p.d.f, $q = 5$ (d) c.d.f, $q = 5$

Fig. 5. The distributions of CCI in the traditional (AP) and our proposed (IP) schemes.

The probability of occurrence in the IP case is 0.00011 while that of the AP scheme is 0.0156, i.e., approximately two orders of magnitude reduction in the probability of occurrence. On the other hand, the storage efficiency of the IP scheme in this case is 1.592 bits/cell, a 20.4% reduction from the traditional 2 bits/cell for MLC. Finally, the ratio of the two probabilities of occurrence is $\alpha^2 q^3 (1-\alpha)/(q-1)^2$. For $q \geq 4$, $\alpha^2 q^3 (1-\alpha)/(q-1)^2 \leq 1$ requires that $\alpha \geq (q-1)/q$ or $\alpha \leq \left(\sqrt{4q-3}+1\right)/2q$. It is always possible to have an α that reduces the probability of occurrence of the unwanted pattern. Remember that the maximum storage efficiency of the IP scheme is achieved at $\alpha = (q-1)/q$, hence the emergence of the trade-off: by sacrificing some of the storage efficiency, the probability of occurrence of the CCI prone pattern can be reduced. Under this model, the condition of $\alpha \geq (q-1)/q$ is the opposite of the condition under which the average CCI is reduced in (13) for the model considered in the previous subsection.

V. Detection

In this section, we introduce two detection methods for IP: fixed reference detection and dynamic reference detection. To aid our discussion, we will begin by presenting the following channel model. There are different factors contributing to the distortion of the threshold voltage of cells. In addition to CCI, the major sources of distortion are programming noise, random telegraph noise and retention noise (charge leakage with time). The overall behaviour is a non-stationary, data dependent noise that is hard to express analytically [28]. However, many empirical results have shown that a Gaussian distribution provides a good approximation to the threshold voltage distribution [7]. For the sake of our analysis, we assume the flash memory channel to be equivalent to a zero mean Gaussian channel with variance that depends only on the number of P/E cycles. The output of a cell i initially set to $x_i \in Q$ is then given by

$$y_i = x_i + z_i, \tag{14}$$

where $z_i \sim \mathcal{N}(0, \sigma^2)$. A pattern $\mathbf{X}$ written into the cells is distorted by noise resulting in an output $\mathbf{Y}$. However, in practice, and as explained in section II, $\mathbf{Y}$ is not readily available but only a quantized version of it, namely $\hat{\mathbf{X}}(\mathbf{Y}, \mathbf{t})$. This means that the read vector $\mathbf{t}$ is what defines the detector, and it is the only factor that can be controlled. In the next two subsections, we will discuss fixed and dynamic reference detection, respectively.

A. Fixed Reference Detection

The AP scheme (equiprobable input) is similar to a q-ary PAM communication system [29] and is optimally detected using read references that are the mid-points between the nominal values of adjacent states

$$\mathbf{t}^*_{\mathbf{AP}} = \left[\frac{s_0+s_1}{2}, \ldots, \frac{s_{q-2}+s_{q-1}}{2}\right]. \tag{15}$$

Only the first element of (15) is needed to obtain an estimate of the indices of programmed cells, and we refer to this reference as the *index read reference*. In the case of IP, to detect the indices using a fixed reference, the same reference as AP can be used. However, the probability of a cell being programmed is in general different from it being erased. In this case, the optimal index read reference must be adjusted. The probability that a programmed cell is detected as erased or vice-versa is given by

$$\begin{aligned}\Pr(\text{error}) = {} & \frac{n-k}{n}\left(1-\Phi\left(\frac{t-s_0}{\sigma}\right)\right) \\ & + \frac{k}{n(q-1)}\sum_{\ell=1}^{q-1}\Phi\left(\frac{t-s_\ell}{\sigma}\right).\end{aligned} \tag{16}$$

The optimal index read reference t^*_{f} should satisfy $\frac{dP(error)}{dt} = 0$. More specifically, it satisfies

$$\begin{aligned}&\frac{n-k}{n}\left(\exp\left(-\frac{\left(t^*_{\mathrm{f}}-s_0\right)^2}{2\sigma^2}\right)\right) \\ &\quad = \frac{k}{n(q-1)}\sum_{\ell=1}^{q-1}\exp\left(-\frac{\left(t^*_{\mathrm{f}}-s_\ell\right)^2}{2\sigma^2}\right).\end{aligned} \tag{17}$$

We assume that an error will most likely shift a cell from its initial state into one of the immediate neighbouring states [15], [16], i.e., if an erased cell is detected as programmed it will most likely be in state s_1. Therefore, we can approximate t^*_{f} by considering non-zero only the first term of the summation. It follows that

$$t^*_{\mathrm{f}} = \frac{s_0+s_1}{2} + \frac{\sigma^2}{s_1-s_0}\ln\frac{(q-1)(n-k)}{k}. \tag{18}$$

Note that for s_j's sufficiently far apart, and low values of σ, $t^*_{\mathrm{f}} \simeq (s_0+s_1)/2$.

B. Dynamic Reference Detection

In the previous section, the read references were fixed and known in advance. Interestingly, the fact that the number of programmed cells is fixed to k can be exploited more efficiently. Intuitively, a necessary requirement for the index read reference to correctly detect the activation pattern is to detect exactly k cells in the programmed state. For this purpose, we suggest a dynamic index read reference that is adjusted for each word-line being read such that exactly k cells are above it. This choice will be superior to the fixed one (as we show in the next section), because in the fixed case, it is possible to detect more or less than k cells as programmed, which results in an error (without further processing). This dynamic reference will be a random variable that changes for each word-line. In practice its implementation can be done by means of a trial and repeat process, until exactly k cells are detected as active, i.e., the voltages are sorted by descending order and the first k cells are considered programmed. The dynamic read obviously increases the read time, but provides several advantages that will be discussed in later sections. After detecting an estimate $\hat{\mathcal{J}}$ of the activation pattern, $\hat{\mathcal{J}}$ is mapped back into a sub-stream $B_{\hat{\mathcal{J}}}$ (ideally, equal to $B_{\mathcal{J}}$). Knowing the activation pattern, the estimate $\hat{\mathbf{F}}$ of $\mathbf{F}$ is obtained by performing a traditional read using fixed references (e.g., (15)) to detect its symbols. This estimate is then mapped back to a stream $B_{\hat{\mathbf{F}}}$ (ideally, equal to $B_{\mathbf{F}}$). Note the sequential operation where the programmed cells have to be detected before reading $\mathbf{F}$.

VI. Performance of Index Programming

In this section we analyze the probabilities that the sub-streams $B_{\mathcal{J}}$ and $B_{\mathbf{F}}$ are detected with errors, i.e., $\Pr\left(\hat{\mathcal{J}} \neq \mathcal{J}\right)$ and $\Pr\left(\hat{\mathbf{F}} \neq \mathbf{F}\right)$, respectively. First we make the analogy between paging in AP and a new paging paradigm invoked by IP where two new logical units (pages) are defined. We derive the expressions of the page error rates for each page of both schemes using the detectors introduced in the previous section. Finally, we present our simulation results and discuss the advantages of our proposed scheme.

A. Paging Under Index Programming

The IP scheme divides a stream B of b bits into two sub-streams, $B_{\mathcal{J}}$ and $B_{\mathbf{F}}$, in a way that resembles paging in AP. While in AP each page stores n bits, in IP, the activation pattern stores b_1 bits, and the voltages of the programmed cells store b_2 bits as defined in section III. Throughout the rest of the paper we will refer to the sub-stream stored in the activation pattern ($B_{\mathcal{J}}$) as the *index page*, and to the sub-stream stored in the actual voltages of the programmed cells ($B_{\mathbf{F}}$) as the *amplitude page*. In this case, the IP scheme will always have two pages for all $q > 2$ as opposed to AP where the number of pages is p for $q = 2^p$. In section VIII, we will see how the IP scheme can be designed to have $p+1$ pages for $q = 2^p + 1$.

B. Page Error Rates

In the case of AP, when the Gray mapping shown in Fig. 2 and the optimal read vector $\mathbf{t}^*_{\mathbf{AP}}$ are used, the probability of error in the MSB and LSB pages (Pr(EMSB) and Pr(ELSB), respectively) can be written as follows

$$\begin{aligned}\Pr(\text{EMSB}) &= 1 - \prod_{m=1}^{n}\Pr\left(\hat{m}_i = m_i\right) \\ &= 1 - \left(1 - \frac{1}{2}\left(\Phi\left(-\frac{A}{2\sigma}\right) + \Phi\left(-\frac{3A}{2\sigma}\right)\right)\right)^n,\end{aligned} \tag{19}$$

and

$$\begin{aligned}\Pr(\text{ELSB}) &= 1-\prod_{m=1}^{n}\Pr\left(\hat{\ell}_i=\ell_i\right)\\ &= 1-\left(1-\frac{1}{2}\left(\Phi\left(\frac{5A}{2\sigma}\right)-\Phi\left(\frac{3A}{2\sigma}\right)\right.\right.\\ &\quad\left.\left.+2\Phi\left(-\frac{A}{2\sigma}\right)\right)\right)^{n},\end{aligned}\tag{20}$$

where m_i and ℓ_i represent the i-th bits of the MSB and LSB pages, respectively. $A=s_{i+1}-s_i$ is assumed constant for all i and $\Phi(x)$ is the c.d.f of the standard normal distribution. For the IP scheme, we define $\mathcal{E}=\{e_1,\ldots,e_{n-k}\}=\{1,\ldots,n\}\backslash\mathcal{I}$ to be the index set of the erased cells when $\mathcal{I}$ is that of the programmed ones. When a fixed index read reference is used, the decision on each cell is made independently of the other cells and $\Pr\left(\hat{\mathcal{I}}=\mathcal{I}\right)$ can be obtained as follows

$$\begin{aligned}\Pr\left(\hat{\mathcal{I}}=\mathcal{I}\right) &= \Pr\left(\mathbf{Y}_{\mathcal{E}}\preceq t_1\cap\mathbf{Y}_{\mathcal{I}}\succ t_1\right)\\ &= P_{0,0}^{n-k}\Pr\left(\mathbf{Y}_{\mathcal{I}}\succ t_1|\mathbf{Y}_{\mathcal{E}}\preceq t_1\right)\\ &= P_{0,0}^{n-k}\prod_{j=1}^{k}\left(1-\Pr\left(y_{i_j}\preceq t_1\right)\right)\\ &= P_{0,0}^{n-k}\left(1-\frac{1}{q-1}\sum_{m=1}^{q-1}P_{0,m}\right)^{k},\end{aligned}\tag{21}$$

where

$$\begin{aligned}P_{i,j} &= \Pr\left(\hat{x}=s_i|x=s_j\right)\\ &= \frac{1}{\sigma\sqrt{2\pi}}\int_{t_i}^{t_{i+1}}\exp\left(-\frac{(y-s_j)^2}{2\sigma^2}\right)dy\\ &= \Phi\left(\frac{t_{i+1}-s_j}{\sigma}\right)-\Phi\left(\frac{t_i-s_j}{\sigma}\right),\end{aligned}\tag{22}$$

and the $\preceq$ and $\succ$ symbols are the element wise $\leq$ and $>$ symbols, respectively. Note that $\mathbf{Y}_{\mathcal{E}}$ and $\mathbf{Y}_{\mathcal{I}}$ are the sub-vectors of $\mathbf{Y}$ indexed by $\mathcal{E}$ and $\mathcal{I}$, respectively. The probability of error is then $\Pr\left(\hat{\mathcal{I}}\neq\mathcal{I}\right)=1-\Pr\left(\hat{\mathcal{I}}=\mathcal{I}\right)$. When a dynamic index read reference is employed, the activation pattern is detected in error if the maximum voltage among the initially erased cells is greater than the minimum voltage among the initially programmed cells, i.e.,

$$\Pr\left(\hat{\mathcal{I}}\neq\mathcal{I}\right)=\Pr\left(\min_{j\in\mathcal{I}}\{y_j\}\leq\max_{j\in\mathcal{E}}\{y_j\}\right).\tag{23}$$

Assuming the distributions of these minimum and maximum are known, by conditioning on $\max_{j\in\mathcal{E}}\{y_j\}$ we obtain

$$\Pr\left(\hat{\mathcal{I}}\neq\mathcal{I}\right)=\int_{-\infty}^{+\infty}f_{\max_{j\in\mathcal{E}}\{y_j\}}(x)\,F_{\min_{j\in\mathcal{I}}\{y_j\}}(x)\,dx,\tag{24}$$

where $F_{\min_{j\in\mathcal{I}}\{y_j\}}(x)$ is the c.d.f of the mimimum voltage among programmed cells and $f_{\max_{j\in\mathcal{E}}\{y_j\}}(x)$ is the p.d.f of the maximum voltage among the erased cells. Since the noise is i.i.d, we can use order statistics to calculate the p.d.f of $f_{\max_{j\in\mathcal{E}}\{y_j\}}(x)$. Furthermore because the $q-1$ programmed states are i.i.d between them, we can also use order statistics [30] to calculate the c.d.f of $F_{\min_{j\in\mathcal{I}}\{y_j\}}(x)$. This leads to

$$\begin{aligned}f_{\max_{j\in\mathcal{E}}\{y_j\}}(x) &= \frac{n-k}{\sigma\sqrt{2\pi}}\exp\left(-\frac{(x-s_0)^2}{2\sigma^2}\right)\\ &\quad\times\Phi^{n-k-1}\left(\frac{x-s_0}{\sigma}\right),\end{aligned}\tag{25}$$

and

$$F_{\min_{j\in\mathcal{I}}\{y_j\}}(x)=1-\left[1-\frac{1}{(q-1)}\sum_{\ell=1}^{q-1}\Phi\left(\frac{x-s_\ell}{\sigma}\right)\right]^{k}.\tag{26}$$

When fixed detection is employed it is easy to see that $\Pr\left(\hat{\mathbf{F}}=\mathbf{F}\right)$ is equivalent to $\Pr\left(\hat{\mathbf{X}}=\mathbf{X}\right)$, hence we can write

$$\begin{aligned}\Pr\left(\hat{\mathbf{F}}=\mathbf{F}\right) &= \prod_{i=1}^{n}\Pr\left(\hat{x}_i=x_i\right)\\ &= \left(\sum_{j=0}^{q-1}\Pr\left(x_i=s_j\right)P_{j,j}\right)^{n},\end{aligned}\tag{27}$$

where $\Pr\left(x_i=s_j\right)=1-\alpha$ for $j=0$ and $\frac{\alpha}{q-1}$ for $j\in\{1,\ldots,q-1\}$. Since $\hat{\mathbf{F}}$ is detected after detecting the activation pattern, we can write the following for dynamic detection

$$\begin{aligned}\Pr\left(\hat{\mathbf{F}}=\mathbf{F}\right) &= \Pr\left(\hat{\mathcal{I}}=\mathcal{I}\right)\Pr\left(\hat{\mathbf{F}}=\mathbf{F}|\hat{\mathcal{I}}=\mathcal{I}\right)\\ &\quad+\Pr\left(\hat{\mathcal{I}}\neq\mathcal{I}\right)\Pr\left(\hat{\mathbf{F}}=\mathbf{F}|\hat{\mathcal{I}}\neq\mathcal{I}\right).\end{aligned}$$

As an approximation, we assume that it is very unlikely to correctly detect $\mathbf{F}$ when the activation pattern is erroneously detected. This means that the second term of (28) can be ignored and we can write

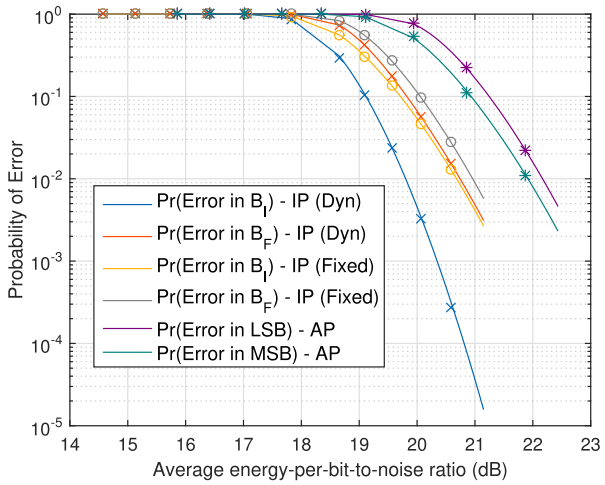
$$\Pr\left(\hat{\mathbf{F}}=\mathbf{F}\right)\approx\Pr\left(\hat{\mathcal{I}}=\mathcal{I}\right)\Pr\left(\hat{\mathbf{F}}=\mathbf{F}|\hat{\mathcal{I}}=\mathcal{I}\right),\tag{28}$$

where

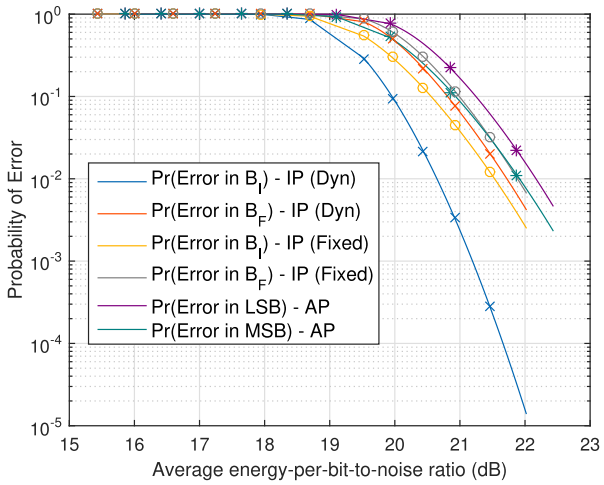
$$\begin{aligned}\Pr\left(\hat{\mathbf{F}}=\mathbf{F}|\hat{\mathcal{I}}=\mathcal{I}\right) &= \prod_{i=1}^{k}\Pr\left(\hat{f}_i=f_i|f_i\in Q\backslash s_0\right)\\ &= \left(\sum_{j=1}^{q-1}\Pr\left(f_i=s_j|f_i\in Q\backslash s_0\right)\Pr\left(\hat{f}_i=s_j|f_i=s_j\right)\right)^{k}\\ &= \left(\frac{1}{q-1}\sum_{j=1}^{q-1}P_{j,j}\right)^{k}\end{aligned}\tag{29}$$

where $\Pr\left(x_i=s_j\right)=1/(q-1)$ for $j\in\{1,\ldots,q-1\}$ and $P_{1,1}=\Phi\left((t_2-s_j)/\sigma\right)$, while for $j=2,3$, $P_{j,j}$ is defined as in (22).

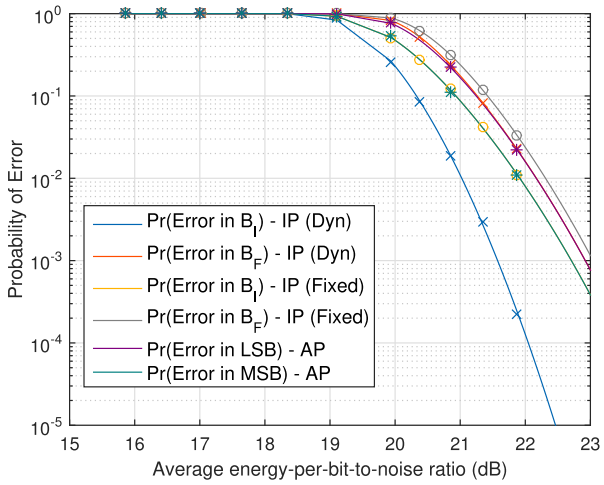
The validity of the above approximation as well as that of the theoretical expressions derived in this subsection is verified in the next section by comparing them to our simulation results.



(a) $\alpha = 0.5$ and $\eta = 1.792$ bits/cell.



(b) $\alpha = 0.675$ and $\eta = 1.979$ bits/cell.



(c) $\alpha = 0.75$ and $\eta = 1.9995$ bits/cell.

Fig. 6. Theoretical (solid lines) and simulated (markers) performance of the AP and IP schemes in Gaussian noise, $n = 16383$ and $q = 4$.

C. Power Consumption Analysis

In both the AP and IP schemes, a memory block is initially in the erased state. We assume that in order to program a cell into any state $s_i \in \{s_1, \ldots, s_{q-1}\}$, an amount of energy $\mathrm{E}_i = |s_0 - s_i|^2$ is required. This amount reflects the fact that more energy is needed to program cells into higher voltages. Since the AP scheme uses equally likely symbols, the average energy consumption per cell is given by

$$\bar{\mathrm{E}}_{AP} = \frac{1}{q}\sum_{i=1}^{q-1} \mathrm{E}_i. \tag{30}$$

On the other hand, the IP scheme changes the distribution of the input and the average energy consumption per cell is given by:

$$\bar{\mathrm{E}}_{IP} = \frac{\alpha}{q-1}\sum_{i=1}^{q-1} \mathrm{E}_i. \tag{31}$$

Finally, we define a parameter that encompasses the energy consumption and the storage efficiency as well as the noise power. We call this parameter the average energy-per-bit-to-noise ratio (AEBNR) and define it as follows:

$$\bar{\gamma}_{AP} = \frac{1}{\lfloor \log_2 q \rfloor}\frac{\bar{\mathrm{E}}_{AP}}{\sigma^2}, \tag{32}$$

$$\bar{\gamma}_{IP} = \frac{n}{b_1 + b_2}\frac{\bar{\mathrm{E}}_{IP}}{\sigma^2}. \tag{33}$$

This parameter allows for a fair comparison between the error performances of AP and IP.

D. Simulation Results

To simulate the proposed scheme we set $n = 16383$ cells on each word-line, $q = 4$ and set the states to be equally spaced with $[s_0, s_1, s_2, s_3] = [1, 1.75, 2.5, 3.25]$. We consider a channel with Gaussian noise, and simulate the performance of IP with dynamic and fixed reference detection as well as the performance of traditional AP. The results for $\alpha = 0.5$, 0.675 and 0.75 are shown in Figs. 6a, 6b and 6c, respectively (markers) along with their theoretical counterparts evaluated

TABLE I
PERFORMANCE GAIN OF IP AT 10^{-2}

Gain of 'x' over MSB/LSB	$\alpha = 0.5, \eta = 1.79$	$\alpha = 0.675, \eta = 1.979$	$\alpha = 0.75, \eta = 1.9995$
index page (dyn)	$2.11/2.36$ dB	$1.27/1.52$ dB	$1.05/1.34$ dB
amplitude page (dyn)	$1.17/1.42$ dB	$0.2/0.45$ dB	$-0.29/-0.00$ dB
index page (fixed)	$1.28/1.48$ dB	$0.38/0.64$ dB	$-0.00/0.28$ dB
amplitude page (fixed)	$0.95/1.2$ dB	$0.03/0.28$ dB	$-0.45/-0.16$ dB

in equations (19), (20), (21), (24), (27) and (28) (solid lines), as a function of AEBNR (in dB). Since in AP the MSB page (green) shows more error resiliency than the LSB page (purple) we will compare it to the index page while we compare the LSB page to the amplitude page. For $\alpha = 0.5$ both index and amplitude pages are more resilient to errors than the MSB and LSB pages in AP. This robustness comes at the expense of a 10.4% reduction in storage efficiency. Furthermore, the index page performs better with dynamic detection than with fixed detection, as does the amplitude page. Dynamic detection can provide 0.9 dB of advantage over fixed detection to maintain a 10^{-2} error rate in the index page. For $\alpha = 0.675$ (Fig.6b), a slight 1% reduction in storage efficiency compared to AP is observed. The index page of IP still performs better than the MSB page for both dynamic and fixed detection. The amplitude page of IP performs better than the MSB page for dynamic detection and almost the same for fixed detection with a slight advantage for the first in the high AEBNR region and for the second in the low AEBNR region. For $\alpha = 0.75$, the storage efficiency is almost the same as AP (0.025% reduction). Fig.6c shows that, using IP, it is possible to replace the MSB and LSB page of traditional AP by the index and amplitude pages, respectively, such that the index page performs better than the MSB page (1 dB gain), and the amplitude page performs similarly to the LSB page using dynamic detection. While less complex, fixed detection is not beneficial for high storage efficiency regimes since the index and MSB page performs similarly while the amplitude page performs slightly worse than the LSB page. It is very important to note that the dynamic detection of the index page has the advantage that the gap between its performance and the MSB page grows larger with AEBNR. Table I summarizes the difference between IP and AP in the required average energy-per-bit-to-noise ratio to achieve a 10^{-2} page error rate. A positive performance gain indicates a better performance of the IP page.

To provide a sensible estimation of the endurance gain provided by the IP scheme, we assume that the noise power is a function of only the number of P/E cycles. According to [27], the noise power scales with the number of P/E cycles in an approximate power law fashion, i.e. $\sigma^2 = aN^\beta$ where a is a constant and N is the number of P/E cycles. In this case a reduction of g dB in the required average energy-to-noise-ratio when IP is employed translates into an endurance extension of the device by a factor of $10^{\frac{g}{10\beta}}$ in cycles. For example, a reduction of 2.27 dB corresponds to approximately 2.5 times more P/E cycles before the error probability rises above 10^{-2} when $\beta = 0.6$, which is a typical power law according to [27]. The above results prove the potential of our proposed scheme in making at least one page more robust to errors.

VII. DISCUSSION

The results of the previous section clearly have shown the potential advantages of IP. However, it is worth noting that the Gaussian model is simplistic compared to a real flash memory channel, which is non stationary, data dependent and includes CCI. Consequently, the results underestimate the advantage of IP because they ignore its CCI reduction capability and exaggerate the performance of AP since the optimal read references are typically not readily available as it is the case in our simulations. Furthermore, dynamic detection does not require any knowledge about the channel to provide its superior performance. Hence, dynamic detection mitigates the need to estimate the channel characteristics in order to optimally detect the index page. Finally, it is worth mentioning that when performing a cycle, a cell left in the erased state will not suffer from wear-out, while cells in other states will. Consequently, and similar to the results in [21], an endurance gain can be obtained from IP by using an appropriate choice of k.

It is very important to clarify the point that IP, as it is presented in this contribution, is an uncoded programming technique, the performance of which could be enhanced with ECC. Indeed, one might envisage an IP strategy based on coding theory principles whereby an expurgated subset of the total available index set is chosen for IP based on minimum distance properties. Alternatively (or additionally), IP in this light could be viewed as an inner code in a concatenated scheme, with the outer code being a high rate LDPC code, BCH code, or even tamper-proof code such as the recently uncovered so-called malleable codes. In this sense, IP is not meant to replace ECC but to be used alongside it. IP can significantly reduce the raw bit error rates without suffering from a significant rate loss.

In our simulations, we considered only the page error rate without addressing the bit error rate. The bit error rate of the IP scheme will depend on the mapping employed to map information bits into activation patterns and vice-versa. Since there are $2^{\lfloor \log_2 \binom{n}{k} \rfloor}$ information messages and $\binom{n}{k}$ activation patterns, there are $D = \binom{n}{k} - 2^{\lfloor \log_2 \binom{n}{k} \rfloor}$ invalid activation patterns that the detector using a dynamic reference might detect. This will result in an activation pattern that cannot

be mapped back to an information message. This issue is a problem also in the recent OFDM with Index Modulation technique, and one recent solution was proposed by [25] where all activation patterns are considered valid by having two possible sizes (b_1 or $b_1 + 1$ bits) for information messages. Since it is beyond the scope of this paper to propose a good and efficient mapping, simulations were performed by generating random activation patterns (combinations) immediately instead of generating random data and mapping it into an activation pattern. In this light, the shown performance of IP can be considered as an upper bound on the actual performance provided a technique like the one in [25] is not applied. Otherwise it can be assumed that this scheme [25], is implicitly considered. The mapping of bits into activation patterns and vice-versa can be seen as a constant weight coding that maps between unconstrained binary b_1-tuples and constant weight binary n-tuples of weight k. A lot of work has been done on constant weight coding [31]–[33], with special focus on balanced codes ($k = n/2$) [34]–[36]. In the original OFDM-IM scheme [24], a mapping that is based on the combinatorial number system of degree k [37, p. 27-30] was employed. This method finds for each integer Z, k binomial coefficients $\binom{n}{c_i}$ such that $\sum_{i=1}^{k} \binom{n}{c_i} = Z$. For large n and k, this method requires evaluating large binomial coefficients, which makes it inefficient. Furthermore, this mapping is not well-behaved and a small change in the activation pattern due to noise might cause a huge number of bit errors.

VIII. Binary Amplitude Pages in IP

In section VI-A, we described how IP creates two logical pages on each word-line: the index page and the amplitude page. Here, we explain how the amplitude page can be divided into several logical pages that are written and read in a fashion that is very similar to AP when $q = 2^p + 1$.

In practice, binary pages within a word-line can be read independently, but can not be programmed independently. As an example, consider an MLC device using Gray labelling (Fig.2). Writing '1' in the MSB page is done by keeping a cell in the first voltage window ('11'), whereas writing '0' requires shifting it into the third window ('00'). To write into the LSB page, the MSB page is first read and the voltages of its cells are adjusted into the corresponding window to reflect a '0' or a '1' in the LSB positions. In general, it is necessary to have the number of levels be a power of two, i.e., $q = 2^p$ ($p = 1, 2,$ and 3 for SLC, MLC and TLC, respectively) in order for the AP scheme to maintain this independent binary paging structure. In the proposed IP scheme, and as explained in section VI-A, the word-line will always be divided into two pages. In IP, the binary paging structure is maintained by having $q = 2^p + 1$ instead of 2^p levels. The index page is programmed by selecting k cells accordingly and setting them to the first programmed (non-erased) state. This page should always be written before the other pages the same way the MSB page has to always be written before the LSB page in MLC. The result is a set of k cells that can be programmed to any of the 2^p programmed states. This means that the k cells can be dealt with as a word-line with k cells, each storing p bits so that they can be sequentially programmed in the same manner as AP explained above. Hence the total number of pages is $p + 1$. Fig. 7 illustrates the paging operation of IP when $q = 5$. First the index page is written by shifting k cells from the erased state (E) to state '11' where the k cells are chosen according to the input bits. To write the first amplitude page (MSB-IP), these k cells are first detected by applying the fixed reference t_1 or the more complex (but more robust) dynamic detector to identify the k programmed cells. Then for each of the k cells, it is either kept in its state or shifted to state '00', depending on the bit to be written on the MSB-IP page in that cell. Finally, to program the second amplitude page (LSB-IP), both the index and the MSB-IP pages need to be read and then the cells' voltages are adjusted to reflect a '0' or a '1' in the LSB-IP page.

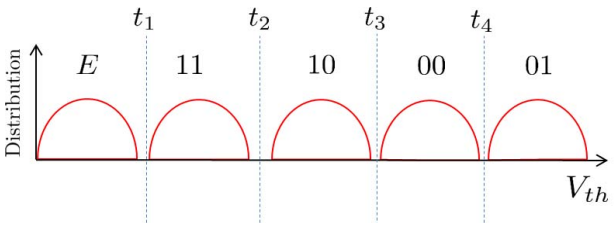


Fig. 7. Diagram showing the states and read references for $q = 5$.

The read process is performed in a very similar way to that of AP. For the example above, the index page is read by detecting the k programmed cells (using fixed or dynamic reference). The information bits can then be retrieved by de-mapping the activation pattern. To read the MSB-IP page, the programmed cells should be detected first and so that t_3 can be applied to identify their MSB bits. To read the LSB-IP page, the programmed cells are first detected, and then t_2 and t_4 are applied to identify the LSB bits. When a fixed reference is used to detect the programmed cells, the process turns out to be very similar to reading the pages of a TLC device. In a TLC device, a single reference is required to read the first page, two for the second and three for the third page. It is important to note that in this case t_1 is applied when reading any page, as opposed to traditional reading where different pages need a different set of references. This is an advantage when several pages on the same word-line are to be read simultaneously. In general, when $q = 2^p + 1$ it is possible to divide a word-line into $p + 1$ logical pages using IP the same way as described above.

IX. Summary, Conclusion and Future Directions

In this paper we have introduced the new IP scheme, where the number of cells that are programmed in each word-line is fixed to k. We have derived the storage efficiency of the new scheme and concluded that a slightly reduced efficiency and additional complexity are the price that must be paid for enhanced error resiliency which translates into endurance gain. We have analyzed the CCI effect of this scheme and deduced that there is a trade-off between storage efficiency and the degree of CCI reduction. We have introduced two methods for detecting the proposed scheme and have shown how they can provide an advantage over AP. We have shown

that dynamic detection will provide a more robust and channel-independent detection compared to fixed detection, at the expense of added complexity. We discussed how $p+1$ logical pages can be constructed using IP with $q = 2^p + 1$. Finally, we have introduced briefly the problem of mapping between information bits and activation patterns.

This paper showed the potential of IP, but also shed light on some of its problems. Future possible directions include analyzing the implementation complexity and the error performance on real devices, solving the problem of mismatch between the number of information messages and that of activation patterns, applying error correcting codes over the set of activation patterns and considering different generalizations of the scheme.

ACKNOWLEDGMENT

The authors would like to thank the Toshiba Telecommunications Research Laboratory and its directors for supporting this work. Particular thanks go to M. Ismail for his constructive feedback on the work.

REFERENCES

[1] J. Brewer and M. Gill, Eds., *Nonvolatile Memory Technologies With Emphasis on Flash: A Comprehensive Guide to Understanding and Using Flash Memory Devices*. Hoboken, NJ, USA: Wiley, 2008.

[2] L. M. Grupp, J. D. Davis, and S. Swanson, "The bleak future of NAND flash memory," in *Proc. 10th USENIX Conf. File Storage Technol.*, 2012, p. 2.

[3] Y. Koh, "NAND flash scaling beyond 20 nm," in *Proc. IEEE Int. Memory Workshop (IMW)*, May 2009, pp. 1–3.

[4] R. Katsumata *et al.*, "Pipe-shaped BiCS flash memory with 16 stacked layers and multi-level-cell operation for ultra high density storage devices," in *Proc. IEEE Symp. VLSI Technol.*, Jun. 2009, pp. 136–137.

[5] G. Dong, N. Xie, and T. Zhang, "On the use of soft-decision error-correction codes in NAND flash memory," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 58, no. 2, pp. 429–439, Feb. 2011.

[6] G. Dong, N. Xie, and T. Zhang, "Enabling NAND flash memory use soft-decision error correction codes at minimal read latency overhead," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 60, no. 9, pp. 2412–2421, Sep. 2013.

[7] 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.

[8] T. Parnell, N. Papandreou, T. Mittelholzer, and H. Pozidis, "Modelling of the threshold voltage distributions of sub-20 nm NAND flash memory," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2014, pp. 2351–2356.

[9] H. Wang, T.-Y. Chen, and R. D. Wesel, "Histogram-based flash channel estimation," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2015, pp. 283–288.

[10] B. Peleato, R. Agarwal, J. M. Cioffi, M. Qin, and P. H. Siegel, "Adaptive read thresholds for NAND flash," *IEEE Trans. Commun.*, vol. 63, no. 9, pp. 3069–3081, Sep. 2015.

[11] H. Zhou, A. Jiang, and J. Bruck, "Error-correcting schemes with dynamic thresholds in nonvolatile memories," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul./Aug. 2011, pp. 2143–2147.

[12] F. Sala, R. Gabrys, and L. Dolecek, "Dynamic threshold schemes for multi-level non-volatile memories," *IEEE Trans. Commun.*, vol. 61, no. 7, pp. 2624–2634, Jul. 2013.

[13] Y. Wu and A. Jiang, "Position modulation code for rewriting write-once memories," *IEEE Trans. Inf. Theory*, vol. 57, no. 6, pp. 3692–3697, Jun. 2011.

[14] A. Jiang, R. Mateescu, M. Schwartz, and J. Bruck, "Rank modulation for flash memories," *IEEE Trans. Inf. Theory*, vol. 55, no. 6, pp. 2659–2673, Jun. 2009.

[15] V. Taranalli, H. Uchikawa, and P. H. Siegel, "Channel models for multi-level cell flash memories based on empirical error analysis," *IEEE Trans. Commun.*, vol. 64, no. 8, pp. 3169–3181, Aug. 2016.

[16] Y. Cai, E. F. Haratsch, O. Mutlu, and K. Mai, "Error patterns in MLC NAND flash memory: Measurement, characterization, and analysis," in *Proc. IEEE Design, Autom. Test Eur. Conf. Exhibit. (DATE)*, Mar. 2012, pp. 521–526.

[17] A. Berman and Y. Birk, "Constrained flash memory programming," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul./Aug. 2011, pp. 2128–2132.

[18] S. Kayser and P. H. Siegel, "Constructions for constant-weight ICI-free codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun./Jul. 2014, pp. 1431–1435.

[19] S. Buzaglo, P. H. Siegel, and E. Yaakobi, "Coding schemes for inter-cell interference in flash memory," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2015, pp. 1736–1740.

[20] Y. Cassuto, M. Schwartz, V. Bohossian, and J. Bruck, "Codes for asymmetric limited-magnitude errors with application to multilevel flash memories," *IEEE Trans. Inf. Theory*, vol. 56, no. 4, pp. 1582–1595, Apr. 2010.

[21] A. Jagmohan, M. Franceschini, L. A. Lastras-Montaño, and J. Karidis, "Adaptive endurance coding for NAND flash," in *Proc. IEEE Globecom Workshops*, Dec. 2010, pp. 1841–1845.

[22] J. Wang, S. Jia, and J. Song, "Generalised spatial modulation system with multiple active transmit antennas and low complexity detection scheme," *IEEE Trans. Wireless Commun.*, vol. 11, no. 4, pp. 1605–1615, Apr. 2012.

[23] D. Tsonev, S. Sinanovic, and H. Haas, "Enhanced subcarrier index modulation (SIM) OFDM," in *Proc. IEEE GLOBECOM Workshops (GC Wkshps)*, Dec. 2011, pp. 728–732.

[24] E. Başar, U. Aygölü, E. Panayırcı, and H. V. Poor, "Orthogonal frequency division multiplexing with index modulation," *IEEE Trans. Signal Process.*, vol. 61, no. 22, pp. 5536–5549, Nov. 2013.

[25] A. I. Siddiq, "Low complexity OFDM-IM detector by encoding all possible subcarrier activation patterns," *IEEE Commun. Lett.*, vol. 20, no. 3, pp. 446–449, Mar. 2016.

[26] P. Pavan, R. Bez, P. Olivo, and E. Zanoni, "Flash memory cells—An overview," *Proc. IEEE*, vol. 85, no. 8, pp. 1248–1271, Aug. 1997.

[27] G. Dong, Y. Pan, N. Xie, C. Varanasi, and T. Zhang, "Estimating information-theoretical NAND flash memory storage capacity and its implication to memory system design space exploration," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 9, pp. 1705–1714, Sep. 2012.

[28] H. Yassine, J. Coon, M. Ismail, and H. Fletcher, "Towards an analytical model of NAND flash memory and the impact on channel decoding," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2016, pp. 1–6.

[29] J. G. Proakis and D. G. Manolakis, *Introduction to Digital Signal Processing*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1988.

[30] H. A. David and H. N. Nagaraja, *Order Statistics*. Hoboken, NJ, USA: Wiley, 1981.

[31] T. V. Ramabadran, "A coding scheme for m-out-of-n codes," *IEEE Trans. Commun.*, vol. 38, no. 8, pp. 1156–1163, Aug. 1990.

[32] A. E. Brouwer, J. B. Shearer, N. J. A. Sloane, and W. D. Smith, "A new table of constant weight codes," *IEEE Trans. Inf. Theory*, vol. 36, no. 6, pp. 1334–1380, Nov. 1990.

[33] V. Skachek and K. A. S. Immink, "Constant weight codes: An approach based on Knuth's balancing method," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 5, pp. 909–918, May 2014.

[34] D. Knuth, "Efficient balanced codes," *IEEE Trans. Inf. Theory*, vol. 32, no. 1, pp. 51–53, Jan. 1986.

[35] J. H. Weber and K. A. S. Immink, "Knuth's balanced codes revisited," *IEEE Trans. Inf. Theory*, vol. 56, no. 4, pp. 1673–1679, Apr. 2010.

[36] K. A. S. Immink and J. H. Weber, "Very efficient balanced codes," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 2, pp. 188–192, Feb. 2010.

[37] E. F. Beckenbach, G. Pólya, D. H. Lehmer, and M. Phister, *Applied Combinatorial Mathematics*. New York, NY, USA: Wiley, 1964.

Hachem Yassine received the B.Sc. in electrical engineering from the Lebanese University, Lebanon, in 2012, and the M.Sc. in communication networks and signal processing from the University of Bristol in 2013. He is currently pursuing the D.Phil. degree with the Engineering Science Department, University of Oxford. His research interests are information theory, signal processing, and their application to reliable data storage systems.

Justin P. Coon (S'02–M'05–SM'10) received the B.Sc. degree (Hons.) in electrical engineering from the Calhoun Honours College, Clemson University, USA, and the Ph.D. degree in communications from the University of Bristol, U.K., in 2000 and 2005, respectively. In 2004, he joined Toshiba Research Europe Ltd., as a Research Engineer, in their Bristol-based Telecommunications Research Laboratory (TRL), where he was involved in the research on a broad range of communication technologies and theories, including single and multi-carrier modulation techniques, estimation and detection, diversity methods, system performance analysis, and networks. He held the position of Research Manager from 2010 to 2013, during which time he led all theoretical and applied research on the physical layer with TRL. He was a Visiting Fellow with the School of Mathematics, University of Bristol, from 2010 to 2012, and was a Reader with the Department of Electrical and Electronic Engineering, University of Bristol, from 2012 to 2013. He joined the University of Oxford in 2013, where he is currently an Associate Professor with the Department of Engineering Science and a Tutorial Fellow of the Oriel College.

He is currently the Technical Manager of the EU FP7 Project DIWINE. He has authored over 100 papers in leading international journals and conferences, and is a named inventor on over 30 patents. His research interests include communication theory, information theory and network theory. He was a recipient of TRL's Distinguished Research Award for his work on block-spread CDMA, aspects of which have been adopted as mandatory features in the 3GPP LTE Rel-8 standard. He received the Award for Outstanding Contribution in 2014. He was also a co-recipient of two best paper awards for work presented at the ISWCS '13 and the EuCNC'14. He has served as an Editor of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS from 2007 to 2013, the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY from 2013 to 2016, and the IEEE WIRELESS COMMUNICATIONS LETTERS since 2016.

David E. Simmons received the degree in mathematics from the University of Central Lancashire in 2011, the M.Sc. degree in communications engineering from the University of Bristol, and the D.Phil. degree from the University of Oxford in 2016, with a focus on amplify- and-forward relay networks. He is currently a Post-Doctoral Research Associate with the University of Oxford. His research interests include classical/quantum information theory and communication theory. He was a recipient of a best paper award.