Pilot Assisted Adaptive Thresholding for Sneak-Path Mitigation in Resistive Memories with Failed Selection Devices

Zehui Chen, *Student Member, IEEE,* Clayton Schoeny, *Student Member, IEEE,* and Lara Dolecek, *Senior Member, IEEE*

***Abstract*—Resistive random-access memory (ReRAM) with the crossbar structure is one promising candidate to be used as a next generation non-volatile memory device. In a crossbar ReRAM, in which a memristor is positioned on each row-column intersection, the sneak-path problem is one of the main challenges for a reliable readout. The sneak-path problem can be solved with additional selection devices. When some selection devices fail short, the sneak-path problem re-occurs. The re-occurred sneak-path problem is addressed in this paper. The re-occurred sneak-path event can be described combinatorially and its adverse effect can be modeled as a parallel interference. Based on a simple pilot construction, we probabilistically characterize the inter-cell dependency of the re-occurred sneak-path events. Utilizing this dependency, we propose adaptive thresholding schemes for resistive memory readout using side information provided by pilot cells. This estimation theoretic approach effectively reduces the bit-error rate while maintaining low redundancy overhead and low complexity.**

***Index Terms*—Resistive memory, sneak-path, selection devices, adaptive thresholding, inter-cell dependency.**

I. Introduction

Crossbar Resistive random-access memory (ReRAM), in which a memristor is positioned on each row-column intersection of the crossbar structure, is considered to be a promising candidate to be used as a next generation non-volatile memory device because of its many unique advantages including a simple structure and high density [2]. One fundamental problem in purely passive crossbar ReRAM that demands detailed attention is the *sneak-path* problem [3]. When a cell in a crossbar array is read, a voltage is applied to the memristor and the resistance is measured to determine whether it is in Low-Resistance State (logic 1) or High-Resistance State (logic 0). Sneak paths are undesirable paths in parallel to the selected cell; they traverse through unselected cells. This problem is especially severe when a cell in High-Resistance State (logic 0) is read because parallel low resistances, due to sneak-paths, lower the resistance measured from the cell at High-Resistance State, thus causing difficulties in distinguishing between the High-Resistance State and the Low-Resistance State. The sneak-path problem is commonly solved by adding a selection device in series of each memory cell [4]. Adding diodes results in the 1D1R structure [5], [6]; adding transistors results in the 1T1R structure [7]; and adding selectors results in the 1S1R structure [8].

Although introducing selection devices can eliminate the sneak-path problem, selection devices are also prone to failure [9], [10]. As modern storage devices have ultra high reliability requirements, it is of interest to study crossbar resistive memories with failed selection devices. A study on crossbar resistive memories with failed selection devices can also aid a memory architecture designer in understanding how selection device reliability affects the memory reliability. In this work, we first focus on 1D1R structured arrays with shorted diodes, and then we generalize the results to 1S1R structured arrays with shorted selectors [11]. The case that diodes/selectors fail open is not considered in this work. In this case, the information stored in the cells to which the open diodes/selectors are connected, is lost, and this case is best treated with erasure correcting codes.

In a 1D1R/1S1R structured array, the sneak-path problem re-occurs when some diodes/selectors fail short, motivating the study on the re-occurred sneak-path problem and approaches to mitigate it. Similar to what is observed in [12], the events that cells on the same row/column are affected by the re-occurred sneak path problem are highly dependent. This dependency causes natural difficulties for standard coding theoretic solutions which assume independent bit-errors. Investigation of alternative techniques which can utilize this dependency is therefore of interest.

In this paper, our goal is to study the re-occurred sneak-path problem and to provide simple yet effective approaches to mitigate it. When viewing the effect of the re-occurred sneak-path problem as a *parallel interference*, the dependency between cells can be utilized to mitigate the re-occurred sneak-path problem. From an estimation theoretic point of view, we utilize the dependency between cells to provide improved estimation schemes based on our probabilistic characterization of the inter-cell dependency. We propose adaptive thresholding schemes that adaptively change the threshold using side information provided by pilot cells. With a simple pilot construction and our proposed adaptive thresholding schemes, we demonstrate up to 50% reduction in bit-error rate (BER) comparing to the fixed threshold scheme. In addition to ReRAM that is built out of memristors, our approaches are also applicable to other memory with crossbar structure, such as the Phase

Z. Chen, C. Schoeny, and L. Dolecek are with the Electrical and Computer Engineering Department, University of California, Los Angeles, Los Angeles, CA 90095, USA. (email: chen1046@ucla.edu; cschoeny@ucla.edu; dolecek@ee.ucla.edu).

This research is supported in part by a grant from UC MEXUS and an NSF-BSF grant no.1718389. Partial results in this paper were presented at the IEEE Information Theory Workshop, Guangzhou, China, Nov. 2018 (reference [1]).

Changing Memory (PCM).

The content of this paper is organized as follows. Section II provides modeling of the re-occurred sneak-path event and its adverse effect in two types of memory structure. The pilot construction used in this paper is also introduced in Section II. Section III formally characterizes the dependency of the re-occurred sneak-path event between the pilot cells and the information cells by calculating joint probabilities of the re-occurred sneak-path events among cells. In Section IV, the adaptive thresholding idea is introduced and the thresholds for different cases are derived. Section V provides bit-error rate analysis and present results comparing our proposed scheme with the fixed threshold scheme for various parameters. We present a case study comparing our proposed schemes with a comparable coding theoretic solution in Section VI. We conclude and discuss future research in Section VII.

II. Sneak-path Modeling and the Pilot Construction

In this paper, since it is clear that we are addressing the "re-occurred" sneak-path problem, we drop the repeated attribute "re-occurred" for simplified language. As an initial step, we use the sneak-path definition in [12] with modification to model the sneak-path event. In this work, we restrict ourselves to the sneak-path of length 3 because a sneak-path of length 3 is more likely to occur than sneak-paths of other lengths and the adverse effect of a sneak-path of length 3 is more prominent than the adverse effects of a sneak-path of other lengths. Other factors that affect the occurrence of the sneak-path problem, such as wire resistance, are not considered in this paper and are left for future work. In this section, we first propose the model of a sneak-path event for crossbar resistive memory with the 1D1R structure and then extend the model to the 1S1R structured array. Our models assume that the selection devices fails short i.i.d. with probability p_f due to reliability issues. Note that these models can be easily extended to crossbar resistive memory without any selection device by setting $p_f = 1$. We define $A \in \{0,1\}^{m\times n}$ to be the data matrix representing data stored in a crossbar resistive memory of size $m \times n$, and let A_{ij} denote the bit value at cell (i, j). For simplicity, we assume $m \leq n$ and that m divides n with $n/m = r$. The case when m does not divide n will be briefly discussed at the end of Section II.D. After presenting the modeling of the sneak-path event for the two different crossbar resistive memory structures, we describe a modeling of the adverse effect of a sneak-path event. At the end of this section, we provide the pilot construction that we use throughout this paper.

A. Modeling of the Sneak-path Event for the 1D1R Structure

By our definition, a sneak-path event occurs at cell (i, j) for an array with a 1D1R structure if the following three conditions are met:

1) The bit value stored is 0.
2) There exists at least one combination of $c \in [1, \cdots, m], s \in [1, \cdots, n], c \neq j, r \neq i$ that induces a sneak-path, defined by
$$A_{ic} = A_{sc} = A_{sj} = 1. \tag{1}$$
3) The diode at cell location (s, c) fails short.

We define e_{ij} to be a boolean random variable denoting the occurrence of the sneak-path event at location (i, j), conditioned on the bit value stored at cell (i, j) being 0. That is, $e_{ij} = 1$ if and only if the cell storing a 0 at position (i, j) incurs a sneak-path event. We also refer to the realization of e_{ij} as the sneak-path state of cell (i, j). Note that the sneak-path event is defined only for those cells that store 0's because the adverse effect of a sneak-path event on a cell that stores 1 is not detrimental to the read process. We have the third condition because in a 1D1R structured array, the diode in series prevents current from flowing in the reverse direction. The failure of the diode at cell location (s, c) to short circuit is therefore necessary to allow current flowing along the sneak-path.

B. Modeling of the Sneak-path Event for the 1S1R Structure

The only difference between our definition for the sneak-path event in a 1S1R structured array and the definition in a 1D1R structured array is the third condition. The third condition for a sneak-path event to occur at cell (i, j) in a 1S1R structured array is:

- The selectors at cell location (i, c), (s, c) and (s, j) fail short.

A cell selector is a device that has a high resistance when the voltage across it is below a certain threshold, e.g., when the cell is on the sneak-path. Therefore, all three selectors in series of the three cells on the sneak-path, i.e., cells that store A_{ic}, A_{sc} and A_{sj}, need to fail short for the sneak-path event to occur. Modeling for the 1D1R and 1S1R structures share the same notation. In this paper, when it is necessary to differentiate these two set-ups, it will be clear from the context which one is being considered.

C. Adverse Effect of a Sneak-Path Event

As this is an initial study, and with limited prior work, we focus on a simplified model described as follows. Our modeling of the adverse effect of a sneak-path event is adapted from [11] as a parallel interference. We first define the 0 state resistance of the memristor to be R_0, and the 1 state resistance of the memristor to be R_1. The adverse effect of a sneak-path event is modeled as a parasitic resistor with value R_s that is parallel to the read cell. We then denote r_{ij} to be the measured resistance value of cell (i, j), with an additive noise η. This additive noise captures two noise sources: the noise introduced by the resistance variation of resistive memory cells [13], and the measurement noise introduced by the sensing circuit [11]. To consider the noise introduced by the resistance variation, we first model the noise caused by the variation of HRS resistance and LRS resistance as Gaussian [14], [15]. As shown in [16], parallel combination of two Gaussian random variables can be approximated with another Gaussian random variable. As a result, we also model the

noise caused by the variation of the sneak-path resistance as a Gaussian additive noise added to the parallel combination of the sneak-path resistance and HRS resistance. Collectively, and for mathematical simplicity, we assume η to be Gaussian with mean 0 and variance σ^2, i.e., the same noise is added to the LRS resistance, the HRS resistance, and the parallel combination of the sneak-path resistance and HRS resistance. Adaptation to different noise models, including Gaussians with different variances are discussed at the end of Section IV B.

Together, we have the following model:

$$r_{ij} = \begin{cases} \left(\frac{1}{R_0} + \frac{e_{ij}}{R_s}\right)^{-1} + \eta & \text{when 0 is stored,} \\ R_1 + \eta & \text{when 1 is stored.} \end{cases} \tag{2}$$

In this paper, we assume $R_1 < (1/R_0 + 1/R_s)^{-1}$ to avoid analyzing degenerate scenarios.

D. Pilot Construction

In a crossbar resistive memory, and as noted in [12], it is not hard to observe that the occurrence of a sneak-path event at one cell is *not* independent of the occurrence of a sneak-path event at another cell. For example, knowing that $e_{ij} = 1$ increases the probability of $e_{ij^*} = 1, j^* \in [1, \cdots, n], j^* \neq j$, as well as $e_{i^*j} = 1, i^* \in [1, \cdots, m], i^* \neq i$. This special behavior of resistive memory presents natural difficulties to standard coding theoretic solutions when viewing the occurrence of a sneak-path event as a bit error. However, when viewing the effect of a sneak-path event as a parallel interference, one can utilize this inter-cell dependency to develop better estimation schemes based on side information provided by cells with known bit values.

We note that two cells are correlated the most when they are on the same row or column. It is also observed in [3] that the location of the cell (i and j) does not affect the probability of e_{ij}. Moreover, the event that two cells in the same row (column) simultaneously incur sneak-path events is also independent of the relative position of cells, i.e., the (joint) probability that these two cells simultaneously incur sneak-path events is independent of their positions, given that we do not consider wire resistance in this study. As a result, knowledge of the occurrence of a sneak-path event at a cell provides the same information as for all other cells on the same row (column). Based on this observation, we wish to explore robustness to selection device failure when a cell is allowed to access a few other cells with known states by utilizing their inter-cell dependency. The simplest yet effective case is that each information cell (a cell that can store either 0 or 1), has at least one pilot cell (a cell with known state) on its row and at least one pilot cell on its column. We therefore propose the following pilot construction.

Pilot Construction 1. *Let $A \in \{0,1\}^{m\times n}$ denote the data matrix representing data stored in a crossbar resistive memory array of size $m \times n$, and let A_{ij} denote the bit value at cell (i, j). We assume $m \leq n$ (A is fat) and m divides n with $n/m = r$. We preset all cells (i, j) such that $i \equiv j \pmod m$ to store 0, i.e., $A_{ij} = 0$, if $i \equiv j \pmod m, \forall i \in \{1, ..., m\}, j \in \{1, ..., n\}$. These preset cells are pilot cells and the rest of the cells are information cells.*

For an array of size $m \times n$, to have at least one pilot cell on the row of each information cell, we need at least m pilot cells. To have at least one pilot cell on the column of each information cell, we need at least n pilot cells. Together, to have at least one pilot cell on both row and column of each information cell, we need at least n pilot cells. Therefore the rate is at most $(m-1)/m$. It is easy to check that Pilot construction 1 achieves this upper bound and satisfies our pilot cell requirements.

The general term "pilot cells" can be thought of as the set of some pre-selected cells that are accessed first in order to infer certain characteristics of the array. In lithium battery, pilot cells are tested to determine the charge and discharge current for the entire battery. In [17], pilot cells are used to estimate the sneak current in resistive memory. In our work, pilot cells are used to estimate the sneak-path states of other cells on their rows and columns. Specifically, we use the resistance of pilot cells with known resistance states to infer the probabilities of other cells incurring the sneak-path events. We provide details on the usage of pilot cells in both Section III and Section IV.

With this simple pilot construction, each cell at location (i, j), with $i \not\equiv j \pmod m$, has 1 pilot cell that stores a 0 in its column and r pilot cells that store 0's in its row. We will use these pilot cells for our improved estimation schemes in latter section. Although there are r pilot cells in the row of an information cell, we only use one of them. Here and elsewhere, we define $j' = j \mod m + m\mathbb{1}(j \mod m = 0), i' = m\lfloor j/m \rfloor + i - m\mathbb{1}(j \mod m = 0)$ where $\mathbb{1}(\cdot)$ is the indicator function. For an information cell that stores A_{ij}, with $i \not\equiv j \pmod m$, we refer to the cells that store $A_{j'j}$ and $A_{ii'}$ to be its two reference cells. This construction generalizes the construction in [1] that is only applicable to a square array, i.e., $m = n$. For an array where m does not divide n, Pilot Construction 1 is still applicable. In this case, some of the information cells at the lower right side of the array do not have reference cells in their rows as defined before. Instead, they can use other pilot cells on their rows as the reference cells. The probabilistic characterization for this case could be done using the same techniques that will be discussed in the following section. We omit the explicit discussion of it because, while considerably more tedious, it does not provide further insight.

III. Probabilities and Joint Probabilities of the Sneak path Event

In the previous section, we proposed a pilot construction for the resistive memory array. In order to utilize the known 0's stored in pilot cells for more informed estimation schemes, several important probabilities need to be calculated analytically. First, in order to determine the sneak-path state of a pilot cell, we need $P_0(e_{ij})$ for $i \equiv j \pmod m$. Second, in order to use the sneak-path state of pilot cells for improved estimation schemes, we need $P(e_{ij}|e_{ii'}, A_{ij} = 0)$, $P(e_{ij}|e_{j'j}, A_{ij} = 0)$ and $P(e_{ij}|e_{ii'}, e_{j'j}, A_{ij} = 0)$ for $i \not\equiv j \pmod m$. The three sets of conditional probabilities are used in different estimation

schemes in latter sections. To make a comparison with the scheme that does not use any side information, we also need $P(e_{ij}|A_{ij} = 0)$. Although similar probabilistic characterizations of sneak-path event(s) for a single cell and for two cells can be found in [12], modifications need to be made to consider the pilot construction. In addition, characterizing dependency between three cells is necessary to allow for a more accurate estimation scheme.

To keep the discussion clear, in this section, we simply state the probabilities that are used to obtain the necessary probabilities mentioned above, and we refer readers to Appendix for the lemmas that calculate these probabilities. All probabilities that we stated are conditioned on the usage of our pilot construction, and, for clarity, we omit explicitly specifying this condition. Apart from $P_0(e_{ij})$, the probabilities stated in this section also conditioned on the event $A_{ij} = 0$. We also omit specifying this condition explicitly, and instead use the subscript 0 to highlight that $P_0(e_{ij})$ is not conditioned on a known information cell. We assume that the bit values to be stored in information cells are chosen i.i.d. Bernoulli with parameter q representing the prior probability of a 1 being stored. Note that although we include the indexes i and j in the argument of probabilities, these probabilities are independent of i and j. We only require i and j to satisfy certain conditions based on the context.

All probabilities have two versions, one for the 1D1R structure and one for the 1S1R structure. In this section and the following sections, we do not differentiate them in most cases unless specified. In Appendix, we provide lemmas that calculate the stated probabilities for the 1D1R structure. The corresponding probabilities for the 1S1R structure can be calculated using the following claim, which readily follows from the definitions of the sneak-path event in Sections II.A and II.B.

Claim 1. *For a 1S1R structured array, let $p_f^{(1S1R)}$ be the selector failure probability, and $q^{(1S1R)}$ be the prior probability of a 1 being stored. In order to calculate a certain probability for the 1S1R structured array for which the corresponding equation for the 1D1R structure is already available, it suffices to use that equation for the 1D1R structure with substitution: $p_f = 1$ and $q = q^{(1S1R)} \times p_f^{(1S1R)}$.*

To obtain the set of probabilities $P_0(e_{ij})$, we calculate $P_0(e_{ij} = 0)$ (Lemma 1 in Appendix). To obtain the set of probabilities $P(e_{ij})$, we calculate $P(e_{ij} = 0)$ (Lemma 2 in Appendix). There are two reference cells for each information cell. We first consider the case of using the sneak-path state of only one reference cell. To obtain probabilities of form $P(e_{ij}|e_{ii'})$ and $P(e_{ij}|e_{j'j})$, we first need $P(e_{ij} = 0, e_{ii'} = 0)$ (Lemma 3 in Appendix) and $P(e_{ij} = 0, e_{j'j} = 0)$ (Lemma 4 in Appendix). We also need $P(e_{ii'} = 0)$ (Lemma 5 in Appendix) and $P(e_{j'j} = 0)$ (Lemma 6 in Appendix). When using the sneak-path states of two reference cells, we need $P(e_{ij}|e_{ii'}, e_{j'j})$. To obtain the set of probabilities $P(e_{ij}|e_{ii'}, e_{j'j})$, we first need $P(e_{ij} = 0, e_{ii'} = 0, e_{j'j} = 0)$ (Lemma 7 in Appendix). We also need the set of probabilities $P(e_{ii'}, e_{j'j})$ obtained by calculating $P(e_{ii'} = 0, e_{j'j} = 0)$ (Lemma 8 in Appendix).

IV. Adaptive Thresholding Schemes

With all necessary conditional probabilities calculated, we propose our adaptive thresholding schemes. We propose two schemes, the *Single Reference Scheme* based on a single reference cell, and the *Double Reference Scheme* based on two reference cells. The *Single Reference Scheme* is further separated into two sub-schemes, the *Single Reference (Row) Scheme* that uses the reference cell on the row of an information cell, and the *Single Reference (Column) Scheme* that uses the reference cell on the column of an information cell. We need to consider both the *Single Reference (Row) Scheme* and the *Single Reference (Column) Scheme* because in a strictly non-square array, due to asymmetry, the two reference cells do not provide the same side information. For comparison, we also state the *No Reference Scheme* which uses no side information. For the two adaptive thresholding schemes, first we determine the sneak-path states of the reference cells for a targeted information cell, then, based on these sneak-path states, we choose appropriate thresholds to decide the states of the information cells to be read. All decisions are made through threshold estimators for implementation simplicity. In this section and the next section, the probability density function of the Gaussian noise η in Equation (2) is defined to be $f_\eta(\cdot)$.

In the remainder of this section, we first calculate the threshold of the threshold estimator that determines the sneak-path states of pilot cells. Then, we calculate the thresholds of the threshold estimators that determine the states of information cells, assuming known sneak-path states. Afterwards, we further discuss the proposed adaptive thresholding schemes that are based on the derived threshold estimators.

A. Optimal Threshold Estimation for Pilot Cells

We define τ_0 to be the threshold of the threshold estimator used to determine whether or not a sneak-path event occurs at a pilot cell. For a given resistance measurement r_{ij}, taken for a pilot cell (i, j) where $i \equiv j \pmod m$, the output of this threshold estimator is

$$\hat{e}_{ij} = \begin{cases} 1 & \text{if} \quad 0 \le r_{ij} \le \tau_0, \\ 0 & \text{if} \quad \tau_0 \le r_{ij} \le \infty. \end{cases} \tag{3}$$

Next, we derive the optimal threshold for this threshold estimator. For an arbitrary pilot cell that stores $A_{ij} = 0$ with $i \equiv j \pmod m$, there are two hypotheses, $e_{ij} = 0$ and $e_{ij} = 1$. Based on our modeling in Equation (2), the posterior functions of the two hypotheses can be expressed as,

$$\Lambda_{e_{ij}=0}(r_{ij}) = f_\eta\left(r_{ij} - R_0\right) P_0(e_{ij} = 0), \tag{4}$$

and

$$\Lambda_{e_{ij}=1}(r_{ij}) = f_\eta\left(r_{ij} - \left(\frac{1}{R_0} + \frac{1}{R_s}\right)^{-1}\right) P_0(e_{ij} = 1), \tag{5}$$

where $P_0(e_{ij} = 0)$ and $P_0(e_{ij} = 1)$ can be calculated using Lemma 1 in Appendix.

Minimizing the average error probability of this threshold estimator by Bayes Criterion gives the condition:

$$\Lambda_{e_{ij}=1}(\tau_0) = \Lambda_{e_{ij}=0}(\tau_0). \tag{6}$$

Solving (6) gives the following optimal threshold:

$$\tau_0 = \frac{1}{2}\frac{R_0^2 - \left(\frac{1}{R_0}+\frac{1}{R_s}\right)^{-2} + 2\sigma^2 \log\left(\frac{P_0(e_{ij}=1)}{P_0(e_{ij}=0)}\right)}{R_0 - \left(\frac{1}{R_0}+\frac{1}{R_s}\right)^{-1}}. \tag{7}$$

B. Optimal Threshold Estimation for Information Cells

In this subsection, we calculate the optimal thresholds for the three thresholding schemes, assuming the sneak-path states of the pilot cells are known. The three thresholding schemes are alike and only differ in the side-information used for each scheme. It is thus convenient to use c to denote the side-information, i.e., sneak-path state(s), used in each scheme to describe the three schemes collectively. Depending on the corresponding scheme, c may be nothing, a scalar or a vector. For an arbitrary information cell with $i \not\equiv j \pmod m$, we let $c = \textit{none}$ for the *No Reference Scheme*, $c = e_{ii'}$ for the *Single Reference (Row) Scheme*, $c = e_{j'j}$ for the *Single Reference (Column) Scheme*, and $c = [e_{ii'}, e_{j'j}]$ for the *Double Reference Scheme*.

There are two hypotheses for an information cell (i, j), $A_{ij} = 0$ and $A_{ij} = 1$. We are interested in the threshold estimators that decide between these two hypotheses, while considering different realizations of c. Let $\tau(c)$ be the threshold used for a particular thresholding scheme under a certain realization of the corresponding c. For example, $\tau(\textit{none})$ is the threshold to be used when no sneak-path state of reference cells is used; $\tau(e_{ii'} = 1, e_{j'j} = 1)$ is the threshold to be used when the *Double Threshold Scheme* is used and both reference cells of this information cell incur sneak-path events. Therefore, when reading cell (i, j), the output of the threshold estimator, using the selected threshold $\tau(c)$ where c is known sneak-path state(s) of reference cell(s), is:

$$\hat{A}_{ij} = \begin{cases} 1 & \text{if} \quad 0 \le r_{ij} \le \tau(c), \\ 0 & \text{if} \quad \tau(c) \le r_{ij} \le \infty. \end{cases} \tag{8}$$

Based on the selected thresholding scheme and the realization of c, the posterior functions of each hypothesis as a function of r_{ij} are

$$\begin{aligned}\Lambda_{A_{ij}=0}(r_{ij}) = (1-q)\Bigg[& f_\eta(r_{ij} - R_0)P(e_{ij}=0|c) \\ & + f_\eta\left(r_{ij} - \left(\frac{1}{R_0}+\frac{1}{R_s}\right)^{-1}\right)P(e_{ij}=1|c)\Bigg],\end{aligned} \tag{9}$$

and

$$\Lambda_{A_{ij}=1}(r_{ij}) = q f_\eta(r_{ij} - R_1), \tag{10}$$

where $P(e_{ij} = 0|c)$ and $P(e_{ij} = 1|c)$ can be calculated using Lemmas 2-8 in Appendix.

Minimizing the average error probability of the threshold estimator by Bayes Criterion gives the following condition for the optimal threshold:

$$\Lambda_{A_{ij}=1}(\tau(c)) = \Lambda_{A_{ij}=0}(\tau(c)). \tag{11}$$

Note that unlike in Equation (6) where both sides have simple Gaussian density functions, Equation (11) has the right hand side expressed as an addition of two Gaussian density functions. As a result, Equation (11) has two solutions, and the solution lying in the middle of the two distributions (9) and (10) is the desired threshold that minimizes the error probability. We can solve $\tau(c)$ easily using available numerical methods, e.g., using *vpasolve*($\Lambda_{A_{ij}=1}(\tau(c)) = \Lambda_{A_{ij}=0}(\tau(c))$,$[R_1, R_0]$) in MATLAB.

We quickly remark that in the case that a closed form solution of $\tau(c)$ is needed, one can use the following approximation:

$$\tau(c) \approx \frac{1}{2}\frac{\left(\frac{1}{R_0}+\frac{1}{R_s}\right)^{-2} - R_1^2 + 2\sigma^2 \log\left(\frac{q}{(1-q)P(e_{ij}=1|c)}\right)}{\left(\frac{1}{R_0}+\frac{1}{R_s}\right)^{-1} - R_1}. \tag{12}$$

This approximation follows by replacing the two Gaussian density functions in (9) with a single Gaussian density function, i.e., $(1-q)f_\eta\left(r_{ij} - \left(\frac{1}{R_0}+\frac{1}{R_s}\right)^{-1}\right)P(e_{ij} = 1|c)$, which is closest to the Gaussian density function in (10). As a result, this approximation is valid when R_0 and $\left(\frac{1}{R_0}+\frac{1}{R_s}\right)^{-1}$ are sufficiently apart, and when $f_\eta\left(r_{ij} - \left(\frac{1}{R_0}+\frac{1}{R_s}\right)^{-1}\right)P(e_{ij} = 1|c) \gg f_\eta(r_{ij} - R_0)P(e_{ij} = 0|c)$ around $\left(\frac{1}{R_0}+\frac{1}{R_s}\right)^{-1}$. While being closed-form, this approximation loses the optimality of the threshold estimator. For the BER results in latter section, we use the thresholds that are solved numerically.

Note that although Gaussian additive noise with the same variance and zero mean is used for mathematical simplicity, other models of the additive noise, such as log-normal distribution [18], or Gaussian additive noise with different variances for the HRS, LRS, and sneak-path resistances, can be easily adopted with appropriate changes in the thresholds calculations, e.g., by using different η in (2) and replacing $f_\eta(\cdot)$ with an appropriate density function(s). The case that the diode is partially shorted (ohmic) can be considered as a bias term among the noise modeling for the sneak-path resistances. If the noise distribution is unimodal, the validity of the approximation in (12) follows analogously to the Gaussian case.

Also note that, for mathematical tractability, only one sneak-path of length 3 is considered in this paper. As a result of this simplification, the sneak-path resistance R_s is the dominant term when the sneak-path event occurs. In a more precise model, one storage cell can be affected by multiple sneak-paths with different structures, [19]. Probabilistic characterization of how a single cell is affected by multiple sneak-paths is provided in [19]. Meanwhile, characterization of how multiple cells are affected by differently structured multiple sneak-paths

is still an open problem. To adopt the adaptive thresholding schemes, one would need to consider sneak-path(s) with different structures as separate events. For example, the content within the bracket of Equation (9) would be replaced with the summation of many terms. Each term would be the product between the pdf of the measured resistance given a sneak-path event with a particular length and structure, and the probability of this sneak-path event given the sneak-path state(s) of the reference cell(s). The sneak-path state of a reference cell would also have multiple states (not just 0 and 1) when multiple sneak-paths with different structures are considered. In this paper, given the limited space, we only provide an analysis for the to the elementary case to illustrate the basic idea of adaptive thresholding.

C. Estimation Using Adaptive Thresholding Schemes

Because the failures of selection devices are hard and are typically found during the writing process, we seek to re-estimate the sneak-path states of pilot cells after a write operation:

- Whenever new data is written to the memory, except for the case when the *No Reference Scheme* is used, measure r_{ij} for each pilot cell (i,j) with $i \equiv j \pmod m$. Then use τ_0 to decide $\hat{e}_{ij}$, and store these estimated sneak-path states of pilot cells.

To read an information cell (i,j), i.e., the cell that stores A_{ij} with $i \not\equiv j \pmod m$, the following procedure is performed depending on the selected adaptive thresholding scheme:

- If the *No Reference Scheme* is used, measure r_{ij} and use $\tau(\textit{none})$ to decide $\hat{A}_{ij}$.
- If the *Single Reference (Row or Column) Scheme* is used, retrieve the previously stored $\hat{e}_{ii'}$ or $\hat{e}_{j'j}$. Then measure r_{ij} and use $\tau(e_{ii'} = \hat{e}_{ii'})$ or $\tau(e_{j'j} = \hat{e}_{j'j})$ to decide $\hat{A}_{ij}$.
- If the *Double Reference Scheme* is used, retrieve the previously stored $\hat{e}_{ii'}$ and $\hat{e}_{j'j}$. Then measure r_{ij} and use $\tau(e_{ii'} = \hat{e}_{ii'}, e_{j'j} = \hat{e}_{j'j})$ to decide $\hat{A}_{ij}$.

From the above statements, we notice that the adaptive thresholding schemes are particularly efficient for applications in which memory cells are read more frequently than they are written. In addition, the *Single Reference (Column) Scheme*, in a fat array, possesses a special property amenable for simple implementation: all memory cells on the same column use the same threshold, which depends on the estimated sneak-path state of the pilot cell in this column. Therefore, this threshold could be hardware configured to be used for the entire column.

V. BER Analysis and Results

In Section IV, we proposed adaptive thresholding schemes and the associated threshold estimators. Based on these proposed methods, in this section, we derive an analytical result for the bit-error rate (BER), and present a comparison between different schemes.

To read an information cell (i,j), we assume resistance measurements r_{ij}, $r_{ii'}$ and $r_{j'j}$ are available. Which r is being used depends on which thresholding scheme is being selected. Let $c = [e_{ii'}, e_{j'j}]$ denote the true sneak-path states for the two reference cells, and let C denote the support of c. Let $\hat{c}$ denote the estimated sneak-path state(s) for zero ($\hat{c} = \textit{none}$), one ($\hat{c} = \hat{e}_{ii'}$ or $\hat{c} = \hat{e}_{j'j}$) or two ($\hat{c} = [\hat{e}_{ii'}, \hat{e}_{j'j}]$) reference cells, depending on the selected thresholding scheme, and let $\hat{C}$ be the support of the corresponding $\hat{c}$. Since the sneak-path events involving the two reference cells are only weakly dependent, for the sake of simplified analysis of $P_r(c)$ in Equation (13), and without sacrificing qualitative findings, we shall assume that they are independent. Specifically, since $P_r(c)$ is only a weight term in Equation (13), unlike in the threshold calculations, this calculation is insensitive to the small error introduced by the independence assumption. For the exact evaluation of $P_r(e_{ii'}, e_{j'j})$, one can use $P_r(e_{ii'} = 0, e_{j'j} = 0) = (1-q)P(e_{ii'} = 0, e_{j'j} = 0 | A_{ij} = 0) + qP(e_{ii'} = 0, e_{j'j} = 0 | A_{ij} = 1)$,where $P(e_{ii'} = 0, e_{j'j} = 0 | A_{ij} = 0)$ is calculated in Lemma 8 and $P(e_{ii'} = 0, e_{j'j} = 0 | A_{ij} = 1)$ can be obtained using an argument similar to Lemma 8.

Let $P(\hat{A}_{ij} \neq A_{ij})$ be the average probability of a decision error. We have the following expression:

$$P(\hat{A}_{ij} \neq A_{ij}) = \sum_{c \in C} P_r(c) \sum_{\hat{c} \in \hat{C}} P(\hat{c}|c) P(\hat{A}_{ij} \neq A_{ij} | c, \hat{c}). \tag{13}$$

In Equation (13), $P_r(c)$ is the probability of certain realizations of c, and can be calculated using $P_r(c) = P_0(e_{ij} = e_{ii'}) P_0(e_{ij} = e_{j'j})$; $P(\hat{c}|c)$ is the probability of deciding $\hat{c}$ given c; $P(\hat{A}_{ij} \neq A_{ij} | c, \hat{c})$ is the decision error probability given certain realizations of c and $\hat{c}$, where the latter governs the choice of the threshold. Probabilities $P(\hat{c}|c)$ for the *No Reference Scheme* and the *Single Reference (Row/Column) Scheme* are summarized in Table I.

Probability $P(\hat{c}|c)$ for the *Double Reference Scheme* is calculated as follows using the probabilities in Table I:

$$P([\hat{e}_{ii'}, \hat{e}_{j'j'}]|c) = P(\hat{e}_{ii'}|c) P(\hat{e}_{j'j'}|c). \tag{14}$$

The decision error probability $P(\hat{A}_{ij} \neq A_{ij} | c, \hat{c})$ is given by the following equation:

$$P(\hat{A}_{ij} \neq A_{ij} | c, \hat{c}) = qQ\left(\frac{\tau(\hat{c}) - R_1}{\sigma}\right) + (1-q)\Bigg[P(e_{ij} = 1|c) \times Q\left(\frac{\frac{R_0 R_s}{R_0 + R_s} - \tau(\hat{c})}{\sigma}\right) + P(e_{ij} = 0|c) Q\left(\frac{R_0 - \tau(\hat{c})}{\sigma}\right)\Bigg], \tag{15}$$

where q is the prior probability of 1 being stored. In the above equations, $Q(\cdot)$ is the Q-function, i.e., $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty \exp(-\frac{u^2}{2}) du$. These probabilities are obtained using our modeling, the estimators introduced in Section IV, and simple estimation theory.

With the theoretical expression for the bit-error rate derived, we evaluate the results for different schemes. In the following evaluations, we use prior probability $q = 0.5$ and the following resistance values, which are the same as in [11]: $R_1 = 100\Omega$, $R_0 = 1000\Omega$, $R_s = 250$ (the value of the parameter R_s will be changed for the upcoming Figure 4) and $q = 0.5$. We vary the parameters σ, n and p_f to test their influence on the performance of our proposed schemes.

TABLE I

No Reference Scheme				
$P(none\|c)$	1			
Single Reference (Row) Scheme				
case	$\hat{e}_{ii'}=0, e_{ii'}=0$	$\hat{e}_{ii'}=1, e_{ii'}=1$	$\hat{e}_{ii'}=1, e_{ii'}=0$	$\hat{e}_{ii'}=0, e_{ii'}=1$
$P(\hat{e}_{ii'}\|c)$	$Q\left(\frac{\tau_0-R_0}{\sigma}\right)$	$Q\left(\frac{\frac{R_0R_s}{R_0+R_s}-\tau_0}{\sigma}\right)$	$Q\left(\frac{R_0-\tau_0}{\sigma}\right)$	$Q\left(\frac{\tau_0-\frac{R_0R_s}{R_0+R_s}}{\sigma}\right)$
Single Reference (Column) Scheme				
case	$\hat{e}_{ii'}=0, e_{ii'}=0$	$\hat{e}_{ii'}=1, e_{ii'}=1$	$\hat{e}_{ii'}=1, e_{ii'}=0$	$\hat{e}_{ii'}=0, e_{ii'}=1$
$P(\hat{e}_{ii'}\|c)$	$Q\left(\frac{\tau_0-R_0}{\sigma}\right)$	$Q\left(\frac{\frac{R_0R_s}{R_0+R_s}-\tau_0}{\sigma}\right)$	$Q\left(\frac{R_0-\tau_0}{\sigma}\right)$	$Q\left(\frac{\tau_0-\frac{R_0R_s}{R_0+R_s}}{\sigma}\right)$

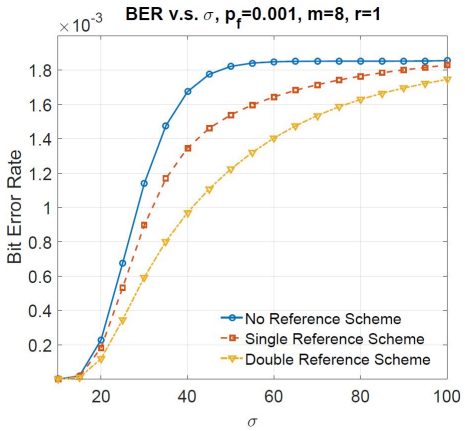


Fig. 1: BER for a 1D1R structured square array with various noise magnitude.

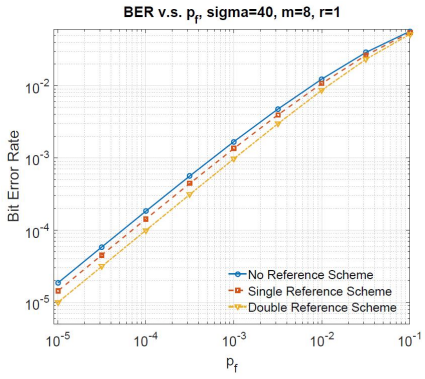


Fig. 2: BER for a 1D1R structured square array with various p_f.

We first evaluate results for a square array, i.e., $r = 1$. In this case, there is no need to differentiate the *Single Reference (Row) Scheme* and the *Single Reference (Column) Scheme*. We first set $m = 8$, $p_f = 0.001$, and vary σ from 10 to 100. The results are shown in Figure 1. We observe that both the *Single Reference Scheme* and the *Double Reference Scheme* offer noticeable improvements in terms of BER compared with the *No Reference Scheme*. Using the *Double Reference Scheme* provides better BER than using the *Single Reference Scheme*. We observe that under moderate noise (σ from 20 to 40), we can get consistently over 40% reduction in BER by using the *Double Reference Scheme*, and over 20% reduction in BER by using the *Single Reference Scheme*. In the high noise regime ($\sigma > 40$), all three schemes start to saturate and the improvement offered by our proposed schemes becomes smaller. However, by using our proposed schemes, the saturation of BER becomes slower. For example, for a targeted BER of 1.6×10^{-3}, in terms of the noise standard deviation σ, 50% more noise can be tolerated by using the *Single Reference Scheme* and almost 100% more noise can be tolerated by using the *Double Reference Scheme*.

Next, we set $\sigma = 40$, $m = 8$, and vary p_f from 1×10^{-1} to 1×10^{-5}. The results are shown in Figure 2. We observe that as the diode becomes more reliable, i.e. as p_f becomes smaller, the improvement resulting from our proposed schemes increases. We conjecture that as the diodes become more reliable, the sneak-path events become more rare so that the information provided by the reference cells impacts the threshold more profoundly. For example, for $p_f = 1 \times 10^{-1}$, knowing $e_{ii'} = 1$ only changes $P(e_{ij} = 1|none) = 0.3017$ to $P(e_{ij} = 1|e_{ii'} = 1) = 0.5609$. In contrast, for $p_f = 1 \times 10^{-4}$, knowing $e_{ii'} = 1$ changes $P(e_{ij} = 1|none) = 0.0003749$ to $P(e_{ij} = 1|e_{ii'} = 1) = 0.4168$. The relative larger change of $P(e_{ij} = 1|c)$ in the latter case shifts the threshold more towards the left thus reducing the estimation error more effectively.

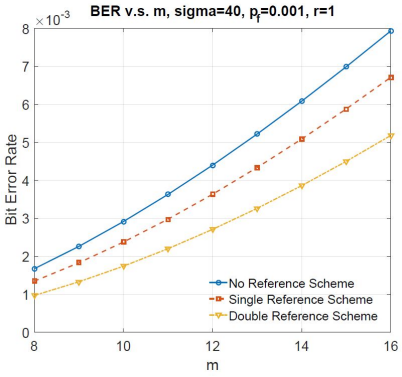


Fig. 3: BER for a 1D1R structured square array with various array dimension.

Next, we set $\sigma = 40$, $p_f = 0.001$ and vary m from 8 to 16 to test the performance of our proposed schemes in arrays with different sizes. The results are shown in Figure 3. We observe

consistent improvement in using the proposed schemes in the range of tested array size. We also observe that as the array size gets larger, the relative improvement using our proposed scheme decreases by a small amount, akin to the reasoning for decrease in performance improvement in the case of increased p_f. For a large array, one can divide the large array into small sub-arrays to keep the relative larger improvement. However, this approach induces a larger circuit overhead and a larger storage overhead, resulting in the familiar trade-off between reliability and overhead observed in e.g., channel coding. Also, for a large array, i.e., large m, the probabilities in Appendix are hard to compute due to the combinatorial nature of these calculations. This difficulty can potentially limit the usage of the proposed schemes but it can be mitigated in two ways. First, these probabilities are only computed once for an array of a given size so these probabilities along with the induced thresholds can be precomputed at the design stage. Second, if the exact probabilities are prohibitively difficult to compute, one can use the approximations mentioned at the end of the Appendix to compute the approximated probabilities and then the induced suboptimal thresholds. This simplified approach still maintains most of the relative gain of the adaptive thresholding schemes while reducing the burden of computing those probabilities exactly.

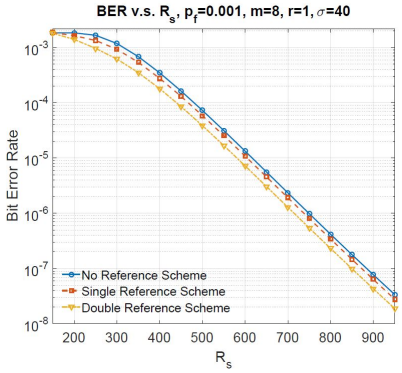


Fig. 4: BER for a 1D1R structured square array with various sneak-path resistance.

We also test the sensitivity of our proposed schemes under different values of the sneak-path resistance R_s, which is governed by cell nonlinearity [20]. We set $\sigma = 40$, $p_f = 0.001$ and vary R_s from 150 to 950. As shown in Figure 4, the improvement using our proposed schemes is consistent throughout this range, except when R_s is very small, i.e., in the case when $\frac{R_0 R_s}{R_0+R_s}$ is very close to R_1. When $\frac{R_0 R_s}{R_0+R_s}$ is very close to R_1, knowledge of the sneak-path state(s) of reference cell(s) does not change threshold significantly. As a result, the adaptive thresholding schemes are ineffective. From this result, we can also infer the performance of our proposed scheme for other values of the R_0, R_1 parameters, as the Q-function in the BER calculation is invariant to the same scaling of R_0, R_1, R_s, and σ.

Next, we turn our attention to the performance of the proposed schemes for an array with non-unity aspect ratio, i.e., for $r \neq 1$. We first set $m = 8$, $r = 2$, $p_f = 0.001$ and vary σ from 10 to 100. The results are shown in Figure

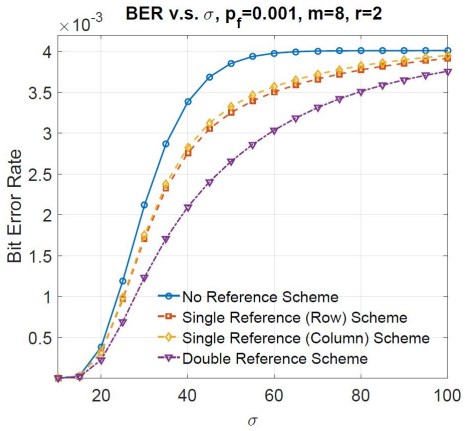


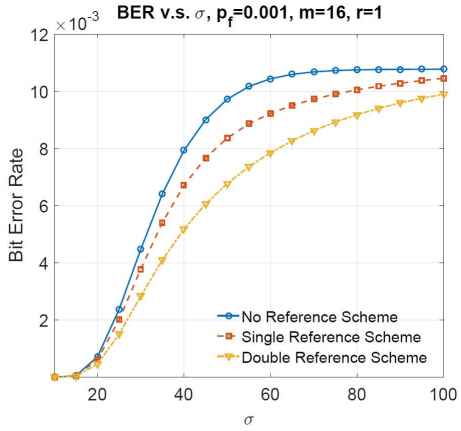
Fig. 5: BER for a 1D1R structured 8×16 array with various noise magnitude.

5. From the results, we note that the *Single Reference (Row) Scheme* outperforms *Single Reference (Column) Scheme*. This observation suggests that the reference cell in the row of an information cell provides more information than the reference cell in the column of this information cell. Intuitively, with more available information cells in the row than in the column, an information cell is more likely to incur a sneak-path event given that we know a sneak-path event occurs at the reference cell in its row than given that we know a sneak-path event occurs at the reference cell in its column. For example, for the parameters in Figure 5, we have $P(e_{ij} = 0, e_{ii'} = 0) = 0.9959$ and $P(e_{ij} = 0, e_{j'j} = 0) = 0.9955$.

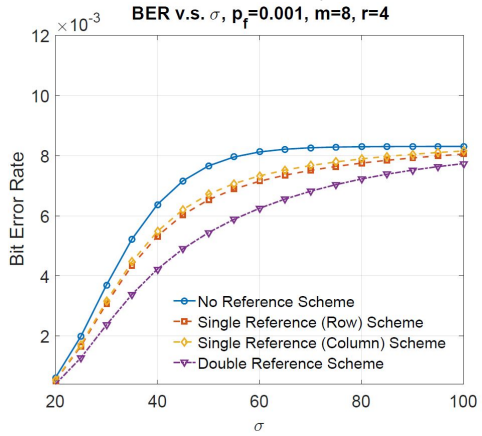
Next, we present the results for two arrays with the same total number of cells, one array being a square array and the other array being a non-square array. We set $\sigma = 40$, $p_f = 0.001$. The square array is 16×16 with 240 information cells, and the non-square array is 8×32 with 224 information cells. The results are reported in Figure 6. We observe that although the non-square array has fewer information cells relative to the square array, the non-square array is less susceptible to bit-errors relative to the square array under the same amount of noise and the same thresholding scheme. Intuitively, this can be partially explained by the observation that a decrease in one dimension (here from 16 to 8), which decreases the probability of a sneak-path event, has a stronger effect on the occurrence of sneak-path event than the same simultaneous multiplicative increase in the other dimension (here from 16 to 32), which in contrast increases the probability of a sneak-path event. For example, the extreme case of the array of size 1×256 is indeed free of sneak-path events.

Finally, we report the performance for an array with a 1S1R structure. We set $m = 8$, $p_f = 0.1$, and vary σ from 10 to 100. We choose $p_f = 0.1$ because the 1S1R structure is less susceptible to the sneak-path event compared to the 1D1R structure. 1S1R structure with $p_f = 0.1$ gives $P(e_{ij} = 0)$, comparable to $P(e_{ij} = 0)$ for 1D1R structure with $p_f = 0.001$. The result is reported in Figure 7.

We observe that although the *Single Reference Scheme* and the *Double Reference Scheme* still offer improvements in terms of BER, the improvements are incremental. This observations



(a) 16×16 array



(b) 8×32 array

Fig. 6: BER for two 1D1R structured arrays with same number of cells.

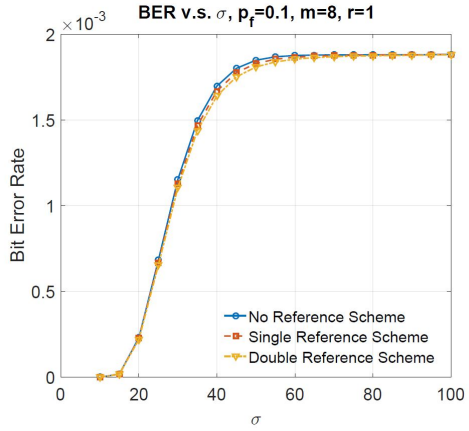


Fig. 7: BER for a 1S1R structured 8×8 array with various noise magnitude.

suggests that, in the 1S1R structured array, with the stronger isolation provided by the cell selectors, the dependency of the sneak-path event between the reference cells and the information cell is weaker than this dependency is in the 1D1R structured array. For example, in a 1S1R structured array, with $m = 8$, $r = 1$ and $p_f = 10^{-0.5}$, we get $P(e_{ij} = 0) = 0.8896$, $P(e_{ij} = 0|e_{ii'} = 0) = 0.8961$, $P(e_{ij} = 0|e_{ii'} = 0, e_{j'j} = 0) = 0.8325$, and $P(e_{ij} = 0|e_{ii'} = 1, e_{j'j} = 1) = 0.3899$; in a 1D1R structured array, with $m = 8$, $r = 1$ and $p_f = 10^{-1.5}$, we get $P(e_{ij} = 0) = 0.8961$, $P(e_{ij} = 0|e_{ii'} = 0) = 0.9433$, $P(e_{ij} = 0|e_{ii'} = 0, e_{j'j} = 0) = 0.9709$, and $P(e_{ij} = 0|e_{ii'} = 1, e_{j'j} = 1) = 0.1208$. It is clear from this example that with comparable $P(e_{ij} = 0)$, in an array with the 1S1R structure, the reference cells provide less side information than the amount of side information provided by reference cells in an array with the 1D1R structure.

VI. A Comparison with the BCH $(239, 255)$ Code

The adaptive thresholding approach proposed in this paper is not a direct substitute to the channel coding approaches. First, the adaptive thresholding approach requires extra read operations and multiple threshold detectors while the channel coding approaches require a channel encoder/decoder. Second, this adaptive thresholding approach, which is estimation theoretic, reduces the BER for a certain noise magnitude while some channel coding techniques have guaranteed error-correction capability.

A comparison between the adaptive thresholding approach and the channel coding approaches is nonetheless valuable, as it can help in better understanding the property of this special problem (the re-occurrence of sneak-path due to selection device failure) and can provide insight into the memory design. We therefore present the following case study.

We consider 1D1R structured 16×16 arrays. For a 16×16 array, 16 cells are used as pilot cells when our proposed pilot construction is used. We compare our adaptive thresholding schemes using the pilot construction with the standard BCH $(239, 255)$ code, which corrects up to 2 bit-errors also also uses 16 bits of redundancy. To make the BCH code compatible with the array architecture, we assume that the BCH $(239, 255)$ code can correct up to 2 bit-errors among 256 (instead of 255) bits. In our simulations, we set $p_f = 0.001, R_0 = 1000, R_1 = 100, R_s = 250, q = 0.5$ and vary σ. Note that for the BCH coded array without the pilot construction, we also use the optimal threshold, which is calculated similar to $\tau(none)$, to determine the state of a cell.

In Figure 8, under both linear and log scale, we report our simulation results of the raw bit-error rate (RBER) for an array with pilot cells, with the No Reference Scheme and the Double Reference Scheme. We also report the RBER and undetectable bit-error rate (UBER) for an array that is coded with the BCH $(239, 255)$ code. To gain further insight on how the inter-cell dependency of sneak-path events affects the coding performance, we also report a lower bound of the UBER for BCH $(239, 255)$ coded array assuming bit-errors are independent.

It is first worth noting that by simply using the pilot construction without adaptively changing the threshold, i.e., using the No Reference Scheme, we get better RBER performance comparing to the RBER of the array without the pilot construction. This can be explained as follows: by using the pilot construction, we effectively reduce the fraction of cells with LRS. The pilot construction can be considered as a

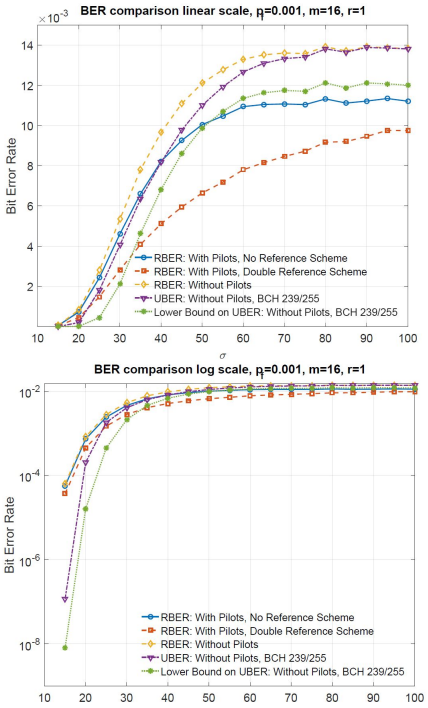


Fig. 8: A Comparison with the BCH $(239, 255)$ Code

simple shaping code in this case. As a result, the probability of a sneak-path event is also reduced, as noted in [12]. The reduced sneak-path probability therefore reduces the RBER. Second, by comparing the UBER of the BCH $(239, 255)$ code with its lower bound, which is obtained using the RBER in the BCH coded array under the independence assumption, we note that the bit-errors are highly dependent. In the rare case that a diode fails, all cells are more prone to error due to the adverse effect of the sneak-path. Most of bit-errors are concentrated in this case and are therefore dependent.

Next, we compare the performance of the coding approach and the adaptive thresholding approach in two representative noise regimes. In the low noise regime ($\sigma < 25$), even if the sneak-path event occurs, the number of bit-errors is usually less than 3, i.e., the number is within the error-correction capability of the BCH $(239, 255)$ code. We observe that in this case, the BCH $(239, 255)$ code reduces BER very effectively. The implication for the memory design is that if the additive noise is small, e.g., there is a large read margin, the sneak-path problem is not severe and it can be handled with an appropriate ECC. In the high noise regime we studied, when a diode fails, the number of bit-errors is often beyond the error-correction capability of the BCH $(239, 255)$ code because the adverse effect of the sneak-path event dominates in this setting. In this case, we observe a much smaller coding gain compared to the improvement achieved by using the Double Reference Scheme with pilot cells. Note that in this high noise regime, while this simple error-correction code is ineffective, the Double Reference Scheme only reduces the number of errors but does not eliminate them entirely, in most cases. This observation opens up the possibility of combining our proposed adaptive thresholding schemes with more sophisticated error-correction code over large (multiple) array(s). In this sense, the adaptive thresholding schemes can be viewed as a preprocessing set that provides a lower RBER for the subsequent ECC solution.

VII. Conclusion

In this paper, utilizing the inter-cell dependency of the re-occurred sneak-path events, we provide light-weight estimation theoretic schemes to mitigate the re-occurred sneak-path problem in resistive memory with failed selection devices. For future theoretical study, the adaptive thresholding technique can be combined with more sophisticated error-correction code. The proposed schemes can also serve as the starting point to studying the incorporation of more precise modeling, including but not limited to the noise model that is not necessarily Gaussian, multiple sneak-paths with different structures, and the ohmic failure modeling of the diodes. SPICE simulation with real memristor model can be also done to test our adaptive thresholding schemes.

Appendix

In the appendix, under the modeling of the sneak-path event of a 1D1R structured array, we calculate the probabilities introduced in Section III. Note that although some of the latter lemmas we present may look complicated, they in fact rely on elementary combinatorics of the sneak-path event defined by our modeling. We include Figure 10 that contains some of the useful notation and indexes used in the proof of following lemmas. We comment on possible approximations for the following lemmas at the end of the appendix. We use the binomial coefficient $\binom{n}{k}$ that is defined for $0 \leq k \leq n$ and equals 0 otherwise.

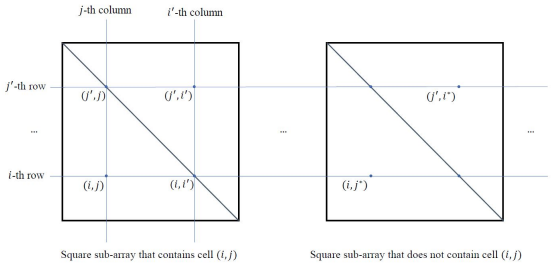


Fig. 9: Some notation and indexes used in the proofs.

Lemma 1. *Without conditioning on any known information cell, the probability that a pilot cell (cell that stores A_{ij} with $i = j'$,i.e., $j = i'$) does not incur a sneak-path event is:*

$$P_0(e_{ij} = 0) = \sum_{u=0}^{m-1} \binom{m-1}{u} q^u (1-q)^{m-1-u} \left[P_0(e_{ij} = 0|u)\right]^r, \tag{16}$$

where

$$P_0(e_{ij} = 0|u) = \sum_{v=0}^{m-1} \sum_{k=0}^{\min(u,v)} \left[\binom{u}{k} \binom{m-1-u}{v-k} \times q^v (1-q)^{m-1-v} (1-p_f q)^{uv-k} \right]. \tag{17}$$

Proof: This probability is derived by conditioning on certain parameters and then summing, over all possible configurations, the probabilities that each of these configurations does not cause a sneak-path event at cell (i,j) with $i = j'$, i.e., $j = i'$. The term configuration, here and elsewhere, refers to an array (sub-array) with known information (depending on the parameters) on the row(s) and column(s) of cell(s) of interest, i.e., cell(s) which we want to not incur a sneak-path event. We first condition on the number of 1's on the j-th column (u), i.e., we select u rows, with an index set I_u, that have 1's on the j-th column. We then divide the fat array into r square sub-arrays, and consider each square sub-array separately, conditioned on the selected u rows. This condition allows us to consider each square sub-array independently. For each square sub-array, we condition on the number of 1's on the i-th row (v), i.e., we select v columns, with an index set I_v, that have 1's on the i-th row. Here and elsewhere in the appendix, since we consider the square sub-arrays separately, all index sets have indexes that represent the positions in the square sub-array. Therefore, we define all index sets to be subsets of $I_{all} = \{1, \cdots, m\} \backslash \{i, j'\}$. For each square sub-array, we also condition on $k = |I_u \cap I_v|$. For each configuration with parameters u, v and k, in order to guarantee no sneak-path event at cell (i,j), we need the $uv - k$ information cells, lying on the intersection of the u rows and v columns, to either store a 0 or store a 1 with a non-failing diode (with probability $1 - p_f q$). ∎

In the following lemmas, the probabilities are all conditioned on $A_{ij} = 0$ (omitted for clarity) where $i \neq j'$, i.e., $j \neq i'$.

Lemma 2. *The probability that an information cell does not incur a sneak-path event is:*

$$P(e_{ij}=0) = \sum_{u=0}^{m-2} \left\{ \binom{m-2}{u} q^u (1-q)^{m-2-u} \times P^{(1)}(e_{ij}=0|u) \left[P^{(2)}(e_{ij}=0|u)\right]^{r-1} \right\}, \tag{18}$$

where

$$P^{(1)}(e_{ij}=0|u) = \sum_{v=0}^{m-2} \sum_{k=0}^{\min(u,v)} \left\{ \binom{u}{k} \binom{m-2-u}{v-k} \times q^v (1-q)^{m-2-v} (1-p_f q)^{uv-k} \right\}, \tag{19}$$

and

$$\begin{aligned} &P^{(2)}(e_{ij}=0|u) \\ =&qP^{(1)}(e_{ij}=0|u)(1-p_f q)^u + (1-q)P^{(1)}(e_{ij}=0|u). \end{aligned} \tag{20}$$

Proof: We first condition on selecting u rows that have 1's on the j-th column. Again, we divide the fat array into r square sub-arrays. For an information cell, the probabilities that no sneak-path event occurs at cell (i,j), conditioned on u selected rows, are different for the square sub-array that contains cell (i,j) and the $r-1$ square sub-arrays that do not contain cell (i,j), and thus need to be considered separately. We use the superscripts (1) and (2) to denote this difference in our probability calculation. The calculation of $P^{(1)}(e_{ij}=0|u)$ is similar to $P_0(e_{ij}=0|u)$ in Lemma 1 except $|I_{all}| = m-2$. The calculation of $P^{(2)}(e_{ij}=0|u)$ is divided into two cases, and the probability for each case can be calculated by utilizing $P^{(1)}(e_{ij}=0|u)$. In the case that the cell (i, j^*), $j^* \equiv j \pmod m$, stores a 0, the configuration of this square sub-array is the same as the configuration of the square sub-array that contains cell (i,j). In the case that the cell (i,j^*) stores a 1, we additionally require u cells on the intersection of j^*-th column and the u rows to either store a 0 or store a 1 with a non-failing diode. ∎

Lemma 3. *Let $P(e_{ij}=0, e_{ii'}=0)$ denote the joint probability that both an information cell that stores 0 and the reference cell on the same row do not incur sneak-path events simultaneously. We have:*

$$\begin{aligned} P(e_{ij}=0, e_{ii'}=0) = &\sum_{u=0}^{m-2} \sum_{u'=0}^{m-2} \sum_{o=0}^{\min(u,u')} \left\{ \binom{m-2}{u} \binom{u}{o} \right. \\ &\times \binom{m-2-u}{u'-o} q^{u+u'} (1-q)^{2m-4-u-u'} \\ &\times \Big[(1-q) P^{(1)}(e_{ij}=0, e_{ii'}=0|u,u',o,A_{j'i'}=0) \\ &\times \left[P^{(2)}(e_{ij}=0, e_{ii'}=0|u,u',o,A_{j'i'}=0)\right]^{r-1} \\ &+ q P^{(1)}(e_{ij}=0, e_{ii'}=0|u,u',o,A_{j'i'}=1) \\ &\left. \times \left[P^{(2)}(e_{ij}=0, e_{ii'}=0|u,u',o,A_{j'i'}=1)\right]^{r-1} \Big] \right\}, \end{aligned} \tag{21}$$

where

$$\begin{aligned} &P^{(1)}(e_{ij}=0, e_{ii'}=0|u,u',o,A_{j'i'}=0) = \\ &\sum_{v=0}^{m-2} \sum_{k=0}^{\min(v,\, u+u'-o)} \left\{ \binom{u+u'-o}{k} \binom{m-2-u-u'+o}{v-k} \right. \\ &\left. \times q^v (1-q)^{m-2-v} (1-p_f q)^{v(u+u'-o)-k} \right\}, \end{aligned} \tag{22}$$

$$\begin{aligned} &P^{(2)}(e_{ij}=0, e_{ii'}=0|u,u',o,A_{j'i'}=0) \\ &= qP^{(1)}(e_{ij}=0, e_{ii'}=0|u,u',o,A_{j'i'}=0)(1-p_f q)^{u+u'-o} \\ &+ (1-q)P^{(1)}(e_{ij}=0, e_{ii'}=0|u,u',o,A_{j'i'}=0), \end{aligned} \tag{23}$$

$$\begin{aligned} &P^{(1)}(e_{ij}=0, e_{ii'}=0|u,u',o,A_{j'i'}=1) \\ &= \sum_{v=0}^{m-2} \sum_{k=0}^{\min(v,u+u'-o)} \left\{ \binom{u+u'-o}{k} \binom{m-2-u-u'+o}{v-k} \right. \\ &\left. \times q^v (1-q)^{m-2-v} (1-p_f q)^{v(u+u'-o+1)-k} \right\}, \end{aligned} \tag{24}$$

and

$$\begin{aligned} &P^{(2)}(e_{ij}=0, e_{ii'}=0|u,u',o,A_{j'i'}=1) \\ &= qP^{(1)}(e_{ij}=0, e_{ii'}=0|u,u',o,A_{j'i'}=1)(1-p_f q)^{u+u'-o} \\ &+ (1-q)P^{(1)}(e_{ij}=0, e_{ii'}=0|u,u',o,A_{j'i'}=1), \end{aligned} \tag{25}$$

Proof: To calculate $P(e_{ij}=0, e_{ii'}=0)$, we condition on selecting u rows, with an index set I_u, that have 1's on the j-th

column. We also select u' rows, with an index set $I_{u'}$, that have 1's on the i'-th column and let $o = |I_u \cap I_{u'}|$. Conditioned on these three parameters, we further divide the calculation into four cases. We separately consider whether or not the square sub-array contains cell (i,j) and whether or not cell (j',i') stores a 0. For the two cases that the square sub-array contains cell (i,j), we select v columns, with an index set I_v, that have 1's on the i-th row. We also condition on $k = |I_v \cap (I_u \cup I_{u'})|$. For each configuration with parameters u, u', o, v and k, the number of information cells that need to either store a 0 or store a 1 with a non-failing diode is $v(u+u'-o)-k$ when $A_{j'i'}=0$, and $v(u+u'-o+1)-k$ when $A_{j'i'}=1$. For the two cases that the square sub-array does not contain cell (i,j), we consider two sub-cases, conditioning on the value stored in cell (i,j^*) with $j^* \equiv j \pmod m$. When $A_{ij^*}=0$, these two configurations are the same as their counterparts when the square sub-array contains cell (i,j). When $A_{ij^*}=1$, the number of additional information cells, for which we need them to either store a 0 or store a 1 with a non-failing diode, is $u+u'-o$ when $A_{j'i'}=0$ and $A_{j'i'}=1$. ∎

Lemma 4. *Let $P(e_{ij}=0, e_{j'j}=0)$ denote the joint probability that both an information cell that stores 0 and the reference cell on the same column do not incur sneak-path events simultaneously. We have:*

$$\begin{aligned} P(e_{ij}=0,e_{j'j}=0) &= \sum_{u=0}^{m-2}\Bigg\{\binom{m-2}{u}q^u(1-q)^{m-2-u} \\ &\times\Bigg[(1-q)P^{(1)}(e_{ij}=0,e_{j'j}=0|u,A_{j'i'}=0) \\ &\times\Big[P^{(2)}(e_{ij}=0,e_{j'j}=0|u,A_{j'i'}=0)\Big]^{r-1} \\ &+qP^{(1)}(e_{ij}=0,e_{j'j}=0|u,A_{j'i'}=0) \\ &\times\Big[P^{(2)}(e_{ij}=0,e_{j'j}=0|u,A_{j'i'}=1)\Big]^{r-1}\Bigg]\Bigg\}, \end{aligned} \tag{26}$$

where

$$\begin{aligned} &P^{(1)}(e_{ij}=0,e_{j'j}=0|u,A_{j'i'}=0) \\ &=\sum_{k=0}^{u}\sum_{w=0}^{m-2-u}\sum_{v=0}^{k+w}\sum_{o=0}^{v}\binom{u}{k}\binom{m-2-u}{w}\binom{k+w}{v}\binom{v}{o} \\ &\times q^{k+w+o}(1-q)^{2m-4-k-w-o}(1-p_fq)^{(k+w)u-k}, \end{aligned} \tag{27}$$

$$\begin{aligned} &P^{(2)}(e_{ij}=0,e_{j'j}=0|u,A_{j'i'}=0) \\ &=q^2P^{(1)}(e_{ij}=0,e_{j'j}=0|u,A_{j'i'}=0)(1-p_fq)^{2u} \\ &+2q(1-q)P^{(1)}(e_{ij}=0,e_{j'j}=0|u,A_{j'i'}=0)(1-p_fq)^{u} \\ &+(1-q)^2P^{(1)}(e_{ij}=0,e_{j'j}=0|u,A_{j'i'}=0), \end{aligned} \tag{28}$$

$$\begin{aligned} &P^{(1)}(e_{ij}=0,e_{j'j}=0|u,A_{j'i'}=1) \\ &=P^{(1)}(e_{ij}=0,e_{j'j}=0|u,A_{j'i'}=0)(1-p_fq)^{u}, \end{aligned} \tag{29}$$

and,

$$\begin{aligned} &P^{(2)}(e_{ij}=0,e_{j'j}=0|u,A_{j'i'}=1) \\ &=P^{(2)}(e_{ij}=0,e_{j'j}=0|u,A_{j'i'}=0). \end{aligned} \tag{30}$$

Proof: To calculate $P(e_{ij}=0,e_{j'j}=0)$, we condition on selecting u rows, with an index set I_u, that have 1's on the j-th column. Next, conditioning on the selected u rows, we consider the same four cases as in the previous proof. For the two cases that the square sub-array contains cell (i,j), we condition on these four parameters: k, w, v and o. Let v be the number of columns, with an index set I_v, that have 1's on the i-th row and let v' be the number of columns, with an index set $I_{v'}$, that have 1's on the j'-th row. We let $k = |I_u \cap (I_v \cup I_{v'})|$, $w = |(I_v \cup I_{v'}) \backslash I_u|$ and $o = |I_v \cap I_{v'}|$. Note that we do not condition on v' explicitly but instead use $v' = k+w-v+o$. For each configuration with parameters u, k, w, v and o, the number of information cells that need to either store a 0 or store a 1 with a non-failing diode is $u(k+w)-k$ when $A_{j'i'}=0$ and $u(k+w+1)-k$ (additional u) when $A_{j'i'}=1$. For the case that the square sub-array does not contain cell (i,j) and $A_{j'i'}=0$, four sub-cases are considered for the values stored in cell (i,j^*) with $j^* \equiv j \pmod m$ and cell (j',i^*) with $i^* \equiv i' \pmod m$. When $A_{ij^*}=0$ and $A_{j'i^*}=0$, the configuration is the same as the configuration when the square sub-array contains cell (i,j). When $A_{ij^*}=1$, $A_{j'i^*}=0$ or $A_{ij^*}=0$, $A_{j'i^*}=1$, the number of additional information cells that need to either store a 0 or store a 1 with a non-failing diode is u. When $A_{ij^*}=1$ and $A_{j'i^*}=1$, the number of additional information cells that need to either store a 0 or store a 1 with a non-failing diode is $2u$. For the case that the square sub-array does not contain cell (i,j) and $A_{j'i'}=1$, the configuration is the same as the configuration when the square sub-array does not contain cell (i,j) and $A_{j'i'}=0$. ∎

Lemma 5. *Let $P(e_{ii'}=0)$ be the probability that the reference cell on the row of an information cell, which stores $A_{ij}=0$, incurs a sneak-path event. We have:*

$$\begin{aligned} P(e_{ii'}=0) &= (1-q)P(e_{ij}=0)+q\sum_{u=0}^{m-2}\Bigg\{\binom{m-2}{u}q^u(1-q)^{m-2-u} \\ &\times P^{(1)}(e_{ii'}=0|u,A_{j'i'}=1)\Big[P^{(2)}(e_{ii'}=0|u,A_{j'i'}=1)\Big]^{r-1}\Bigg\}, \end{aligned} \tag{31}$$

where

$$\begin{aligned} &P^{(1)}(e_{ii'}=0|u,A_{j'i'}=1)=\sum_{v=0}^{m-2}\sum_{k=0}^{\min(u,v)}\Bigg\{\binom{u}{k}\binom{m-2-u}{v-k} \\ &\times q^v(1-q)^{m-2-v}(1-p_fq)^{(u+1)v-k}\Bigg\}, \end{aligned} \tag{32}$$

and

$$\begin{aligned} &P^{(2)}(e_{ii'}=0|u,A_{j'i'}=1)=qP^{(1)}(e_{ii'}=0|u,A_{j'i'}=1) \\ &\times(1-p_fq)^u+(1-q)P^{(1)}(e_{ii'}=0|u,A_{j'i'}=1). \end{aligned} \tag{33}$$

Proof: We again separate the calculation of $P(e_{ii'}=0)$ into four cases. When $A_{j'i'}=0$, we have the same configuration as in the proof of $P(e_{ij}=0)$ in Lemma 2. When $A_{j'i'}=1$, we condition on selecting u rows that have 1's on the i'-th column and divide the fat array into square sub-arrays. For the square sub-array that contains cell (i,i'), the calculation is similar to the calculation of $P^{(1)}(e_{ij}=0|u)$ in Lemma 2 except we have $u+1$ rows that have 1's on the i'-th column because $A_{j'i'}=1$. For the square sub-array that does not contain cell (i,i'), we consider whether or not the cell (i,j^*) with $j^* \equiv j \pmod m$ stores a 0. When $A_{ij^*}=1$,

the number of additional information cells that need to either store a 0 or store a 1 with a non-failing diode is u. When $A_{ij^*}=0$, the configuration is the same as the configuration when the square sub-array contains cell (i,i'). ∎

Lemma 6. *Let $P(e_{j'j}=0)$ be the probability that the reference cell on the column of an information cell, which stores $A_{ij}=0$, incurs a sneak-path event. We have:*

$$P(e_{j'j}=0)=(1-q)P(e_{ij}=0)+q\sum_{u=0}^{m-2}\Bigg\{\binom{m-2}{u}\times q^u(1-q)^{m-2-u}P^{(1)}(e_{j'j}=0|u,A_{j'i'}=1)\times\Big[P^{(2)}(e_{j'j}=0|u,A_{j'i'}=1)\Big]^{r-1}\Bigg\}, \tag{34}$$

where

$$P^{(1)}(e_{j'j}=0|u,A_{j'i'}=1)=P^{(1)}(e_{ij}=0|u)(1-p_fq)^u, \tag{35}$$

and

$$P^{(2)}(e_{j'j}=0|u,A_{j'i'}=1)=P^{(2)}(e_{ij}=0|u). \tag{36}$$

Proof: We again separate the calculation of $P(e_{ii'}=0)$ into four cases. When $A_{j'i'}=0$, we have the same configuration as in the calculation of $P_{ij}=0$ in Lemma 2. When $A_{j'i'}=1$, we condition on selecting u rows that have 1's on the i'-th column and divide the fat array into square sub-arrays. For the square sub-array that contains cell (i,i'), the calculation is similar to the calculation of $P^{(1)}(e_{ij}=0|u)$ in Lemma 2 except we have one additional column that has an 1 on the j'-th row because $A_{j'i'}=1$. This additional column therefore requires additional u information cells to either store a 0 or store a 1 with a non-failing diode. For the square sub-array that does not contain cell (i,i'), the calculation is the same as the calculation of $P^{(2)}(e_{ij}=0|u)$ since they have the same configuration. ∎

Lemma 7. *The probability that an information cell and its two reference cells do not incur sneak-path events simultaneously is given by:*

$$P(e_{ij}=0,e_{ii'}=0,e_{j'j}=0)=\sum_{u=0}^{m-2}\sum_{u'=0}^{m-2}\sum_{o=0}^{\min(u,u')}\Bigg\{\binom{m-2}{u}\times\binom{u}{o}\binom{m-2-u}{u'-o}q^{u+u'}(1-q)^{2m-4-u-u'}\times\Bigg[(1-q)P^{(1)}(e_{ij}=0,e_{ii'}=0,e_{j'j}=0|u,u',o,A_{j'i'}=0)\times\Big[P^{(2)}(e_{ij}=0,e_{ii'}=0,e_{j'j}=0|u,u',o,A_{j'i'}=0)\Big]^{r-1}+qP^{(1)}(e_{ij}=0,e_{ii'}=0,e_{j'j}=0|u,u',o,A_{j'i'}=1)\times\Big[P^{(2)}(e_{ij}=0,e_{ii'}=0,e_{j'j}=0|u,u',o,A_{j'i'}=1)\Big]^{r-1}\Bigg]\Bigg\}, \tag{37}$$

where

$$P^{(1)}(e_{ij}=0,e_{ii'}=0,e_{j'j}=0|u,u',o,A_{j'i'}=0)=\sum_{k=0}^{u}\sum_{k^*=0}^{u'-o}\sum_{w=0}^{m-2-u-u'+o}\sum_{v=0}^{k+k^*+w}\sum_{\substack{v^*=\max(0,\\ v-k-w)}}^{\min(v,k^*)}\sum_{\substack{v'=k+k^*\\ +w-v}}^{k+k^*+w}\Bigg[\binom{u}{k}\binom{u'-o}{k^*}\binom{m-2-u-u'+o}{w}\times\binom{k^*}{v^*}\binom{k+w}{v-v^*}\binom{v}{v-k-k^*-w+v'}q^{v+v'}(1-q)^{2m-4-v-v'}\times(1-p_fq)^{uv+uv'+vu'-ov-u(v-k-k^*-w+v')-k-v^*}\Bigg], \tag{38}$$

$$P^{(2)}(e_{ij}=0,e_{ii'}=0,e_{j'j}=0|u,u',o,A_{j'i'}=0)=\Big[(1-q)^2+q(1-q)(1-p_fq)^{u+u'-o}+q(1-q)(1-p_fq)^u+q^2(1-p_fq)^{2u}\times(1-p_fq)^{u'-o}\Big]P^{(1)}(e_{ij}=0,e_{ii'}=0,e_{j'j}=0|u,u',o,A_{j'i'}=0), \tag{39}$$

$$P^{(1)}(e_{ij}=0,e_{ii'}=0,e_{j'j}=0|u,u',o,A_{j'i'}=1)=\sum_{k=0}^{u}\sum_{k^*=0}^{u'-o}\sum_{w=0}^{m-2-u-u'+o}\sum_{v=0}^{k+k^*+w}\sum_{\substack{v^*=\max(0,\\ v-k-w)}}^{\min(v,k^*)}\sum_{\substack{v'=k+k^*\\ +w-v}}^{k+k^*+w}\Bigg[\binom{u}{k}\binom{u'-o}{k^*}\binom{m-2-u-u'+o}{w}\times\binom{k^*}{v^*}\binom{k+w}{v-v^*}\binom{v}{v-k-k^*-w+v'}q^{v+v'}(1-q)^{2m-4-v-v'}(1-p_fq)^{uv+uv'+vu'-ov-u(v-k-k^*-w+v')-k-v^*}(1-p_f)^{o+v+v'-k-k^*-w}\Bigg], \tag{40}$$

and

$$P^{(2)}(e_{ij}=0,e_{ii'}=0,e_{j'j}=0|u,u',o,A_{j'i'}=1)=\sum_{k=0}^{u}\sum_{k^*=0}^{u'-o}\sum_{w=0}^{m-2-u-u'+o}\sum_{v=0}^{k+k^*+w}\sum_{\substack{v^*=\max(0,\\ v-k-w)}}^{\min(v,k^*)}\sum_{\substack{v'=k+k^*\\ +w-v}}^{k+k^*+w}\Bigg\{\binom{u}{k}\binom{u'-o}{k^*}\times\binom{m-2-u-u'+o}{w}\binom{k^*}{v^*}\binom{k+w}{v-v^*}\times\binom{v}{v-k-k^*-w+v'}q^{v+v'}(1-q)^{2m-4-v-v'}\times(1-p_fq)^{uv+uv'+vu'-ov-u(v-k-k^*-w+v')-k-v^*}\times(1-p_f)^{v+v'-k-k^*-w}\Big[(1-q)^2+q(1-q)(1-p_fq)^{u+u'-o}+q(1-q)(1-p_fq)^u+q^2(1-p_fq)^{2u+u'-o}\Big]\Bigg\}. \tag{41}$$

Proof: To calculate $P(e_{ij}=0,e_{ii'}=0)$, we condition on selecting u rows, with an index set I_u, that have 1's on the j-th column. We also select u' rows, with an index set $I_{u'}$, that have 1's on the i'-th column, and we let $o=|I_u\cap I_{u'}|$. Next, we consider the same four cases that are considered in the previous proofs. Let v be the number of columns, with an

index set I_v, that have 1's on the i-th row and let v' be the number of columns, with an index set $I_{v'}$, that have 1's on the j'-th row. We let k be $|(I_v \cup I_{v'}) \cap I_u|$ and let k^* be $|(I_v \cup I_{v'}) \cap (I_{u'} \backslash I_u)|$. We also let w be $|(I_v \cup I_{v'}) \backslash (I_{u'} \cup I_u)|$ and let v^* be $|I_v \cap (I_{u'} \backslash I_u)|$. For the cases when the square sub-array contains cell (i,j) and $A_{j'i'} = 0$, for each configuration with parameters v, v', k, k^*, w and v^*, the number of information cells that need to either store a 0 or store a 1 with a non-failing diode is $uv + uv' + vu' - ov - u(v - k - k^* - w + v') - k - v^*$ by inclusion and exclusion. For the cases when the square sub-array contains cell (i,j) and $A_{j'i'} = 1$, additionally, these $o + v + v' - k - k^* - w = |I_u \cap I_{u'}| + |I_v \cap I_{v'}|$ cells on the j'-th row and the i'-th column that we know store 1's can not have a failed diode . For the two cases when the square sub-array does not contain cell (i,j), four sub-cases are considered for the values stored in cell (i, j^*) with $j^* \equiv j \pmod m$ and cell (j', i^*) with $i^* \equiv i' \pmod m$. When $A_{j'i'} = 0$, the number of additional information cells that need to either store a 0 or store a 1 with a non-failing diode is 0, u, $u + u' - o$ and $2u + u' - o$ for the sub-cases $A_{ij^*} = 0, A_{j'i^*} = 0$, $A_{ij^*} = 0, A_{j'i^*} = 1$, $A_{ij^*} = 1, A_{j'i^*} = 0$ and $A_{ij^*} = 1, A_{j'i^*} = 1$, respectively. When $A_{j'i'} = 1$, additionally, these $v + v' - k - k^* - w = |I_v \cap I_{v'}|$ cells on the j'-th row that we know store 1's can not have a failed diode. ■

Lemma 8. *Let $P(e_{ii'} = 0, e_{j'j} = 0)$ be the probability that both reference cells of an information cell, which stores $A_{ij} = 0$, incurs sneak-path events simultaneously. We have:*

$$
\begin{aligned}
P(e_{ii'} = 0, e_{j'j} = 0) = \sum_{u=0}^{m-2} \sum_{u'=0}^{m-2} \sum_{o=0}^{\min(u,u')} \Bigg\{ & \binom{m-2}{u} \binom{u}{o} \\
& \times \binom{m-2-u}{u'-o} q^{u+u'} (1-q)^{2m-4-u-u'} \\
& \times \Bigg[(1-q) P^{(1)}(e_{ii'} = 0, e_{j'j} = 0 | u, u', o, A_{j'i'} = 0) \\
& \times \Big[P^{(2)}(e_{ii'} = 0, e_{j'j} = 0 | u, u', o, A_{j'i'} = 0) \Big]^{r-1} \\
& + q P^{(1)}(e_{ii'} = 0, e_{j'j} = 0 | u, u', o, A_{j'i'} = 1) \\
& \times \Big[P^{(2)}(e_{ii'} = 0, e_{j'j} = 0 | u, u', o, A_{j'i'} = 1) \Big]^{r-1} \Bigg] \Bigg\},
\end{aligned} \tag{42}
$$

where

$$
\begin{aligned}
P^{(1)}(e_{ii'} = 0, e_{j'j} = 0 | u, u', o, A_{j'i'} = 0) = & \sum_{k_1=0}^{o} \sum_{k_2=0}^{u-o} \sum_{k_3=0}^{u'-o} \\
& \sum_{w=0}^{\substack{m-2-u \\ -u'-o}} \sum_{v_1=0}^{k_2} \sum_{v_2}^{k_2-v_1} \sum_{v_3}^{k_3} \sum_{\substack{v=v_1 \\ +v_2+v_3}}^{\substack{k_1+k_2 \\ +k_3+w}} \sum_{\substack{v'=k_1+k_2+k_3 \\ +w-v+v_1}}^{\substack{k_1+k_2 \\ +k_3+w}} \Bigg\{ \binom{o}{k_1} \binom{u-o}{k_3} \\
& \binom{m-2-u-u'+o}{w} \binom{k_2}{v_1} \binom{k_2-v_1}{v_2} \binom{k_3}{v_3} \\
& \times \binom{v-v_1-v_2}{v+v'-k_1-k_2-k_3-w-v_1} \\
& \times \binom{k_1+w}{v-v_1-v_2-v_3} q^{v+v'} (1-q)^{2m-4-v-v'} \\
& \times (1-p_f q)^{uv'+vu'-o(v+v'-k_1-k_2-k_3-w)-k_1-(k_2-v_2)-v_3} \Bigg\},
\end{aligned} \tag{43}
$$

$$
\begin{aligned}
P^{(2)}(e_{ii'} = 0, e_{j'j} = 0 | u, u', o, A_{j'i'} = 0) = & \Big[(1-q)^2 \\
& + q(1-q)(1-p_f q)^u + q(1-q)(1-p_f q)^{u'} + q^2 (1-p_f q)^{u+u'} \Big] \\
& \times P^{(1)}(e_{ii'} = 0, e_{j'j} = 0 | u, u', o, A_{j'i'} = 0),
\end{aligned} \tag{44}
$$

$$
\begin{aligned}
P^{(1)}(e_{ii'} = 0, e_{j'j} = 0 | u, u', o, A_{j'i'} = 1) = & \sum_{k_1=0}^{o} \sum_{k_2=0}^{u-o} \sum_{k_3=0}^{u'-o} \\
& \sum_{w=0}^{\substack{m-2-u \\ -u'-o}} \sum_{v_1=0}^{k_2} \sum_{v_2}^{k_2-v_1} \sum_{v_3}^{k_3} \sum_{\substack{v=v_1 \\ +v_2+v_3}}^{\substack{k_1+k_2 \\ +k_3+w}} \sum_{\substack{v'=k_1+k_2+k_3 \\ +w-v+v_1}}^{\substack{k_1+k_2 \\ +k_3+w}} \Bigg\{ \binom{o}{k_1} \\
& \times \binom{u-o}{k_3} \binom{m-2-u-u'+o}{w} \binom{k_2}{v_1} \binom{k_2-v_1}{v_2} \binom{k_3}{v_3} \\
& \times \binom{v-v_1-v_2}{v+v'-k_1-k_2-k_3-w-v_1} \binom{k_1+w}{v-v_1-v_2-v_3} \\
& \times q^{v+v'} (1-q)^{2m-4-v-v'} \\
& \times (1-p_f q)^{uv'+vu'-o(v+v'-k_1-k_2-k_3-w)-k_1-(k_2-v_2)-v_3} \\
& \times (1-p_f)^{o+v+v'-k_1-k_2-k_3-w} \Bigg\},
\end{aligned} \tag{45}
$$

and

$$P^{(2)}(e_{ii'}=0,e_{j'j}=0|u,u',o,A_{j'i'}=1)=\sum_{k_1=0}^{o}\sum_{k_2=0}^{u-o}\sum_{k_3=0}^{u'-o}$$
$$\sum_{w=0}^{m-2-u\atop -u'-o}\sum_{v_1=0}^{k_2}\sum_{v_2}^{k_2-v_1}\sum_{v_3}^{k_3}\sum_{v=v_1\atop +v_2+v_3}^{k_1+k_2\atop +k_3+w}\sum_{v'=k_1+k_2+k_3\atop +w-v+v_1}^{k_1+k_2\atop +k_3+w}\left\{\binom{o}{k_1}\binom{u-o}{k_3}\right.$$
$$\times\binom{m-2-u-u'+o}{w}\binom{k_2}{v_1}\binom{k_2-v_1}{v_2}\binom{k_3}{v_3}$$
$$\times\binom{v-v_1-v_2}{v+v'-k_1-k_2-k_3-w-v_1}\binom{k_1+w}{v-v_1-v_2-v_3}q^{v+v'}$$
$$\times(1-q)^{2m-4-v-v'}(1-p_fq)^{uv'+vu'-o(v+v'-k_1-k_2-k_3-w)}$$
$$\times(1-p_fq)^{-k_1-(k_2-v_2)-v_3}(1-p_f)^{v+v'-k_1-k_2-k_3-w}\Big[(1-q)^2$$
$$\left.+q(1-q)(1-p_fq)^{u}+q(1-q)(1-p_fq)^{u'}+q^2(1-p_fq)^{u+u'}\Big]\right\}. \tag{46}$$

Proof: To calculate $P(e_{ij}=0,e_{ii'}=0)$, we condition on selecting u rows, with an index set I_u, that have 1's on the j-th column. We also select u' rows, with an index set $I_{u'}$, that have 1's on the i'-th column and we let $o=|I_u\cap I_{u'}|$. Next, we consider the same four cases that are considered in the previous proofs. Let v be the number of columns, with an index set I_v, that have 1's on the i-th row and let v' be the number of columns, with an index set $I_{v'}$, that have 1's on the j'-th row. We let $k_1=|(I_u\cap I_{u'})\cap(I_v\cup I_{v'})|$, $k_2=|(I_u/I_{u'})\cap(I_v\cup I_{v'})|$, $k_3=|(I_{u'}/I_u)\cap(I_v\cup I_{v'})|$, $w=|I_v\cup I_{v'}/(I_u\cup I_{u'})|$, $v_1=|(I_u/I_{u'})\cap(I_v\cap I_{v'})|$, $v_1=|(I_u/I_{u'})\cap(I_v/I_{v'})|$ and $v_3=|(I_{u'}/I_u)\cap I_v|$. For the cases when the square sub-array contains cell (i,j) and $A_{j'i'}=0$, for each configuration with the above parameters, the number of information cells that need to either store a 0 or store a 1 with a non-failing diode is $uv'+vu'-o(v+v'-k_1-k_2-k_3-w)-k_1-(k_2-v_2)-v_3$ by noting that $|Iv\cap I_{v'}|=v+v'-k_1-k_2-k_3-w$ and $|(I_u\cap I_{v'})\cup(I_v\cap I_{u'})|=k_1+k_2-v_2+v_3$. For the cases when the square sub-array contains cell (i,j) and $A_{j'i'}=1$, additionally, these $|I_u\cap I_{u'}|+|I_v\cap I_{v'}|$ cells on the j'-th row and the i'-th column that we know store 1's can not have a failed diode . For the two cases when the square sub-array does not contain cell (i,j), four sub-cases are considered for the values stored in cell (i,j^*) with $j^*\equiv j\pmod m$ and cell (j',i^*) with $i^*\equiv i'\pmod m$. When $A_{j'i'}=0$, the number of additional information cells that need to either store a 0 or store a 1 with a non-failing diode is 0, u, u' and $u+u'$ for the sub-cases $A_{ij^*}=0,A_{j'i^*}=0$, $A_{ij^*}=0,A_{j'i^*}=1$, $A_{ij^*}=1,A_{j'i^*}=0$ and $A_{ij^*}=1,A_{j'i^*}=1$, respectively. When $A_{j'i'}=1$, additionally, these $|I_v\cap I_{v'}|$ cells on the j'-th row that we know store 1's can not have a failed diode. ■

All lemmas in the appendix are verified with Monte Carlo simulations. These probabilities can be approximated with simplified calculations by neglecting the fact that the array has preset pilot cells, therefore avoiding the somewhat complicated enumeration through the overlapping indexes between the row indexes and column indexes. The approximated versions of Lemma 2 and Lemma 4 with $p_f=1$ can be found in [12].

REFERENCES

[1] Z. Chen, C. Schoeny, and L. Dolecek, "Coding assisted adaptive thresholding for sneak-path mitigation in resistive memories," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Guangzhou, China, Nov., 2018.

[2] D. B. Strukov, G. S. Snider, D. R. Stewart *et al.*, "The missing memristor found," *nature*, vol. 453, no. 7191, p. 80, 2008.

[3] M. A. Zidan, H. A. H. Fahmy, M. M. Hussain *et al.*, "Memristor-based memory: The sneak paths problem and solutions," *Microelectronics Journal*, vol. 44, no. 2, pp. 176–183, 2013.

[4] Y. Deng, P. Huang, B. Chen, X. Yang, B. Gao, J. Wang, L. Zeng, G. Du, J. Kang, and X. Liu, "RRAM crossbar array with cell selection device: A device and circuit interaction study," *IEEE Trans. Electron Devices*, vol. 60, no. 2, pp. 719–726, 2013.

[5] Y. Zhang, Z. Duan, R. Li, C.-J. Ku, P. I. Reyes, A. Ashrafi, J. Zhong, and Y. Lu, "Vertically integrated ZnO-Based 1D1R structure for resistive switching," *Journal of Physics D: Applied Physics*, vol. 46, no. 14, p. 145101, 2013.

[6] X. Tran, B. Gao, J. Kang, L. Wu, Z. Wang, Z. Fang, K. Pey, Y. Yeo, A. Du, B. Nguyen *et al.*, "High performance unipolar AlO y/HfO x/Ni based RRAM compatible with Si diodes for 3D application," in *2011 Symposium on VLSI Technology-Digest of Technical Papers*. IEEE, 2011, pp. 44–45.

[7] Y. Y. Chen, L. Goux, S. Clima, B. Govoreanu, R. Degraeve, G. S. Kar, A. Fantini, G. Groeseneken, D. J. Wouters, and M. Jurczak, "Endurance/Retention Trade-off on HfO2 Metal Cap 1T1R Bipolar RRAM," *IEEE Transactions on electron devices*, vol. 60, no. 3, pp. 1114–1121, 2013.

[8] J.-J. Huang, Y.-M. Tseng, W.-C. Luo, C.-W. Hsu, and T.-H. Hou, "One selector-one resistor (1S1R) crossbar array for high-density flexible memory applications," in *2011 International Electron Devices Meeting*. IEEE, 2011, pp. 31–7.

[9] E. A. Amerasekera and F. N. Najm, *Failure mechanisms in semiconductor devices*. Wiley New York, 1997.

[10] M. Handbook, "Electronic reliability design handbook," in *MIL-HDBK-338, DoD*, 1988.

[11] Y. Ben-Hur and Y. Cassuto, "Detection and coding schemes for parallel interference in resistive memories," in *Proc. IEEE Int.l Conf. on Commun. (ICC)*, Paris, France, May, 2017, pp. 1–7.

[12] Y. Cassuto, S. Kvatinsky, and E. Yaakobi, "Information-theoretic sneak-path mitigation in memristor crossbar arrays," *IEEE Trans. Inf. Theory*, vol. 62, no. 9, pp. 4801–4813, 2016.

[13] A. Chen and M.-R. Lin, "Variability of resistive switching memories and its impact on crossbar array performance," in *Proc. IEEE Rel. Physics Symp. (IRPS)*, Monterey, CA, April 2011, pp. MY–7.

[14] X. Cao, X. Li, X. Gao *et al.*, "All-ZnO-based transparent resistance random access memory device fully fabricated at room temperature," *Journal of Physics D: Applied Physics*, vol. 44, no. 25, p. 255104, 2011.

[15] M. Wang, W. Luo, Y. Wang *et al.*, "A novel Cu x Si y O resistive memory in logic technology with excellent data retention and resistance distribution for embedded applications," in *Proc. IEEE Symp. on VLSI Technol. (VLSIT)*, Honolulu, HI, June 2010, pp. 89–90.

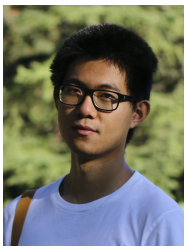
[16] Z. Chen, C. Schoeny, and L. Dolecek, "Hamming distance computation in unreliable resistive memory," *IEEE Transactions on Communications*, vol. 66, no. 11, pp. 5013–5027, 2018.

[17] R. Naous, M. A. Zidan, A. Sultan *et al.*, "Pilot assisted readout for passive memristor crossbars," *Microelectronics Journal*, vol. 54, pp. 48–58, 2016.

[18] X. Guan, S. Yu, H.-S. P. Wong *et al.*, "On the switching parameter variation of metal-oxide RRAM Part I: Physical modeling and simulation methodology," *IEEE Transactions on Electron Devices*, vol. 59, no. 4, pp. 1172–1182, 2012.

[19] Y. Ben-Hur and Y. Cassuto, "Detection and coding schemes for sneak-path interference in resistive memory arrays," *IEEE Transactions on Communications*, vol. 67, no. 6, pp. 3821–3833, 2019.

[20] J. Joshua Yang, M.-X. Zhang, M. D. Pickett *et al.*, "Engineering nonlinearity into memristors for passive crossbar applications," *Applied Physics Letters*, vol. 100, no. 11, p. 113501, 2012.



Zehui Chen (S'17) is a Ph.D. candidate in the Electrical and Computer Engineering Department at the University of California, Los Angeles (UCLA). He received his M.S. degrees in Electrical Engineering from UCLA in 2018. He received his B.S. degree (cum laude) in Electrical Engineering from Purdue University, West Lafayette in 2016. He is currently a graduate student researcher at the Laboratory for Robust Information Systems (LORIS) at UCLA. His research interests include coding theory, information theory and their applications in emerging memory medium such as ReRAM and PCM.

Clayton Schoeny (S'09, M'19) received his Ph.D. in the Electrical & Computer Engineering Department at the University of California, Los Angeles (UCLA) where he was a recipient of the 2018 Distinguished PhD Dissertation Award in Signals & Systems. He received his B.S. (cum laude) and M.S. degrees in Electrical Engineering from UCLA in 2012 and 2014, respectively. His research interests include coding theory and information theory, and he is associated with the LORIS and CoDESS labs. He is a recipient of the Henry Samueli Excellence in Teaching Award, the 2016 Qualcomm Innovation Fellowship, and the UCLA Dissertation Year Fellowship. He is currently a Data Scientist at Fair Financial Corp.

Lara Dolecek (S'05–M'10–SM'12) is a Full Professor with the Electrical and Computer Engineering Department and Mathematics Department (courtesy) at the University of California, Los Angeles (UCLA). She holds a B.S. (with honors), M.S. and Ph.D. degrees in Electrical Engineering and Computer Sciences, as well as an M.A. degree in Statistics, all from the University of California, Berkeley. She received the 2007 David J. Sakrison Memorial Prize for the most outstanding doctoral research in the Department of Electrical Engineering and Computer Sciences at UC Berkeley. Prior to joining UCLA, she was a postdoctoral researcher with the Laboratory for Information and Decision Systems at the Massachusetts Institute of Technology. She received IBM Faculty Award (2014), Northrop Grumman Excellence in Teaching Award (2013), Intel Early Career Faculty Award (2013), University of California Faculty Development Award (2013), Okawa Research Grant (2013), NSF CAREER Award (2012), and Hellman Fellowship Award (2011). With her research group and collaborators, she received numerous best paper awards. Her research interests span coding and information theory, graphical models, statistical methods, and algorithms, with applications to emerging systems for data storage and computing. She currently serves as an Associate Editor for IEEE Transactions on Information Theory and as the Secretary of the Information Theory Society. Prof. Dolecek has served as a consultant for a number of companies specializing in data communications and storage.