Endurance-Limited Memories with Informed Decoder

Yeow Meng Chee[*], **Michal Horovitz**[†], **Alexander Vardy**[‡¶], **Van Khu Vu**[¶], and **Eitan Yaakobi**[§]

[*] Department of Industrial Systems Engineering and Management, National University of Singapore, Singapore

[†] Computer Science Department, Tel-Hai College, and The Galilee Research Institute - Migal, Upper Galilee, Israel

[‡] Department of Electrical and Computer Engineering, University of California San Diego, La Jolla, CA 92093, USA

[¶] School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore

[§] Department of Computer Science, Technion — Israel Institute of Technology, Haifa, 32000 Israel

Emails: pvocym@nus.edu.sg, horovitzmic@telhai.ac.il, avardy@ucsd.edu, vankhu001@ntu.edu.sg, yaakobi@cs.technion.ac.il

***Abstract*—Non-volatile *resistive memories*, such as *phase change memories* and *resistive random access memories*, have attracted significant attention recently due to their scalability, speed, and rewritability. However, in order to use these memories in large-scale memory and storage systems, the *limited endurance* deficiency of these memories must be addressed. In a recent paper, we proposed a new coding scheme, called *endurance-limited memories* (*ELM*) codes, which increases the endurance of these memories by limiting the number of cell programming operations. Namely, an ℓ*-change* t*-write ELM code* is a coding scheme that allows to write t messages into some n binary cells while guaranteeing that the number of times each cell is programmed is at most ℓ. There are several models of these codes which depend upon the information that is available to the encoder and the decoder before each write. This information can be one of the following three options: 1. the number of times each cell has been programmed, 2. only the memory state before programming, or 3. no information is available on the cells' state or previous writes. In this paper, we study the models in which the decoder knows on each write the number of times each cell has been programmed before the last write, while for the encoder we consider the aforementioned three possibilities.**

I. Introduction

Resistive memories, such as *resistive random access memories* (*ReRAM*) and *phase-change memories* (*PCM*) have the potential to be widely used in the future as a universal memory. These memories are abundant with several important advantages, such as speed, density, non-volatile, and rewritability. However, one of their main bottlenecks is the limited *write endurance*, which must be solved in order to make these memories widely used.

Previous works have offered different solutions to overcome this limitation. For example, in [16] a technique has been applied for mellow writes which reduce the wearout of the cells during programming, and in [8] the authors using locally repairable codes (LRC) to get small rewriting locality. Other solutions used coding schemes to correct stuck-at cells; see e.g. [9], [11], [15]. Recently, in [2], we proposed a new family of codes, called *endurance-limited memory* (*ELM*) codes. The model under this paradigm assumes that there are n cells and each cell stores a binary value by the cell's resistance. The goal is to write t messages sequentially in the cells, while the resistance of each cell can be changed at most some $\ell \geqslant 1$ times. Note that for $\ell = 1$, we get the classical problem of *write-once memory* (*WOM*) codes [3], [6], [7], [10], [13], [14].

There are several models of ELM codes that can be studied [2]. These models are distinguished by the information that is available to the encoder and the decoder before each write. Specifically, for the encoder/decoder this information can be one of the following three options:

1) *Informed All* (*EIA/DIA*) is the case where the number of times each cell has been programmed so far is known.
2) *Informed Partially* (*EIP/DIP*) is the case where only the cell state is known before programming.
3) *Uninformed* (*EU/DU*) is the case where no information on the cells is available before programming.

Thus, by considering all combinations of the above three cases for the encoder and the decoder, it is possible to define and study nine models, $EX : DY$, where $X, Y \in \{IA, IP, U\}$. The EIA models have been studied by the authors in [2], and this paper advances this study to cover the DIA models as well.

The rest of the paper is organized as follows. First, the DIA models are formally defined in Section II. In Section III, we review the capacity region and maximum sum-rates results for the EIA models that were determined in [2]. Then, we proceed to study the capacity region of the DIA models. The capacity region of the EIP:DIA model is studied in Section IV and Section V presents capacity achieving codes for this model. Then, Section VI compares between the EIP:DIA and the EIA models. Lastly, in Section VII we study the EU:DIA model.

II. Definitions and Preliminaries

In this section we formally define the three DIA ELM models which are studied in the paper and state several simple observations. The definition of all nine models can be found in [2].

For a positive integer a, the set $\{0, \dots, a-1\}$ is defined by $[a]$. We assume that the number of cells is n, and we use the vector notation $\boldsymbol{c} \in [2]^n$ to represent the encoded vector of the n memory cells, and the vector $\boldsymbol{v} \in [\ell+1]^n$, which will be called the *cell-program-count vector*, to represent the number of times each cell has been programmed. Note that the state of a cell is the parity of the number of times it was programmed. Thus, if the encoder (or the decoder) knows the cell-program-count vector $\boldsymbol{v}$, in particular it knows the *cell-state vector* as well. For a vector $\boldsymbol{v} \in [\ell+1]^n$, we denote by $\langle \boldsymbol{v} \rangle_2$ the length-n binary vector which satisfies $(\langle \boldsymbol{v} \rangle_2)_k = \boldsymbol{v}_k (\text{mod } 2)$ for all $k \in [n]$, and we say that $\langle \boldsymbol{v} \rangle_2$ equals to $\boldsymbol{v}$ modulo 2. For two length-n vectors $\boldsymbol{a}, \boldsymbol{b}$, $\boldsymbol{a}+\boldsymbol{b}$ is the vector obtained by point-wise addition. If $\boldsymbol{a}$ and $\boldsymbol{b}$ are binary vectors $\boldsymbol{a} \oplus \boldsymbol{b}$ is the vector obtained by point-wise addition modulo 2.

According to the constraint in which each cell can be programmed at most some ℓ times, we assume that if the encoder attempts to program a cell more than ℓ times then its value will not be changed. For the EIA models, it is assumed that the encoder will not attempt to program a cell that has already been programmed ℓ times before. We see this as an extension of the WOM model for $\ell = 1$. These models will be defined both for the zero-error and the ϵ-error cases.

For a cell-program-count vector $\boldsymbol{v} \in [\ell+1]^n$ and a new vector $\boldsymbol{c} \in [2]^n$ to be programmed to the cells, we define the result of programming the new vector $\boldsymbol{c}$ by $N(\boldsymbol{v}, \boldsymbol{c}) \in [\ell+1]^n$ and $f(\boldsymbol{v}, \boldsymbol{c}) \in [2]^n$, such that, $N(\boldsymbol{v}, \boldsymbol{c})$ is the new cell-program-count vector after programming $\boldsymbol{c}$, $f(\boldsymbol{v}, \boldsymbol{c})$ is the new cell-state vector, and they are formally defined as follows. For all $k \in [n]$, $N(\boldsymbol{v}, \boldsymbol{c})_k = \boldsymbol{v}_k$ if $\boldsymbol{c}_k = \boldsymbol{v}_k (\text{mod } 2)$, and otherwise $N(\boldsymbol{v}, \boldsymbol{c})_k = \max\{\ell, \boldsymbol{v}_k + 1\}$. Similarly, $f(\boldsymbol{v}, \boldsymbol{c})_k = \boldsymbol{c}_k$ if $\boldsymbol{v}_k < \ell$, and otherwise $f(\boldsymbol{v}, \boldsymbol{c})_k = \boldsymbol{v}_k (\text{mod } 2)$. Note that $\langle N(\boldsymbol{v}, \boldsymbol{c}) \rangle_2 = f(\boldsymbol{v}, \boldsymbol{c})$, i.e., $f(\boldsymbol{v}, \boldsymbol{c})$ equals to $N(\boldsymbol{v}, \boldsymbol{c})$ modulo 2.

Definition 1. *An $[n, t, \ell; M_1, \ldots, M_t]^{EX:DIA,\boldsymbol{p}_e}$ ℓ-change t-write **endurance-limited memory (ELM)** code with error probability vector $\boldsymbol{p}_e = (p_{e_1}, \ldots, p_{e_t})$, where $X \in \{IA, IP, U\}$, is a coding scheme comprising of n binary cells and is defined by t encoding and decoding maps $(\mathcal{E}_j, \mathcal{D}_j)$ for $1 \leqslant j \leqslant t$. For the map $\mathcal{E}_j$, $Im(\mathcal{E}_j)$ is its image, where by definition $Im(\mathcal{E}_0) = \{(0, \ldots, 0)\}$.*

Furthermore, for $j \in [t+1]$, let N_j and $Im^*(\mathcal{E}_j)$ be the set of all state-program-count vectors, cell-state vectors which can be obtained after the first j writes, respectively. Formally, for $j \geqslant 1$, $N_j = \{N(\boldsymbol{v}, \boldsymbol{c}) : \boldsymbol{c} \in Im(\mathcal{E}_j), \boldsymbol{v} \in N_{j-1}\}$, where $N_0 = \{(0, \ldots, 0)\}$, and $Im^*(\mathcal{E}_j) = \{\langle \boldsymbol{v} \rangle_2 : \boldsymbol{v} \in N_j\}$. Note that for the EIA models $Im^*(\mathcal{E}_j) = Im(\mathcal{E}_j)$. For a message m we denote by $Ind_m(x)$ the indicator function, where $Ind_m(x) = 0$ if $m = x$, and otherwise $Ind_m(x) = 1$. The three DIA models are defined as follows for all $1 \leqslant j \leqslant t$.

(1) If $X = IA$ then $\mathcal{E}_j : [M_j] \times N_{j-1} \mapsto [2]^n$, such that for all $(m, \boldsymbol{v}) \in [M_j] \times N_{j-1}$ it holds that $\boldsymbol{v} + (\langle \boldsymbol{v} \rangle_2 \oplus \mathcal{E}_j(m, \boldsymbol{v})) \in [\ell+1]^n$, and

$$\mathcal{D}_j : \{(\mathcal{E}_j(m, \boldsymbol{v}), \boldsymbol{v}) : m \in [M_j], \boldsymbol{v} \in N_{j-1}\} \mapsto [M_j],$$

where

$$\sum_{(m,\boldsymbol{v}) \in [M_j] \times N_{j-1}} Pr(m) Pr(\boldsymbol{v}) \cdot Ind_m \left(\mathcal{D}_j(\mathcal{E}_j(m, \boldsymbol{v}), \boldsymbol{v})\right) \leqslant p_{e_i}.$$

(2) If $X = IP$ then $\mathcal{E}_j : [M_j] \times Im^*(\mathcal{E}_{j-1}) \mapsto [2]^n$, and

$$\mathcal{D}_j : \{(f(\boldsymbol{v}, \mathcal{E}_j(m, \langle \boldsymbol{v} \rangle_2)), \boldsymbol{v}) : m \in [M_j], \boldsymbol{v} \in N_{j-1}\} \mapsto [M_j],$$

where

$$\sum_{(m,\boldsymbol{v}) \in [M_j] \times N_{j-1}} Pr(m) Pr(\boldsymbol{v}) \cdot Ind_m \left(\mathcal{D}_j(f(\boldsymbol{v}, \mathcal{E}_j(m, \langle \boldsymbol{v} \rangle_2)), \boldsymbol{v})\right) \leqslant p_{e_i}.$$

(3) If $X = U$ then $\mathcal{E}_j : [M_j] \mapsto [2]^n$, and

$$\mathcal{D}_j : \{f(\boldsymbol{v}, \mathcal{E}_j(m)), \boldsymbol{v}) : m \in [M_j], \boldsymbol{v} \in N_{j-1}\} \mapsto [M_j],$$

where

$$\sum_{(m,\boldsymbol{v}) \in [M_j] \times N_{j-1}} Pr(m) Pr(\boldsymbol{v}) \cdot Ind_m \left(\mathcal{D}_j(f(\boldsymbol{v}, \mathcal{E}_j(m)), \boldsymbol{v})\right) \leqslant p_{e_i}.$$

If $p_{e_j} = 0$ for all $1 \leqslant j \leqslant t$, then the code is called a zero-error ELM code and is denoted by $[n, t, \ell; M_1, \ldots, M_t]^{EX:DIA,z}$.

Similarly, it is possible to define $[n, t, \ell; M_1, \ldots, M_t]^{EX:DY,g}$ ELM codes for $g \in \{z, \epsilon\}$ and $Y \in \{IP, U\}$ for the cases where the decoder is either partially informed or uniformed. See [2] for the formal definition of these models.

The *rate* on the j-th write of an $[n, t, \ell; M_1, \ldots, M_t]^{EX:DY}$ ELM code, $X, Y \in \{IA, IP, U\}$, is defined as $R_j = \frac{\log M_j}{n}$, and the *sum-rate* is the sum of the individual rates on all writes, $R_{sum} = \sum_{j=1}^{t} R_j$. A rate tuple $\mathbf{R} = (\mathcal{R}_1, \ldots, \mathcal{R}_t)$ is called *ϵ-error achievable* in model $EX : DY$, if for all $\epsilon > 0$ there exists an $[n, t, \ell; M_1, \ldots, M_t]^{EX:DY,\boldsymbol{p}_e}$ ELM code with error probability vector $\boldsymbol{p}_e = (p_{e_1}, \ldots, p_{e_t}) \leqslant (\epsilon, \ldots, \epsilon)$, such that $\frac{\log M_i}{n} \geqslant \mathcal{R}_i - \epsilon$. The rate tuple $\mathbf{R}$ will be called *zero-error achievable* if for all $1 \leqslant j \leqslant t$, $p_{e_j} = 0$. The *ϵ-error capacity region* of the $EX : DY$ model is the set of all ϵ-error achievable rates tuples, that is,

$$\mathcal{C}_{t,\ell}^{EX:DY,\epsilon} = \{(R_1, \ldots, R_t) | (R_1, \ldots, R_t) \text{ is } \epsilon\text{-error achievable}\},$$

and the *ϵ-error maximum sum-rate* will be denoted by $\mathcal{R}_{t,\ell}^{EX:DY,\epsilon}$. The zero-error capacity region $\mathcal{C}_{t,\ell}^{EX:DY,z}$ and the zero-error maximum sum-rate $\mathcal{R}_{t,\ell}^{EX:DY,z}$ are defined similarly. For $\mathbf{R} = (\mathcal{R}_1, \ldots, \mathcal{R}_t)$ and $\mathbf{R}' = (\mathcal{R}'_1, \ldots, \mathcal{R}'_t)$, we say that $\mathbf{R} \leqslant \mathbf{R}'$ if $\mathcal{R}_j \leqslant \mathcal{R}'_j$ for all $1 \leqslant j \leqslant t$, and $\mathbf{R} < \mathbf{R}'$ if $\mathbf{R} \leqslant \mathbf{R}'$ and $\mathbf{R} \neq \mathbf{R}'$.

According to these definitions it is easy to verify the following relations. For $g \in \{z, \epsilon\}, X, Y \in \{IA, IP, U\}$,

$$\mathcal{C}_{t,\ell}^{EU:DY,g} \subseteq \mathcal{C}_{t,\ell}^{EIP:DY,g} \subseteq \mathcal{C}_{t,\ell}^{EIA:DY,g},$$
$$\mathcal{C}_{t,\ell}^{EX:DU,g} \subseteq \mathcal{C}_{t,\ell}^{EX:DIP,g} \subseteq \mathcal{C}_{t,\ell}^{EX:DIA,g},$$
$$\mathcal{C}_{t,\ell}^{EX:DY,z} \subseteq \mathcal{C}_{t,\ell}^{EX:DY,\epsilon}.$$

Similar connections hold for the maximum sum-rates.

Note that if $\ell \geqslant t$ then all problems are trivial since it is possible to program all cells on each write, so the capacity region in all models is $[0, 1]^t$ and the maximum sum-rate is t. For $\ell = 1$ we get the classical and well-studied WOM codes [3], [6], [7], [10], [13], [14]. In this case we also notice that the IA and IP models are the same for both the encoder and the decoder. The capacity region and the maximum sum-rate in most of these cases are known; see e.g. [6], [7], [10], [14]. In the rest of this paper, and unless stated otherwise, we assume that $1 \leqslant \ell < t$. We note that there is a strong connection between the codes studied in this paper and non-binary WOM codes as described in [2].

III. The EIA Models

The capacity region and the maximum sum-rate of the EIA models for both zero-error and ϵ-error cases, were explored in [2]. In this section we review these results in order to compare them with the EIP models which are studied in the paper.

For $1 \leqslant j \leqslant t$ and $i \in [\ell+1]$, let $p_{j,i} \in [0, 1]$, be the probability to program a cell on the j-th write, given that this cell has been already programmed i times. We define $p_{j,\ell} = 0$. Let $Q_{j,i}$ be the probability that a cell has been programmed exactly i times on the first j writes. Formally, $Q_{j,i}$ is defined recursively by using the probabilities $p_{j,i}$ and $p_{j,i-1}$ as follows.

$$Q_{j,i} = \begin{cases} Q_{j-1,i}(1 - p_{j,i}) + Q_{j-1,i-1} p_{j,i-1}, & \text{if } i > 0, \\ Q_{j-1,i}(1 - p_{j,i}), & \text{if } i = 0, \end{cases} \quad (1)$$

where $Q_{0,0} = 1$ and $Q_{0,i} = 0$ for $i > 0$.

In this paper $h(x)$, $H(X)$ is the binary entropy function where $0 \leqslant x \leqslant 1$, X is a random variable, respectively. The rates region $\mathcal{C}_{t,\ell}$ is defined as follows.

$$\mathcal{C}_{t,\ell} = \Big\{(R_1, \ldots, R_t) | \forall 1 \leqslant j \leqslant t : R_j \leqslant \textstyle\sum_{i=0}^{\min\{\ell,j\}-1} Q_{j-1,i} h(p_{j,i}),$$
$$\forall i \in [\ell] : p_{j,i} \in [0, 0.5], \text{ and } Q_{j,i} \text{ is defined in (1)}\Big\}. \quad (2)$$

It is readily verified that the maximum sum-rate is achieved with $p_{j,i} = 0.5$ for all $t - j + 1 \geqslant \ell - i$.

Corollary 2. *For all t and ℓ, $\mathcal{C}_{t,\ell}$ is the capacity region for all the EIA models for both zero-error and ϵ-error cases, and is denoted by $\mathcal{C}_{t,\ell}^{EIA}$. The maximum sum-rate in all the EIA models is $\mathcal{R}_{t,\ell}^{EIA} = \log \sum_{i=0}^{\ell} \binom{t}{i}$.*

IV. Capacity of the EIP:DIA model

In this section we discuss the capacity region and the maximum sum-rates of the EIP:DIA model. Recall that if $\ell = 1$ then by definition, EIP is equivalent to EIA and this model is equivalent to the known WOM model. Thus, in this section we assume that $\ell > 1$.

Denote by $\boldsymbol{c}_j$, $j \in [t+1]$, the length-n binary vector which represents the memory state after the j-th write, where $\boldsymbol{c}_0 = \mathbf{0}$. Note that on the j-th write, both the encoder and the decoder know the state of the memory $\boldsymbol{c}_{j-1}$ before writing the new data. Therefore, programming the current memory state, $\boldsymbol{c}_j$, is equivalent to writing a length-n binary vector which represents the difference between these two states, $\boldsymbol{c}_j \oplus \boldsymbol{c}_{j-1}$.

Let X_j be a length-n binary vector, where $X_{j,k} = 1$ if and only if the k-th cell is intended to be programmed on the j-th write. Similarly, Y_j, is a length-n binary vector, where $Y_{j,k} = 1$ if and only if the value of the k-th cell was successfully changed on the j-th write, that is, $Y_j = \boldsymbol{c}_j \oplus \boldsymbol{c}_{j-1}$.

For $1 \leqslant j \leqslant t$ and $i \in [\ell+1]$, we define the probabilities $p_{j,0}$, $p_{j,1}$, and $Q_{j,i}$ as follows. $p_{j,k}$ is the probability of programming a cell on the j-th write given that the value of this cell was k, $k \in \{0, 1\}$, and $Q_{j,i}$ is the probability of a cell to be programmed exactly i times after the first j writes. Additionally, let $Q_{j,e}, Q_{j,o}$ be the probability of a cell to be programmed an even, odd number of times after the first j writes, respectively. Formally, $Q_{j,i}$, $Q_{j,e}$, and $Q_{j,o}$ are defined recursively by using the probabilities $p_{j',0}$ and $p_{j',1}$ for $j' \leqslant j$. We now assume that ℓ is even. The case of an odd ℓ is defined similarly. We define $Q_{j,i}$ for $j > 0$ as follows. For even $i \geqslant 0$,

$$Q_{j,i} = \begin{cases} Q_{j-1,i-1} \cdot p_{j,1} + Q_{j-1,i} \cdot (1 - p_{j,0}), & \text{if } 0 < i \leqslant \ell. \\ Q_{j-1,i} \cdot (1 - p_{j,0}), & \text{if } i = 0, \end{cases} \quad (3)$$

and for odd $i > 0$, $Q_{j,i} = Q_{j-1,i-1} \cdot p_{j,0} + Q_{j-1,i} \cdot (1 - p_{j,1})$. The base $j = 0$, is $Q_{0,0} = 1$ and $Q_{0,i} = 0$ for $i > 0$. Furthermore, let $Q_{j,e} = \sum_{i=0}^{\ell/2} Q_{j,2i}$ and $Q_{j,o} = \sum_{i=1}^{\ell/2} Q_{j,2i-1}$.

Next, we define the rates region $\widetilde{\mathcal{C}}_{t,\ell}^{lb}$, which will be shown to be achieved, and the rates region $\widetilde{\mathcal{C}}_{t,\ell}^{ub}$ which will be shown to be an upper bound for the capacity region. That is, $\widetilde{\mathcal{C}}_{t,\ell}^{lb} \subseteq \mathcal{C}_{t,\ell}^{EIP:DIA,\epsilon} \subseteq \widetilde{\mathcal{C}}_{t,\ell}^{ub}$. In addition, it will be shown that $\widetilde{\mathcal{C}}_{t,\ell}^{ub} \subseteq \mathcal{C}_{t,\ell}$ for all t, ℓ. We present here the definition for even ℓ, while the odd case case can be defined similarly.

$$\widetilde{\mathcal{C}}_{t,\ell}^{lb} = \Big\{ (\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_t) | \forall 1 \leqslant j \leqslant t : \\ \mathcal{R}_j \leqslant Q_{j-1,o} h(p_{j,1}) + Q_{j-1,e} h(p_{j,0}) - Q_{j-1,\ell}, \\ p_j \in [0, 0.5] \text{ and } Q_{j,e}, Q_{j,o}, Q_{j,\ell} \text{ are defined above.} \Big\}, \quad (4)$$

$$\widetilde{\mathcal{C}}_{t,\ell}^{ub} = \Big\{ (\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_t) | \forall 1 \leqslant j \leqslant t : \\ \mathcal{R}_j \leqslant Q_{j-1,o} h(p_{j,1}) + (Q_{j-1,e} - Q_{j-1,\ell}) h(p_{j,0}), \\ p_j \in [0, 0.5] \text{ and } Q_{j,e}, Q_{j,o}, Q_{j,\ell} \text{ are defined above.} \Big\}. \quad (5)$$

For example, by substituting $p_{3,0} = p_{3,1} = 0.5$ in Equations (2), (4), and (5) for $t = 3$ and $\ell = 2$ we obtained the following region.

$$\widetilde{\mathcal{C}}_{3,2}^{lb} = \widetilde{\mathcal{C}}_{3,2}^{ub} = \mathcal{C}_{3,2} = \Big\{ (R_1, R_2, R_3) | R_1 \leqslant h(p_{1,0}), \\ R_2 \leqslant (1 - p_{1,0}) \cdot h(p_{2,0}) + p_{1,0} \cdot h(p_{2,1}), \\ R_3 \leqslant (1 - p_{1,0} \cdot p_{2,1}), \\ p_{1,0}, p_{2,0}, p_{2,1} \in [0, 0.5] \Big\},$$

The next theorem presents the main result of this section.

Theorem 3. *For all $t > \ell$, $\widetilde{\mathcal{C}}_{t,\ell}^{lb} \subseteq \mathcal{C}_{t,\ell}^{EIP:DIA,\epsilon} \subseteq \widetilde{\mathcal{C}}_{t,\ell}^{ub}$.*

Proof Sketch: For the direct part we should prove that for each $\epsilon > 0$ and $(\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_t) \in \widetilde{\mathcal{C}}_{t,\ell}^{lb}$, there exists an $[n, t; M_1, \ldots, M_t]_{t,\ell}^{EIP:DIA,\boldsymbol{p}_e}$ ELM code, where for all $1 \leqslant j \leqslant t$, $\frac{\log M_j}{n} \geqslant \mathcal{R}_j - \epsilon$ and $\boldsymbol{p}_e = (p_{e_1}, \ldots, p_{e_t}) \leqslant (\epsilon, \ldots, \epsilon)$. We use the well-known random channel-coding theorem [4, p. 200] on each write.

For the converse part, let $S_1, \ldots, S_t$ be independent random variables, where S_j is uniformly distributed over the messages set $[M_j]$, and $\hat{S}_j$ is the decoding result on the j-th write. Let V_j be an independent random variable on N_j, the set of the cell-programs-count vectors after the first j writes. The data processing yields the following Markov chain:

$$S_j | V_{j-1} — X_j | V_{j-1} — Y_j | V_{j-1} — \hat{S}_j | V_{j-1},$$

and therefore, $I(X_j; Y_j | V_{j-1}) \geqslant I(S_j; \hat{S}_j | V_{j-1})$. Additionally, applying Fano's inequality [4, p. 38] yields $I(S_j; \hat{S}_j | V_{j-1}) \geqslant \log(M_j) - H(p_{e_j}) - p_{e_j} \log(M_j)$. Let L be an index random variable, which is uniformly distributed over the index set $[n]$. We have that

$$\frac{1}{n} I(X_j; Y_j | V_{j-1}) \leqslant \sum_{i=0}^{\ell-1} Pr(V_{j-1,L} = i) H(Y_{j,L} | V_{j-1,L} = i).$$

Now, we set $p_{j,0} = Pr(X_{j,L} = 1 | V_{j-1,L} \mod 2 = 0)$ and similarly $p_{j,1} = Pr(X_{j,L} = 0 | V_{j-1,L} \mod 2 = 1)$. Thus, for even $i < \ell$, $H(Y_{j,L} | V_{j-1,L} = i) = H(p_{j,0})$, and for odd $i < \ell$, $H(Y_{j,L} | V_{j-1,L} = i) = H(p_{j,1})$. We also define for $i \in [\ell+1]$ $Q_{j,i} = Pr(V_{j,L} = i)$ and we note that $Q_{j,i}$ can be calculated as in Equation (3). Additionally, we use the notations $Q_{j,o}$ and $Q_{j,e}$ as defined above. Finally, we conclude that

$$\frac{\log(M_j)}{n} - \epsilon_j \leqslant \frac{1}{n} I(X_j; Y_j | V_{j-1}) \\ \leqslant Q_{j-1,o} h(p_{j,1}) + (Q_{j-1,e} - Q_{j-1,\ell}) h(p_{j,0}),$$

where $\epsilon_j = \frac{H(p_{e_j}) + p_{e_j} \log(M_j)}{n}$, and the converse part is implied. ■

The next theorem proves that if $t = \ell + 1$ or $t = \ell + 2$, then there is strict equality between the two regions $\widetilde{\mathcal{C}}_{t,\ell}^{lb}$ and $\widetilde{\mathcal{C}}_{t,\ell}^{ub}$.

Theorem 4. *If $t \in \{\ell+1, \ell+2\}$ then $\widetilde{\mathcal{C}}_{t,\ell}^{lb} = \widetilde{\mathcal{C}}_{t,\ell}^{ub} = \mathcal{C}_{t,\ell}^{EIP:DIA,\epsilon}$.*

Proof: If we use the same probabilities $p_{j,0}, p_{j,1}$, $1 \leqslant j \leqslant t$, for these two regions, then the gap between the rates on the j-th write is $Q_{j-1,\ell} \cdot h(p_{j,i})$, where $i \in [2]$, and $i = \ell \pmod 2$. For $j < \ell$, it holds that $Q_{j,\ell} = 0$. Thus, we can assign $Q_{j-1,\ell} = 0$ in $\widetilde{\mathcal{C}}_{t,\ell}^{lb}$ and in $\widetilde{\mathcal{C}}_{t,\ell}^{ub}$ for all $j < t$. For $j = t$, the maximum sum-rate is obtained for $p_{t,0} = p_{t,1} = 0.5$. Thus, if $t = \ell + 1$ we get $\widetilde{\mathcal{C}}_{t,\ell}^{lb} = \widetilde{\mathcal{C}}_{t,\ell}^{ub}$. However, if $t = \ell + 2$ then for $j = t - 1 = \ell + 1$, and $i = \ell \pmod 2$, $i \in [2]$ the maximum sum-rate is obtained for $p_{j,i} = 0.5$. Thus, $\widetilde{\mathcal{C}}_{t,\ell}^{lb}$ and $\widetilde{\mathcal{C}}_{t,\ell}^{ub}$ are exactly the same region, and this is the capacity region of the EIP:DIA model for the ϵ-error case. ■

On the other hand, the next theorem proves that if $t > \ell + 2$ then the two regions are not the same.

Theorem 5. *For all $t > \ell + 2$, $\widetilde{\mathcal{C}}_{t,\ell}^{lb} \subsetneq \widetilde{\mathcal{C}}_{t,\ell}^{ub}$, and the maximum sum-rate in $\widetilde{\mathcal{C}}_{t,\ell}^{lb}$ is smaller than the maximum sum-rate obtained in $\widetilde{\mathcal{C}}_{t,\ell}^{ub}$.*

Proof: By Theorem 3, we have $\widetilde{\mathcal{C}}_{t,\ell}^{lb} \subseteq \widetilde{\mathcal{C}}_{t,\ell}^{ub}$. Now, we prove that $\widetilde{\mathcal{C}}_{t,\ell}^{lb} \neq \widetilde{\mathcal{C}}_{t,\ell}^{ub}$. Let $\widetilde{\mathbf{R}}^{ub} = (\widetilde{\mathcal{R}}_1^{ub}, \ldots, \widetilde{\mathcal{R}}_t^{ub}) \in \widetilde{\mathcal{C}}_{t,\ell}^{ub}$ be a maximal rate tuple (that is, there does not exists $\mathbf{R}' \in \widetilde{\mathcal{C}}_{t,\ell}^{ub}$ such that $\widetilde{\mathbf{R}}^{ub} < \mathbf{R}'$) where $\widetilde{\mathcal{R}}_j^{ub} \neq 0$ for all $1 \leqslant j \leqslant \ell + 3$. We denote the probabilities which achieve $\widetilde{\mathbf{R}}^{ub}$ by $\widetilde{p}_{j,i}^{ub}, 1 \leqslant j \leqslant t, i \in [2]$, and $\widetilde{Q}_{j,i}^{ub}$, $1 \leqslant j \leqslant t, i \in [\ell+1] \cup \{e, o\}$, is defined above in this section (see Equation (3)) using the $\widetilde{p}_{j',i'}^{ub}$ probabilities. We can assume without loss of generality that $\widetilde{Q}_{\ell,\ell}^{ub} > 0$ and $\widetilde{p}_{\ell+1,i}^{ub} < 0.5$ where $i = \ell \pmod 2$, $i \in [2]$ (otherwise, $\widetilde{\mathbf{R}}^{ub}$ does not achieve maximum sum-rate). We can prove by induction on j that for obtaining $\widetilde{\mathbf{R}}^{ub}$ in $\widetilde{\mathcal{C}}_{t,\ell}^{lb}$ using the probabilities $\widetilde{p}_{j,i}^{lb}$, we must set $\widetilde{p}_{j,i}^{lb} = \widetilde{p}_{j,i}^{ub}$. Thus, $\widetilde{Q}_{j,i}^{lb} = \widetilde{Q}_{j,i}^{ub}$ for all j, i, where $\widetilde{Q}_{j,i}^{lb}$ is defined as $\widetilde{Q}_{j,i}^{ub}$ (see Equation (3)) using the $\widetilde{p}_{j',i'}^{lb}$ probabilities, and hence $\widetilde{\mathcal{R}}_{\ell+1}^{lb} < \widetilde{\mathcal{R}}_{\ell+1}^{ub}$. The theorem for the capacity region is an immediate conclusion, and by choosing $\widetilde{\mathbf{R}}^{ub}$ as a rate tuple which achieves the maximum sum-rate in $\widetilde{\mathcal{C}}^{ub}_{t,\ell}$, we get the result regarding the maximum sum-rate. ■

In Section V we present capacity achieving codes for the EIP:DIA model for the case of $\ell = 2$. Then, in Section VI we compare between the the EIP:DIA model which was discussed in this section, and the EIA models which were presented in Section III.

V. Constructions for the EIP:DIA Model

Our goal in this section is to construct a family of capacity achieving t-write ℓ-change EIP:DIA ELM codes, for $\ell = 2$. Before presenting our main result, we introduce the following family of EU:DI WOM codes [7]. These codes will be used as an important component in our construction of EIP:DIA ELM codes. The EU:DI WOM codes were defined in [7] as follows.

Definition 6. *An $[n, 2; M_1, M_2]_2^{EU:DI,(p_{e_1},p_{e_2})}$ two-write binary EU:DI WOM code is a coding scheme comprising of n bits. It consists of two pairs of encoding and decoding maps $(\mathcal{E}_1^{EU:DI}, \mathcal{D}_1^{EU:DI})$ and $(\mathcal{E}_2^{EU:DI}, \mathcal{D}_2^{EU:DI})$. For the map $\mathcal{E}_i^{EU:DI}$, $Im(\mathcal{E}_i^{EU:DI})$ is its image and $Im^*(\mathcal{E}_i^{EU:DI})$ is the cell-state vectors which can be obtained after the i-th write. The encoding and decoding maps are defined as follows. $\mathcal{E}_i^{EU:DI} : [M_i] \mapsto [2]^n$ and $\mathcal{D}_i^{EU:DI} : Im^*(\mathcal{E}_{i-1}^{EU:DI}) \times Im^*(\mathcal{E}_i^{EU:DI}) \mapsto [M_2]$ such that for all $m \in [M_i]$,*

$$\sum_{(m,\boldsymbol{c})\in[M_i]\times Im^*(\mathcal{E}_{i-1}^{EU:DI})} Pr(m)Pr(\boldsymbol{c})\cdot Ind_m\big(\mathcal{D}_i^{EU:DI}(\max\{\boldsymbol{c},\mathcal{E}_i^{EU:DI}(m)\},\boldsymbol{c})\big)\leqslant p_{e_i}.$$

We note that there exists a family of two-write binary EU:DI WOM codes which achieve the rates pair $\mathbf{R} = (R_1, R_2)$ for all $R_1 \leqslant h(p_1)$ and $R_2 \leqslant (1 - p_1)h(p_2)$ such that $0 \leqslant p_1, p_2 \leqslant 1$ [14]. Several explicit constructions of such codes were presented in [7]. In these presented codes, $Im(\mathcal{E}_1)^{EU:DI}$ is a subset of all the length-n binary vectors of Hamming weight at most $\lfloor p_1 \cdot n\rfloor$. We refer to these codes as $[n, 2; M_1, M_2]_2^{EU:DI,(0,p_{e_2})}(p_1, p_2)$ Using this family of WOM codes, we can construct a family of capacity achieving three-write two-change EIP:DIA ELM codes.

Construction 7. *Given $p_{j,i} \in [0, 0.5]$ where $1 \leqslant j \leqslant 3$ and $i \in [2]$, and a $[Q_{2,e}n, 2; M_1', M_2']_2^{EU:DI,(0,\epsilon)}(\frac{Q_{2,2}}{Q_{2,e}}, 0.5)$ WOM code, ($Q_{2,2} = p_{1,0}p_{2,1}$ and $Q_{2,e} = p_{1,0}p_{2,1} + (1 - p_{1,0})(1 - p_{2,0})$)[1], we construct an $[n, 3, 2; M_1, M_2, M_3]^{EIP:DIA,(0,0,\epsilon)}$ ELM code such that $M_1 = \binom{n}{\lfloor p_{1,0}n\rfloor}$, $M_2 = \binom{\lfloor p_{1,0}n\rfloor}{\lfloor p_{1,0}p_{2,1}n\rfloor}\cdot\binom{n-\lfloor p_{1,0}n\rfloor}{\lfloor p_{2,0}(n-p_{1,0}n)\rfloor}$, and $M_3 = \lfloor 2^{n(1-p_{1,0}p_{2,1})}\rfloor$.*

1) ***First write:*** *We use the pair of encoder/decoder of a constant weight code of length n, weight $\lfloor p_{1,0}n\rfloor$ to encode/decode in the first write. The rate is $R_1 = h(p_{1,0})$.*
2) ***Second write:*** *The idea is to write a binary constant weight code of length $\lfloor p_{1,0}n\rfloor$, weight $\lfloor p_{1,0}p_{2,1}n\rfloor$ into all positions with value 1 after the first write and write a binary constant weight code of length $\lfloor n - p_{1,0}n\rfloor$, weight $\lfloor(n-p_{1,0}n)p_{2,0}\rfloor$ into all positions with value 0 after the first write. Since the decoder knows all previous states of all cells, it can decode the two constant weight codes separately to get the original information. The value M_2 is the product of the sizes of the two constant weight codes. The rate is $R_2 = p_{1,0}h(p_{2,1}) + (1 - p_{1,0})h(p_{2,0})$.*
3) ***Third write:*** *Let $\mathbf{c}_2$ be the cell-state vector after the second write. The encoder on the third write does not have the information of the vector $N_2 \in [3]^n$, but it knows the positions of all cells with value 1, that is, the cells which were programmed exactly once. The encoder can write any word of length $\lfloor(n - p_{1,0}n)p_{2,0}\rfloor + \lfloor p_{1,0}n\rfloor - \lfloor p_{1,0}p_{2,1}n\rfloor$ into these positions. For the rest of all cells with value 0, the encoder does not know which cells are programmed twice and which cells are not programmed while the decoder has this information. The idea is to write on these cells as in the second write of the EU:DI WOM code. Since the decoder has full information on the cells, it decodes the message on the cells with value 0, by considering cells which were programmed twice in the first two writes as programmed cells in WOM. The rate on this write equals to the weighted sum of the rates from the two codes. The first code which contains all words of length $\lfloor(n - p_{1,0}n)p_{2,0}\rfloor + \lfloor p_{1,0}n\rfloor - \lfloor p_{1,0}p_{2,1}n\rfloor$ contributes to the rate $R_{3,1} = (1 - p_{1,0})p_{2,0} + p_{1,0}(1 - p_{2,1})$. The second code which is the code of the second write of the EU:DI WOM code applied on the cells with value 0, provides $R_{3,2} = (1 - p_{1,0})(1 - p_{2,0})$. Hence, the rate on the third write is $R_3 = R_{3,1} + R_{3,2} = 1 - p_{1,0}p_{2,1}$.*

By the above construction, we can achieve for the EIP:DIA model any rate tuple from $\mathcal{C}_{3,2}$, and hence we can conclude that $\mathcal{C}_{3,2}^{EIA} = \mathcal{C}_{3,2}^{EIP:DIA,\epsilon}$. Furthermore, the technique in Construction 7 can be generalized for all t when $\ell = 2$. Thus, $\mathcal{C}_{t,2}^{EIA} = \mathcal{C}_{t,2}^{EIP:DIA,\epsilon}$ for all t as summarized in Corollary 10. Note that for $t = 3$ we can simplify the third write in Construction 7 by applying the second write of $[n, 2; M_1', M_2']_2^{EU:DI,(0,\epsilon)}(p_{1,0}p_{2,1}, 0.5)$ WOM code. However, we choose to present here the more complicated construction in order to clarify the generalization in Construction 8. Due to lack of space, we only sketch the idea of the general construction.

Construction 8. *Given $p_{j,i} \in [0, 0.5]$ where $1 \leqslant j \leqslant t$ and $i \in [2]$, let C_k be a $[Q_{k-1,e}n, 2; M_{k1}, M_{k2}]_2^{EU:DI,(0,\epsilon)}(\frac{Q_{k-1,2}}{Q_{k-1,e}}, p_{k,0})$ WOM code, $3 \leqslant k \leqslant t$, where $M_{k2} = 2^{Q_{k-1,e}nR_{k2}}$ and $R_{k2} = (1 - \frac{Q_{k-1,2}}{Q_{k-1,e}})h(p_{k,0})$.*
We construct an $[n, t, 2; M_1, \ldots, M_t]^{EIP:DIA,(0,0,\epsilon,\ldots,\epsilon)}$ ELM code such that the values of M_1 and M_2 are exactly as in Construction 7, and $M_j = 2^{nR_j}$ where $R_j = Q_{j-1,e}R_{j2} + Q_{j-1,1}h(p_{j,1})$ for $3 \leqslant j \leqslant t$. The construction is as follows.

1) *The first two writes follow the ones from Construction 7.*
2) *On the j-th write, where $j \geqslant 3$, the encoder knows the binary cell-state vector $\mathbf{c}_{j-1}$ and does not have information about the cell-program-count vector N_{j-1} which the decoder knows. For any given δ and ϵ, the weight of $\mathbf{c}_{j-1}$ is $w(\mathbf{c}_{j-1}) \in [(Q_{j-1,1} - \delta)n, (Q_{j-1,1} + \delta)n]$ with high probability ($p \geqslant 1 - \epsilon$). In all the positions with value 1, the encoder can write a word of length $w(\mathbf{c}_{j-1})$ and weight approximately $(1 - p_{j,1})w(\mathbf{c}_{j-1})$.For the rest of the cells, all positions with value 0, the encoder can write according the the second write of C_j, an EU:DI WOM code with probability $p_2 = p_{j,0}$.*

VI. The EIP:DIA Model Vs. the EIA Models

In this section we compare between the EIP:DIA model and the EIA models. The capacity of the EIA models, $\mathcal{C}_{t,\ell}^{EIA:DY,g}$ for $g \in \{z, \epsilon\}$ and $Y \in \{IA, IP, U\}$ was stated in Section III to be equal to $\mathcal{C}_{t,\ell}$, while in Section IV we presented upper and lower bounds for the capacity of the EIP:DIA model for the ϵ-error case, $\widetilde{\mathcal{C}}_{t,\ell}^{lb} \subseteq \mathcal{C}_{t,\ell}^{EIP:DIA,\epsilon} \subseteq \widetilde{\mathcal{C}}_{t,\ell}^{ub}$. For $t \in \{\ell + 1, \ell + 2\}$ it was proved in Theorem 4 that these bounds are tight.

[1]We assume here and in the rest of the section that n is sufficiently large so that $Q_{j,e}n$ is an integer number for all $2 \leqslant j \leqslant t - 1$.

The next theorem proves that for $t > \ell \geqslant 3$ the capacity regions $\mathcal{C}_{t,\ell}^{EIP:DIA,\epsilon}$ and $\mathcal{C}_{t,\ell}^{EIA}$ are different, while for $\ell = 2$ these regions were shown to be the same using Construction 8.

Theorem 9. *For $t > \ell \geqslant 3$, $\widetilde{\mathcal{C}}_{t,\ell}^{ub} \subsetneq \mathcal{C}_{t,\ell}$. Hence $\mathcal{C}_{t,\ell}^{EIP:DIA,\epsilon} \subsetneq \mathcal{C}_{t,\ell}^{EIA}$.*

Proof: It can be readily verified that $\widetilde{\mathcal{C}}_{t,\ell}^{ub} \subseteq \mathcal{C}_{t,\ell}$. Next, we prove by induction on t, ℓ that for $t > \ell \geqslant 3$, $\widetilde{\mathcal{C}}_{t,\ell}^{ub} \subsetneq \mathcal{C}_{t,\ell}$. The base of the induction is $t = 4, \ell = 3$. The maximum sum-rate in $\mathcal{C}_{4,3}$ is obtained for $p_{2,0} = p_{3,1} = 0.5$, $p_{3,2} < p_{3,0} = 0.5$ (for completeness, $p_{4,i} = 0.5$ for $i \in [\ell]$, $p_{1,0} = 0.467$,$p_{2,1} = 0.429$, and $p_{3,2} = 0.333$, and the rate tuple is $(0.997, 0.993, 0.984, 0.933)$). Thus, in order to achieve this rate tuple in $\widetilde{\mathcal{C}}_{t,\ell}^{ub}$ the probabilities denoted by $\widetilde{p}_{j,0}^{ub}, \widetilde{p}_{j,1}^{ub}$ must be as follows. For achieving $\mathcal{R}_1$ we should set $\widetilde{p}_{1,0}^{ub} = p_{1,0}$. Then, for $\mathcal{R}_2$ we must set $\widetilde{p}_{2,0}^{ub} = p_{2,0} = 0.5$, $\widetilde{p}_{2,1}^{ub} = p_{2,1}$ (otherwise, if we set $\widetilde{p}_{2,1}^{ub} \neq p_{2,1}$ then a bigger rate will be achieved in the second write, but it will decrease the sum of the rates in the next writes). For $j = 3$, we must set $\widetilde{p}_{3,1}^{ub} = p_{3,1} = 0.5$, and then there does not exist a value $\widetilde{p}_{3,0}^{ub}$ which achieves $\mathcal{R}_3$, since instead of two different probabilities $p_{3,0}$ and $p_{3,2}$ in $\mathcal{C}_{4,3}$, there exists only one parameter $\widetilde{p}_{3,0}^{ub}$ in $\widetilde{\mathcal{C}}_{4,3}^{ub}$.

For the induction step, we assume $\mathbf{R} = (\mathcal{R}_1, \ldots, \mathcal{R}_t) \in \mathcal{C}_{t,\ell} \setminus \widetilde{\mathcal{C}}_{t,\ell}^{ub}$, and we prove the claim for two different cases; $(t+1, \ell)$ and $(t+1, \ell+1)$.

For $(t+1, \ell)$ we have $\mathbf{R}' = (\mathcal{R}_1, \ldots, \mathcal{R}_t, 0) \in \mathcal{C}_{t+1,\ell}$. Assume by contradiction that $\mathcal{C}_{t+1,\ell} = \widetilde{\mathcal{C}}_{t+1,\ell}^{ub}$. Then, $\mathbf{R}' = (\mathcal{R}_1, \ldots, \mathcal{R}_t, 0) \in \widetilde{\mathcal{C}}_{t+1,\ell}^{ub}$, which implies that $\mathbf{R} = (\mathcal{R}_1, \ldots, \mathcal{R}_t) \in \widetilde{\mathcal{C}}_{t,\ell}^{ub}$ in contradiction.

For $(t+1, \ell+1)$ we have $\mathbf{R}' = (\mathcal{R}_1, \ldots, \mathcal{R}_t, 1) \in \mathcal{C}_{t+1,\ell+1}$, where $Q_{t,\ell+1} = 0$. Assume by contradiction that $\mathcal{C}_{t+1,\ell+1} = \widetilde{\mathcal{C}}_{t+1,\ell+1}^{ub}$. Then, $\mathbf{R}' = (\mathcal{R}_1, \ldots, \mathcal{R}_t, 1) \in \widetilde{\mathcal{C}}_{t+1,\ell+1}^{ub}$, with $Q_{t,\ell+1} = 0$ (otherwise, $\mathcal{R}_{t+1} < 1$). Hence, we can conclude that $\mathbf{R} = (\mathcal{R}_1, \ldots, \mathcal{R}_t) \in \widetilde{\mathcal{C}}_{t,\ell}^{ub}$, which it is a contradiction. ∎

We note that by the proof of the induction base case in Theorem 9 we can conclude also that $\mathcal{R}_{4,3}^{EIP:DIA,\epsilon} < \mathcal{R}_{4,3}^{EIA}$, but we can not deduce it for all $t > \ell \geqslant 3$. We can summarize this section in the following corollary.

Corollary 10 *For all $t > \ell$ the following holds*

$$\widetilde{\mathcal{C}}_{t,\ell}^{lb} \subseteq \mathcal{C}_{t,\ell}^{EIP:DIA,\epsilon} \subseteq \widetilde{\mathcal{C}}_{t,\ell}^{ub} \subseteq \mathcal{C}_{t,\ell} = \mathcal{C}_{t,\ell}^{EIA}.$$

- *For $t \in \{\ell+1, \ell+2\}$, $\widetilde{\mathcal{C}}_{t,\ell}^{lb} = \mathcal{C}_{t,\ell}^{EIP:DIA,\epsilon} = \widetilde{\mathcal{C}}_{t,\ell}^{ub}$.*
- *For $t > \ell+2$, $\widetilde{\mathcal{C}}_{t,\ell}^{lb} \subsetneq \widetilde{\mathcal{C}}_{t,\ell}^{ub}$.*
- *For $t \geqslant \ell = 2$ all these regions are equal, in particular, $\mathcal{C}_{t,2}^{EIP:DIA,\epsilon} = \mathcal{C}_{t,2}^{EIA}$.*
- *For $t > \ell \geqslant 3$, $\widetilde{\mathcal{C}}_{t,\ell}^{ub} \subsetneq \mathcal{C}_{t,\ell}$, and hence $\mathcal{C}_{t,\ell}^{EIP:DIA,\epsilon} \subsetneq \mathcal{C}_{t,\ell}^{EIA}$.*

VII. Capacity of the EU:DIA model

In this section we discuss the capacity region and the maximum sum-rate of the EU:DIA model. The proofs in this section are similar to those in Section IV, and due to lack of space are omitted.

For $1 \leqslant j \leqslant t$ and $i \in [\ell+1]$, we define the probabilities p_j, and $Q_{j,i}$ as follows. p_j is the probability of programming a cell on the j-th write, and $Q_{j,i}$ is the probability of a cell to be programmed exactly i times in the first j writes. Additionally, let $Q_{j,e}$ and $Q_{j,o}$ be the probability of a cell to be programmed even and odd times in the first j writes, respectively. Formally, $Q_{j,i}$, $Q_{j,e}$, and $Q_{j,o}$ are defined recursively by using $p_{j'}$ probabilities. For $j \geqslant 1$,

$$Q_{j,i} = \begin{cases} Q_{j-1,i-1} \cdot p_j + Q_{j-1,i} \cdot (1-p_j), & \text{if } 0 < i \leqslant \ell. \\ Q_{j-1,i} \cdot (1-p_j), & \text{if } i = 0, \end{cases} \quad (6)$$

where $Q_{0,0} = 1$ and $Q_{0,i} = 0$ for $i > 0$.

Next, we define the regions $\overline{\mathcal{C}}_{t,\ell}^{lb}$ and $\overline{\mathcal{C}}_{t,\ell}^{ub}$, which are lower and upper bounds for the capacity region $\mathcal{C}_{t,\ell}^{EU:DIA,\epsilon}$, respectively.

$$\overline{\mathcal{C}}_{t,\ell}^{lb} = \Big\{(\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_t) | \forall 1 \leqslant j \leqslant t : \mathcal{R}_j \leqslant h(p_j) - Q_{j-1,\ell} \quad (7)$$
$$p_j \in [0, 0.5],\ Q_{j,\ell} \text{ is defined above}\Big\}.$$

$$\overline{\mathcal{C}}_{t,\ell}^{ub} = \Big\{(\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_t) | \forall 1 \leqslant j \leqslant t : \mathcal{R}_j \leqslant h(p_j) - Q_{j-1,\ell} h(p_j), \quad (8)$$
$$p_j \in [0, 0.5],\ Q_{j,\ell} \text{ is defined above}\Big\}.$$

We conclude the results regarding the capacity of the $EU : DIA$ model in the following corollary.

Corollary 11 *For all $t \geqslant \ell$ the following holds*

$$\overline{\mathcal{C}}_{t,\ell}^{lb} \subseteq \mathcal{C}_{t,\ell}^{EU:DIA,\epsilon} \subseteq \overline{\mathcal{C}}_{t,\ell}^{ub} \subseteq \widetilde{\mathcal{C}}_{t,\ell}^{lb} \subseteq \mathcal{C}_{t,\ell}^{EIP:DIA}.$$

- *For $t = \ell+1$, $\ell \geqslant 2$, $\overline{\mathcal{C}}_{t,\ell}^{lb} = \mathcal{C}_{t,\ell}^{EU:DIA,\epsilon} = \overline{\mathcal{C}}_{t,\ell}^{ub}$.*
- *For $t \geqslant \ell+2$, $\ell \geqslant 2$, $\overline{\mathcal{C}}_{t,\ell}^{lb} \subsetneq \overline{\mathcal{C}}_{t,\ell}^{ub}$.*
- *For all $t > \ell \geqslant 2$, $\overline{\mathcal{C}}_{t,\ell}^{ub} \subsetneq \widetilde{\mathcal{C}}_{t,\ell}^{lb}$, and hence $\mathcal{C}_{t,\ell}^{EU:DIA,\epsilon} \subsetneq \mathcal{C}_{t,\ell}^{EIP:DIA,\epsilon}$.*
- *For $t \in \{\ell+1, \ell+2\}$, $\ell \geqslant 2$, $\mathcal{R}_{t,\ell}^{EU:DIA,\epsilon} < \mathcal{R}_{t,\ell}^{EIP:DIA,\epsilon}$.*

References

[1] Y. M. Chee, T. Etzion, H. M. Kiah and A. Vardy, "Cooling codes: Thermal-management coding for high-performance interconnects," *Proc. IEEE Int. Symp. on Inform. Theory*, Aachen, Germany, Jun. 2017.

[2] Y. M. Chee, M. Horovitz, A. Vardy, V. K. Vu and E. Yaakobi, "Codes for Endurance-Limited Memories," *Proc. IEEE Int. Symp. on Inform. Theory and Its App.*, Singapore, October , 2018

[3] Y. M. Chee, H. M. Kiah, A. Vardy, and E. Yaakobi, "Explicit Constructions of Finite-Length WOM Codes", *Proc. IEEE Int. Symp. on Inform. Theory*, Aachen, Germany, Jun. 2017.

[4] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 1st Edition. New York: Wiley-Interscience, 1991.

[5] F. Fu and A. J. H. Vinck, "On the capacity of generalized write-once memory with state transitions described by an arbitrary directed acyclic graph," *IEEE Trans. Inform. Theory*, vol. 45, no. 1, pp. 308–313, Sep. 1999.

[6] C. Heegard, "On the capacity of permanent memory," *IEEE Trans. Inform. Theory*, vol. 31, no. 1, pp. 34–42, Jan. 1985.

[7] M. Horovitz and E. Yaakobi, "On the capacity of write-once memories," *IEEE Trans. Inform. Theory*, vol. 63, no. 8, pp. 5124–5137, Aug. 2017.

[8] Y. Kim, A. A. Sharma, R. Mateescu, S. H. Song, Z. Z. Bandic, J. A. Bain, and B. V. K. Vijaya Kumar, "Locally rewritable codes for resistive memories," *IEEE J. Selected Areas in Comm.*, vol. 34, no. 9, pp. 2470–2485, Sep. 2016.

[9] R. Maddah, R. Melhem, and S. Cho, "RDIS: Tolerating many stuck-at aaults in resistive memory", *IEEE Trans. Computers*, vol. 64, no. 3, pp. 847–861, Mar. 2015.

[10] R. L. Rivest and A. Shamir, "How to reuse a write-once memory," *Inform. and Contr.*, vol. 55, no. 1–3, pp. 1–19, Dec. 1982.

[11] S. Schechter, G.H. Loh, K. Strauss, and D. Burger, "Use ECP, not ECC, for hard failures in resistive memories," *Proc. of the 37th Annual Int. Symp. on Comp. Arch.*, pp. 141–152, Saint-Malo, France, 2010.

[12] A. Shpilka, "New constructions of WOM codes using the Wozencraft ensemble", *IEEE Trans. Inform. Theory,* Vol. 59, No. 7, 2013.

[13] A. Shpilka, "Capacity-Achieving Multiwrite WOM Codes", *IEEE Trans. Inform. Theory*, Vol. 60, No. 3, pp.1481–1487, 2014.

[14] J. K. Wolf, A. D. Wyner, J. Ziv, and J. Korner, "Coding for a write-once memory," *AT&T Bell Labs. Tech. J.*, vol. 63, no. 6, pp. 1089–1112, 1984.

[15] C. Xu, D. Niu, Y. Zheng, S. Yu, and Y. Xie, "Impact of cell failure on reliable cross-point resistive memory design," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 20, no. 4, pp. 63:1–63:21, Sep. 2015.

[16] L. Zhang, B. Neely, D. Franklin, D. Strukov, Y. Xie, and F. T. Chong, "Mellow writes: Extending lifetime in resistive memories through selective slow write backs", *2016 ACM/IEEE 43rd Annual Int. Symp. on Comp. Arch.*, pp. 519–531, Jun. 2016.