Codes for Endurance-Limited Memories

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

[*] School of Physical and Mathematical Sciences, Nanyang Technological University, 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
[§] Department of Computer Science, Technion — Israel Institute of Technology, Haifa, 32000 Israel
Emails:{**ymchee,vankhu001**}**@ntu.edu.sg, horovitzmic@telhai.ac.il, avardy@ucsd.edu, yaakobi@cs.technion.ac.il**

***Abstract*—*Resistive memories*, such as *phase change memories* and *resistive random access memories* have attracted significant attention in recent years due to their better scalability, speed, rewritability, and yet non-volatility. However, their *limited endurance* is still a major drawback that has to be improved before they can be widely adapted in large-scale systems.**

In this work, in order to reduce the wearout of the cells, we propose a new coding scheme, called *Endurance-Limited Memories* (*ELM*) code, that 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 each cell is programmed at most ℓ times. In case $\ell = 1$ then these codes coincide with the well-studied *write-once memory* (*WOM*) codes. We study four models of these codes which depend upon whether the encoder knows, on each write, the number of times each cell was programmed or only knows its state. For the decoder, we consider two cases which depend upon whether the decoder knows the previous state of the memory or not. For two of these models we fully characterize the capacity regions and present partial results for another model. Although only one of the four models is suitable for resistive memories, we consider all four in order to carry out a complete information-theory study of endurance-limited codes.

I. Introduction

Emerging resistive memory technologies, such as *resistive random access memories* (*ReRAM*) and *phase-change memories* (*PCM*), have the potential to be the future's universal memories. They combine several important attributes starting from the speed of SRAM, the density of DRAM, and the non-volatility of flash memories. However, they fall short in their *write endurance*, which significantly increases their Bit Error Rate (BER). Hence, solving the limited endurance of these memories is crucial before they can be widely adapted in large-scale systems.

Resistive memories are nonvolatile memories which are composed of cells. The information is stored in the cells by changing their resistance. They combine both properties of DRAM and flash memories. Similarly to flash memories and unlike DRAM they are nonvolatile memories and thus they do not require refresh operations. Furthermore, like DRAM and unlike flash memories they are rewritable without an erase operation. The main challenge that has remained to be solved in order to make these memories a legitimate candidate as a universal memory is their limited write endurance, which is the goal on this paper.

In order to combat the limited write endurance in resistive memories, this paper proposes to study a new family of codes, called *Endurance-Limited Memory* (*ELM*) codes. Assume there are n binary cells and t messages that are required to be stored in these cells sequentially. Assume also that each cell can be programmed at most $\ell \geqslant 1$ times. Then, we seek to find the set of achievable rates, i.e., the capacity region, and design code constructions for this model. Note that for $\ell = 1$, we get the classical problem of write-once memory (WOM) codes [2], [5], [6], [9], [12], [13].

Consider for example the setup in which $\ell = 2$ and three writes, i.e., $t = 3$. A naive solution is to use a two-write WOM code for the first two writes and then write n more bits on the third write. The maximum sum-rate using this solution will be $\log(3) + 1 = \log(6)$, while, as will be shown in the paper, the maximum sum-rate in this case is $\log(7)$. The intuition behind this is as follows. Let p_1 be the probability to program a cell on the first write, so we assume that $p_1 n$ cells are programmed. Then, on the second and third writes we have a two-write WOM code problem for the $p_1 n$ programmed cells, and for the $(1 - p_1)n$ non-programmed cells, we can write twice on them so no coding is needed. The maximum sum-rate is achieved for $p_1 = 3/7$ and $p_2 = 1/3$. However, it is still a challenging task to design codes that can approach sum-rate of $\log(7)$.

There are several models of ELM codes which can be studied. These models are distinguishable by the information that is available to the encoder and the decoder. In particular, for the encoder we consider three cases which depend upon whether the encoder knows the number of times each cell was programmed, *Encoder Informed All* (*EIA*), only the current state of the cell, *Encoder Informed Partially* (*EIP*), or no information about the cells state, *Encoder Uninformed* (*EU*). The decoder will also have three cases, corresponding to the same information that is available to the encoder. However, in this work we consider only four models, where the encoder can be Informed All (EIA) or Informed Partially (EIP) and decoder is Informed Partially (and in short Informed (*DI*)) or Uninformed (*DU*). We note that from the practical point of view, only the EIA:DU model suits the memories architecture, however we comprehensively study all models in order to provide a rigorous information-theoretic study of these codes.

Previous works have offered different solutions to combat the write endurance of resistive memories. In [7], the authors proposed to use Locally Repairable Codes (LRC) in order to construct codes with small rewriting locality in order to mitigate both the problems endurance and power consumption. In [15], the authors proposed mellow writes, a technique which is targeted to reduce the wearout of the writes rather than reducing the number of writes. Lastly, several other works proposed coding schemes which correct stuck-at cells; see e.g. [8], [10], [14].

The rest of this paper is organized as follows. In Section II, we formally define the models studied in this paper and discuss some basic observations. In Section III, we study the capacity region of the EIA:DI model. In section IV, we carry the same task for the EIA:DU model to show that its capacity region is the same as the EIA:DI model. Then, in Section V, we present several results on the EIP:DU model. Due to the lack of space, some proofs and details in the paper are omitted. The EIP:DI model will be studied in the full version of the paper.

Copyright (C) 2018 by IEICE

II. Definitions and Preliminaries

For a positive integer a, the set $\{0, \ldots, a-1\}$ is defined by $[a]$. A vector $\boldsymbol{c} \in [2]^n$ will be called a *cell-state vector*. The vector $\boldsymbol{c} = \max\{\boldsymbol{c}_1, \boldsymbol{c}_2\}$ is defined by $c_i = \max\{c_{1,i}, c_{2,i}\}$ for all $1 \leqslant i \leqslant n$. The complement of a vector $\boldsymbol{c}$ is denoted by $\overline{\boldsymbol{c}}$. The all ones vector will be denoted by $\mathbf{1}$. In the models studied in this paper we assume that each cell can be programmed at most ℓ times, so if the encoder attempts to program a cell more than ℓ times then its value will not be changed. We see this as an extension of the WOM model for $\ell = 1$. Furthermore, since we study in this paper only the zero-error case, the codes we present will indeed satisfy this constraint.

Definition 1. *An $[n, t, \ell; M_1, \ldots, M_t]^{EX:DY}$ ℓ-change t-write ELM code, where $EX \in \{EIA, EIP\}$, $DY \in \{DI, DU\}$, 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 $Im(\mathcal{E}_0) = \{(0, \ldots, 0)\}$. For a cell-state vector $\boldsymbol{c}$, we let $N(\boldsymbol{c}) \in [\ell+1]^n$ be the vector indicating the number of times each cell was programmed. Furthermore, the set $N(Im(\mathcal{E}_j))$ is defined by $N(Im(\mathcal{E}_j)) = \{N(\boldsymbol{c}) : \boldsymbol{c} \in Im(\mathcal{E}_j)\}$. The encoding and decoding maps are defined as follows.*

(1) If $(EX, DY) = (EIA, DI)$ then for all $1 \leqslant j \leqslant t$,
$\mathcal{E}_j : [M_j] \times N(Im(\mathcal{E}_{j-1})) \mapsto [2]^n$,
$\mathcal{D}_j : \{(\mathcal{E}_j(m, N(\boldsymbol{c})), \boldsymbol{c}) : m \in [M_j], \boldsymbol{c} \in Im(\mathcal{E}_{j-1})\} \mapsto [M_j]$,
such that for all $(m, \boldsymbol{c}) \in [M_j] \times Im(\mathcal{E}_{j-1})$ it holds that $N(\mathcal{E}_j(m, N(\boldsymbol{c}))) \in [\ell+1]^n$ and $\mathcal{D}_j(\mathcal{E}_j(m, N(\boldsymbol{c})), \boldsymbol{c}) = m$.

(2) If $(EX, DY) = (EIP, DI)$ then for all $1 \leqslant j \leqslant t$,
$\mathcal{E}_j : [M_j] \times Im(\mathcal{E}_{j-1}) \mapsto [2]^n$,
$\mathcal{D}_j : \{(\mathcal{E}_j(m, \boldsymbol{c}), \boldsymbol{c}) : m \in [M_j], \boldsymbol{c} \in Im(\mathcal{E}_{j-1})\} \mapsto [M_j]$,
such that for all $(m, \boldsymbol{c}) \in [M_j] \times Im(\mathcal{E}_{j-1})$ it holds that $N(\mathcal{E}_j(m, \boldsymbol{c})) \in [\ell+1]^n$ and $\mathcal{D}_j(\mathcal{E}_j(m, \boldsymbol{c}), \boldsymbol{c}) = m$.

(3) If $(EX, DY) = (EIA, DU)$ then for all $1 \leqslant j \leqslant t$,
$\mathcal{E}_j : [M_j] \times N(Im(\mathcal{E}_{j-1})) \mapsto [2]^n$,
$\mathcal{D}_j : Im(\mathcal{E}_j) \mapsto [M_j]$,
such that for all $(m, \boldsymbol{c}) \in [M_j] \times Im(\mathcal{E}_{j-1})$ it holds that $N(\mathcal{E}_j(m, N(\boldsymbol{c}))) \in [\ell+1]^n$ and $\mathcal{D}_j(\mathcal{E}_j(m, \boldsymbol{c})) = m$.

(4) If $(EX, DY) = (EIP, DU)$ then for all $1 \leqslant j \leqslant t$,
$\mathcal{E}_j : [M_j] \times Im(\mathcal{E}_{j-1}) \mapsto [2]^n$,
$\mathcal{D}_j : Im(\mathcal{E}_j) \mapsto [M_j]$,
such that for all $(m, \boldsymbol{c}) \in [M_j] \times Im(\mathcal{E}_{j-1})$ it holds that $N(\mathcal{E}_j(m, \boldsymbol{c})) \in [\ell+1]^n$ and $\mathcal{D}_j(\mathcal{E}_j(m, \boldsymbol{c})) = m$.

For all $EX \in \{EIA, EIP\}$, $DY \in \{DI, DU\}$, the *rate* of an $[n, t, \ell; M_1, \ldots, M_t]^{EX:DY}$ ELM code on the j-th write 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 rates tuple $(R_1, \ldots, R_t)$ is called *achievable* in model EX:DY if for any $\epsilon > 0$ there exists an $[n, t, \ell; M_1, \ldots, M_t]_{\ell}^{EX:DY}$ ELM code such that $R_j \geqslant \frac{\log M_j}{n} - \epsilon$ for all $1 \leqslant j \leqslant t$. The *capacity region* of the EX:DY model is the set of all achievable rates tuples, $\mathcal{C}_{t,\ell}^{EX:DY} = \{(R_1, \ldots, R_t) | (R_1, \ldots, R_t) \text{ is achievable}\}$, and the *maximum sum-rate* will be denoted by $\mathcal{R}_{t,\ell}^{EX:DY}$. Note that the cell state is the parity of the number of times it was programmed. Thus, if the encoder (or the decoder) knows the vector $N(\boldsymbol{c})$, in particular it knows the cell-state vector $\boldsymbol{c}$. According to these definitions it is easy to verify the following relations

$$\mathcal{C}_{t,\ell}^{EIP:DU} \subseteq \mathcal{C}_{t,\ell}^{EIA:DU} \subseteq \mathcal{C}_{t,\ell}^{EIA:DI},$$
$$\mathcal{C}_{t,\ell}^{EIP:DU} \subseteq \mathcal{C}_{t,\ell}^{EIP:DI} \subseteq \mathcal{C}_{t,\ell}^{EIA:DI},$$

and similar connections hold for the maximum sum-rate.

For $\ell \geqslant t$ 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 [2], [5], [6], [9], [12], [13]. In this case we notice that the EIA and EIP models are the same. The capacity regions and maximum sum-rate are also known; see e.g. [5], [9], [13]. In the rest of this paper, and unless stated otherwise, we assume that $1 \leqslant \ell \leqslant t$. We conclude this section with the following remark.

Remark 1. The codes studied in this paper can be viewed as a generalization of WOM codes. Similarly to the research on WOM codes, we also consider all four models in Definition 1 even though only the fourth model, in which (EX,DY)=(EIP,DU), is suitable for Resistive memories. This extensive study not only provides a complete information-theoretic investigation of these codes, but we also use the results of the EIA:DI and EIA:DU models in order to derive an upper bound on the capacity of the EIP:DU model. Furthermore, we note that there is a strong connection between ELM codes and non-binary WOM codes. In fact, we can treat every cell as an $(\ell+1)$-ary cell where it is only possible to increase its level by one on each write and its maximum level is ℓ. While we will take advantage of existing WOM codes in order to construct ELM codes, unfortunately it is not possible to use existing capacity results on WOM codes such as the ones from [4]. The model in [4] assumes that the transitions between the cell levels are transitive, in the sense that if the transitions $i \to j$ and $j \to k$ are legal so is the transition $i \to k$. However, this condition does not hold in the model of this work. Further connections between these models will be analyzed in the full version of this paper.

III. Capacity of the EIA:DI Model

In this section we study the capacity region and maximum sum-rate of the EIA:DI model. Denote by $\boldsymbol{c}_i$, $i \in [t+1]$, the binary length-n vector which represents the cell-state vector after the i-th write, where $\boldsymbol{c}_0 = \mathbf{0}$. Recall that in the EIA:DI model on the i-th write the encoder knows the number of times each cell was programmed before the i-th write. That is, the encoder receives as an input a length-n vector $N(\boldsymbol{c}_{i-1}) \in [\ell+1]^n$ that represents the number of cell programs so far. Since the decoder is informed, on the i-th write it knows the state of the memory before and after the i-th write, that is, $\boldsymbol{c}_{i-1}$ and $\boldsymbol{c}_i$.

Next we define for all t and ℓ the region $\mathcal{C}_{t,\ell}$ and in Theorem 2, we prove that this is the capacity region of the EIA:DI model. That is, we prove $\mathcal{C}_{t,\ell}^{EIA:DI} = \mathcal{C}_{t,\ell}$. 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 $p_{j,\ell} = 0$ for $1 \leqslant j \leqslant t$. We 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 $p_{j,i}$ and $p_{j,i-1}$ as follows.

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

where for $j = 0$ we set $Q_{0,0} = 1$ otherwise $Q_{0,i} = 1$. The rates region $\mathcal{C}_{t,\ell}$ is defined as follows.

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

Note that for $\ell = 1$ it is possible to verify that we get the capacity region of WOM [5], [9], [13]. The next theorem proves that this holds also for $2 \leqslant \ell \leqslant t-1$.

Copyright (C) 2018 by IEICE

Theorem 2. *$\mathcal{C}_{t,\ell}$ is the capacity region of the ℓ-change t-write ELM in the EIA:DI model. That is, $\mathcal{C}_{t,\ell}^{EIA:DI} = \mathcal{C}_{t,\ell}$.*

Proof: On the j-th write, both the encoder and the decoder know the cell-state of the memory before writing the new data, $\boldsymbol{c}_{j-1}$, and 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, $\boldsymbol{c}_j + \boldsymbol{c}_{j-1}$.

Let X_j be a length-n binary vector, where $X_{j,k} = 1$ if and only if the k-th cell is tried 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 k-th cell was actually programmed on the j-th write, that is, $Y_j = \boldsymbol{c}_j + \boldsymbol{c}_{j-1}$. Additionally, on the j-th write, the encoder knows an additional vector $N_{j-1} = (x_1, \dots, x_n) \in [\ell+1]^n$, where x_k represents the number of times that the k-th cell was programmed after the first $j-1$ writes. Note that N_0 is the all-zero vector.

The rest of the proof consists of two parts. The first part, called the direct part, proves $\mathcal{C}_{t,\ell} \subseteq \mathcal{C}_{t,\ell}^{EIA:DI}$, and in the second, called the converse part, we prove $\mathcal{C}_{t,\ell}^{EIA:DI} \subseteq \mathcal{C}_{t,\ell}$. The direct part can be proved by the results from Section IV for the EIA:DU model, since by the definition of the models $C_{t,\ell}^{EIA:DU} \subseteq C_{t,\ell}^{EIA:DI}$, and by Theorem 10 we get $C_{t,\ell} \subseteq C_{t,\ell}^{EIA:DU}$. However, we can prove the direct part explicitly for the EIA:DI model by random coding technique. Due to lack of space, we omit this part.

For the converse part we need to prove that if there exists an $[n,t,\ell;M_1,\dots,M_t]^{EIA:DI}$ ELM code, then

$$\left(\frac{\log M_1}{n}, \frac{\log M_2}{n}, \dots, \frac{\log M_t}{n}\right) \in \mathcal{C}_{t,\ell}.$$

Let $S_1,\dots,S_t$ 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. The data processing inequality yields the following Markov chain:

$$S_j|N_{j-1} \text{ --- } X_j|N_{j-1} \text{ --- } Y_j|N_{j-1} \text{ --- } \hat{S}_j|N_{j-1}$$

and therefore, $I(X_j;Y_j|N_{j-1}) \geqslant I(S_j;\hat{S}_j|N_{j-1})$. Additionally, since we discuss the zero-error case we have that $S_j = \hat{S}_j$ and S_j is independent on N_{j-1}, then we have $I(S_j;\hat{S}_j|N_{j-1}) = H(S_j)$. Let L be an index random variable, which is uniformly distributed over the index set $[n]$. Since L is independent of all other random variables we get

$$\begin{aligned}
\frac{1}{n}I(X_j;Y_j|N_{j-1}) &\leqslant \frac{1}{n}H(Y_j|N_{j-1}) \overset{(a)}{\leqslant} \frac{1}{n}\textstyle\sum_{k=0}^{n-1} H(Y_{j,k}|N_{j-1,k})\\
&\overset{(b)}{=} H(Y_{j,L}|N_{j-1,L},L) \overset{(c)}{\leqslant} H(Y_{j,L}|N_{j-1,L})\\
&= \textstyle\sum_{i=0}^{\ell} Pr(N_{j-1,L}=i)H(Y_{j,L}|N_{j-1,L}=i)\\
&\overset{(d)}{=} \textstyle\sum_{i=0}^{\ell-1} Pr(N_{j-1,L}=i)H(Y_{j,L}|N_{j-1,L}=i),
\end{aligned}$$

where steps (a) and (c) follow from the fact that entropy of a vector is not greater than the sum of the entropies of its components, and conditioning does not increase the entropy. Step (b) follows from the fact that

$$\begin{aligned}
H(Y_{j,L}|N_{j-1,L},L) &= \textstyle\sum_{k=0}^{n-1} Pr(L=k)H(Y_{j,k}|N_{j-1,L},L=k)\\
&= \frac{1}{n}\textstyle\sum_{k=0}^{n-1} H(Y_{j,k}|N_{j-1,k}),
\end{aligned}$$

and step (d) follows from $H(Y_{j,L}|N_{j-1,L}=\ell) = 0$.

Now, we set $p_{j,i} = Pr(Y_{j,L}=1|N_{j-1,L}=i)$, and thus we can conclude that $Q_{j,i} = Pr(N_{j,L}=i)$ where $Q_{j,i}$ is calculated in Equation (1), and then

$$\begin{aligned}
\frac{\log(M_j)}{n} &\leqslant \frac{1}{n}I(X_j;Y_j|N_{j-1})\\
&\leqslant \textstyle\sum_{i=0}^{\ell-1} Pr(N_{j-1,L}=i)H(Y_{j,L}|N_{j-1,L}=i)\\
&= \textstyle\sum_{i=0}^{\ell-1} Q_{j-1,i}h\left(p_{j,i}\right),
\end{aligned}$$

and the converse part is implied. ■

We note that the capacity region of the ϵ-error case in this model is the same as for the zero-error case, where the converse part for the ϵ-error case can be proved by using Fano's inequality [3, pp. 38]. This result will be presented in details in the longer version of this paper.

Next, we seek to present the capacity region of the EIA:DI model in a recursive form. While we see this representation of the capacity region more intuitive, it will also help us in finding the maximum sum-rate in this model. For all $t \geqslant 1$ and $\ell \geqslant 1$, let $\widehat{\mathcal{C}}_{t,\ell}$ be the following region which is defined recursively.

$$\begin{aligned}
\widehat{\mathcal{C}}_{t,\ell} = \Big\{ &(R_1,\dots,R_t) | R_1 \leqslant h(p), p \in [0,1],\\
&\text{for } 2 \leqslant j \leqslant t, R_j \leqslant p \cdot R'_j + (1-p)\cdot R''_j,\\
&(R'_2,\dots,R'_t) \in \widehat{\mathcal{C}}_{t-1,\ell-1} \text{ and } (R''_2,\dots,R''_t) \in \widehat{\mathcal{C}}_{t-1,\ell} \Big\},
\end{aligned} \tag{3}$$

where $\widehat{\mathcal{C}}_{t,0} = \emptyset$, and for all $\ell \geqslant t$ we set $\widehat{\mathcal{C}}_{t,\ell} = \widehat{\mathcal{C}}_{t,t} = [0,1]^t$.

Theorem 3. *For all t and ℓ, $\widehat{\mathcal{C}}_{t,\ell} = \mathcal{C}_{t,\ell}$.*

Proof: For the first direction, we prove by induction on t that for all $\ell \leqslant t-1$, if $\mathbf{R} = (R_1,\dots,R_t) = \widehat{\mathcal{C}}_{t,\ell}$ then $\mathbf{R} \in \mathcal{C}_{t,\ell}^{EIA:DI}$. Since $\mathcal{C}_{t,\ell}^{EIA:DI} = \mathcal{C}_{t,\ell}$, we conclude that $\widehat{\mathcal{C}}_{t,\ell} \subseteq \mathcal{C}_{t,\ell}$.

The base of the induction, $t=1$, is readily verified. For the step, let $\mathbf{R} = (R_1,R_2,\dots,R_t) \in \widehat{\mathcal{C}}_{t,\ell}$ such that $R_1 = h(p)$ for $p \in [0,1]$ and for $2 \leqslant j \leqslant t$ $R_j = p_1 \cdot R'_j + (1-p_1)\cdot R''_j$ where $(R'_2,R'_3,\dots,R'_t) \in \widehat{\mathcal{C}}_{t-1,\ell-1}$ and $(R''_2,R''_3,\dots,R''_t) \in \widehat{\mathcal{C}}_{t-1,\ell}$. By the induction hypothesis, $(R'_2,R'_3,\dots,R'_t) \in \mathcal{C}_{t-1,\ell-1}^{EIA:DI}$ and $(R''_2,R''_3,\dots,R''_t) \in \mathcal{C}_{t-1,\ell}^{EIA:DI}$. Thus, we have two codes: C_1 - an $(\ell-1)$-change $(t-1)$-write ELM code which achieves the rates tuple $(R'_2,R'_3,\dots,R'_t)$ and C_2 - an ℓ-change $(t-1)$-write ELM code which achieves the rates tuple $(R''_2,R''_3,\dots,R''_t)$. Then, we can design an ℓ-change t-write ELM code, such that on the first write the encoder program a cell with probability p for $p \in [0,1]$, and then on the next writes it applies C_1 for the cells that were programmed on the first write, and C_2 for the other cells. Thus, the rate tuple $\mathbf{R}$ is achieved.

The second direction, $\mathcal{C}_{t,\ell} \subseteq \widehat{\mathcal{C}}_{t,\ell}$, is proved by induction on t, that is, for each $t \geqslant 1$ we prove that $\mathcal{C}_{t,\ell} \subseteq \widehat{\mathcal{C}}_{t,\ell}$ for all $1 \leqslant \ell \leqslant t$. The base of the induction, $t=1$ and $\ell=1$, is trivial. The induction assumption is that for each $1 \leqslant \ell' \leqslant t-1$, $\mathcal{C}_{t-1,\ell'} \subseteq \widehat{\mathcal{C}}_{t-1,\ell'}$. For the step, let $\mathbf{R} = (R_1,R_2,\dots,R_t) \in \mathcal{C}_{t,\ell}$ which is achieved by the probabilities $p_{j,i}$. Denote by $\mathbf{R}' = (R'_2,R'_3,\dots,R'_t) \in \mathcal{C}_{t-1,\ell-1}$ the rates tuple which is attained by the probabilities $p'_{j,i} = p_{j+1,i+1}$, and by $\mathbf{R}'' = (R''_2,R''_3,\dots,R''_t) \in \mathcal{C}_{t-1,\ell}$ the rates tuple which is attained by the probabilities $p''_{j,i} = p_{j+1,i}$. Recall that we define $\widehat{\mathcal{C}}_{t-1,t} = \widehat{\mathcal{C}}_{t-1,t-1}$, and $\widehat{\mathcal{C}}_{t-1,0} = \emptyset$. It can be easily verified that for all j, $2 \leqslant j \leqslant t$, $R_j = p_{1,1}R'_j + (1-p_{1,1})R''_j$. By the induction hypothesis, $\mathbf{R}' \in \widehat{\mathcal{C}}_{t-1,\ell-1}$ and $\mathbf{R}'' \in \widehat{\mathcal{C}}_{t-1,\ell}$. Thus, by defining $p = p_{1,1}$ we get a recursive form for $\mathbf{R}$, and we can conclude $\mathcal{C}_{t,\ell} \subseteq \widehat{\mathcal{C}}_{t,\ell}$. ■

Using the result from Theorem 3, it is possible to find the maximum sum-rate of this model, $\mathcal{R}_{t,\ell}^{EIA:DI}$. Note that since every cell can be programmed at most ℓ times, it is possible to show that $\mathcal{R}_{t,\ell}^{EIA:DI} \leqslant \log\sum_{i=0}^{\ell}\binom{t}{i}$. The next theorem assures that this upperbound is indeed tight.

Theorem 4. *For all t and ℓ, $\mathcal{R}_{t,\ell}^{EIA:DI} = \log\sum_{i=0}^{\ell}\binom{t}{i}$, and this value is achieved for $p = p_{1,1} = \frac{\sum_{i=0}^{\ell-1}\binom{t-1}{i}}{\sum_{i=0}^{\ell}\binom{t}{i}}$, where p and $p_{1,1}$ are defined in $\widehat{\mathcal{C}}_{t,\ell}$ and $\mathcal{C}_{t,\ell}$, respectively. For example, if $\ell = 2$ the maximum sum-rate is achieved for $p = p_{1,1} = \frac{2t}{t^2+t+2}$.*

Copyright (C) 2018 by IEICE

IV. The EIA:DU Model

In this section, we study the EIA:DU model, that is, encoder informed all and decoder uninformed. Our main contribution is a construction of a capacity-achieving ℓ-change t-write EIA:DU-ELM code, which assures that the capacity regions of the EIA:DI and EIA:DU models are the same, that is, $\mathcal{C}_{t,\ell}^{EIA:DIA} = \mathcal{C}_{t,\ell}^{EIA:DI}$.

Let us start with the first non-trivial case when $t = 3$ and $\ell = 2$. From Section III, we know that the capacity region of the two-change three-write EIA:DI model is given by

$$\mathcal{C}_{3,2}^{EIA:DI} = \mathcal{C}_{3,2} = \Big\{ (R_1, R_2, R_3) | R_1 \leqslant h(p_{1,0}), \\ R_2 \leqslant (1-p_{1,0}) \cdot h(p_{2,0}) + p_{1,0} \cdot h(p_{2,1}), \\ R_3 \leqslant (1-p_{1,0} \cdot p_{2,1}), p_{1,0}, p_{2,0}, p_{2,1} \in [0,1] \Big\}.$$

Since $\mathcal{C}_{3,2}^{EIA:DI}$ is an upper bound on the achievable rate region of $\mathcal{C}_{3,2}^{EIA:DU}$, we have that $\mathcal{C}_{3,2}^{EIA:DU} \subseteq \mathcal{C}_{3,2}^{EIA:DI}$. The following theorem states that any point in the above capacity region is also achievable, thereby we get that $\mathcal{C}_{3,2}^{EIA:DU} = \mathcal{C}_{3,2}^{EIA:DI}$.

Theorem 5. *For any $\epsilon > 0$ and $p_{1,0}, p_{2,0}, p_{2,1} \in [0,1]$, there exists an explicit construction of a two-change three-write EIA:DU-ELM code satisfying $R_1 \geqslant h(p_{1,0}) - \epsilon$, $R_2 \geqslant (1-p_{1,0}) \cdot h(p_{2,0}) + p_{1,0} \cdot h(p_{2,1}) - \epsilon$ and $R_3 \geqslant (1-p_{1,0} \cdot p_{2,1}) - \epsilon$.*

Before presenting our construction for two-change three-write EIA:DU-ELM codes, we introduce the following family of WOM codes. We then use these WOM codes as component codes in our construction of EIA:DU-ELM codes.

Definition 6. *An $[n, 2; M_1, M_2]_q^{EI:DU}$ two-write q-ary EI:DU-WOM code is a coding scheme comprising of n q-ary bits. It consists of two pairs of encoding and decoding maps $(\mathcal{E}_{q,1}, \mathcal{D}_{q,1})$ and $(\mathcal{E}_{q,2}, \mathcal{D}_{q,2})$ which are defined as follows:*

(1) $\mathcal{E}_{q,1} : [M_1] \mapsto [q]^n$ and $\mathcal{D}_{q,1} : Im(\mathcal{E}_{q,1}) \mapsto [M_1]$ such that for all $m_1 \in [M_1]$, $\mathcal{D}_{q,1}(\mathcal{E}_{q,1}(m_1)) = m_1$.

(2) $\mathcal{E}_{q,2} : [M_2] \times Im(\mathcal{E}_{q,1}) \mapsto [q]^n$ and $\mathcal{D}_{q,2} : Im(\mathcal{E}_{q,2}) \mapsto [M_2]$ such that for all $(m_2, \boldsymbol{c}) \in [M_2] \times Im(\mathcal{E}_{q,1})$, $\mathcal{E}_{q,2}(m_2, \boldsymbol{c}) \geqslant \boldsymbol{c}$ and $\mathcal{D}_{q,2}(\mathcal{E}_{q,2}(m_2, \boldsymbol{c})) = m_2$.

We say that $\boldsymbol{p} = (p_0, p_1, \ldots, p_{m-1})$ is a *probability vector* if $\sum_{i=0}^{m-1} p_i = 1$ and $p_i \geqslant 0$ for all $i \in [m]$. For two positive integers n, q and a probability vector $\boldsymbol{p} = (p_0, p_1, \ldots, p_{q-1})$, we denote by $\mathcal{B}(n, \boldsymbol{p})$ the set of all length-n q-ary vectors of constant composition $\boldsymbol{w} = (w_0, \ldots, w_{q-1})$, where $w_i = p_i \cdot n$ for $i \in [q]$ [1]. Let $p_{j,i \to k}$ be the probability that on the j-th write, a cell in state i is programmed to state $k \geqslant i$.

A family of two-write q-ary capacity-achieving EI:DU-WOM codes was constructed recently by Shpilka [12]. Particularly, given $\epsilon > 0$ and probability vectors $\boldsymbol{p}_{1,0}, \boldsymbol{p}_{2,0}, \ldots, \boldsymbol{p}_{2,q-2}$, Shpilka [12] constructed a family of capacity achieving two-write q-ary EI:DU-WOM codes that match these probability vectors on the first and second writes. We state this result formally.

Lemma 7. *[12] For all $(j, i) \in \{(1,0), (2,0), (2,1), \ldots, (2, q-2)\}$, let $\boldsymbol{p}_{j,i} = (p_{j,i\to i}, p_{j,i \to i+1}, \ldots, p_{j,i\to q-1})$ be a probability vector. Then, there exists an $[n, 2; M_1, M_2]_q^{EI:DU}$ two-write q-ary EI:DU-WOM code satisfying:*

- *$Im(\mathcal{E}_{q,1}) \subseteq \mathcal{B}(n, \boldsymbol{p}_{1,0})$ and the rate $R_1 = \frac{\log M_1}{n} \geqslant h(\boldsymbol{p}_{1,0}) - \epsilon$.*
- *For all $\boldsymbol{c}_1 \in Im(\mathcal{E}_{q,1})$, $m_2 \in [M_2]$, and $\boldsymbol{c}_2 = \mathcal{E}_{q,2}(m_2, \boldsymbol{c}_1)$, the following condition holds. For $i \in [q]$, let $\boldsymbol{c}_2^i$ be a length-$w_{1,i}$, $w_{1,i} = n \cdot p_{1,0\to i}$, substring of $\boldsymbol{c}_2$ at all locations k such that $\boldsymbol{c}_1[k] = i$. Then, $\boldsymbol{c}_2^i \in \mathcal{B}(w_{1,i}, \boldsymbol{p}_{2,i})$. Furthermore, the rate $R_2 \geqslant \sum_{i=0}^{q-2} p_{1,0\to i} \cdot h(\boldsymbol{p}_{2,i}) - \epsilon$.*

We refer to the family of WOM codes from Lemma 7 as $[n, 2; M_1, M_2]_q^{EI:DU}(\boldsymbol{p}_{1,0}, \boldsymbol{p}_{2,0}, \ldots, \boldsymbol{p}_{2,q-2})$. For the case $q = 2$, for shorthand, given $p_{1,0\to 1} = p$ we denote these codes by $[n, 2; M_1, M_2]^{EI:DU}(p)$ (where $p_{2,0\to 1} = 1/2$). Furthermore, using cooling codes, Chee et al. [2] provided the following family of binary WOM codes.

Lemma 8. *[2] For all $p \in [0, 1/2]$, there exists a two-write binary WOM code $[n, 2; M_1, M_2]^{EI:DU}(p)$ such that $M_1 = \sum_{i=0}^{\tau} \binom{n}{i}$ and $M_2 = 2^{n-\tau-1}$, where $\tau + 1 = p \cdot n$. Therefore, for any $\epsilon > 0$, there exists n such that $R_1 = \frac{\log M_1}{n} \geqslant h(p) - \epsilon$ and $R_2 = \frac{\log M_2}{n} \geqslant 1 - p - \epsilon$.*

We are now ready to present a construction of two-change three-write EIA:DU-ELM codes which establishes the result in Theorem 5.

Construction 9. *Given $p_{1,0}, p_{2,0}, p_{2,1} \in [0,1]$. We use the following two WOM codes:*

1) *Let $\boldsymbol{p}_{1,0} = (p_{1,0\to 0}, p_{1,0\to 1}, p_{1,0\to 2}) = (1-p_{1,0}, p_{1,0}, 0)$, $\boldsymbol{p}_{2,0} = (p_{2,0\to 0}, p_{2,0\to 1}, p_{2,0\to 2}) = (0, p_{2,0}, 1-p_{2,0})$, and $\boldsymbol{p}_{2,1} = (p_{2,1\to 1}, p_{2,1\to 2}) = (1-p_{2,1}, p_{2,1})$. Let C_1 be an $[n, 2; M_1, M_2]_3^{EI:DU}(\boldsymbol{p}_{1,0}, \boldsymbol{p}_{2,0}, \boldsymbol{p}_{2,1})$ two-write ternary EI:DU-WOM code from Lemma 7 with two pairs of encoder/decoder $(\mathcal{E}_{3,1}, \mathcal{D}_{3,1})$ and $(\mathcal{E}_{3,2}, \mathcal{D}_{3,2})$.*
2) *Let $\rho_1 = p_{1,0} \cdot p_{2,1}$, and C_2 be an $[n, 2; M_1', M_3]^{EI:DU}(\rho_1)$ two-write binary EI:DU-WOM code from Lemma 8 with two pairs of encoder/decoder $(\mathcal{E}_{2,1}, \mathcal{D}_{2,1})$ and $(\mathcal{E}_{2,2}, \mathcal{D}_{2,2})$.*

We construct an $[n, 3, 2; M_1, M_2, M_3]^{EIA:DU}$ two-change three-write ELM code where its three pairs of encoder/decoder mappings $(\mathcal{E}_j^{EIA:DU}, \mathcal{D}_j^{EIA:DU})$ for $j = 1, 2, 3$ are defined as follows.

***First write:** $\mathcal{E}_1^{EIA:DU}(m_1) = \mathcal{E}_{3,1}(m_1)$ for all $m_1 \in [M_1]$. Similarly, $\mathcal{D}_1^{EIA:DU}(\mathcal{E}_1^{EIA:DU}(m_1)) = \mathcal{D}_{3,1}(\mathcal{E}_{3,1}(m_1) = m_1$. Note that since we chose the probability to program level 2 in the first write of C_1 to be zero, the output of the encoder $\mathcal{E}_{3,1}$ is indeed a binary vector, so $\mathcal{E}_1^{EIA:DU}$ is well defined.*

***Second write:** The idea is to use the second write encoder $\mathcal{E}_{3,2}$ of C_1 with the probability vectors $\boldsymbol{p}_{2,0}$ and $\boldsymbol{p}_{2,1}$, and notice that here we write all cells to levels 1 or 2. Then, we can view this "ternary word" as a binary word. Let $\boldsymbol{c}_1 = (c_{1,1}, \ldots, c_{1,n}) \in Im(\mathcal{E}_1^{EIA:DU})$ be the cell-state vector after the first write, and note that this is a binary vector. The encoder/decoder $(\mathcal{E}_2^{EIA:DU}, \mathcal{D}_2^{EIA:DU})$ are defined formally as follows. For all $(m_2, \boldsymbol{c}_1) \in [M_2] \times Im(\mathcal{E}_1^{EIA:DU})$,*

$$\boldsymbol{c}_2 = \mathcal{E}_2^{EIA:DU}(m_2, \boldsymbol{c}_1) = \boldsymbol{c}_2'(\text{mod} 2),$$

where $\boldsymbol{c}_2' = \mathcal{E}_{3,2}(m_2, \boldsymbol{c}_1) \in [3]^n$. Furthermore, for all $\boldsymbol{c}_2 \in Im(\mathcal{E}_2^{EIA:DU})$,

$$\mathcal{D}_2^{EIA:DU}(\boldsymbol{c}_2) = \mathcal{D}_{3,2}(\boldsymbol{c}_2') = m_2,$$

where $\boldsymbol{c}_2' = 2 \cdot \boldsymbol{1} - \boldsymbol{c}_2$, that is, $c_{2,i}' = 1$ if $c_{2,i} = 1$ and $c_{2,i}' = 2$ if $c_{2,i} = 0$.

***Third write:** Let $\boldsymbol{c}_2$ be the cell-state vector after the second write. We note that the encoder on the third write knows the vector $N(\boldsymbol{c}_2) \in [3]^n$ but the decoder does not have this information. Among the n cells, there are $\rho_1 \cdot n$ cells which have been programmed twice, where $\rho_1 = p_{1,0} \cdot p_{2,1}$, and therefore (only) these cells cannot be programmed on this write. Hence, the encoder can interpret the vector $N(\boldsymbol{c}_2)$ as a length-n binary vector indicating for each cell whehter it can be programmed on this write. We denote this vector by $\boldsymbol{c}_2'$, so $\boldsymbol{c}_2'[i] = 1$ if and only if $N(\boldsymbol{c}_2)[i] = 2$. We will use the code C_2 to encode and decode*

[1] We assume here that p_i is a rational number and n is large enough such that $p_i \cdot n$ is an integer for $i \in [q]$

Copyright (C) 2018 by IEICE

on this write. Specifically, the encoder/decoder mappings are defined as follows: for all $m_3 \in [M_3]$ *and* $\boldsymbol{c}_2 \in Im(\mathcal{E}_2^{EIA:DU})$,

$$\mathcal{E}_3^{EIA:DU}(m_3, N(\boldsymbol{c}_2)) = \overline{\mathcal{E}_{2,2}(m_3, \boldsymbol{c}_2')}.$$

Furthermore, for all $\boldsymbol{c}_3 \in Im(\mathcal{E}_3^{EIA:DU})$,

$$\mathcal{D}_3^{EIA:DU}(\boldsymbol{c}_3) = \mathcal{D}_{2,2}(\overline{\boldsymbol{c}_3}).$$

We are now ready to present the proof of Theorem 5.

Proof of Theorem 5: For any $\epsilon > 0$ and $p_{1,0}, p_{2,0}, p_{2,1} \in [0,1]$, we choose the codes C_1 and C_2 in Construction 9 to satisfy

$$R_1(C_1) \geqslant h(\boldsymbol{p}_{1,0}) - \epsilon = h(p_{1,0}) - \epsilon,$$

since $\boldsymbol{p}_{1,0} = (1-p_{1,0}, p_{1,0}, 0)$,

$$R_2(C_1) \geqslant p_{1,0\to 0} \cdot h(\boldsymbol{p}_{2,0}) + p_{1,0\to 1} \cdot h(\boldsymbol{p}_{2,1}) - \epsilon$$
$$\geqslant (1-p_{1,0}) \cdot h(p_{2,0}) + p_{1,0} \cdot h(p_{2,1}) - \epsilon,$$

and

$$R_2(C_2) \geqslant 1 - \rho_1 - \epsilon = 1 - p_{1,0} \cdot p_{2,1} - \epsilon.$$

The result follows from the fact that the rates-tuple of the two-change three-write ELM code is $(R_1(C_1), R_2(C_1), R_2(C_2))$. ■

The solution for the case $t=3, \ell=2$ can be generalized for any t and ℓ, so we get the following result. We skip the details due to the lack of space.

Theorem 10. *For all* t *and* ℓ, $\mathcal{C}_{t,\ell} \subseteq C_{t,\ell}^{EIA:DU}$, *that is, for any* $\epsilon > 0$ *and a rates* t*-tuple* $(R_1, \dots, R_t) \in C_{t,\ell}^{EIA:DI}$, *there exists an* ℓ*-change* t*-write ELM code* C *such that its rate on the* j*-th write satisfies* $R_j(C) \geqslant R_j - \epsilon$ *for all* $1 \leqslant j \leqslant t$.

By Theorems 2 and 10, and by the definition of the models we can conclude the EIA models.

Corollary 11. *For all* t *and* ℓ, $C_{t,\ell}^{EIA:DU} = C_{t,\ell}^{EIA:DI} = \mathcal{C}_{t,\ell}$ *and* $\mathcal{R}_{t,\ell}^{EIA:DU} = \mathcal{R}_{t,\ell}^{EIA:DI} = \log \sum_{i=0}^{\ell} \binom{t}{i}$.

We note that the capacity region for the ϵ-error in these models is equal to the zero-error capacity. The details about the ϵ-error case will be presented in a full version of this paper.

V. The EIP:DU Model

In this section we study the EIP:DU model and in particular focus on its sum-rate. First, we note that since $\mathcal{C}_{t,\ell}^{EIP:DU} \subseteq \mathcal{C}_{t,\ell}^{EIA:DU}$ we also have that $\mathcal{R}_{t,\ell}^{EIP:DU} \leqslant \mathcal{R}_{t,\ell}^{EIA:DU} = \log \sum_{i=0}^{\ell} \binom{t}{i}$. Since we conjecture that at this case there is no equality between these two cases, our goal in this section is to provide constructions with the highest sum-rate we can get. We first present a general construction which is based upon WOM codes that already achieves high sum-rates and we then show how to improve it for the case $t=3, \ell=2$.

The next construction provides a family of ℓ-change t-write EIP:DU-ELM codes.

Construction 12. *Let* $(k_1, \dots, k_\ell)$ *be such that* $1 \leqslant k_i \leqslant t$ *for* $1 \leqslant i \leqslant \ell$ *and* $\sum_{i=1}^{\ell} k_i = t$. *Let* C_i *be a binary* k_i*-write EI:DU-WOM code for* $1 \leqslant i \leqslant \ell$ *with sum-rate* $\mathcal{R}_i$. *An* $[n, t, \ell; M_1, \dots, M_t]^{EIP:DU}$ ℓ*-change* t*-write ELM code with sum-rate* $\mathcal{R} = \sum_{i=1}^{\ell} \mathcal{R}_i$ *is constructed as follows.*

- *On the first* k_1 *writes, we use the* k_1*-write WOM code* C_1.
- *On the following* k_2 *writes, we use the* k_2*-write WOM code* C_2, *by writing the complement of the cell-state vectors on each write.*
- *We continue this process iteratively for the following* $\ell - 2$ *WOM codes.*

The maximum sum-rate of the ELM codes from Construction 12 is $\mathcal{R}_{max} = \sum_{i=1}^{\ell} \log(k_i + 1)$ and it will be achieved when each of the codes $C_i, 1 \leqslant i \leqslant \ell$ will be ϵ-close to its maximum sum-rate $\log(k_i+1)$. Hence, in order to maximize the sum-rate, our goal is to maximize the value of $\sum_{i=1}^{\ell} \log(k_i+1)$ given that $\sum_{i=1}^{\ell} k_i = t$. Assume that $t = k \cdot \ell + r$, $r \in [\ell]$, then this maximum value will be achieved for $k_1 = \dots = k_r = k+1, k_{r+1} = \dots = k_\ell = k$.

Corollary 13. *For all* t *and* ℓ, *where* $t = k \cdot \ell + r$, $r \in [\ell]$,

$$\mathcal{R}_{t,\ell}^{EIP:DU} \geqslant r \cdot \log(k+2) + (\ell - r) \cdot \log(k+1)$$
$$= \ell \log\left(\left\lfloor \frac{t}{\ell} \right\rfloor + 1\right) + (t \bmod \ell) \log\left(1 + \frac{1}{\lfloor \frac{t}{\ell} \rfloor + 1}\right).$$

Recall that the upper bound of the maximum sum-rate $\mathcal{R}^{EIP:DU} \leqslant \mathcal{R}_{t,\ell}^{EIA:DU} = \log \sum_{i=0}^{\ell} \binom{t}{i}$. The following result shows for $\ell = 2$ the sum-rate of the ELM code constructed in Construction 12 is already close to the upper bound.

Proposition 14. *For* $\ell = 2$ *and* $t \geqslant 3$, $\mathcal{R}_{t,2}^{EIP:DU} \geqslant \mathcal{R}_{t,2}^{EIA:DU} - 1$.

For example, for $t=3$ and $\ell=2$, we get that the maximum achievable sum-rate of the codes in Construction 12 is $\log 6 \approx 2.585$, while the upper bound is $\log 7 \approx 2.807$. Lastly, we report here on another construction we have for this case which provides a family of two-change three-write EIP:DU-ELM codes which achieve the sum-rate of roughly 2.64. However, these codes work for the ϵ-error case and not for the zero-error case, which we studied in the paper.

VI. Acknowledgement

The research of Y. M. Chee is partially funded by the Singapore Ministry of Education under grants MOE2017-T3-1-007 and MOE2015-T2-2-086. The research of Alexander Vardy was supported in part by the National Science Foundation under Grants CCF-1405119 and CCF-1719139. The research of E. Yaakobi is partially supported by the German Science Foundation (GIF) Grant I-1356-407.6/2016.

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, H. M. Kiah, A. Vardy, and E. Yaakobi, "Explicit constructions of finite-length WOM codes", *Proc. IEEE Int. Symp. on Inform. Theory,* Aachen, Germany, pp. 2870–2874, Jun. 2017.

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

[4] 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.

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

[6] 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.

[7] Y. Kim et al.,"Locally rewritable codes for resistive memories," *IEEE J. Selected Areas in Comm.*, vol. 34, no. 9, pp. 2470–2485, Sep. 2016.

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

[9] 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.

[10] 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.

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

[12] A. Shpilka, "Capacity-achieving multiwrite WOM codes", *IEEE Trans. Inform. Theory*, Vol. 60, No. 3, pp.1481–1487, Mar. 2014.

[13] 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.

[14] 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.

[15] L. Zhang et al., "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.

Copyright (C) 2018 by IEICE