Maximizing the Storage Capacity of Nonvolatile Memories

Anxiao (Andrew) Jiang
Computer Science and Eng. Dept.
Texas A&M University
College Station, TX 77843
ajiang@cse.tamu.edu

Hongchao Zhou
Electrical Engineering Dept.
California Institute of Technology
Pasadena, CA 91125
hzhou@caltech.edu

Jehoshua Bruck
Electrical Engineering Dept.
California Institute of Technology
Pasadena, CA 91125
bruck@caltech.edu

Abstract—We propose variable-level cell, a new data representation scheme, for nonvolatile memories (including flash memories, phase-change memories, etc.). We derive its storage capacity, and analyze its performance on rewriting data.

I. Introduction

For nonvolatile memories (NVMs) – including flash memories, phase-change memories (PCMs), memristors, *etc.*, – maximizing the storage capacity is a key challenge. The existing method is to use multi-level cells (MLCs) of more and more levels, where a cell of q discrete levels can store $\log_2 q$ bits [1]. Flash memories with 4 and 8 levels have been used in products, and MLCs with 16 levels have been demonstrated in prototypes. For PCMs, cells with 4 or more levels have been in development. How to maximize the number of levels in cells is a most important topic for study.

The number of levels that can be programmed into cells is seriously constrained by the noise in cell programming and by cell heterogeneity [1]. We explain it with flash memories as an example, and the concepts can be naturally extended to PCMs and memristors. A flash memory uses the charge stored in floating-gate cells to store data, where the amount of charge in a cell is quantized into q values to represent q discrete levels. Cell programming – the operation of injecting charge into cells – is a noisy process, which means that the actual increase in the cell levels can deviate substantially from the target value. And due to the block erasure property, – which means that to remove charge from any cell, a whole block of about 10^5 cells must be erased together to remove all their charge, – during the writing procedure, the cell levels are only allowed to monotonically increase using charge injection. That makes it infeasible to correct over-injection errors [1]. Beside cell-programming noise, the difficulty in programming is also caused by cell heterogeneity, which means that even when the same voltage is used to program different cells, the increments in the different cells' levels can differ substantially, due to the heterogeneity in cell material and geometry [6]. Since memories use parallel programming for high write speed, a common voltage is used to program many cells during a programming step, which cannot be adjusted for individual cells [1], [6]. As cell sizes scale down, the cell heterogeneity will be even more significant [1].

The storage capacity of MLC is limited by the worst-case performance of cell-programming noise and cell heterogeneity [1], [6]. We illustrate it in Fig. 1 (a). A *safety gap* is needed to separate two adjacent levels to prevent errors after programming. The charge level for an individual cell has a random distribution due to the cell-programming noise [1], [6]. The actual value of the charge level varies from one write to another. Due to cell heterogeneity, the charge-level distributions of different cells in the same level shift away from each other, which widens the overall charge-level distribution of the level [1], [6]. Since MLC uses fixed levels for storage, it needs to accommodate the worst-case programming performance: the charge-level range for a level is set to be sufficiently wide to accommodate not only the worst-case programming noise for each cell, but also the worst-case cell heterogeneity. That limits the number of levels in MLC.

In this paper, we introduce a new storage scheme named *variable-level cells (VLC)* for maximum storage capacity. It has two unique properties: the number of levels is not fixed, and the positions of the levels are chosen adaptively during programming. More specifically, we program the levels sequentially from low to high. After level i is programmed, we program level $i+1$ such that the gap between the two adjacent levels is at least the required safety gap. (There are many ways to differentiate the cells in different levels. For example, we can require the cells of the same level to have charge levels within δ from each other, and require cells in different levels to have charge levels at least Δ away from each other, for appropriately chosen parameters δ, Δ.) We program as many levels into the cells as possible until the highest programmed level reaches the physical limit.

The VLC scheme places the levels as compactly as possible, and maximizes the number of programmed levels, which is determined by the *actual* instead of the *worst-case* programming performance. It is illustrated in Fig. 1 (b). Note that for a set of cells programmed in parallel, their heterogeneity is usually not as significant as the worst-case heterogeneity of all memory cells, which helps narrow the actual charge-level range for a level [1]. Furthermore, the actual cell-programming noise is often not as large as its worst-case value, which further narrows the actual range of charge levels for the level. The VLC scheme places level $i+1$ as low as possible based on the actual position of level i. The better the actual programming performance is, the more levels we write into the cells.

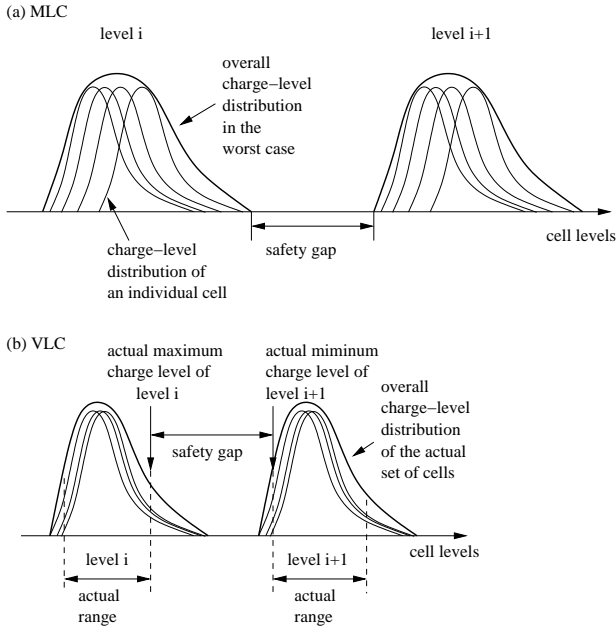The VLC scheme shifts data representation into the stochas-

Fig. 1. Charge-level distribution of (a) MLC; (b) VLC.

tic regime, because the number of levels actually used is not determined in advance. New coding schemes are needed for this new paradigm. In this paper, we derive the storage capacity of VLC. We also study rewriting codes, which are important for improving the longevity of flash memories and PCMs [4], [5], and present bounds for achievable rates.

The rest of the paper is organized as follows. In Section II, the storage capacity of VLC is derived. In Section III, the capacity region of data rewriting is studied. In Section IV, concluding remarks are presented.

II. DISCRETE MODEL AND CAPACITY OF VLC

In this section, we present a probabilistic model for VLC, and derive its capacity.

A. Discrete Model for VLC

For a storage scheme, it is key to have a discrete model that not only enables efficient code designs, but is also robust to the physical implementation of the scheme. In this paper, we use the following simple probabilistic model for VLC.

Let q denote the maximum number of levels we can program into cells, and call the q levels level 0, level 1, $\cdots$, level $q-1$. Let n denote the number of cells, and for $i = 1, 2, \cdots, n$, denote the level of the ith cell by $c_i \in \{0, 1, \cdots, q-1\}$. Before writing, all cells are at level 0. Let

$$L = (\ell_1, \ell_2, \cdots, \ell_n) \in \{0, 1, \cdots, q-1\}^n$$

denote the *target levels*, which means that for $i = 1, \cdots, n$, we plan to program c_i as ℓ_i. [1] To program cells to the target

[1]Since VLC uses the relative positions of charge levels to store data, we usually require for $i = 0, 1, \cdots, \max_{1 \le j \le n} \ell_j$, at least one cell is assigned to level i. However when $n \to \infty$, this constraint has a negligible effect on the code rate. So when we analyze capacity, this constraint can be neglected.

levels L, we first program level 1 (namely, push some cells from level 0 to level 1), then program level 2, level 3, $\cdots$, until we reach a certain level i such that its charge levels are so close to the physical limit that we will not be able to program level $i+1$. All the cells that should belong to levels $1, 2, \cdots, i$ are successfully programmed to those levels. The cells that should belong to levels $\{i+1, i+2, \cdots, \max_{1 \le j \le n} \ell_j\}$ are still in level 0 (together with the cells that should belong to level 0). So the final cell levels are

$$L_i \triangleq (c'_1, c'_2, \cdots, c'_n),$$

where for $j = 1, \cdots, n$, $c'_j = \ell_j$ if $1 \le \ell_j \le i$, and $c'_j = 0$ otherwise.

For $i = 1, 2, \cdots, q-1$, let p_i denote the probability that level i can be programmed *given* that levels $1, 2, \cdots, i-1$ are successfully programmed. (And for convenience, define $p_q = 0$.) Let T denote the target levels, and S denote the *written* levels. So when $T = L \in \{0, 1, \cdots, q-1\}^n$, for $i = 0, 1, \cdots, q-1$, we have

$$Pr\{S = L_i\} = (1 - p_{i+1}) \prod_{j=1}^{i} p_j.$$

We define the *capacity of VLC* by

$$C = \lim_{n \to \infty} \frac{1}{n} \max_{P_T(t)} I(T; S),$$

where $P_T(t)$ is the probability distribution of T, and $I(T; S)$ is the mutual information of T and S. [2]

B. Capacity of VLC

We now derive the capacity of VLC. We first present a channel model for a single cell. Let X denote the target level for a cell, and let Y denote the actual state of the cell after writing. Clearly, $X \in \{0, 1, \cdots, q-1\}$. The level X can be successfully programmed with probability $p_1 p_2 \cdots p_X$ if $X \ge 1$, and with probability $p_1 p_2 \cdots p_{q-1}$ if $X = 0$; and if so, we get $Y = X$. It is also possible that level X is not successfully programmed. For $i = 0, 1, \cdots, q-2$, the highest programmed level will be level i with probability

$$(1 - p_{i+1}) \prod_{j=1}^{i} p_j;$$

and if so, the cells with target levels in

$$\{0, i+1, i+2, \cdots, q-1\}$$

will all remain in level 0. In that case, if $X = 0$ or $i+1 \le X \le q-1$, we denote that state of the cell after writing (namely, Y) by

$$E_{\{0, i+1, i+2, \cdots, q-1\}}$$

[2]Here we view the n cells as *one* symbol for the channel, and normalize its capacity by the number of cells. The capacity defined this way equals the *expected* number of bits a cell can store.

and call it a *partial erasure*, because it is infeasible to tell which level in $\{0, i+1, i+2, \cdots, q-1\}$ is the target level of the cell. So we have

$$Y \in \{0, 1, \cdots, q-1\} \cup \{E_{\{0,1,2,\cdots,q-1\}}, E_{\{0,2,3,\cdots,q-1\}}, \cdots, E_{\{0,q-1\}}\}.$$

We call the channel the *partial-erasure channel*. Examples of the channel for $q = 2, 3$ are shown in Fig. 2, where the states in rectangles are the partial erasures. (We can see that when $q = 2$, the channel is the same as the binary erasure channel (BEC) with erasure probability $1 - p_1$.)
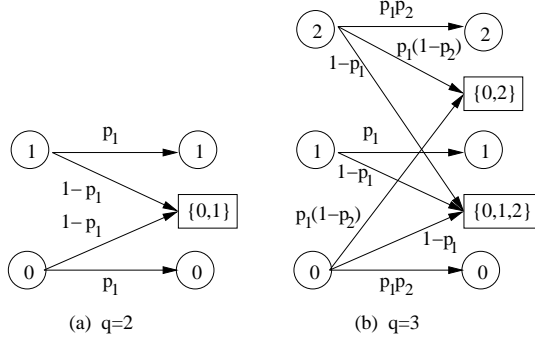

Fig. 2. Partial-erasure channel for q levels. (a) $q = 2$. (b) $q = 3$.

Let us define $A_1, A_2, \cdots, A_{q-1}$ recursively:

$$A_{q-1} = 2^{p_{q-1}};$$

and for $i = q - 2, q - 3, \cdots, 1$,

$$A_i = (1 + A_{i+1})^{p_i}.$$

Example 1. *Consider VLC constant-weight codes with $q = 5$. We have*

$$A_4 = 2^{p_4},$$
$$A_3 = (1 + 2^{p_4})^{p_3},$$
$$A_2 = \left(1 + (1 + 2^{p_4})^{p_3}\right)^{p_2},$$
$$A_1 = \left(1 + \left(1 + (1 + 2^{p_4})^{p_3}\right)^{p_2}\right)^{p_1}.$$

$\square$

Lemma 2. *Let $x \in [0, 1]$ and $y \in [0, 1]$ be given numbers. Let*

$$\mu^* = \frac{1}{1 + 2^{\frac{y}{x}}}.$$

Then

$$\max_{\mu \in [0,1]} xH(\mu) + y(1 - \mu)$$
$$= xH(\mu^*) + y(1 - \mu^*)$$
$$= \log_2\left(1 + 2^{\frac{y}{x}}\right)^x.$$

Proof: Define $f(\mu) \triangleq xH(\mu) + y(1 - \mu)$. Then

$$f(\mu) = y - \frac{1}{\ln 2}\left(x\mu \ln \mu + x(1 - \mu)\ln(1 - \mu) + y\mu \ln 2\right).$$

So

$$f'(\mu) = -\frac{1}{\ln 2}\left(x \ln \frac{\mu}{1 - \mu} + y \ln 2\right),$$

where $f'(\mu)$ is the derivative of $f(\mu)$. By setting $f'(\mu) = 0$, we get

$$\mu = \frac{1}{1 + 2^{\frac{y}{x}}} \triangleq \mu^*.$$

And we get

$$f(\mu^*) = \log_2\left(1 + 2^{\frac{y}{x}}\right)^x.$$

$\blacksquare$

Lemma 3. *The capacity of the partial-erasure channel for q levels is*

$$\log_2 A_1$$

bits per cell.

Proof: The capacity of the partial-erasure channel is $max_{P_X(x)} I(X; Y)$, where $P_X(x)$ is the probability distribution for X. For $i = 2, 3, \cdots, q$, we define Ch_i to be a partial-erasure channel with i levels and the following alternation of notations:

1) Its i levels – from low to high – are denoted by levels $0, q - i + 1, q - i + 2, \cdots, q - 1$ (instead of levels $0, 1, \cdots, i - 1$);
2) The probabilities that the highest programmed level will be level $q - i + 1$, level $q - i + 2$, $\cdots$, level $q - 1$ are $p_{q-i+1}, p_{q-i+1}p_{q-i+2}, \cdots, p_{q-i+1}p_{q-i+2}\cdots p_{q-1}$, respectively (instead of $p_1, p_1p_2, \cdots, p_1p_2\cdots p_{i-1}$).

Let $\bar{X}_i$ and $\bar{Y}_i$ denote the input and output symbols to the channel Ch_i, respectively. (Clearly, we have $X = \bar{X}_q$ and $Y = \bar{Y}_q$.) We now prove the following claim by induction:

- Claim ♣: For $i = 2, 3, \cdots, q$, we have

$$\max_{P_{\bar{X}_i}(x)} I(\bar{X}_i; \bar{Y}_i) = \log_2 A_{q-i+1}.$$

First, consider the base case $i = 2$. The channel Ch_2 is a binary erasure channel with erasure probability $1 - p_{q-1}$, and its capacity is p_{q-1}. We have $A_{q-1} = 2^{p_{q-1}}$, so $\log_2 A_{q-1} = p_{q-1}$. So claim ♣ holds for $i = 2$.

As the inductive step, consider $i \geq 3$. We have

$$\bar{X}_i \in \{0, q - i + 1, q - i + 2, \cdots, q - 1\}$$

and

$$\bar{Y}_i \in \{0, q - i + 1, q - i + 2, \cdots, q - 1\} \cup \{E_{\{0,q-i+1,\cdots,q-1\}}, E_{\{0,q-i+2,\cdots,q-1\}}, \cdots, E_{\{0,q-1\}}\}.$$

For convenience, in the following equation we use $P(x)$ to denote $P_{\bar{X}_i}(x)$, use $P(y)$ to denote $P_{\bar{Y}_i}(y)$, use $P(x, y)$ to denote the joint distribution $P_{\bar{X}_i, \bar{Y}_i}(x, y)$, and use $P(y|x)$ to denote the conditional distribution $P_{\bar{Y}_i|\bar{X}_i}(y|x)$. Define $\mu \triangleq P(x = q - i + 1)$. We have

$$I(\bar{X}_i; \bar{Y}_i)$$
$$= \sum_x \sum_y P(x, y) \log \frac{P(x,y)}{P(x)P(y)}$$
$$= \sum_x P(x) \sum_y P(y|x) \log \frac{P(y|x)}{P(y)}$$

$$
\begin{aligned}
=\ & P(x = q - i + 1)P(y = q - i + 1 | x = q - i + 1) \\
& \cdot \log \frac{P(y = q - i + 1 | x = q - i + 1)}{P(y = q - i + 1)} + \\
& P(x = q - i + 1)P(y = E_{\{0, q-i+1, \cdots, q-1\}} | x = q - i + 1) \\
& \cdot \log \frac{P(y = E_{\{0, q-i+1, \cdots, q-1\}} | x = q - i + 1)}{P(y = E_{\{0, q-i+1, \cdots, q-1\}})} \\
& + \sum_{x \in \{0, q-i+2, \cdots, q-1\}} P(x) \sum_y P(y|x) \log \frac{P(y|x)}{P(y)} \\
=\ & \mu p_{q-i+1} \log \frac{p_{q-i+1}}{\mu p_{q-i+1}} + \mu(1 - p_{q-i+1}) \log \frac{1 - p_{q-i+1}}{1 - p_{q-i+1}} + \\
& \sum_{x \in \{0, q-i+2, \cdots, q-1\}} P(x) \sum_y P(y|x) \log \frac{P(y|x)}{P(y)} \\
=\ & p_{q-i+1} \mu \log \frac{1}{\mu} + \\
& \sum_{x \in \{0, q-i+2, \cdots, q-1\}} P(x) \sum_y P(y|x) \log \frac{P(y|x)}{P(y)} \\
=\ & p_{q-i+1} \mu \log \frac{1}{\mu} + P(x \neq q - i + 1) \\
& \cdot \sum_{x \in \{0, q-i+2, \cdots, q-1\}} P(x | x \neq q - i + 1) \\
& \cdot [P(y = E_{\{0, q-i+1, \cdots, q-1\}} | x) \log \frac{P(y = E_{\{0, q-i+1, \cdots, q-1\}} | x)}{P(y = E_{\{0, q-i+1, \cdots, q-1\}})} \\
& + \sum_{y \neq E_{\{0, q-i+1, \cdots, q-1\}}} P(y|x) \log \frac{P(y|x)}{P(y)}] \\
=\ & p_{q-i+1} \mu \log \frac{1}{\mu} + (1 - \mu) \\
& \cdot \sum_{x \in \{0, q-i+2, \cdots, q-1\}} P(x | x \neq q - i + 1) \\
& \cdot \sum_{y \neq E_{\{0, q-i+1, \cdots, q-1\}}} P(y|x) \log \frac{P(y|x)}{P(y)} \\
=\ & p_{q-i+1} \mu \log \frac{1}{\mu} + p_{q-i+1}(1 - \mu) \\
& \cdot \sum_{x \in \{0, q-i+2, \cdots, q-1\}} P(x | x \neq q - i + 1) \\
& \cdot \sum_{y \neq E_{\{0, q-i+1, \cdots, q-1\}}} P(y|x, y \neq E_{\{0, q-i+1, \cdots, q-1\}}) \\
& \cdot \log \frac{p_{q-i+1} P(y|x, y \neq E_{\{0, q-i+1, \cdots, q-1\}})}{(1 - \mu) p_{q-i+1} P(y|x \neq q - i + 1, y \neq E_{\{0, q-i+1, \cdots, q-1\}})} \\
=\ & p_{q-i+1} \mu \log \frac{1}{\mu} + p_{q-i+1}(1 - \mu) \\
& \cdot \sum_{x \in \{0, q-i+2, \cdots, q-1\}} P(x | x \neq q - i + 1) \\
& \cdot \sum_{y \neq E_{\{0, q-i+1, \cdots, q-1\}}} P(y|x, y \neq E_{\{0, q-i+1, \cdots, q-1\}}) \\
& \cdot \log \frac{1}{1 - \mu} + p_{q-i+1}(1 - \mu) \\
& \cdot \sum_{x \in \{0, q-i+2, \cdots, q-1\}} P(x | x \neq q - i + 1) \\
& \cdot \sum_{y \neq E_{\{0, q-i+1, \cdots, q-1\}}} P(y|x, y \neq E_{\{0, q-i+1, \cdots, q-1\}}) \\
& \cdot \log \frac{P(y|x, y \neq E_{\{0, q-i+1, \cdots, q-1\}})}{P(y|x \neq q - i + 1, y \neq E_{\{0, q-i+1, \cdots, q-1\}})}
\end{aligned}
$$

By defining B as

$$
\begin{aligned}
B \triangleq\ & \sum_{x \in \{0, q-i+2, \cdots, q-1\}} P(x | x \neq q - i + 1) \\
& \cdot \sum_{y \neq E_{\{0, q-i+1, \cdots, q-1\}}} P(y|x, y \neq E_{\{0, q-i+1, \cdots, q-1\}}) \\
& \cdot \log \frac{P(y|x, y \neq E_{\{0, q-i+1, \cdots, q-1\}})}{P(y|x \neq q - i + 1, y \neq E_{\{0, q-i+1, \cdots, q-1\}})},
\end{aligned}
$$

we get

$$
\begin{aligned}
& I(\bar{X}_i; \bar{Y}_i) \\
=\ & p_{q-i+1} \mu \log \frac{1}{\mu} + p_{q-i+1}(1 - \mu) \log \frac{1}{1 - \mu} + \\
& p_{q-i+1}(1 - \mu)B \\
=\ & p_{q-i+1} (H(\mu) + (1 - \mu)B)
\end{aligned}
$$

We see that B is actually the mutual information between the input and output symbols of the channel Ch_{i-1}, namely $B = I(\bar{X}_{i-1}; \bar{Y}_{i-1})$. By the induction assumption, the maximum value of B is $\log_2 A_{q-i+2}$. So

$$
\begin{aligned}
& \max_{P_{\bar{X}_i}(x)} I(\bar{X}_i; \bar{Y}_i) \\
=\ & \max_{\mu \in [0,1]} p_{q-i+1} (H(\mu) + (1 - \mu) \log_2 A_{q-i+2}).
\end{aligned}
$$

By Lemma 2,

$$
\max_{\mu \in [0,1]} H(\mu) + (1 - \mu) \log_2 A_{q-i+2} = \log_2 \left(1 + A_{q-i+2} \right).
$$

So

$$
\max_{P_{\bar{X}_i}(x)} I(\bar{X}_i; \bar{Y}_i) = \log_2 \left(1 + A_{q-i+2} \right)^{p_{q-i+1}} = \log_2 A_{q-i+1}.
$$

So claim ♣ is proved. Since $X = \bar{X}_q$ and $Y = \bar{Y}_q$, we have

$$
\max_{P_X(x)} I(X; Y) = \log_2 A_{q-q+1} = \log_2 A_1.
$$

That completes the proof. ∎

Theorem 4. *The capacity of VLC is*

$$
C = \log_2 A_1.
$$

Proof: Let $T = (x_1, \cdots, x_n) \in \{0, 1, \cdots, q - 1\}^n$ denote the *target levels* of the n cells, and $S = (y_1, \cdots, y_n) \in \{0, 1, \cdots, q - 1, E_{\{0,1,\cdots,q-1\}}, E_{\{0,2,\cdots,q-1\}}, \cdots, E_{\{0,q-1\}}\}^n$ denote the *written levels* of the n cells. Note that the requirement for every level to have at least one cell has a negligible effect on the capacity, because we can satisfy the requirement by assigning q auxiliary cells $a_0, a_1, \cdots, a_{q-1}$ to the q levels, where for $i = 0, 1, \cdots, q - 1$, we let auxiliary cell a_i's target level be level i. As $n \to \infty$, the q auxiliary cells do not affect the code's rate. So in the following, we can assume that the set of values that T can take are exactly the set $\{0, 1, \cdots, q - 1\}^n$. Namely, every cell's target level can be freely chosen from the set $\{0, 1, \cdots, q - 1\}$. We also assume the q auxiliary cells exist without loss of generality (w.l.o.g.).

Let $h \in \{0, 1, \cdots, q - 1\}$ denote the highest programmed level. $Pr\{h = 0\} = 1 - p_1$, and for $i = 1, 2, \cdots, q - 1$, $Pr\{h = i\} = p_1 p_2 \cdots p_i$. The value of h can be determined after writing this way: h is the highest written level of the q auxiliary cells. Note that the random variable h is independent of the n target levels $x_1, x_2, \cdots, x_n$; and for $i = 1, \cdots, n$, the value of y_i is determined by x_i and h. So $\max_{P_T(t)} I(T; S) = n \max_{P_{x_i}(x)} I(x_i; y_i) = n \max_{P_X(x)} I(X; Y) = n \log_2 A_1$, where X, Y are the input and output symbols of the partial-erasure channel. Since the capacity of VLC is $C = \lim_{n \to \infty} \frac{1}{n} \max_{P_T(t)} I(T; S)$ (where we see every VLC group of n cells as one symbol for the channel, and the channel has infinitely many such symbols), we have $C = \log_2 A_1$. ∎

III. CAPACITY REGION FOR REWRITING DATA IN VLC

In this section, we study codes for rewriting data in VLC, and bound its achievable rates. There has been extensive study on rewriting codes for flash memories and PCMs (for both single-level cells (SLCs) and MLCs) for achieving longer memory lifetime [4], [5]. In the well known write-once memory (WOM) model, the cell levels can only increase when data are rewritten [3]. For flash memories and PCMs, the model describes the behavior of cells between two global erasure operations. Since erasures reduce the quality of cells, it is highly desirable to avoid them. Given the number of rewrites,

T, our objective is to maximize the rates of the code for the T rewrites, when cell levels can only increase for rewriting.

We study the achievable rates for rewriting in VLC. Note that unlike MLC, which are deterministic, the highest programmable level of a VLC group is a random variable. So we need to define code rates accordingly.

Consider a VLC group of n cells, whose highest programmable level is a random variable $h \in \{1, 2, \cdots, q-1\}$. (We assume $h \geq 1$ – namely $p_1 = 1$ – for the convenience of presentation. The analysis can be extended to $h \geq 0$.) Note that the value of h remains unknown until level h is programmed. To simplify rate analysis, we suppose that there are q auxiliary cells $a_0, a_1, \cdots, a_{q-1}$ in the same VLC group, whose target levels are $0, 1, \cdots, q-1$, respectively. For $i = 1, \cdots, h$, when level i is programmed, the auxiliary cell a_i will be raised to level i and always remain there. If $h < q-1$, after level h is programmed (at which point we find that level $h+1$ cannot be programmed), we push $a_{h+1}, \cdots, a_{q-1}$ to level h, too. So having more than one auxiliary cell in a level i indicates $h = i$. For sufficiently large n, the q auxiliary cells have a negligible effect on the code rate.

Now consider N VLC groups $G_1, G_2, \cdots, G_N$, each of n cells. (For capacity analysis, we consider $N \to \infty$.) For $i = 1, \cdots, N$, denote the highest programmable level of G_i by

$$h_i \in \{1, \cdots, q-1\},$$

and denote its cells by

$$(c_{i,1}, \cdots, c_{i,n}).$$

Here $h_1, \cdots, h_N$ are i.i.d. random variables, where for $1 \leq i \leq N$ and $1 \leq j \leq q-1$,

$$Pr\{h_i = j\} = p_1 p_2 \cdots p_j (1 - p_{j+1}).$$

(Note $p_1 = 1$ and $p_q \triangleq 0$.) If the *target level* of cell $c_{i,j}$ is $\ell_{i,j}$, we will program it to level

$$\min\{\ell_{i,j}, h_i\}.$$

Then if $h_i < q-1$ and the *written level* of cell $c_{i,j}$ is h_i, we say that the cell is in the *partially-erased state*

$$E_{h_i},$$

since its target level could be any value in $\{h_i, h_i + 1, \cdots, q-1\}$. In addition, for any two vectors $\mathbf{x} = (x_1, x_2, \cdots, x_k)$ and $\mathbf{y} = (y_1, y_2, \cdots, y_k)$, we say

$$\mathbf{x} \leq \mathbf{y}$$

if $x_i \leq y_i$ for $i = 1, \cdots, k$.

Definition 5. *A*

$$(T, V_1, V_2, \cdots, V_T)$$

rewriting code for the N VLC groups consists of T pairs of encoding and decoding functions

$$\{(f_t, g_t)\}_{t=1}^T,$$

with the message index sets

$$I_t = \{1, 2, \cdots, V_t\},$$

the encoding functions

$$f_t : I_t \times \{0, 1, \cdots, q-1\}^{Nn} \to \{0, 1, \cdots, q-1\}^{Nn},$$

and the decoding functions

$$g_t : \{0, 1, \cdots, q-1\}^{Nn} \to I_t.$$

Let $\mathbf{x}_0^{Nn} = (0, 0, \cdots, 0) \in \{0, 1 \cdots, q-1\}^{Nn}$. *Given any sequence of T messages*

$$m_1 \in I_1, m_2 \in I_2, \cdots, m_T \in I_T,$$

for the T rewrites, the target levels for the cells

$$(c_{1,1}, \cdots, c_{1,n}, c_{2,1}, \cdots, c_{2,n}, \cdots\cdots, c_{N,1}, \cdots, c_{N,n})$$

are

$$\mathbf{x}_1^{Nn} = f_1\left(m_1, \mathbf{x}_0^{Nn}\right),$$

$$\mathbf{x}_2^{Nn} = f_2\left(m_2, \mathbf{x}_1^{Nn}\right),$$

$$\vdots$$

$$\mathbf{x}_T^{Nn} = f_T\left(m_T, \mathbf{x}_{T-1}^{Nn}\right),$$

respectively, where

$$\mathbf{x}_{t-1}^{Nn} \leq \mathbf{x}_t^{Nn}$$

for $t = 1, \cdots, T$. However, while the target cell levels for the t-th rewrite (for $t = 1, \cdots, T$) are $\mathbf{x}_t^{Nn} = (\ell_{1,1}, \cdots, \ell_{1,n}, \ell_{2,1}, \cdots, \ell_{2,n}, \cdots\cdots, \ell_{N,1}, \cdots, \ell_{N,n})$, the written cell levels are $\mathbf{y}_t^{Nn} = \left(\ell'_{1,1}, \cdots, \ell'_{1,n}, \ell'_{2,1}, \cdots, \ell'_{2,n}, \cdots\cdots, \ell'_{N,1}, \cdots, \ell'_{N,n}\right)$, where $\ell'_{i,j} = \min\{\ell_{i,j}, h_i\}$. For decoding, it is required that for $t = 1, \cdots, T$, we have

$$Pr\{g_t\left(\mathbf{y}_t^{Nn}\right) = m_t\} \to 1$$

as $N \to \infty$.

For $t = 1, \cdots, T$, define

$$R_t = \frac{1}{Nn} \log_2 V_t.$$

Then $(R_1, R_2, \cdots, R_T)$ is called the rate vector of the code. □

We call the closure of the set of all rate vectors the *capacity region*, and denote it by $\mathcal{A}_T$. We present its inner/outer bounds.

A. Inner Bound to Capacity Region

We consider a *sub-channel code* for VLC. Let

$$c_1, c_2, \cdots, c_N$$

be N cells, one from each of the N VLC groups. (The Nn cells in the N VLC groups can be partitioned into n such "sub-channels.") We define the rewriting code for the N cells in the same way as in Definition 5 (by letting $n = 1$). We denote its capacity region by $\tilde{\mathcal{A}}_T$. Clearly, for any given n, we have $\tilde{\mathcal{A}}_T \subseteq \mathcal{A}_T$.

Let $\mathcal{L} = \{0, 1, \cdots, q-1\}$ denote the set of target levels. Let $\mathcal{E} = \{E_1, E_2, \cdots, E_{q-2}\}$ denote the set of partially-erased states. Then $\mathcal{L} \cup \mathcal{E}$ are written levels. For two random variables X, Y taking values in $\mathcal{L}$, we say "$X \Rightarrow Y$" if

$$Pr\{X = x, Y = y\} = 0$$

for any $0 \le y < x \le q-1$. Let random variables $S_1, S_2, \cdots, S_T$ form a Markov chain that takes values in $\mathcal{L}$. We say

$$S_1 \Rightarrow S_2 \Rightarrow \cdots \Rightarrow S_T$$

if $S_{t-1} \Rightarrow S_t$ for $t = 2, 3, \cdots, T$. For $i = 1, 2, \cdots, T$, let $(s_{i,0}, s_{i,1}, \cdots, s_{i,q-1})$ denote the probability distribution where

$$s_{i,j} = Pr\{S_i = j\}$$

for $j = 0, 1, \cdots, q-1$.

Given the random variables $S_1, S_2, \cdots, S_T$, we define $\alpha_{i,j}$ and $B_{i,j}$ (for $i = 1, 2, \cdots, T$ and $j = 1, 2, \cdots, q-2$) as follows. Let

$$\alpha_{i,j} = \left(\sum_{k=j}^{q-1} s_{i,k} \right) \left(\prod_{k=2}^{j} p_k \right) (1 - p_{j+1}).$$

We define $B_{i,j}$ to be a random variable taking values in

$$\{j, j+1, \cdots, q-1\},$$

where

$$Pr\{B_{i,j} = k\} = s_{i,k} / \left(\sum_{\ell=j}^{q-1} s_{i,\ell} \right)$$

for $k = j, j+1, \cdots, q-1$. We now present an inner bound to $\tilde{\mathcal{A}}_T$. Since $\tilde{\mathcal{A}}_T \subseteq \mathcal{A}_T$, it is also an inner bound to $\mathcal{A}_T$.

Theorem 6. *Define* $\mathcal{D}_T = \{(R_1, R_2, \cdots, R_T) \in \mathbb{R}^T \mid there$
exist Markov-chain random variables $S_1, S_2, \cdots, S_T$ *taking values in* $\{0, 1, \cdots, q-1\}$, *such that* $S_1 \Rightarrow S_2 \Rightarrow \cdots \Rightarrow S_T$ *and*

$$
\begin{aligned}
R_1 &\le H(S_1) - \Sigma_{i=1}^{q-2} \alpha_{1,i} H(B_{1,i}), \\
R_2 &\le H(S_2|S_1) - \Sigma_{i=1}^{q-2} \alpha_{2,i} H(B_{2,i}), \\
&\vdots \\
R_T &\le H(S_T|S_{T-1}) - \Sigma_{i=1}^{q-2} \alpha_{T,i} H(B_{T,i}).\}
\end{aligned}
$$

Then, we have $\mathcal{D}_T \subseteq \tilde{\mathcal{A}}_T$.

Proof: Suppose $S_1, S_2, \cdots, S_T$ are Markov-chain random variables that take values in $\{0, 1, \cdots, q-1\}$, and that

$S_1 \Rightarrow S_2 \Rightarrow \cdots \Rightarrow S_T$. For any constant $\epsilon > 0$ (which can be arbitrarily small), we set

$$V_1 = 2^{N[H(S_1) - \Sigma_{i=1}^{q-2} \alpha_{1,i} H(B_{1,i}) - 2\epsilon]},$$

$$V_t = 2^{N[H(S_t|S_{t-1}) - \Sigma_{i=1}^{q-2} \alpha_{t,i} H(B_{t,i}) - 2\epsilon]}, \quad \text{for } t = 2, 3, \cdots, T.$$

We will prove that when N is sufficiently large, there exists an $(T, V_1, V_2, \cdots, V_T)$ rewriting code for the N cells $c_1, c_2, \cdots, c_N$.

We first consider the case $T = 2$. Let $T_{S_1}^N$ denote the vectors in $\mathcal{L}^N = \{0, 1, \cdots, q-1\}^N$ of *type*

$$(s_{1,0}, s_{1,1}, \cdots, s_{1,q-1}).$$

That is,

$$
\begin{aligned}
T_{S_1}^N = \{&(x_1, x_2, \cdots, x_N) \in \mathcal{L}^N \mid \text{for } i = 0, 1, \cdots, q-1, \\
&\frac{|\{j | 1 \le j \le N, x_j = i\}|}{N} = s_{1,i}\}.
\end{aligned}
$$

Similarly, let $T_{S_2}^N$ denote the vectors in $\mathcal{L}^N$ of type

$$(s_{2,0}, s_{2,1}, \cdots, s_{2,q-1}).$$

We construct two sets $\tilde{T}_{S_1}^N$ and $\tilde{T}_{S_2}^N$ using random coding as follows. From the set $T_{S_1}^N$, we uniformly randomly select

$$\frac{\left| T_{S_1}^N \right|}{2^{N[\Sigma_{i=1}^{q-2} \alpha_{1,i} H(B_{1,i}) + \epsilon]}}$$

elements, and denote the selected subset by $\tilde{T}_{S_1}^N$. Similarly, from the set $T_{S_2}^N$, we uniformly randomly select

$$\frac{\left| T_{S_2}^N \right|}{2^{N[\Sigma_{i=1}^{q-2} \alpha_{2,i} H(B_{2,i}) + \epsilon]}}$$

elements, and denote the selected subset by $\tilde{T}_{S_2}^N$.

We first prove the following property:

- Property ♣: $\forall~ \mathbf{x} \in \tilde{T}_{S_1}^N$, if we program the N cells $c_1, \cdots, c_N$ with $\mathbf{x}$ as the target levels, let $\mathbf{y} \in (\mathcal{L} \cup \mathcal{E})^N$ denote their written levels. Then with high probability (which approaches 1 as N approaches infinity), the vector $\mathbf{y}$ can be correctly decoded to $\mathbf{x}$.

To prove Property ♣, consider the channel model for a cell c_i, with its target level

$$X \in \mathcal{L}$$

as the input symbol and its written level

$$Y \in \mathcal{L} \cup \mathcal{E}$$

as the output symbol. We have $Pr\{Y = 0 | X = 0\} = 1$; for $i = 1, 2, \cdots, q-2$, we have

$$Pr\{Y = i | X = i\} = p_2 p_3 \cdots p_{i+1}$$

and for $j = 1, 2, \cdots, i$,

$$Pr\{Y = E_j | X = i\} = p_2 p_3 \cdots p_j (1 - p_{j+1});$$

and we have

$$Pr\{Y = q - 1|X = q - 1\} = p_2 p_3 \cdots p_{q-1}$$

and for $j = 1, 2, \cdots, q - 2$,

$$Pr\{Y = E_j|X = q - 1\} = p_2 p_3 \cdots p_j(1 - p_{j+1}).$$

The channel model for $q = 6$ is illustrated in Fig. 3.

We can see that if X has the same distribution as the random variable S_1, then for $i = 1, 2, \cdots, q - 2$,

$$Pr\{Y = E_i\} = \left(\sum_{j=i}^{q-1} s_{1,j}\right)\left(\prod_{j=2}^{i} p_j\right)(1 - p_{i+1}) = \alpha_{1,i};$$

also, for $i = 1, 2, \cdots, q - 2$ and $j = i, i + 1, \cdots, q - 1$,

$$Pr\{X = j|Y = E_i\} = \frac{s_{1,j}}{\sum_{k=i}^{q-1} s_{1,k}} = Pr\{B_{1,i} = j\}.$$

For any $i \in \mathcal{L}$, if $Y = i$, then $X = i$ and $H(X|Y = i) = 0$. So we have

$$
\begin{aligned}
&H(X|Y) \\
=\ & \sum_{z \in \mathcal{L} \cup \mathcal{E}} Pr\{Y = z\}H(X|Y = z) \\
=\ & \sum_{z \in \mathcal{E}} Pr\{Y = z\}H(X|Y = z) \\
=\ & \sum_{i=1}^{q-2} \alpha_{1,i}H(B_{1,i})
\end{aligned}
$$

Since

$$
\begin{aligned}
& \lim_{N \to \infty} \left(log_2\left|\tilde{T}_{S_1}^N\right|\right)/N \\
=\ & \lim_{N \to \infty} \left(log_2\left|T_{S_1}^N\right|\right)/N - [\textstyle\sum_{i=1}^{q-2} \alpha_{1,i}H(B_{1,i}) + \epsilon] \\
=\ & H(S_1) - \textstyle\sum_{i=1}^{q-2} \alpha_{1,i}H(B_{1,i}) - \epsilon \\
=\ & H(X) - H(X|Y) - \epsilon \\
=\ & I(X;Y) - \epsilon \\
<\ & I(X;Y),
\end{aligned}
$$

when $N \to \infty$, with probability one we can decode $\mathbf{x}$ from $\mathbf{y}$ based on their joint typicality. So Property $\clubsuit$ is true. Using the same analysis, we get the following property for $\tilde{T}_{S_2}^N$:

- Property $\spadesuit$: $\forall\ \mathbf{x} \in \tilde{T}_{S_2}^N$, if we program the N cells $c_1, \cdots, c_N$ with $\mathbf{x}$ as the target levels, let $\mathbf{y} \in (\mathcal{L} \cup \mathcal{E})^N$ denote their written levels. Then with high probability (which approaches 1 as N approaches infinity), the vector $\mathbf{y}$ can be correctly decoded to $\mathbf{x}$.

We now discuss the encoding and decoding of the $T = 2$ writes. For the first write, we choose V_1 different elements

$$\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{V_1} \in \tilde{T}_{S_1}^N,$$

and set the encoding function as

$$f_1(i) = \mathbf{x}_i.$$

To write data $i \in I_1 = \{1, 2, \cdots, V_1\}$, we program the cells with $\mathbf{x}_i$ as their target levels. Let $\mathbf{y} \in (\mathcal{L} \cup \mathcal{E})^N$ denote the written levels. We set the decoding function $g_1(\mathbf{y})$ as follows: it first recovers $\mathbf{x}_i$ from $\mathbf{y}$ based on joint typicality (which succeeds with high probability by Property $\clubsuit$), then maps $\mathbf{x}_i$ to i. So the first write succeeds with high probability.

Consider the second write. Let

$$\{F_1, F_2, \cdots, F_{V_2}\}$$

be a partition of the set $\tilde{T}_{S_2}^N$. Namely, $\cup_{i=1}^{V_2} F_i = \tilde{T}_{S_2}^N$ and for any $i \neq j$, $F_i \cap F_j = \emptyset$. We first show that the following property holds:

- Property $\diamondsuit$: There exists a partition $\{F_1, F_2, \cdots, F_{V_2}\}$ of the set $\tilde{T}_{S_2}^N$ such that for any $\mathbf{u} \in \tilde{T}_{S_1}^N$ and any $v \in I_2 = \{1, 2, \cdots, V_2\}$, there exists a vector $\mathbf{x} \in F_v$ such that $\mathbf{u} \leq \mathbf{x}$.

To prove Property $\diamondsuit$, we use the method of random coding. For every $\mathbf{z} \in T_{S_2}^N$, associate it with an index $r_{\mathbf{z}}$ that is uniformly randomly and independently chosen from the message index set $I_2 = \{1, 2, \cdots, V_2\}$. For $i = 1, 2, \cdots, V_2$, define

$$F_i = \{\mathbf{z} \in \tilde{T}_{S_2}^N \mid r_{\mathbf{z}} = i\}.$$

Then $\{F_1, F_2, \cdots, F_{V_2}\}$ form a partition of the set $\tilde{T}_{S_2}^N$.

For any $\mathbf{u} \triangleq (u_1, u_2, \cdots, u_N) \in T_{S_1}^N$, define the set of conditional typical sequences $T_{S_2|S_1}^N(\mathbf{u})$ as

$$
\begin{aligned}
T_{S_2|S_1}^N(\mathbf{u}) =\ & \{\ (v_1, v_2, \cdots, v_N) \in \mathcal{L}^N \mid \forall\ (a, b) \in \mathcal{L}^2, \\
& \frac{|\{i|1 \leq i \leq N, u_i = a, v_i = b\}|}{N} = Pr\{S_1 = a, S_2 = b\}\},
\end{aligned}
$$

and define $G(\mathbf{u})$ as

$$G(\mathbf{u}) = \{\mathbf{v} \in T_{S_2}^N \mid \mathbf{u} \leq \mathbf{v}\}.$$

Since $S_1 \Rightarrow S_2$, we have $T_{S_2|S_1}^N(\mathbf{u}) \subseteq G(\mathbf{u})$. By the property of typical sequences [2], [3], we have

$$|G(\mathbf{u})| \geq \left|T_{S_2|S_1}^N(\mathbf{u})\right| \geq (N+1)^{-q^2} 2^{NH(S_2|S_1)}.$$

For any $v \in I_2 = \{1, 2, \cdots, V_2\}$ and $\mathbf{u} \in \tilde{T}_{S_1}^N$, we get

$$
\begin{aligned}
& Pr\{F_v \cap G(\mathbf{u}) = \emptyset\} \\
=\ & Pr\{\text{for every } \mathbf{z} \in G(\mathbf{u}), \text{ either } r_{\mathbf{z}} \neq v \text{ or } \mathbf{v} \notin \tilde{T}_{S_2}^N\} \\
=\ & \left(1 - \frac{1}{V_2 \cdot 2^{N[\sum_{i=1}^{q-2} \alpha_{2,i}H(B_{2,i}) + \epsilon]}}\right)^{|G(\mathbf{u})|} \\
\leq\ & exp\{-\frac{|G(\mathbf{u})|}{V_2 \cdot 2^{N[\sum_{i=1}^{q-2} \alpha_{2,i}H(B_{2,i}) + \epsilon]}}\} \\
\leq\ & exp\{-\frac{(N+1)^{-q^2} 2^{NH(S_2|S_1)}}{2^{N[H(S_2|S_1) - \sum_{i=1}^{q-2} \alpha_{2,i}H(B_{2,i}) - 2\epsilon] + N[\sum_{i=1}^{q-2} \alpha_{2,i}H(B_{2,i}) + \epsilon]}}\} \\
=\ & exp\{-\frac{(N+1)^{-q^2} 2^{NH(S_2|S_1)}}{2^{N[H(S_2|S_1) - \epsilon]}}\} \\
=\ & exp\{-(N+1)^{-q^2} 2^{N\epsilon}\}
\end{aligned}
$$

By the union bound, we get

$$
\begin{aligned}
& Pr\{\exists v \in I_2 \text{ and } \mathbf{u} \in \tilde{T}_{S_1}^N \text{ such that } F_v \cap G(\mathbf{u}) = \emptyset\} \\
\leq\ & V_2 \cdot \left|\tilde{T}_{S_1}^N\right| \cdot exp\{-(N+1)^{-q^2} 2^{N\epsilon}\} \\
\leq\ & q^{2N} \cdot exp\{-(N+1)^{-q^2} 2^{N\epsilon}\} \\
\to\ & 0 \quad \text{as } N \to \infty.
\end{aligned}
$$

This implies that Property $\diamondsuit$ is true.

We now describe the encoding and decoding functions of the second write. Let $\{F_1, F_2, \cdots, F_{V_2}\}$ be a partition of the
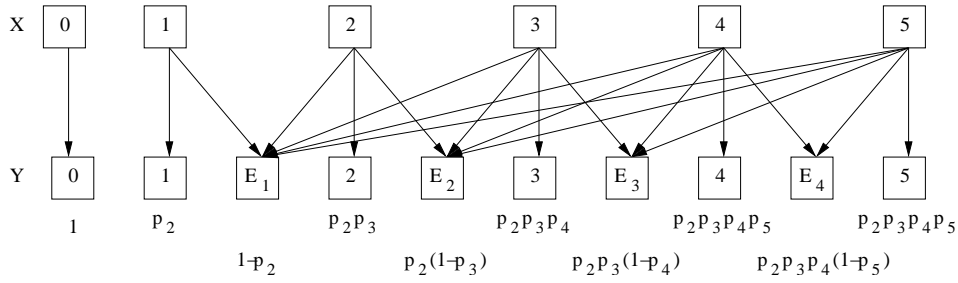
Fig. 3. Channel model for $q = 6$ levels, with X, Y as input and output symbols, respectively. Here the probability $Pr\{Y = j | X = i\}$ is labelled beside the symbol j for Y.)

set $\tilde{T}_{S_2}^N$ that has the property described in Property $\diamondsuit$. For any $\mathbf{u} \in \tilde{T}_{S_1}^N$ and any $v \in I_2 = \{1, 2, \cdots, V_2\}$, there exists a vector $\mathbf{x}_v(\mathbf{u}) \in F_v$ such that $\mathbf{u} \leq \mathbf{x}_v(\mathbf{u})$. We set the encoding function $f_2(v, \mathbf{u}')$ as follows: given the written levels $\mathbf{u}'$ of the first write, it first recovers the target levels $\mathbf{u}$ of the first write (which succeeds with high probability by Property $\clubsuit$), then sets $f_2(v, \mathbf{u}') = \mathbf{x}_v(\mathbf{u})$. When the target cell levels of the first write are $\mathbf{u}$, to write data $v \in I_2$ in the second write, we program the cells with $\mathbf{x}_v(\mathbf{u})$ as their target levels. Let $\mathbf{y} \in (\mathcal{L} \cup \mathcal{E})^N$ denote the written levels. We set the decoding function $g_2(\mathbf{y})$ as follows: it first recovers $\mathbf{x}_v(\mathbf{u})$ from $\mathbf{y}$ based on joint typicality (which succeeds with high probability by Property $\spadesuit$), then maps $\mathbf{x}_v(\mathbf{u})$ to v. So the second write succeeds with high probability.

The above proof for $T = 2$ can be easily generalized to the proof for general T. The encoding and decoding functions for the tth write (for $t = 3, 4, \cdots, T$) can be defined in the same way as for the second write. So we get the conclusion. ∎

Note that if $p_2 = p_3 = \cdots = p_{q-1} = 1$ (namely, every cell can be programmed to the highest level $q - 1$ with guarantee), we get $\alpha_{i,j} = 0$ for all i, j. Consequently, the set of achievable rates presented in the above theorem, $\mathcal{D}_T$, becomes $\mathcal{D}_T = \{(R_1, R_2, \cdots, R_T) \in \mathbb{R}^T \mid$ there exist Markov-chain random variables $S_1, S_2, \cdots, S_T$, such that $S_1 \Rightarrow S_2 \Rightarrow \cdots \Rightarrow S_T$ and $R_1 \leq H(S_1)$, $R_2 \leq H(S_2|S_1)$, $\cdots$, $R_T \leq H(S_T|S_{T-1})\}$, which is exactly the capacity region of MLC with q levels [3].

B. Outer Bound to Capacity Region

To derive an outer bound to the capacity region $\mathcal{A}_T$, we consider the rewriting code as defined in Definition 5, but with an additional property: the highest reachable levels $h_1, h_2, \cdots, h_N$ for the N VLC groups are known in advance. Thus the encoding and decoding functions can use that information. Let $\mathcal{A}_T^*$ denote its capacity region. Clearly, $\mathcal{A}_T^* \supseteq \mathcal{A}_T$, so it is an outer bound to $\mathcal{A}_T$.

Theorem 7. *Define* $\mathcal{G}_T = \{(R_1, R_2, \cdots, R_T) \in \mathbb{R}^T \mid$ *for* $i = 1, 2, \cdots, q-1$, *there exist*

$$(r_{1,i}, r_{2,i}, \cdots, r_{T,i}) \in \mathbb{R}^T$$

and Markov-chain random variables $S_{1,i}, S_{2,i}, \cdots, S_{T,i}$ *taking values in* $\{0, 1, \cdots, i\}$, *such that*

$$S_{1,i} \Rightarrow S_{2,i} \Rightarrow \cdots \Rightarrow S_{T,i},$$

$$\begin{aligned} r_{1,i} &\leq H(S_{1,i}), \\ r_{2,i} &\leq H(S_{2,i}|S_{1,i}), \\ &\vdots \\ r_{T,i} &\leq H(S_{T,i}|S_{T-1,i}), \end{aligned}$$

and for $j = 1, 2, \cdots, T$,

$$R_j = \sum_{k=1}^{q-1} p_1 p_2 \cdots p_k (1 - p_{k+1}) r_{j,k}. \ \}$$

Let $\mathcal{C}_T$ *be the closed set generated by* $\mathcal{G}_T$. *Then, we have*

$$\mathcal{A}_T^* = \mathcal{C}_T.$$

Proof: For $i = 1, 2, \cdots, q - 1$, let Q_i be the indices of the VLC groups whose highest reachable levels are all level i. That is,

$$Q_i = \{j \in \{1, 2, \cdots, N\} \mid h_j = i\} \subseteq \{1, 2, \cdots, N\}.$$

Also, define

$$\gamma_i = p_1 p_2 \cdots p_i (1 - p_{i+1}).$$

(As before, $p_q \triangleq 0$.) Clearly,

$$\frac{|Q_i|}{N} \to \gamma_i$$

with high probability as $N \to \infty$.

We first prove that all rate vectors $(R_1, R_2, \cdots, R_T)$ in $\mathcal{G}_T$ are achievable rate vectors. It is known that for WOM of $i + 1$ levels [3], the rate vector

$$(r_{1,i}, r_{2,i}, \cdots, r_{T,i})$$

is achievable for T writes if and only if there exist Markov-chain random variables $S_{1,i}, S_{2,i}, \cdots, S_{T,i}$ taking values in $\{0, 1, \cdots, i\}$ such that

$$S_{1,i} \Rightarrow S_{2,i} \Rightarrow \cdots \Rightarrow S_{T,i}$$

and

$$r_{1,i} \leq H(S_{1,i}),$$

$$r_{2,i} \le H(S_{2,i}|S_{1,i}),$$

$$\cdots,$$

$$r_{T,i} \le H(S_{T,i}|S_{T-1,i}).$$

So for $i = 1, 2, \cdots, q-1$, we can use the cells in the VLC groups indexed by Q_i to achieve T writes with the rate vector $(r_{1,i}, r_{2,i}, \cdots, r_{T,i})$. Together, the N VLC groups achieve T writes with the rate vector $(R_1, R_2, \cdots, R_T)$.

Next, we prove the converse. Given a $(T, V_1, V_2, \cdots, V_T)$ code, we need to show that

$$\left(\frac{1}{Nn} \log_2 V_1, \frac{1}{Nn} \log_2 V_2, \cdots, \frac{1}{Nn} \log_2 V_T \right) \in \mathcal{G}_T.$$

We use the same technique of proof as in [3] (Theorem 3.1). For $t = 1, 2, \cdots, T$, let f_t, g_t denote the encoding and decoding functions of the code for the t-th write, respectively.

Let $W_1, W_2, \cdots, W_T$ be independent random variables that are uniformly distributed over the message index set $I_t = \{1, 2, \cdots, V_t\}$ (for $t = 1, 2, \cdots, T$), respectively. Let $\mathbf{Y}_0^{Nn} \triangleq \{0, 0, \cdots, 0\}$ denote the all-zero vector of length Nn. Then for $t = 1, 2, \cdots, T$, define

$$\mathbf{Y}_t^{Nn} = (Y_{t,1}, Y_{t,2}, \cdots, Y_{t,Nn})$$

as

$$\mathbf{Y}_t^{Nn} = f_t \left(W_t, \mathbf{Y}_{t-1}^{Nn} \right).$$

That is, $\mathbf{Y}_t^n$ denotes the cell levels after the t-th write. It is not hard to see that

$$H(W_t) = H(\mathbf{Y}_t^{Nn}|\mathbf{Y}_{t-1}^{Nn})$$

for $t = 1, 2, \cdots, T$.

For $i = 1, 2, \cdots, q-1$, let

$$\tilde{Q}_i \subseteq \{1, 2, \cdots, Nn\}$$

denote the indices of the cells whose highest reachable levels are all i, and let L_i be an independent random variable that is uniformly distributed over the index set $\tilde{Q}_i$. (Specifically, the indices for cells in VLC group G_1 are $\{1, 2, \cdots, n\}$, the indices for cells in G_2 are $\{n+1, n+2, \cdots, 2n\}$, and so on.) Let L be an independent random variable that is uniformly distributed over the index set $\{1, 2, \cdots, Nn\}$. We get

$$\frac{1}{Nn} \log_2 V_t = \frac{1}{Nn} H(W_t) = \frac{1}{Nn} H(\mathbf{Y}_t^{Nn}|\mathbf{Y}_{t-1}^{Nn})$$
$$\le \frac{1}{Nn} \sum_{i=1}^{Nn} H(Y_{t,i}|Y_{t-1,i})$$
$$= \sum_{i=1}^{Nn} Pr\{L = i\} H(Y_{t,L}|Y_{t-1,L}, L = i)$$
$$= \sum_{i=1}^{q-1} \gamma_i \sum_{j \in \tilde{Q}_i} Pr\{L = j|L \in \tilde{Q}_i\} H(Y_{t,L}|Y_{t-1,L}, L = j)$$
$$= \sum_{i=1}^{q-1} \gamma_i H(Y_{t,L_i}|Y_{t-1,L_i}, L_i)$$
$$\le \sum_{i=1}^{q-1} \gamma_i H(Y_{t,L_i}|Y_{t-1,L_i})$$

For $i = 1, 2, \cdots, q-1$, define a set of new random variables $S_{1,i}, S_{2,i}, \cdots, S_{T,i}$ taking values in $\{0, 1, \cdots, i\}$, whose joint probability distribution is defined as

$$Pr\{S_{1,i} = j_1, S_{2,i} = j_2, \cdots, S_{T,i} = j_T\}$$
$$= Pr\{Y_{1,L_i} = j_1\} Pr\{Y_{2,L_i} = j_2|Y_{1,L_i} = j_1\} \cdots$$
$$Pr\{Y_{T,L_i} = j_T|Y_{T-1,L_i} = j_{T-1}\}.$$

Define $S_{0,i} \triangleq 0$. It is not hard to see that $S_{1,i}, S_{2,i}, \cdots, S_{T,i}$ form a Markov chain, and for any $t \in \{1, 2, \cdots, T\}$ the random variables $(S_{t-1,i}, S_{t,i})$ and (Y_{t-1,L_i}, Y_{t,L_i}) have the same probability distribution. So $H(S_{1,i}) = H(Y_{1,L_i})$ and for $t = 2, 3, \cdots, T$,

$$H(S_{t,i}|S_{t-1,i}) = H(Y_{t,L_i}|Y_{t-1,L_i}).$$

Since $Y_{t-1,L_i} \Rightarrow Y_{t,L_i}$ for $t = 2, 3, \cdots, T$, we have $S_{1,i} \Rightarrow S_{2,i} \Rightarrow \cdots \Rightarrow S_{T,i}$. Therefore for $t = 1, 2, \cdots, T$,

$$\frac{1}{Nn} \log_2 V_t \le \sum_{i=1}^{q-1} \gamma_i H(S_{t,i}|S_{t-1,i}).$$

So we have

$$\left(\frac{1}{Nn} \log_2 V_1, \frac{1}{Nn} \log_2 V_2, \cdots, \frac{1}{Nn} \log_2 V_T \right) \in \mathcal{G}_T.$$

That completes the converse part of the proof. So $\mathcal{A}_T^* = \mathcal{C}_T$. ∎

Let

$$M_T \triangleq \max\{ \sum_{t=1}^{T} R_t | (R_1, R_2, \cdots, R_T) \in \mathcal{A}_T \}$$

denote the maximum total rate of all rewriting codes for VLC. It is known that for WOM (i.e., MLC) of $i+1$ levels, the maximum total rate over T writes is $\log_2 \binom{T+i}{i}$ [3]. By Theorem 7, we get

$$M_T \le \max\{ \sum_{t=1}^{T} R_t | (R_1, R_2, \cdots, R_T) \in \mathcal{A}_T^* \}$$
$$= \sum_{k=1}^{q-1} p_1 p_2 \cdots p_k (1 - p_{k+1}) \log_2 \binom{T+k}{k}.$$

IV. CONCLUSION

This paper introduces a new data representation scheme, variable-level cells, for nonvolatile memories. By adaptively choosing the number and positions of levels in cells, higher storage rates can be achieved. The storage capacity of the VLC scheme is proved. Coding for rewriting data is also analyzed for the VLC scheme, and both inner and outer bounds to the capacity region of rewriting are presented.

ACKNOWLEDGMENT

REFERENCES

[1] J. E. Brewer and M. Gill (*Ed.*), *Nonvolatile memory technologies with emphasis on flash*, John Wiley & Sons, Inc., Hoboken, New Jersey, 2008.

[2] I. Csiszar and J. Korner, *Information Theory: Coding Theorems for Discrete Memoryless Systems*, New York: Academic, 1981.

[3] F. Fu and A. J. Han Vinck, "On the capacity of generalized write-once memory with state transitions described by an arbitrary directed acyclic graph," *IEEE Transactions on Information Theory*, vol. 45, no. 1, pp. 308-313, 1999.

[4] A. Jiang, V. Bohossian, and J. Bruck, "Floating codes for joint information storage in write asymmetric memories," *Proc. IEEE International Symposium on Information Theory (ISIT)*, Nice, France, Jun. 2007, pp. 1166–1170.

[5] L. A. Lastras-Montano, M. Franceschini , T. Mittelholzer, J. Karidis and M. Wegman, "On the lifetime of multilevel memories," *Proc. IEEE International Symposium on Information Theory (ISIT)*, Seoul, Korea, 2009, pp. 1224-1228.

[6] H. T. Lue *et al.*, "Study of incremental step pulse programming (ISPP) and STI edge effect of BE-SONOS NAND flash," *Proc. IEEE Int. Symp. on Reliability Physics*, vol. 30, no. 11, pp. 693-694, May 2008.