Coding for Write ℓ-step-up Memories

Yeow Meng Chee*, Han Mao Kiah†, A. J. Han Vinck‡ Van Khu Vu†, and Eitan Yaakobi§
*Department of Industrial Systems Engineering and Management, National University of Singapore, Singapore
†School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore
‡ Institute of Digital Signal Processing, University of Duisburg-Essen, 47057 Duisburg, Germany
§Computer Science Department, Technion—Israel Institute of Technology, Haifa, Israel
Emails: pvocym@nus.edu.sg, {hmkiah,vankhu001}@ntu.edu.sg, han.vinck@uni-due.de, yaakobi@cs.technion.ac.il

***Abstract*—In this work, we propose and study a new class of non-binary rewriting codes, called *write ℓ-step-up memories* (*WℓM*) codes. From an information-theoretic point of view, this coding scheme is a generalization of non-binary *write-once memories* (WOM) codes. From a practical point of view, this coding scheme can be used not only to increase the lifetime of flash memories but also mitigate their over-shooting problem. We first provide an exact formula for the capacity region and the maximum sum-rate of WℓM codes. Lastly, we present several explicit constructions of high-rate WℓM codes with efficient encoding/decoding algorithms.**

I. Introduction

Write-once memory (WOM) is a binary information storage medium where the state of each cell can be changed from 0 to 1 but not vice versa. In 1982, Rivest and Shamir [1] studied a coding scheme to reuse write-once memory which is known in literature as WOM codes. Later, Fiat and Shamir [2] studied a coding scheme for generalized write-once memory where each cell has q levels that can only be increased. This coding scheme is called non-binary *WOM codes*. In the 1980's and 1990's, a few different models of WOM codes were studied due to its beauty from an information-theoretical point of view and its applications in punch cards and optical disks [2]–[6]. Furthermore, there are several versions of WOM codes, including *write uni-directional memory* [7], *write isolated memory* [8], and defective channel [9], [10]. Recently, WOM codes have gained significant attention thanks to their applications in flash memories [11]–[17].

Flash memory, invented in 1984 by Masuoka [18], has become a popular non-volatile storage technology due to its high capacity, low power consumption, and low cost. However, flash memory still faces several major challenges such as limited lifetime and noisy programming. The fundamental data storage element in flash memories is a floating-gate (FG) transistor, known as a *cell*. In a single level cell (SLC), each cell has two levels and thus is able to store a single bit, and in a multilevel cell (MLC), each cell has $q > 2$ levels and is able to store $\log_2 q$ bits. While it is possible to increase a cell level by injecting an appropriate charge amount into the cell, it is impossible to decrease a cell level without first erasing a whole block of cells. This erasure operation is not only expensive in terms of time but also reduces the lifetime of flash memories. Each block of cells in a flash memory can be erased a limited number of times. To improve the lifetime of flash memory, it is possible to use WOM codes to write multiple messages before the whole block is erased.

*This work was done while Y. M. Chee was with the School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore.

On top the limited lifetime of flash memory, it is also challenging to precisely program the cells. One of the main obstacles in achieving accurate programming is over-shooting. When a large amount of charge is injected into a cell, it may be over charged than necessary and thus an error occurs. One approach to overcome this difficulty is by using coding schemes such as asymmetric error-correcting codes [19] and rank-modulation codes [20].

The goal of this paper is to initiate the study of a coding scheme which combats both limited-endurance and overshooting in flash memory. One approach to meet this challenge is by using error-correcting codes together with WOM codes as was studied in [12], [21], [22]. However, in order to directly reduce the over-shooting problem, this work proposes to use a generalization of non-binary WOM codes where each cell has q levels that can only be increased by at most some $\ell < q$ levels. These codes are called *write ℓ-step-up memories (WℓM) codes*. Similarly to WOM codes, there are four models of WℓM codes which depend upon whether the encoder and the decoder are informed or uninformed with the previous state of the memory [3] and each model can be investigated in two cases: ϵ-error and zero-error. In this paper, we focus on the more practical model where the encoder is informed and the decoder is uninformed for the zero-error case. Although this paper is the first to study WℓM codes, there are several closely related models such as codes for *endurance-limited memories (ELM)* [23] and *write-constrained memories (WCM)* codes [24]. Results on the exact capacity region and the maximum sum-rate of WℓM codes in some cases can be accomplished using known techniques. However, we are not aware of any explicit construction of high-rate WℓM codes in the literature. Before we present these results, we introduce some necessary notations and definitions.

A. Notations and Definitions

In this section, we define WℓM codes formally and present several related definitions that will be used throughout the paper. For a positive integer a, the set $\{0, \dots, a-1\}$ is defined by $[a]$. A vector $\boldsymbol{c} \in [q]^n$ will be called a *cell-state vector*. The complement of a binary vector $\boldsymbol{c}$ is denoted by $\overline{\boldsymbol{c}}$. That is $\boldsymbol{c} + \overline{\boldsymbol{c}} = \boldsymbol{1}$, where $\boldsymbol{1}$ is the all ones vector. For two vectors $\boldsymbol{x} = (x_1, \dots, x_n)$ and $\boldsymbol{y} = (y_1, \dots, y_n)$, $\boldsymbol{x} \geqslant \boldsymbol{y}$ if and only if $x_i \geqslant y_i$ for all $1 \leqslant i \leqslant n$. The vector $\boldsymbol{c} = \max\{\boldsymbol{x}, \boldsymbol{y}\}$ is defined by $c_i = \max\{x_i, y_i\}$ for all $1 \leqslant i \leqslant n$. A vector $\boldsymbol{p} = (p_1, \dots, p_n)$ is called a probability vector if $0 \leqslant p_i \leqslant 1$ for all $1 \leqslant i \leqslant n$ and $\sum_{i=1}^{n} p_i = 1$. Unless stated otherwise, all logarithms in this paper are taken according to base 2. For each probability vector $\boldsymbol{p}$, we define the entropy function $H(\boldsymbol{p}) = -\sum_{i=1}^{n} p_i \log p_i$. When $n = 2$ and $\boldsymbol{p} = (1-p, p)$, the entropy function is $h(p) = -p\log p - (1-p)\log(1-p)$. We first recall the definition of non-binary WOM codes.

Definition 1. *An $[n, t; M_1, \ldots, M_t]_q$ q-ary t-write WOM code is a coding scheme comprising of q-ary cells and is defined by t pairs of encoding and decoding maps $\mathcal{E}_j$ and $\mathcal{D}_j$ for $1 \leqslant j \leqslant t$. For the map $\mathcal{E}_j$, its image is $Im(\mathcal{E}_j)$ for $1 \leqslant j \leqslant t$. By definition, $Im(\mathcal{E}_0) = \{(0, \ldots, 0)\}$. For $1 \leqslant j \leqslant t$, the encoding and decoding maps are defined as follows.*

(1)
$$\mathcal{E}_j : [M_j] \times Im(\mathcal{E}_{j-1}) \mapsto [q-1]^n,$$
such that for all $(m, \boldsymbol{c}) \in [M_j] \times Im(\mathcal{E}_{j-1})$ it holds that $\mathcal{E}_j(m, \boldsymbol{c}) \geqslant \boldsymbol{c}$.

(2)
$$\mathcal{D}_j : Im(\mathcal{E}_j) \mapsto [M_j],$$
such that $\mathcal{D}_j(\mathcal{E}_j(m, \boldsymbol{c})) = m$ for all $(m, \boldsymbol{c}) \in [M_j] \times [q-1]^n$.

Next, q-ary t-write WℓM codes are defined.

Definition 2. *An $[n, t, \ell; M_1, \ldots, M_t]_q$ q-ary t-write WℓM code is a coding scheme comprising of q-ary cells and is defined by t pairs of encoding and decoding maps $\mathcal{E}_j$ and $\mathcal{D}_j$ for $1 \leqslant j \leqslant t$. For the map $\mathcal{E}_j$, its image is $Im(\mathcal{E}_j)$ for $1 \leqslant j \leqslant t$. By definition, $Im(\mathcal{E}_0) = \{(0, \ldots, 0)\}$. For $1 \leqslant j \leqslant t$, the encoding and decoding maps are defined as follows.*

(1)
$$\mathcal{E}_j : [M_j] \times Im(\mathcal{E}_{j-1}) \mapsto [q-1]^n,$$
such that for all $(m, \boldsymbol{c}) \in [M_j] \times Im(\mathcal{E}_{j-1})$ it holds that $\boldsymbol{c}' = (c'_1, \ldots, c'_n) = \mathcal{E}_j(m, \boldsymbol{c})$, where $c'_i - c_i \in [\ell]$ for all $1 \leqslant i \leqslant n$.

(2)
$$\mathcal{D}_j : Im(\mathcal{E}_j) \mapsto [M_j],$$
such that $\mathcal{D}_j(\mathcal{E}_j(m, \boldsymbol{c})) = m$ for all $(m, \boldsymbol{c}) \in [M_j] \times [q-1]^n$.

The following definitions apply both for WOM and WℓM codes. The *rate* on the j-th write is the ratio between the number of written bits and the number of cells, that is, $R_j = \frac{\log M_j}{n}$. The *sum-rate* is the sum of all rates on t writes, that is, $R_{sum} = \sum_{j=1}^{t} R_j$. In q-ary t-writes WℓM code, a rate tuple $(R_1, \ldots, R_t)$ is said to be achievable if for any $\epsilon > 0$, there exists an $[n, t, \ell; M_1, \ldots, M_t]_q$ q-ary t-write WℓM code such that $R_j - \epsilon \leqslant \frac{\log M_j}{n}$ for all $1 \leqslant j \leqslant t$. The *capacity region* of q-ary t-writes WℓM is the set of all achievable rate tuples and is denoted by $\mathbb{C}_{q,t,\ell}$, and the *maximum sum-rate* is denoted by $\mathcal{R}_{q,t,\ell}$.

From the definitions above, we observe that WOM codes are a special case of WℓM codes when $\ell = q-1$. There are several more known families of codes in the literature which are closely related to WℓM code and are reviewed next.

B. Related Work

First, we recall that the proposed WℓM codes are a generalization of non-binary WOM codes where every increase in cell level is at most some value ℓ. Gabrys et al. [25] studied a relevant model where every cell level increase is at least ℓ. WℓM codes are also closely related to the recently studied *endurance limited* (*ELM*) codes [23]. In ELM, each cell has two states which can be changed at most some b times. For every cell, its *count-vector* is a vector which indicates the number of times each cell changed its state. When a cell changes its state in ELM, the corresponding coordinate in the count-vector is increased by one. Hence, this count-vector is similar to a word in WℓM code for $\ell = 1$. Yet, these codes are still not identical and furthermore the count-vector is not always available in ELM. There are a few different models in ELM which depend upon whether the encoder and decoder are informed or uninformed. For the formal definitions of these models, we refer to the reference [23]. Let $\mathbb{C}_{t,q-1}^{EIA:DI}$ and $\mathbb{C}_{t,q-1}^{EIA:DU}$ be the capacity region of $(q-1)$-change t-write ELM codes in the $EIA:DI$ and $EIA:DU$ model, respectively. According to these definitions, we can already obtain the following result on the capacity region of q-ary t-write WℓM code with $\ell = 1$.

Proposition 3. $\mathbb{C}_{t,q-1}^{EIA:DI} \subseteq \mathbb{C}_{q,t,1} \subseteq \mathbb{C}_{t,q-1}^{EIA:DU}$.

Furthermore, it is known from [23] that $\mathbb{C}_{t,q-1}^{EIA:DI} = \mathbb{C}_{t,q-1}^{EIA:DU} = \widehat{\mathcal{C}}_{t,q-1}$, where the region $\widehat{\mathcal{C}}_{t,q-1}$ is defined recursively as follows.

$$\widehat{\mathcal{C}}_{t,q-1} = \Big\{ (R_1, \ldots, R_t) | R_1 \leqslant h(p), p \in [0,1], \text{ for } 2 \leqslant j \leqslant t, R_j \leqslant p \cdot R'_j + (1-p) \cdot R''_j, (R'_2, \ldots, R'_t) \in \widehat{\mathcal{C}}_{t-1,q-2} \text{ and } (R''_2, \ldots, R''_t) \in \widehat{\mathcal{C}}_{t-1,q-1} \Big\},$$

where $\widehat{\mathcal{C}}_{t,0} = \emptyset$, and for all $q-1 \geqslant t \geqslant 1$, we set $\widehat{\mathcal{C}}_{t,q-1} = \widehat{\mathcal{C}}_{t,t} = [0,1]^t$. Therefore, for q-ary t-write WℓM when $\ell = 1$, the capacity region is $\mathbb{C}_{q,t,1} = \widehat{\mathcal{C}}_{t,q-1}$ and the maximum sum-rate is $\mathcal{R}_{q,t,1} = \log \sum_{i=0}^{q-1} \binom{t}{i}$, [23].

Recently, Kobayashi et al. [24] studied a coding scheme for *write-constrained memories* (*WCM*) which is a WOM code with costs on the state-transitions. In this coding scheme, each cell-state transition from level i to level j has a cost, denoted by $c(i \to j)$. Using this notation, if we assign $c(i \to j) = 1$ for $j - i \in [\ell]$ and $c(i \to j) = \infty$ otherwise, then we can obtain WℓM codes. The work of [24] extended the results of Fu and Vinck [6] to obtain the capacity region of WCM codes. However, it is not possible to explicitly derive the capacity region $\mathbb{C}_{q,t,\ell}$ using the expression from [24] and constructing explicit capacity-achieving codes still remains an interesting challenge which is addressed in the paper.

II. The Capacity of q-ary WℓM

In this section, we present the capacity region and the maximum sum-rate of WℓM codes. The following rate tuples region is defined recursively,

$$\widehat{\mathcal{C}}_{q,t,\ell} = \Big\{ (R_1, \ldots, R_t) | \boldsymbol{p} = (p_0, \ldots, p_\ell) \text{ is a probability vector } R_1 \leqslant h(\boldsymbol{p}), \text{for } 2 \leqslant j \leqslant t, R_j \leqslant \sum_{i=0}^{\ell} p_i \cdot R_j^i, (R_2^i, \ldots, R_t^i) \in \widehat{\mathcal{C}}_{t-1,q-1-i,\ell} \text{ for } 0 \leqslant i \leqslant \ell \Big\}.$$

The next theorem establishes our result on the capacity of WℓM. The proof will appear in the long version of the paper.

Theorem 4. *For all q, t, ℓ, $\mathbb{C}_{q,t,\ell} = \widehat{\mathcal{C}}_{q,t,\ell}$.*

According to the result in Theorem 4, the maximum sum-rate of q-ary t-write WℓM codes is derived. For all t, q, and ℓ, let

$$B(q,t,\ell) = \sum_{(i_1,\ldots,i_\ell): \sum_{j=1}^{\ell} j \cdot i_j \leqslant q-1} \binom{t}{(t - \sum_{j=1}^{\ell} i_j), i_1, \ldots, i_\ell}.$$

Theorem 5. *For all t, q, and ℓ, $\mathcal{R}_{q,t,\ell} = \log B(q,t,\ell)$.*

Proof: Let $\mathcal{B}_{q,t,\ell}$ be the set of all length-t vectors in $[\ell+1]^t$ such that their Lee-weight is at most $q-1$, that is,

$$\mathcal{B}_{q,t,\ell} = \left\{ \boldsymbol{x} = (x_1,\ldots,x_t) | 0 \leqslant x_i \leqslant \ell \text{ and } \sum_{i=0}^{t} x_i \leqslant q-1 \right\}.$$

The size of the set $\mathcal{B}_{q,t,\ell}$ is

$$|\mathcal{B}_{q,t,\ell}| = B(q,t,\ell) = \sum_{(i_1,\ldots,i_\ell):\sum_{j=1}^{\ell} j\cdot i_j \leqslant q-1} \binom{t}{(t-\sum_{j=1}^{\ell} i_j), i_1, \ldots, i_\ell}.$$

To prove the theorem, we first show that $\mathcal{R}_{q,t,\ell} \leqslant \log B(q,t,\ell)$ by using the size of $\mathcal{B}_{q,t,\ell}$. Let $\mathcal{C}$ be an $[n,t,\ell;M_1,\ldots,M_t]_q$ q-ary t-write WℓM code. For any vector of t messages $(m_1,\ldots,m_t) \in [M_1]\times\cdots\times[M_t]$ that is written to the memory using the code $\mathcal{C}$, we assign a $t\times n$ matrix $A_{m_1,\ldots,m_t}$ such that every row in the matrix is the increase in the levels of all n cells. Note that every entry in the array is an integer in $[\ell+1]$. Moreover, the sum of all entries in each column of the array is at most $q-1$ since the highest level of each cell is $q-1$. Hence, every column in $A_{m_1,\ldots,m_t}$ is a vector in $\mathcal{B}_{q,t,\ell}$. Hence, there are at most $B(q,t,\ell)^n$ distinct possible arrays, so we deduce that $\mathcal{R}_{q,t,\ell} \leqslant \log B(q,t,\ell)$.

Next, it is shown that this upper bound is tight, that is, there is a WℓM code whose sum-rate achieves the upper bound. The proof holds by induction. It is straightforward to verify that it holds for $t=1$. Assume that it holds for $t-1$ and its correctness will be proved for t. From Theorem 4 and the induction assumption, there exists a WℓM code of sum-rate approaching

$$\begin{aligned}\sum_{j=1}^{t} R_j &= h(\boldsymbol{p}) + \sum_{j=1}^{t}\sum_{i=0}^{\ell} p_i \cdot R_j^i \\ &= \sum_{i=0}^{\ell}(p_i \cdot \log 1/p_i) + \sum_{i=0}^{\ell} p_i \cdot \mathcal{R}_{q-i,t-1,\ell} \\ &= \sum_{i=0}^{\ell} p_i \cdot (\log 1/p_i + \log B(q-1-i,t-1,\ell)).\end{aligned}$$

By choosing

$$p_i = \frac{B(q-i,t-1,\ell)}{B(q,t,\ell)}$$

we get

$$(\log 1/p_i + \log B(q-i,t-1,\ell)) = \log B(q,t,\ell)$$

for all $0 \leqslant i \leqslant \ell$. Hence,

$$\sum_{j=1}^{t} R_j = \sum_{i=0}^{\ell} p_i \cdot \log B(q,t,\ell) = \log B(q,t,\ell),$$

and the theorem is proven. ■

Remark 1. q-ary WOM codes could be viewed as a special case of q-ary WℓM codes when $\ell = q-1$. However, when $\ell = 1$, each time the cells are updated in WℓM, we only write a binary vector. In the case $\ell = 1$, we observe that the maximum sum-rate of q-ary t-write WℓM codes is $\mathcal{R}_{q,t,1} \approx (q-1)\log(t+1)$ when $t \gg q$. Comparing to the maximum sum-rate of binary t-write WOM codes, which is $\log(t+1)$, $\mathcal{R}_{q,t,1}$ is $(q-1)$ times larger. Furthermore, the maximum sum-rate of q-ary t-write WOM codes is $\log\binom{q-1+t}{q-1}$. We observe that $\log\binom{q-1+t}{q-1} \approx (q-1)\log(t+q-1) \approx \mathcal{R}_{q,t,1}$ when q is given and t tends to infinity.

III. Constructions

In this section, we present constructions of q-ary t-write WℓM codes. We study here the case of $q=3$, $t=3$ and $\ell=1$, while the extension to other parameters is left for future work. The *weight* of a binary vector $\boldsymbol{x} = (x_1,\ldots,x_n) \in [2]^n$ is defined to be $w(\boldsymbol{x}) = \sum_{i=1}^{n} x_i$ and its support set is denoted by $supp(\boldsymbol{x})$.

Let us first remind the definition of the *convergence rate* of rewriting codes, as was defined in [17]. The convergence rate of a construction of rewriting codes is the minimum length $n(\epsilon)$ in order to be ϵ-close to a rate tuple $(R_1,\ldots,R_t)$ or a sum-rate R. More specifically, it is said that a construction approaches the rate tuple or sum-rate with polynomial, exponential rate if $n(\epsilon)$ is polynomial, exponential in $1/\epsilon$, respectively.

Next, a special family of two-write binary WOM code, called *high-weight two-write binary WOM code*, is presented and will be used in our construction.

Definition 6. *An $[n,2;M_1,M_2]_2$ (w_1,w_2)-high-weight two-write binary WOM code is a coding scheme comprising of n binary bits. It consists of two pairs of encoding and decoding maps $(\mathcal{E}_1,\mathcal{D}_1)$ and $(\mathcal{E}_2,\mathcal{D}_2)$ which are defined as follows:*

(1) $\mathcal{E}_1 : [M_1] \mapsto [2]^n$ and $\mathcal{D}_1 : Im(\mathcal{E}_{q,1}) \mapsto [M_1]$ such that for all $m_1 \in [M_1]$, it holds that $\mathcal{E}_1(m_1) = \boldsymbol{c} \in [2]^n$ and $w(\boldsymbol{c}) = w_1$. Furthermore, $\mathcal{D}_1(\mathcal{E}_1(m_1)) = m_1$.

(2) $\mathcal{E}_2 : [M_2] \times Im(\mathcal{E}_1) \mapsto [2]^n$ and $\mathcal{D}_2 : Im(\mathcal{E}_2) \mapsto [M_2]$ such that for all $(m_2,\boldsymbol{c}) \in [M_2]\times Im(\mathcal{E}_1)$, it holds that $\mathcal{E}_2(m_2,\boldsymbol{c}) = \boldsymbol{c}' \geqslant \boldsymbol{c}$ and $w(\boldsymbol{c}') \geqslant w_2$. Furthermore, $\mathcal{D}_2(\mathcal{E}_2(m_2,\boldsymbol{c})) = m_2$.

In a similar way, (w_1,w_2)-*constant-weight two-write binary WOM codes* are defined if on the second write, $w(\boldsymbol{c}') = w_2$. Without the weight constraint on the two writes, we obtain the classical two-write binary WOM codes.

For $w_1 = (1-p_1)\cdot n$ and $w_2 = (1-p_1p_2)\cdot n$, where $0 \leqslant p_1,p_2 \leqslant 1$, it is possible to show that a rate tuple (R_1,R_2), where $R_1 = h(p_1)$ and $R_2 = p_1 \cdot h(p_2)$, is achievable. For example, a deterministic construction of these codes can be obtained using Shpilka's techniques [14], however with exponential convergence rate. This technique can also be extended for constant-weight two-write binary WOM codes with the same convergence rate.

High-weight two-write binary WOM codes will be an important component code in the following construction of three-write ternary WℓM code with $\ell = 1$.

Construction 7. *Given $p_1,p_2 \in [0,1]$, assume the following codes exist:*

- *Let $C_1(p_1)$ be an $[n,2,M_{1,p_1},M_{2,p_1}]_2$ two-write binary WOM code such that on the first write $w(\boldsymbol{c}) = p_1\cdot n$. The two pairs of encoding/decoding maps are $(\mathcal{E}_{1,p_1},\mathcal{D}_{1,p_1})$ and $(\mathcal{E}_{2,p_1},\mathcal{D}_{2,p_1})$.*
- *Let $C_2(p_1,p_2)$ be an $[n,2,M_{1,p_2}^{hr},M_{2,p_2}^{hr}]_2$ (w_1,w_2)-high-weight two-write binary WOM code such that $w_1 = (1-p_1)\cdot n$ and $w_2 = (1-p_1p_2)\cdot n$. The two pairs of encoding/decoding maps are $(\mathcal{E}_{1,p_2}^{hr},\mathcal{D}_{1,p_2}^{hr})$ and $(\mathcal{E}_{2,p_2}^{hr},\mathcal{D}_{2,p_2}^{hr})$.*

The proposed $[n,3,1;M_1,M_2,M_3]_3$ three-write ternary WℓM code is defined using the three pairs of encoding/decoding maps as follows.

- ***First write:** The idea is to encode a message as a codeword of length n with weight $w_1 = p_1\cdot n$. Hence, the pair of encoder/decoder on the first write of three-write ternary WℓM* *code is the same as the pair of encoder/decoder on the first write of two-write binary WOM code. That is,* $(\mathcal{E}_1, \mathcal{D}_1) = (\mathcal{E}_{1,p_1}, \mathcal{D}_{1,p_1})$. *So,* $M_1 = M_{1,p_1}$ *and the rate is* $R_1 = h(p_1)$.

- ***Second write:*** *Let* $\boldsymbol{c}_1 = (c_{1,1}, c_{1,2}, \ldots, c_{1,n})$ *be the cell-state vector after the first write and* $\overline{\boldsymbol{c}}_1 = \boldsymbol{c}'_1 = (c'_{1,1}, \ldots, c'_{1,n})$ *be its complement.*
Let $M_2 = M_{2,p_1} \cdot M^{hr}_{2,p_2}$. *For each* $m_2 \in [M_2]$, *we can determine the unique pair* (m_{2,p_1}, m'_{2,p_2}) *such that* $m_{2,p_1} \in [M_2, p_1]$ *and* $m'_{2,p_2} \in [M^{hr}_{2,p_2}]$. *Now, we are ready to define the encoder*
$$\mathcal{E}_2 : [M_2] \times Im(\mathcal{E}_1) \to [3]^n$$
on the second write. For each $(m_2, \boldsymbol{c}_1) \in [M_2] \times Im(\mathcal{E}_1)$, *we can determine* $\mathcal{E}_2(m_2, \boldsymbol{c}_1) = \boldsymbol{c}_2 = (c_{2,1}, c_{2,2}, \ldots, c_{2,n})$ *in two steps as follows.*
  - ***Step 1:*** *Let* $\boldsymbol{c}_1$ *be the input to the encoder* $\mathcal{E}_{2,p_1}$ *of the two-write binary WOM code* $C_1(p_1)$. *For each* $m_{2,p_1} \in [M_2, p_1]$, *we obtain* $\mathcal{E}_{2,p_1}(m_{2,p_1}, \boldsymbol{c}) = \boldsymbol{x} = (x_1, \ldots, x_n)$.
  For $1 \leqslant i \leqslant n$, *if* $c_{1,i} = 0$ *then* $c_{2,i} = x_i$.
  - ***Step 2:*** *Let* $\boldsymbol{c}'_1$ *be the input of the encoder* $\mathcal{E}^{hr}_{2,p_2}$ *of* (w_1, w_2)*-high-weight two-write binary WOM code* $C_2(p_1, p_2)$. *For each* $m'_{2,p_2} \in [M^{hr}_{2,p_2}]$, *we obtain* $\mathcal{E}^{hr}_{2,p_2}(m'_{2,p_2}, \boldsymbol{c}'_1) = \boldsymbol{y}' = (y'_1, \ldots, y'_n) \in [2]^n$. *We determine the vector* $\boldsymbol{y} = (y_1, \ldots, y_n) \in [3]^n$ *as follows. For all* $1 \leqslant i \leqslant n$, $y_i = 0$ *if* $c'_{1,i} = y'_i = 1$, $y_i = 2$ *if* $c'_{1,i} = y'_i = 0$, *and* $y_i = 1$ *otherwise.*
  For $1 \leqslant i \leqslant n$, *if* $c_{1,i} = 1$ *then* $c_{2,i} = y_i$.

We define the decoder
$$\mathcal{D}_2 : Im(\mathcal{E}_2) \to [M_2]$$
on the second write as follows.
For each $\boldsymbol{c}_2 = (c_{2,1}, \ldots, c_{2,n}) \in Im(\mathcal{E}_2)$, *we can determine* $\boldsymbol{y}' = (y'_1, \ldots, y'_n) \in [2]^n$ *such that* $y'_i = 0$ *if and only if* $c_{2,i} = 2$. *Using the decoder* $\mathcal{D}^{hr}_{2,p_2}$ *of the* (w_1, w_2)*-high-weight two-write binary WOM code* $C_2(p_1, p_2)$, *we obtain* $\mathcal{D}^{hr}_{2,p_2}(\boldsymbol{y}') = m'_{2,p_2}$. *Furthermore, we also can determine* $\boldsymbol{x} = (x_1, \ldots, x_n) \in [2]^n$ *such that* $x_i = 0$ *if and only if* $c_{2,i} = 0$. *Using the decoder* $\mathcal{D}_{2,p_1}$ *of the two-write binary WOM code* $C_1(p_1)$, *we obtain* $\mathcal{D}_{2,p_1}(\boldsymbol{x}) = m_{2,p_1}$. *From* m'_{2,p_2} *and* m_{2,p_1}, *we can determine the unique* m_2, *and decode by* $\mathcal{D}_2(\boldsymbol{c}_2) = m_2$.

- ***Third write:*** *Let* $\boldsymbol{c}_2 = (c_{2,1}, \ldots, c_{2,n})$ *be the cell-state vector after the second write. We define the binary vector* $\boldsymbol{c}'_2 = (c'_{2,1}, c'_{2,2}, \ldots, c'_{2,n}) \in [2]^n$ *such that* $c'_{2,i} = 1$ *if and only if* $c_{2,i} = 2$. *On the third write, we determine the encoder*
$$\mathcal{E}_3 : [M_3] \times Im(\mathcal{E}_2) \to [3]^n$$
as follows. Let $p_{1,2} = p_1 \cdot p_2$. *Let* $\mathcal{E}_{2,p_{1,2}}$ *be the encoder on the second write of the two-write binary WOM code* $C_1(p_{1,2})$. *For each* $m_3 \in [M_3]$, *we obtain* $\mathcal{E}_{2,p_{1,2}}(m_3, \boldsymbol{c}'_2) = \boldsymbol{z} = (z_1, \ldots, z_n)$. *So* $\mathcal{E}_3(m_3, \boldsymbol{c}_2) = \boldsymbol{c}_3 = (c_{3,1}, \ldots, c_{3,n})$ *can be defined such that* $c_{3,i} = 1$ *if* $z_i = 0$, $c_{3,i} = 0$ *if* $c_{2,i} = 0$ *and* $z_i = 1$, *and* $c_{3,i} = 2$ *otherwise.*
The corresponding decoder
$$\mathcal{D}_3 : Im(\mathcal{E}_3) \to [M_3]$$
on the third write can be defined as follows. For each $\boldsymbol{c}_3 \in Im(\mathcal{E}_3)$, *we determine* $\boldsymbol{z} = (z_1, \ldots, z_n) \in [2]^n$ *such that* $z_i = 0$ *if and only if* $c_{3,i} = 1$. *Using the decoder* $\mathcal{D}_{2,p_{1,2}}$ *of the two-write binary WOM code* $C_1(p_{1,2})$, *we obtain* $\mathcal{D}_3(\boldsymbol{c}_3) = \mathcal{D}_{2,p_{1,2}}(\boldsymbol{z}) = m_3$.

We observe that Construction 7 uses two-write binary WOM codes and high-weight two-write binary WOM codes as important component codes. An interesting question, which is addressed next, is whether this construction can provide codes achieving the maximum sum-rate.

Theorem 8. *If there exists an explicit construction of* (w_1, w_2)*-high-weight two-write binary WOM codes which achieve the rate-tuple* $(R_1, R_2) = (h(p_1), p_1 h(p_2))$ *for any given* $w_1 = (1 - p_1) \cdot n$ *and* $w_2 = (1 - p_1 p_2) \cdot n$, *then there exists an explicit construction of three-write ternary W*ℓ*M codes which achieves the rate-tuple* $(R'_1, R'_2, R'_3) = (h(p_1), 1 - p_1 + p_1 h(p_2), 1 - p_1 p_2)$. *In particular, there exists an explicit construction of three-write ternary W*ℓ*M codes for* $\ell = 1$ *which achieves the maximum sum-rate* $\mathcal{R}_{3,3,1} = \log 7$.

Proof: We note that there exists an explicit construction of two-write binary WOM codes which achieves the rate-tuple $(h(p_1), 1 - p_1)$ for any given $p_1 \in [0, 1/2]$ [17]. Assume that there exists an explicit construction of (w_1, w_2)-high-weight two-write binary WOM codes which achieves the rate-tuple $(h(p_1), p_1 h(p_2))$. In Construction 7, the rate on the first write $R'_1 = h(p_1)$. On Step 1, Step 2 of the second write the rate is $R_{2,1} = \frac{\log M_{2,p_1}}{n} = 1 - p_1$, $R_{2,2} = \frac{\log M^{hr}_{2,p_2}}{n} = p_1 h(p_2)$, respectively. So, the rate on the second write is $R'_2 = R_{2,1} + R_{2,2} = 1 - p_1 + p_1 h(p_2)$. On the third write, the rate is $R'_3 = 1 - p_1 p_2$. Hence, the constructed three-write ternary WℓM codes achieve the rate-tuple $(R'_1, R'_2, R'_3) = (h(p_1), 1 - p_1 + p_1 h(p_2), 1 - p_1 p_2)$, and sum-rate $R_{sum} = R_1 + R_2 + R_3 = h(p_1) + 1 - p_1 + p_1 h(p_2) + 1 - p_1 p_2$. For $p_1 = 3/7$ and $p_2 = 1/3$ we get the maximum sum-rate $\mathcal{R}_{3,3,1} = \log 7$.

Lastly, we observe that in all three writes of the WℓM codes from Construction 7, every step is explicit and is based on two component codes, two-write binary WOM codes and (w_1, w_2)-high-weight two-write binary WOM codes. Therefore, the theorem is proven. ■

It is now possible to conclude with the following corollary.

Corollary 9. *There exists an explicit construction of three-write ternary W*ℓ*M codes for* $\ell = 1$ *which achieves the maximum sum-rate* $\mathcal{R}_{3,3,1} = \log 7$.

The convergence rate of the codes achieving the maximum sum-rate in Corollary 9 is exponential. This follows since they require the construction of (w_1, w_2)-high-weight two-write binary WOM codes and our best construction is based on the techniques from [14], which also have exponential convergence rate. In fact, it is possible to directly construct WℓM codes by the techniques from [14] with exponential convergence rate. However, Construction 7 is beneficial since we believe that finding high-weight two-write binary WOM codes with polynomial convergence rate will be an easier task, and furthermore it enables us to present practical WℓM codes of short block length but yet achieve high sum-rate. This will be accomplished by explicit constructions of (w_1, w_2)-high-weight two-write binary WOM codes.

To do so, we need the following definition from [26], [27].

Definition 10. *For* n, t *and* w *with* $t + w \leqslant n$, *an* (n, t, w)*- low-power cooling (LPC) code C of size* M *is defined as a collection* *of code sets $\{C_1, C_2, \ldots, C_M\}$, where $C_1, C_2, \ldots, C_M$ are disjoint subsets of $\{\boldsymbol{u} \in [2]^n : w(\boldsymbol{u}) \leqslant w\}$ satisfying the following property: for any set $S \subseteq [n]$ of size $|S| = t$ and for $i \in [M]$, there exists a vector $\boldsymbol{u} \in C_i$ with $supp(\boldsymbol{u}) \cap S = \emptyset$.*

From this definition, we obtain the following result. The proof is derived from the definitions of low-power cooling codes and (w_1, w_2)-high-weight two-write binary WOM code. We omit the details of the proof due to the lack of space.

Theorem 11. *Given n, w_1, w_2 such that $w_1 < w_2 < n$, if there exists an $(n, w_1, n - w_2)$-LPC code of size M, then there exists an $[n, 2; M_1, M_2]_2$ (w_1, w_2)-high-weight two-write binary WOM code such that $M_1 = \binom{n}{w_1}$ and $M_2 = M$.*

Recently, LPC codes have been investigated and a few explicit constructions of LPC codes were presented in [26], [27]. We use one of these families of LPC codes to construct an explicit high-weight two-write binary WOM code and thus an explicit three-write ternary WℓM code with $\ell = 1$. Following is an example of asymptotically optimal LPC codes.

Example 1. [26, Corollary 18] Fix $\tau \leqslant 0.687$ and $\omega = (1 - \tau)/2$. Then there exists a family of (n, w_1, w_2)-LPC codes C_n such that $w_1 = \lfloor \tau n \rfloor, w_2 = \lfloor \omega n \rfloor$, and $\lim_{n\to\infty}(\log |C_n|)/n = 1 - \tau$. In other words, the rate of the codes converges to $1 - \tau$. □

From Theorem 11 and Example 1, we obtain the following.

Corollary 12. *For all $0 \leqslant p_1 \leqslant 0.687$, there exists a family of $[n, 2, M_1, M_2]_2$ (w_1, w_2)-high-weight two-write binary WOM codes for $w_1 = p_1 n$ and $w_2 = (n - w_1)/2$, which achieves sum-rate $R = h(p_1) + 1 - p_1$ with polynomial convergence rate.*

If we choose the above $[n, 2, M_1, M_2]_2$ (w_1, w_2)-high-weight two-write binary WOM codes as a component code in Construction 7 with $p_1 = 2 - \sqrt{2} \approx 0.5858$ then the sum-rate of our constructed three-write ternary WℓM codes with $\ell = 1$ approaches $h(p_1) + (1 - p_1) + p_1 + 1 - (1 - p_1)/2 \approx 2.772$. We conclude our result in the following corollary.

Corollary 13. *There exists an explicit construction of three-write ternary WℓM codes of sum-rate approaching 2.772 with polynomial convergence rate.*

As part of our future work, we will extend our technique in Construction 7 to construct q-ary t-write WℓM code for all q, t, and ℓ. Furthermore, these ideas can be leveraged in order to construct the classical q-ary t-write WOM codes achieving maximum sum-rate in general for all q and t. Namely, we established the following theorem.

Theorem 14. *If there exists an explicit construction of constant-weight two-write binary WOM code which achieves the rate-tuple $(h(p_1), p_1 h(p_2))$ for any given parameter $p_1, p_2 \in [0, 1]$ then there exists an explicit construction of q-ary t-write WOM code which achieves the maximum sum-rate.*

Hence, the problem of finding non-binary WOM codes which achieve the maximum sum-rate with polynomial convergence rate can be reduced to the problem of finding constant-weight two-write binary WOM codes with the same property. Due to the lack of space, these results will be discussed in the full version of this work.

IV. Acknowledgement

The research of Y. M. Chee, H. M. Kiah and V. K. Vu is partially funded by the Singapore Ministry of Education under grant MOE2015-T2-2-086. The research of E. Yaakobi is supported in part by the German-Israeli Foundation for Scientific Research and Development (GIF) grant I-1356-407.6/2016.

References

[1] 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.

[2] A. Fiat and A. Shamir, "Generalized write-once memories," *IEEE Trans. Inform. Theory,* vol. IT-30, pp. 470–480, May 1984.

[3] 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.

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

[5] G.D. Cohen, P. Godlewski, and F. Merkx, "Linear binary code for write-once memories," *IEEE Trans. Inform. Theory,* vol. 32, no. 5, pp. 697–700, Oct. 1986.

[6] 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.

[7] F.W. Fu, A.J.H. Vinck, V.K. Wei, and R.W. Yeung, "On the capacity of write uni-directional memories with non-periodic codes," *IEEE Trans. Inform. Theory*, vol. 50, no. 4, pp. 649–656, 2004.

[8] G.D. Cohen and G. Zémor, "Write isolated memories (WIMs)," *Discrete Mathematics*, vol. 114, pp. 105–113, 1993.

[9] A.V. Kuznetsov and B.S. Tsybakov, "Coding in a memory with defective cell," *Probl. Inform. Trans.,* vol. 10, pp. 132–138, 1974.

[10] A.V. Kuznetsov and A.J.H. Vinck, "On the general defective channel with informed encoder and capacities of some constrained memories," *IEEE Trans. Inform. Theory,* vol. 40, pp. 1866–1871, Nov. 1994.

[11] A. Jiang, "On the generalization of error-correcting WOM codes," *Proc. IEEE Int. Symp. Inform. Theory,* pp. 1391–1395, France, June 2007.

[12] E. Yaakobi, P.H. Siegel, A. Vardy, and J.K. Wolf, "Multiple error-correcting WOM-codes," *Proc. IEEE Int. Symp. Inform. Theory,* pp. 1933–1937, USA, 2010.

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

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

[15] R. Gabrys and L. Dolecek, "Constructions of nonbinary WOM codes for multilevel flash memories," *IEEE Trans. Inform. Theory,* vol. 61, no. 4, pp. 1905–1919, 2015.

[16] 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.

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

[18] F. Masuoka, M. Asano, H. Iwahashi, T. Komuro, and S. Tanaka, "A new flash E2PROM cell using triple polysilicon technology," *Proc. Int. Electron Devices Meeting*, pp. 464–467, 1984.

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

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

[21] G. Zémor and G.D. Cohen, "Error-correcting WOM codes," *IEEE Trans. Inform. Theory,* vol. 37, no. 3, pp. 730–734, May 1991.

[22] A. Solomon and Y. Cassuto, "Error-correcting WOM constructions through concatenation and joint design," *Proc. IEEE Int. Symp. Inform. Theory,* pp. 1660–1664, Vail, CO, Jun 2018.

[23] Y. M. Chee, M. Horovitz, A. Vardy, V. K. Vu, and E. Yaakobi, "Codes for endurance-limited memories," *Int. Symp. Inform. Theory and Its Applications,* Singapore, Oct. 2018.

[24] T. Kobayashi, H. Morita, and A. Manada, "On the capacity of write-constrained memories," *IEEE Trans. Inform. Theory,* vol. 64, no. 7, pp. 5101–5109, Jul. 2018.

[25] R. Gabrys, E. Yaakobi, L. Dolecek, P.H. Siegel, A. Vardy, and J.K. Wolf, "Non-binary WOM-codes for multilevel flash memories," *IEEE Inform. Theory Workshop,* pp. 40–44, Brazil, Oct. 2011.

[26] Y. M. Chee, T. Etzion, H. M. Kiah, and A. Vardy, "Cooling codes: Thermal-management coding for high-performance interconnects," *IEEE Trans. Inform. Theory*, vol. 64, no. 4, Apr. 2018.

[27] Y. M. Chee, T. Etzion, H. M. Kiah, A. Vardy, and H. Wei, "Low-power cooling codes with efficient encoding and decoding," *Proc. IEEE Int. Symp. Inform. Theory,* pp. 1655–1659, Vail, CO, 2018.