Covering codes for Multilevel Flash Memories

Kathryn Haymaker
Department of Mathematics
University of Nebraska-Lincoln
Lincoln, NE 68588
Email: s-khaymak1@math.unl.edu

Christine A. Kelley
Department of Mathematics
University of Nebraska-Lincoln
Lincoln, NE 68588
Email: ckelley2@math.unl.edu

***Abstract*—Write-once-memory (WOM) and flash codes are used to increase the number of writes in flash memories in order to improve the lifetime of flash-based storage systems. An early construction method of binary WOM codes used cosets of a binary linear code in the writing process, and the covering radius of the code was used to determine the number of writes possible. In this paper, we show how to combine this method with nonbinary codes for multilevel flash cells, and introduce a new family of ternary WOM codes using the finite Euclidean geometry $EG(m,3)$.**

I. Introduction

Flash memory is a nonvolatile memory device that is composed of blocks of cells. Each cell contains a charge that conveys information, and recent advances have allowed for a single cell to store multiple bits. The process of rewriting on the memory carries varying costs in terms of reliability of the device. In particular, increasing the level of a cell is less costly than decreasing a cell's level, which requires the entire block containing that cell to be erased and reprogrammed. Rewriting codes have been developed in the interest of delaying the erase operation. Such codes in the binary case share features with write once memory (WOM) codes, which were originally designed for applications such as punch cards and optical disks.

In this paper we investigate how to design codes for multilevel flash memories. The main results are two construction methods that combine the coset encoding scheme with nonbinary WOM codes to obtain codes for q-ary flash cells, and a new construction of ternary WOM codes from finite Euclidean geometries. The construction methods have similarities to concatenation but use covering codes for the outer code, and nonbinary WOM codes for the inner code. We show how to apply the coset encoding scheme of [1] and rewrite on the components, and detail the two-step decoding process. We illustrate our construction methods with several examples, and discuss advantages and disadvantages of these constructions.

The paper is organized as follows. We introduce some preliminary definitions and notation in Section II, and summarize current methods for nonbinary WOM code design. In Section III, we present two construction methods using the coset encoding scheme with nonbinary WOM codes. In Section IV, we give a new construction of ternary WOM codes from $EG(m,3)$ and analyze their performance. Section V contains examples of codes resulting from the constructions, and Section VI addresses their error correction capabilities. Section VII concludes the paper.

II. Preliminaries

A. Definitions

A WOM code is a writing scheme that allows different messages to be stored in the same n cells over various time-steps (or "generations"). Thus, a WOM code is characterized by a pair of encoding and decoding maps where the encoding process ("writing") ensures that the levels of the memory cells are non-decreasing. The encoding map takes as input the current state of the memory $\mathbf{c}$ and the new information $\mathbf{v}$ to be represented, and outputs a new *memory state vector* that is component-wise greater than or equal to $\mathbf{c}$, and decodes to the information $\mathbf{v}$. Denote by $\{n,m,t\}_q$ a WOM code that allows t successive writings of m messages on n q-ary cells. Such a code has *sum-rate*

$$R = \frac{\log_2(m^t)}{n},$$

and is called a *fixed-rate WOM code*, since the amount of information that can be represented is constant over all writes. A code that can represent a different amount of information at each write is called a *variable rate WOM code*, and the sum rate is defined analogously, see e.g. [2].

An $[n,k,d]$ linear block code $\mathcal{C}$ has length n, dimension k, and minimum distance d. An $[n,k,d]$ code $\mathcal{C}$ is *maximal* if $\mathcal{C}$ is not a subcode of a code with the same distance. The *covering radius* of a code $\mathcal{C}$ is

the smallest number $R(\mathcal{C})$ such that spheres of radius R centered at the codewords cover the space. Equivalently, $\mathcal{C}$ is maximal if $R(\mathcal{C}) \leq (d-1)$. Throughout the paper we will use $\mathbf{1}$ to denote the all ones vector and $\mathbf{0}$ to denote the all zeros vector.

B. Related work

Although WOM codes were first introduced in the early 1980s [3], interest in these rewriting schemes continued through the 90s [4], and was renewed in 2007 due to the notable link to flash memory applications observed by Jiang [5]. Since that time, many new constructions of binary and nonbinary WOM codes have been proposed, including some capacity-achieving schemes [6]. We will focus on works related to nonbinary WOM codes [7] and binary WOM codes inspired by the coset encoding scheme proposed by Cohen et al. in 1986 [1]. This scheme is closely related to writing on defective memories [8], where the positions with ones are regarded as 'defects' for the second write. We begin by reviewing the key ideas of coset encoding.

C. Binary coset encoding

Let $\mathcal{C}$ be an $[N, K, D]$ binary linear code with covering radius R. Using $\mathcal{C}$, we will encode $(N-K)$ bits on N cells. The messages are associated with syndromes of $\mathcal{C}$, so there are $2^{(N-K)}$ messages on each write. The process is as follows:

1) To encode $\mathbf{s_1} \in \mathbb{F}_2^{(N-K)}$, write a minimum weight vector $\mathbf{y_1} \in \mathbb{F}_2^N$ with syndrome $\mathbf{s_1}$.

2) To encode the next message $\mathbf{s_2}$, find $\mathbf{y_2}$ such that $\mathbf{y_2} + \mathbf{y_1}$ has minimum weight with syndrome $\mathbf{s_2}$ and the support of $\mathbf{y_2}$ and $\mathbf{y_1}$ are disjoint.

3) Repeat this process until the encoding of a new message is no longer possible.

The main result of this method, to be stated next, shows that when $\mathcal{C}$ is a maximal code and satisfies other maximal conditions on shortened versions, then the resulting WOM code guarantees T writes of $(N-K)$ bits, where T is an expression in terms of the minimum distance and covering radius of $\mathcal{C}$, and the number of shortened versions retaining maximality. Specifically, the authors present the following theorem.

Theorem 2.1: [1] Let $\mathcal{C}$ be an $[n, k, d]R$ maximal code. If for some i with $i \leq d^{\perp}$-1,[1] its shortened versions of lengths at least $(n-i)$ remain maximal and of minimum distance d, then at least T writings of $(n-k)$ bits are guaranteed with $T = 2 + \lfloor (i-R)/(d-1) \rfloor$.

[1]Here, $d^{\perp}$ denotes the minimum distance of the dual code of $\mathcal{C}$.

Due to the interest in nonbinary WOM codes, it is natural to ask how the coset encoding scheme works when applied to a nonbinary covering code. In this case, the rewriting capabilities come from the process of making disjoint subsets of the cells nonzero on each write, but once a cell has been programmed, it will never be reprogrammed. Thus, a major advantage of a multilevel memory—that cell levels may be increased multiple times—is not utilized. This drawback is also identified in [2], where nonbinary coset encoding is used to create efficient *binary* WOM codes. These authors and, separately, Wu [9] construct binary WOM codes using ideas similar to coset encoding on the second write of two-write constructions. These modifications avoid the maximality restrictions on the error-correcting code that make coset encoding difficult to apply to an arbitrary code. A different application of covering codes was used in the context of floating codes in [10]. The goal in that application was to obtain bounds on floating codes for large-alphabet messages using existing bounds on floating codes for small-alphabet messages.

III. Constructions

In this section, we present two methods of combining a covering code and a nonbinary WOM code to obtain a nonbinary rewriting code. The constructions are similar to concatenation, where the outer covering code is encoded and decoded using the coset encoding scheme, and the inner code uses encoding and decoding rules of a nonbinary WOM code.

A. Construction I

Let $\mathcal{C}$ be an $[N, K, d]$ binary linear code with covering radius R, and suppose $\mathcal{C}$ is maximal. Suppose with coset encoding, $\mathcal{C}$ produces an $\{N, M, T\}_2$ binary WOM code where $M = 2^{(N-K)}$. Let $\mathcal{W}$ be an $\{n, m, t\}_q$ WOM code on q-ary cells. For both constructions, assume that codewords in $\mathcal{W}$ have distinguishable generations, and that the all-zeros word is not a codeword in $\mathcal{W}$.

We construct a length Nn q-ary WOM code that guarantees Tt writes as follows. View the Nn cells of the memory state vector as N groups of n cells. During the writing process, each group will be in a state of "active", "active saturated", or "inactive". Let x_{ij} denote the number of groups activated on the $((i-1)T + j)$th write[2]. An "i-saturated vector" will be a word of length n that is not in $\mathcal{W}$, and is "less than" any word in generation $i+1$ of $\mathcal{W}$. Given a codeword $\mathbf{c} \in \mathcal{W}$ such that $\mathbf{c}$ is in generation i, the existence of

[2]This corresponds to the jth write of the outer code in the ith iteration, so the inner codewords at this stage will be generation i.

an i-saturated vector $\mathbf{w}^i$ such that $\mathbf{c} < \mathbf{w}^i$ will be a requirement in the first construction detailed next.

Encoding:

1) Given one of $M = 2^{(N-K)}$ messages, coset-encoding using $\mathcal{C}$ produces a length N binary word to be viewed as an indicator vector for which groups will be activated. One of m symbols can now be written on each active group using $\mathcal{W}$. The first write can store $2^{(N-K)}m^{x_{11}}$ messages, where x_{11} is the weight of the memory state vector corresponding to the syndrome message of the outer covering code, and thus represents the number of groups activated in the first write.

2) On writes two through T, first write a 1-saturated vector on each of the active groups. Use the outer code to encode one of M messages—such an encoding activates at least one more group. Write one of m messages from $\mathcal{W}$ on each new active group. On each of these writes, $2^{(N-K)}m^{x_{1j}}$ messages can be represented in total, where x_{1j} is the number of new groups activated during the jth write.

3) For $i = 1, \ldots, t-1$, on the $(iT+1)$ write, first write an iT-saturated vector on any inactive or active groups remaining after the (iT)th write, and call all groups "inactive". As before, any of M messages may be stored by indicating a new set of active groups, and each new "active" group can store one of m messages using a generation $i+1$ word from $\mathcal{W}$. Write $(iT+1)$ can represent $2^{(N-K)}m^{x_{i+1,1}}$ messages.

4) For $i = 1, \ldots, t-1$, for writes $(iT+2)$ to $(iT+T)$, continue in the same way as Step 2, where at the end of each write, each active group is saturated using an appropriate i-saturated vector, and at the end of the $(iT+T)$th write, all groups have an i-saturated vector. On write $(iT+j)$, for $j = 2, 3, \ldots, T$, a total of $2^{(N-K)}m^{x_{i+1,j}}$ messages can be represented.

5) Writing stops once the (Tt)th write is complete.

Decoding:

1) Create an indicator vector $\mathbf{y} \in \mathbb{F}_2^N$ that has $y_i = 1$ if group i is active or active saturated, and $y_i = 0$ if group i is inactive. The inner codewords in the active groups have generation j words from $\mathcal{W}$ for some j, while the active saturated and inactive groups have j-saturated and $(j-1)$-saturated vectors, respectively. Compute the syndrome of $\mathbf{y}$ to reveal the message M that was stored by the outer code.

2) For any group that is active but not saturated, decode the corresponding inner WOM codeword using $\mathcal{W}$.

Remark 3.1: In order to be able to encode the maximum number of messages on write $(iT+j)$, knowledge of x_{ij} should be known beforehand. Thus in most cases, one can only assume $x_{ij} \geq 1$ and encode $2^{(N-K)}m$ messages on an arbitrary write, which gives the lower guaranteed rate in the result below. However, the more general rate is also provided in case knowledge of the coset structure and other information on the outer code messages is available.

The following sum-rate is achieved from the construction.

Theorem 3.2: This method produces a q-ary Tt write WOM code with length Nn and sum-rate

$$r \geq \frac{Tt(N-K) + (\sum x_{ij}) \log_2(m)}{Nn}$$

where the sum is over $i = 1, \ldots, t$ and $j = 1, \ldots, T$. Note that $Tt \leq \sum x_{ij} \leq Nt$, and so in the worst case, assuming just one activated group per write,

$$r = \frac{Tt(N-K) + (Tt) \log_2(m)}{Nn}.$$

B. Construction II

Let $\mathcal{C}$ be an $[N, K, d]$ binary linear code that with coset encoding produces an $\{N, M, T\}_2$ binary WOM code where $M = 2^{(N-K)}$. Let $\mathcal{W}$ be an $\{n, m, t\}_q$ WOM code on q-ary cells with distinguishable codeword generations, and assume that the all-zeros word is not a codeword in $\mathcal{W}$.

We construct a length Nn q-ary WOM code that guarantees $T + t - 1$ writes as follows. View the Nn cells of the memory state vector as N groups of n cells. During the writing process, each group will be in a state of "active", "active saturated", or "inactive". Let x_i denote the number of new groups activated on the ith write, for $i = 1, \ldots, T$, and let $\mathbf{w}$ denote the all-$(q-1)$ vector, called the saturated vector. For convenience, define $x_i = 0$ for $i \leq 0$.

Encoding:

1) Same as Step 1 in Construction I. Given one of M messages, coset encoding using $\mathcal{C}$ produces a length N indicator vector for which groups will be activated. One of m symbols can now be written on each active group using a generation one codeword of $\mathcal{W}$. The first write can store Mm^{x_1} messages.

2) On writes i, where $i = 2, \ldots, T$, use the outer code to encode one of M messages. If the current memory state vector of a group is a generation j codeword in $\mathcal{W}$ where $j < t$, one of m messages may be stored using a generation $j+1$ codeword of $\mathcal{W}$. If the current state of a group is a generation t codeword, write the saturated vector $\mathbf{w}$ on that group. For any new groups activated by the outer code, write one of m messages from $\mathcal{W}$ using a generation one codeword. In total, write i can store $Mm^{(x_{i-t+1}+\cdots+x_i)}$ possible messages.

3) Once T writes have been completed, the outer code can no longer be used. On write $T + t - i$ where $i = 1, \ldots (t-1)$, write one of m messages on each of the $x_{T-i+1} + \cdots + x_T$ active groups remaining that

have current states not in generation t, and saturate any generation t groups using $\mathbf{w}$.

Decoding:
1) Check if there are any generation one codewords of $\mathcal{W}$ in any of the groups. If so, compute the indicator vector of all active and saturated groups, and decode the outer code message using $\mathcal{C}$. If there are no generation one inner codewords, then the message does not contain any outer code message.
2) Among the active groups that are not saturated, decode the inner codewords using $\mathcal{W}$.

The following sum-rate is achieved from the construction.

Theorem 3.3: This method produces a q-ary $T+t-1$ write WOM code of length Nn and sum-rate

$$r \geq \frac{T(N-K) + t\sum_i (x_i)\log_2(m)}{Nn}.$$

In the worst case, assuming just one new activated group per outer code write yields

$$r = \frac{T(N-K) + (Tt)\log_2(m)}{Nn}.$$

An advantage of Construction II over Construction I is the relaxed conditions on the inner WOM code. While the rate and number of writes is inferior, most WOMs can be used as the inner code in Construction II.

Remark 3.4: In this construction, the inactive groups remaining after the T writes of the outer code are never used. Alternatively, an indicator cell may be added to the memory state vector to indicate when the first T writes are complete. This would allow additional messages to be written on those groups using $\mathcal{W}$ on writes $T+1$ through $T+t$.

IV. Ternary WOM codes from $EG(m,3)$

Consider the finite Euclidean geometry of dimension m over $\mathbb{F}_3$, denoted $EG(m,3)$. This incidence structure consists of 3^m points and $\frac{3^m(3^m-1)}{6}$ lines. Each line contains three points, and every point lies on $\frac{3^m-1}{2}$ lines. Given two points $\mathbf{a},\mathbf{b}$, there is a unique line that contains these points, and a unique third point $\mathbf{c}$ that lies on that line. In the following, the vector notation $\mathbf{x} < \mathbf{y}$ means that $x_i \leq y_i$ in every component.

A. Encoding and decoding

We construct a $\{2m, 3^m, 2\}_3$ WOM code from $EG(m,3)$. Each of the 3^m messages is represented by a point in $\mathbb{F}_3^m$. The memory state vector $\mathbf{c}$ will have $2m$ cells. For convenience, we organize the memory state vector in the form $\mathbf{c} = (\mathbf{a},\mathbf{b})$ where $\mathbf{a},\mathbf{b} \in \mathbb{F}_3^m$. Assume the memory cells are each initialized at 0, i.e., the current memory state vector is $\mathbf{c} = (\mathbf{0},\mathbf{0})$. Each cell has $q = 3$ levels, which we will denote with symbols $0, 1$, and 2.

Encoding:
1) First Write: Given message $\mathbf{v} = (v_0,\ldots,v_{m-1}) \in \mathbb{F}_3^m$, if the largest component of $\mathbf{v}$ is one, then set $\mathbf{c} = (\mathbf{v},\mathbf{0})$. Otherwise, find the point $\mathbf{y}$ that shares a line with $\mathbf{v}$ and the point $\mathbf{0} = (0,0)$. If $\mathbf{y}$ has largest component one, set $\mathbf{c} = (\mathbf{0},\mathbf{y})$. Otherwise, choose two points $\mathbf{a},\mathbf{b}$ that form a line with $\mathbf{v}$ where each has largest component one, and set $\mathbf{c} = (\mathbf{a},\mathbf{b})$.

2) Subsequent Writes: Let $\mathbf{c} = (\mathbf{a},\mathbf{b})$ be the current memory state vector, and suppose the message $\mathbf{v}' = (v_0', v_1', \ldots, v_{m-1}')$ is to be stored.

- If $\mathbf{b} = \mathbf{0}$: If $\mathbf{a} < \mathbf{v}'$, set $\mathbf{c} = (\mathbf{v}',\mathbf{0})$. If $\mathbf{a} \not< \mathbf{v}'$, set $\mathbf{c} = (\mathbf{a},\mathbf{b}')$ where $\mathbf{b}'$ is the third point on the unique line containing $\mathbf{v}'$ and $\mathbf{a}$.
- If $\mathbf{a} = \mathbf{0}$ and $\mathbf{b} \neq \mathbf{0}$: Consider the vector $\mathbf{w} = (w_0, w_1, \ldots, w_{m-1})$ where $w_i + v_i \equiv 0 \pmod 3$ for $i = 0, 1, \ldots, m-1$. If $\mathbf{b} < \mathbf{w}$, then set $\mathbf{c} = (\mathbf{0},\mathbf{w})$. If $\mathbf{b} \not< \mathbf{w}$, then set $\mathbf{c} = (\mathbf{a}',\mathbf{b})$ where $\mathbf{a}'$ is the third point on the unique line containing $\mathbf{v}'$ and $\mathbf{b}$.
- If $\mathbf{a} \neq \mathbf{0}$ and $\mathbf{b} \neq \mathbf{0}$: Let $\mathbf{y}$ be the third point on the unique line containing $\mathbf{v}'$ and $\mathbf{a}$, and let $\mathbf{x}$ be the third point on the unique line containing $\mathbf{v}'$ and $\mathbf{b}$. If $\mathbf{b} < \mathbf{y}$ and $\mathbf{a} \not< \mathbf{x}$, set $\mathbf{c} = (\mathbf{a},\mathbf{y})$. If $\mathbf{a} < \mathbf{x}$ and $\mathbf{b} \not< \mathbf{y}$ set $\mathbf{c} = (\mathbf{x},\mathbf{b})$. If both $\mathbf{b} < \mathbf{y}$ and $\mathbf{a} < \mathbf{x}$, then choose a vector in $\{(\mathbf{a},\mathbf{y}),(\mathbf{x},\mathbf{b})\}$ that results in fewer cell increases.
  If none of the above, then consider the $\frac{3^m-1}{2} - 2$ lines incident with $\mathbf{v}'$ that do not contain $\mathbf{a}$ or $\mathbf{b}$. Suppose the ith line consists of points $\{\mathbf{v}', \mathbf{w^i}, \mathbf{z^i}\}$ for $i = 1, \ldots, \left(\frac{3^m-1}{2} - 2\right)$. Let $\mathcal{J} = \{i | (\mathbf{a},\mathbf{b}) < (\mathbf{w^i},\mathbf{z^i})\}$. If $\mathcal{J} \neq \emptyset$ then set $\mathbf{c} = (\mathbf{w^i},\mathbf{z^i})$ for some $i \in \mathcal{J}$ for which the vector $(\mathbf{w^i},\mathbf{z^i})$ has minimum weight. If such a vector does not exist, then the message $\mathbf{v}'$ can not be written.

Note that two writes are guaranteed because the weight of the memory state vector is low after the first write.

Decoding:
If the memory state vector is

$$\mathbf{c} = (a_0, a_1, \ldots, a_{m-1}, b_0, b_1, \ldots, b_{m-1}),$$

then when $b_0 = b_1 = \cdots = b_{m-1} = 0$, decode to the point $(a_0, a_1, \ldots, a_{m-1})$. Otherwise, $\mathbf{c}$ decode to the point $(v_0, v_1, \ldots, v_{m-1})$ such that $v_i + a_i + b_i \equiv 0 \pmod 3$ for $i = 0, 1, \ldots, m-1$.

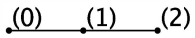

Fig. 1. $EG(1,3)$

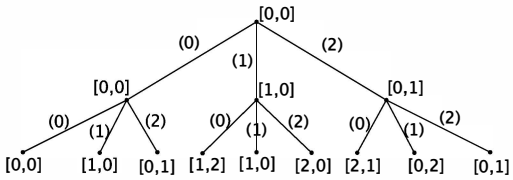

Fig. 2. $EG(1,3)$ WOM code, where the ith layer of the tree corresponds to the ith write.

B. Examples

Example 4.1: Consider $EG(1,3)$, consisting of one line and three points, as shown in Figure 1. The corresponding WOM code has parameters $\{2,3,2\}_3$ and rate $\frac{2\log_2(3)}{2}$. The code based on $EG(1,3)$ has a simple encoding map, shown by the tree in Figure 2. Note that more than two writes are possible in some cases, but we only demonstrate the guaranteed writes in the figure.

Example 4.2: Consider $EG(2,3)$, consisting of nine points and 12 lines in Figure 3. We construct a $\{4,9,2\}_3$ WOM code from $EG(2,3)$. Each of the 9 messages is represented by a point. The memory state vector $\mathbf{c}$ will have four cells, denoted $(\mathbf{a},\mathbf{b})$ where $\mathbf{a},\mathbf{b}\in\mathbb{F}_3^2$. While the code guarantees two writes, often more are possible.

The following is an example message sequence that can obtain five writes:

Info.	(01)	(22)	(21)	(00)	(02)
Writes	[0,1,0,0]	[0,1,1,0]	[0,2,1,0]	[1,2,2,1]	[1,2,2,2]

In contrast, the message sequence below yields two writes:

Info.	(12)	(20)
Writes	[1, 0, 1, 1]	[2, 1, 2, 2]

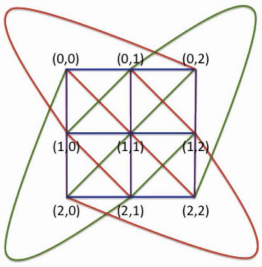

Fig. 3. $EG(2,3)$.

When viewed as a variable-rate WOM code, the $EG(2,3)$ code obtains 3.108 writes, on average. This average was obtained by running 10^6 random message sequences in MatLab and averaging over the number of writes achieved. If we restrict certain bad message sequences such as that above, more writes may be guaranteed, but at a significant cost in the number of messages that can be represented at each generation. Consider the trivial scheme of representing some nonzero message $\mathbf{v}$ on the first write using $(\mathbf{v},\mathbf{0})$, and a nonzero message $\mathbf{v}'$ on the second write using $(\mathbf{v},\mathbf{v}')$. Of course, the $EG(2,3)$ construction only does better than this scheme when it attains three or more writes.

Example 4.3: Consider $EG(3,3)$, consisting of 27 points and 117 lines. The corresponding WOM code has parameters $\{6,27,2\}_3$ and rate $\frac{2\log_2(27)}{6}$.

V. Examples

A. Inner nonbinary WOM codes

Constructions I and II each require specific features of the inner WOM code. The following is an example of a nonbinary WOM code that can be used as the inner code in Construction I, which requires a saturated state between each generation of writes. It is a variation of the simple q-ary t-write WOM code presented in [7] and later in [11]. The idea for two writes is to partition the alphabet for each cell into two groups, $\{0,1,\ldots,\lfloor\frac{q}{2}\rfloor\}$ and $\{\lfloor\frac{q}{2}\rfloor+1,\ldots,(q-1)\}$. On the first write the cells only take values from the first group, while on the second write the cell values all come from the second group, which results in an increase of all cell levels from write one to write two. The variation we provide is to reserve the word $\mathbf{0}$ for the inactive state, the word $\lfloor\frac{q}{2}\rfloor\mathbf{1}$ for the 1-saturated state, and the word $(q-1)\mathbf{1}$ is reserved for the 2-saturated state.

$$\underbrace{0,1,\ldots,\lfloor\frac{q}{2}\rfloor-1,\lfloor\frac{q}{2}\rfloor}_{\text{write one alphabet}}$$

$$\underbrace{\lfloor\frac{q}{2}\rfloor,\lfloor\frac{q}{2}\rfloor+1,\ldots,(q-2),(q-1)}_{\text{write two alphabet}}$$

Denote the write one alphabet by A_1 and the write two alphabet by A_2. Let n be the number of cells in the memory state vector of the WOM code. Note that in the variation we present, on the first write any word in $(A_1)^n$ may be written except $\mathbf{0}$ and $\lfloor\frac{q}{2}\rfloor\mathbf{1}$, which are reserved for the states "inactive" and "1-saturated", respectively. During the second write any word in $(A_2)^n$ may be written except $\lfloor\frac{q}{2}\rfloor\mathbf{1}$ and the 2-saturated state, $(q-1)\mathbf{1}$. This modification allows this WOM code to be used as the inner code in Construction I.

The sum-rate of this WOM code is

$$\frac{\log_2[((\lfloor \frac{q}{2} \rfloor + 1)^n - 2)(\lfloor \frac{q}{2} \rfloor^n - 2)]}{n}.$$

This construction can be generalized to t writes, though large q is necessary for this construction. The following partition of the cell alphabet will yield a variable-rate WOM code with t writes:

$$\underbrace{0, 1, \ldots, \left\lfloor \frac{q}{t} \right\rfloor}_{\text{write 1 alphabet}},$$

$$\underbrace{\left\lfloor \frac{q}{t} \right\rfloor, \ldots, 2\left\lfloor \frac{q}{t} \right\rfloor}_{\text{write 2 alphabet}}$$

$$\vdots$$

$$\underbrace{(t-1)\left\lfloor \frac{q}{t} \right\rfloor, \ldots, (q-1)}_{\text{write t alphabet}}.$$

Note that the saturated states are of the form $\left(\left\lfloor \frac{q}{t} \right\rfloor\right)\mathbf{1}, \left(2\left\lfloor \frac{q}{t} \right\rfloor\right)\mathbf{1}$, etc. Thus, the sum rate of this inner code is

$$\frac{\log_2[((\lfloor \frac{q}{t} \rfloor + 1)^n - 2)(\lfloor \frac{q}{t} \rfloor^n - 2)^{(t-2)}((q - t\lfloor \frac{q}{t} \rfloor)^n - 2)]}{n}.$$

The $EG(3,2)$ code is suitable as an inner code for Construction II since the write generations are distinguishable from the contents of the cells. Note that a slight variation must be used in order to ensure this for all message sequences. Specifically, in the encoding procedure of Section IV, disregard any rule that results in a first generation word $(\mathbf{a}, \mathbf{b})$, with both components nonzero, or a second generation word with a zero component. For example, the steps that state "if $\mathbf{b} < \mathbf{w}$, then set $\mathbf{c} = (\mathbf{0}, \mathbf{w})$," and "if $\mathbf{a} < \mathbf{v}'$, set $\mathbf{c} = (\mathbf{v}', \mathbf{0})$" (i.e., those that indicate a second generation memory state that is indistinguishable from a first generation state) should both be omitted, so that all second generation words have the form $(\mathbf{a}, \mathbf{b})$ such that $\mathbf{a} \neq \mathbf{0}$ and $\mathbf{b} \neq \mathbf{0}$. These rules were originally created to allow for possible third writes and beyond, but since the second and third generation memory state vectors are not distinguishable, we restrict to using the inner WOM code over only two writes. As we discussed above, the simple scheme is just as good, and in the context of Construction II we can use either inner code with the same basic result.

B. Codes from Construction I and II

Note that in both constructions the outer binary code can come from a more general class of WOM codes than those obtained from coset encoding, but possibly at the cost of decoding complexity. Some examples of outer binary WOM codes that result from coset encoding are given in [12]:

- A Hamming code of length $2^r - 1$ produces a $\{2^r - 1, 2^r, 2^{r-2} + 1\}$ WOM code using the coset encoding scheme.
- The length 23 Golay WOM code produces a $\{23, 2^{11}, 3\}$ WOM code using the coset encoding scheme.

TABLE I
PARAMETERS FOR VARIOUS CHOICES OF INNER AND OUTER CODES IN CONSTRUCTION I.

Outer code	Inner code	Nn	q	rate r	Tt
$\{7,8,3\}_2$	$\{1,2,2\}_6$	7	6	3.428; 4.571	6
$\{15,16,5\}_2$	$\{1,2,2\}_6$	15	6	3.333; 4.667	10
$\{7,8,3\}_2$	$\{2,2^2,3\}_{10}$	14	10	3.214; 4.928	9
$\{15,16,5\}_2$	$\{2,2^2,3\}_{10}$	30	10	3; 5	15
$\{7,8,3\}_2$	$\{1,4,2\}_{11}$	7	11	4.285; 6.571	6
$\{15,16,5\}_2$	$\{1,4,2\}_{11}$	15	11	4; 6.667	10
$\{31,32,9\}_2$	$\{1,4,2\}_{11}$	31	11	4.0645; 6.903	18
$\{7,8,3\}_2$	$\{4,4^4,2\}_{11}$	28	11	2.357; 4.643	6
$\{15,16,5\}_2$	$\{4,4^4,2\}_{11}$	60	11	2; 4.667	10
$\{7,8,3\}_2$	$\{2,3^2,3\}_{13}$	14	13	3.966; 6.684	9
$\{15,16,5\}_2$	$\{2,3^2,3\}_{13}$	30	13	3.585; 6.755	15
$\{7,8,3\}_2$	$\{2,2^2,5\}_{16}$	14	16	5.357; 8.214	15
$\{15,16,5\}_2$	$\{2,2^2,5\}_{16}$	30	16	5; 8.333	25
$\{7,8,3\}_2$	$\{2,3^2,4\}_{17}$	14	17	5.285; 8.913	12
$\{15,16,5\}_2$	$\{2,3^2,4\}_{17}$	30	17	4.78; 9.006	20
$\{7,8,3\}_2$	$\{2,2^2,7\}_{22}$	14	22	7.5; 11.5	21
$\{15,16,5\}_2$	$\{2,2^2,7\}_{22}$	30	22	7; 11.667	35

TABLE II
PARAMETERS FOR VARIOUS CHOICES OF INNER AND OUTER CODES IN CONSTRUCTION II.

Outer code	Inner code	Nn	q	rate r	T+t-1
$\{7,8,3\}_2$	$\{1,2,2\}_6$	7	6	2.142; 3.285	4
$\{15,16,5\}_2$	$\{1,2,2\}_6$	15	6	2.0; 3.333	6
$\{7,8,3\}_2$	$\{2,2^2,3\}_{10}$	14	10	1.928; 3.642	5
$\{15,16,5\}_2$	$\{2,3^2,4\}_{17}$	30	17	2.779; 7.006	8
$\{7,8,3\}_2$	$\{2,2^2,7\}_{22}$	14	22	3.642; 7.642	9
$\{15,16,5\}_2$	$\{2,2^2,7\}_{22}$	30	22	3.0; 7.667	13

Table I shows parameters of codes obtained using various inner WOM codes $\mathcal{W}$ of the type given in the previous section. The outer codes are various Hamming codes using the parameters detailed above [12]. Note that the rates are given as a range, using the upper and lower bounds on rate indicated in Section III. Even the upper bound generally lies well below the theoretical upper limit on rate given in [4], indicating that the use of variable rate WOM codes for the inner and outer codes could lead to an improvement in rate. One gain associated with Constructions I and II is that the number of guaranteed writes increases significantly, depending on the constituent codes. Table II shows parameters of codes obtained using Construction II.

VI. Error correction

In addition to obtaining many writes, Constructions I and II may be used for error correction when the constituent WOM codes have some error correction capability. Error correction in the context of a concatenation-type scheme for WOM has been explored in [13], [14], but in the former case the outer code in the concatenation was a traditional error-correcting code rather than a binary WOM code.

Particular types of errors that occur in Constructions I and II are distinguished in the following way: errors that effect the decoding of the outer codeword are called global errors; errors that impact the inner codewords are called local errors. Global errors in the outer code can be of the type (1) active group becomes inactive, or (2) inactive becomes active. Local errors in the inner code are Hamming errors.

The following proposition gives a general result about using codes with some error correction as the inner and outer codes of the constructions.

Proposition 6.1: Let $\mathcal{C}$ be the outer $\{N, M, T\}_2$ E-error-correcting WOM code, and $\mathcal{W}$ be the inner e-error-correcting $\{n, m, t\}_q$ WOM code. Then the code $\mathcal{C} \boxtimes \mathcal{W}$ under either Construction I or II has parameters as indicated in Theorem 3.2 or 3.3, respectively, and can correct at least $(E+1)(e+1)-1$ errors.

Proof: Note that the Constructions detailed in Section III hold the same for the error-correcting case. What remains is to check that all patterns of $(E+1)(e+1)-1$ can be corrected. Indeed, for an inner word to be read in error, at least $e+1$ errors must occur in a subblock of length n. Further, an outer word can be corrected as long as at most E of the N blocks are in error. Thus, as long as no more than $(E+1)(e+1)-1$ cells are in error, the memory contents can be decoded correctly. ■

Many local error patterns will not actually result in a global error. For example, if an active group is read in error as a different active group, a global error does not occur.

Example 6.2: For the outer code, consider a single-error-correcting WOM code (formed from a two-error-correcting BCH code), presented in [15]. The WOM code has parameters $\{63, 6, t\}$, where $t \geq 4$. For the inner WOM code, use an error-correcting WOM code construction in [14] that yields a $\{21, 16, 5\}_{16}$ WOM code that can correct three Hamming errors. Construction II provides a length-1323 WOM code that can correct seven Hamming errors with $q = 16$, and that guarantees eight writes.

VII. Conclusion

We presented two constructions that give a general framework to obtain a nonbinary WOM code from an outer binary code and an inner nonbinary code. In particular, we showed how to use the classical coset encoding scheme on an outer binary code in order to get nonbinary WOM codes from Constructions I and II. The two constructions have tradeoffs in the number of writes versus the amount of information that can be stored at each write. While Construction I allows for more writes, less information can be guaranteed at each generation of the code. Conversely, Construction II allows for more information at each generation, but fewer writes overall. An advantage of Construction II is that it carries fewer restrictions on the properties of the inner nonbinary WOM code that can be used. Extensions include using error-correcting WOM codes as the inner and outer codes, and incorporating variable-rate WOM codes as the component codes.

We also presented a family of two-write ternary WOM codes that can be used as components in Construction II. When used for only two writes, the write generation is recognizable from the memory state vector $\mathbf{c}$. These codes are derived from the structure of $EG(m, 3)$, and can also be viewed as a stand-alone construction of variable-rate WOM codes that result in three writes on average. Other uses of finite geometries may lead to more efficient codes.

Acknowledgements

This work was supported in part by NSA grant H98230-11-1-0156. The United States Government is authorized to reproduce and distribute reprints notwithstanding any copywrite notation herein.

References

[1] G. Cohen, P. Godlewski, F. Merkx, "Linear binary codes for write-once memories," *IEEE Trans. on Inform. Thy*, vol. 32, no. 5, pp. 697-700, 1986.

[2] E. Yaakobi, S. Kayser, P. H. Siegel, A. Vardy, and J. K. Wolf, "Codes for write once memories", *IEEE Trans. on Info. Thy.*, vol. 58, no. 9, pp. 5985-5999, 2012.

[3] R. L. Rivest and A. Shamir, "How to reuse a "write-once" memory," *Inform. Control*, vol. 55, pp.1-19, Oct./Nov./Dec. 1982.

[4] 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 Trans. Inf. Theory*, vol. 45, no.1, pp. 308-313, Jan. 1999.

[5] A. Jiang, "On the generalization of error-correcting WOM codes," *Proc. IEEE ISIT*, June 2007.

[6] A. Shpilka, New constructions of WOM codes using the Wozencraft ensemble, in *Latin American Symp. on Theoretical Informatics*, Arequipa, Peru, April 2012.

[7] R. Gabrys, E. Yaakobi, L. Dolecek, P. H. Siegel, A. Vardy, J. K. Wolf, "Non-binary WOM-Codes for Multilevel Flash Memories," *IEEE Info. Theory Workshop*, Oct., 2011.

[8] A. V. Kuznetsov and A. J. Han Vinck, "On the general defective channel with informed encoder and capacities of some constrained memories," *IEEE Trans. on Info. Theory*, vol. 40, no. 6, Nov. 1994.

[9] Y. Wu, "Low complexity codes for writing write-once memory twice," in *Proc. IEEE ISIT*, July 2010.

[10] A. Jiang, V. Bohossian, J. Bruck, "Rewriting codes for joint information storage in flash memories," *IEEE Trans. on Info. Thy.*, vol. 56, no. 10, pp. 5300-5313, 2010.
[11] A. Shpilka and E. Yaakobi, "High Sum-Rate Three-Write and Non-Binary WOM Codes," *Proc. IEEE ISIT* , July 2012.
[12] G. Cohen, I. Honkala, S. Litsyn, A. Lobstein, *Covering codes*, North-Holland Mathematical Library, Volume 54, Elsevier, Amsterdam: 1997.
[13] Q. Huang, S. Lin, and K. Abdel-Ghaffar, "Error-correcting codes for flash coding", in *Proc. IEEE Information Theory and Applications*, San Diego, Feb. 2011.
[14] K. Haymaker and C.A. Kelley, "Coding strategies for reliable storage in multilevel flash memories," In *Proc. of the Int'l Castle Meeting on Coding Theory and Apps.*, September 2011.
[15] G. Zémor and G. Cohen, "Error-correcting WOM-codes," *IEEE Trans. on Info. Theory*, vol. 37, no. 3, pp. 730 - 734, May, 1991.
[16] Y. Wu and A. Jiang, "Position modulation code for rewriting write-once memories," *IEEE Trans. on Inform. Thy*, vol. 57, no. 6, pp. 3692-3697, 2011.
[17] F. Merkx, "Womcodes constructed with projective geometries," *Traitment du Signal*, vol. 1, no. 2-2, pp. 227-231, 1984.
[18] K. Haymaker and C.A. Kelley, "Geometric WOM codes and coding strategies for multilevel flash memories," *Designs, Codes and Cryptography*, May 2012.
[19] Y. Cassuto, M. Schwartz, V. Bohossian, and J. Bruck, "Codes for asymmetric limited-magnitude errors with application to multilevel flash memories," *IEEE Trans. on Info. Thy.*, vol. 56, no. 4, pp. 1582-1595, Apr. 2010.