Forty-Eighth Annual Allerton Conference
Allerton House, UIUC, Illinois, USA
September 29 - October 1, 2010

Multiple-Write WOM-Codes

Scott Kayser, Eitan Yaakobi, Paul H. Siegel, Alexander Vardy, and Jack K. Wolf
University of California, San Diego La Jolla, CA 92093 − 0401, USA
Emails: {skayser, eyaakobi, psiegel, avardy, jwolf}@ucsd.edu

***Abstract*—A *Write Once Memory (WOM)* is a storage device that consists of *cells* that can take on q possible linearly-ordered values, with the added constraint that rewrites can only increase a cell's value. In the binary case, each cell can change from the level zero to the level one only once. Examples of WOMs include punch cards, optical disks, and more recently flash memories. A length-n, t-write WOM-code is a coding scheme that allows t messages to be stored in n cells. If in the i-th write we write one of M_i messages, then the rate of the i-th write is the ratio of the number of bits written to the WOM to the total number of cells used, i.e., $\log_2(M_i)/n$. The rate of the WOM-code is the sum of all individual rates in all writes.**

In this paper, we review a recent construction of binary two-write WOM-codes. The construction is generalized for two-write WOM-codes with q levels per cell. Then, we show how to use such a code with ternary cells in order to construct three and four-write WOM-codes. This construction is used recursively in order to generate a family of t-write WOM-codes for all t. Another generalized construction is given which provides us with more ways to construct families of WOM-codes. Finally, we give a comparison between our codes and the best known WOM-codes in order to show that the WOM-codes constructed here outperform all previously known WOM-codes for $3 \leqslant t \leqslant 10$.

I. INTRODUCTION

Write-Once Memory, or WOM, is a type of memory with cells that can be written, but not erased. In the binary case, cells can be changed from the '0' state to the '1' state, but cannot be returned to the '0' state. Examples of write-once memories include punch cards, optical discs, and more recently, flash memories. WOM-codes were first introduced by Rivest and Shamir in [11]. These codes allow us to write several times to the WOM without erasing any cells.

A binary $[n, M_1, M_2, ..., M_t, t]$ WOM-code can write t messages on n binary cells, where during the i-th write, $1 \leqslant i \leqslant t$, we write one of M_i possible messages. During each write, a 0 can be changed to a 1, but a 1 cannot be changed to a 0. The rate of the i-th write is the ratio between the number of bits that can be written during that write to the total number of cells used,

$$\mathcal{R}_i = \frac{\log_2 M_i}{n}.$$

The total rate of the WOM-code is the sum of the rates for each write,

$$\mathcal{R} = \sum_{i=1}^{t} \mathcal{R}_i = \frac{\sum_{i=1}^{t} \log_2 M_i}{n}.$$

Note that there are two different problems we can address when searching for WOM-codes: we either require all M_i, for $i = 1, \ldots, t$, to be the same, or we allow them to be different. In this paper, we consider only the second case.

It is proved in [3] and [6] that the capacity region of a binary t-write WOM code is

$$C_t = \Big\{ (\mathcal{R}_1, \ldots, \mathcal{R}_t) \mid \mathcal{R}_1 \leqslant h(p_1), \mathcal{R}_2 \leqslant (1-p_1)h(p_2), \ldots,$$
$$\mathcal{R}_{t-1} \leqslant \Big(\prod_{i=1}^{t-2}(1-p_i)\Big)h(p_{t-1}), \mathcal{R}_t \leqslant \prod_{i=1}^{t-1}(1-p_i),$$
$$\text{where } 0 \leqslant p_1, \ldots, p_{t-1} \leqslant 1/2 \Big\}.$$

It is also proved that the maximum achievable rate for a binary WOM-code with t writes is $\log_2(t+1)$.

The first WOM-code construction, presented by Rivest and Shamir, was designed for the storage of two bits twice using only three cells [11]. In their work, Rivest and Shamir also reported on more WOM-code constructions, including tabular WOM-codes and "linear" WOM-codes. Merkx constructed WOM-codes based on projective geometry [10]. In [2], using binary linear codes, Cohen et al. introduced a "coset-coding" technique that is used to construct WOM-codes, and in [5], an improvement to one of the constructions in [2] was given by Godlewski. Recently, position modulation codes were introduced by Wu and Jiang in order to construct multiple-write WOM-codes [17]. Wu found WOM-codes for two writes in [16] which improved the best rate previously known. In [18], inspired by the coset coding technique of Cohen et al. and Wu's work, a family of two-write WOM-codes was found which further improved the best rate known and was proved to be capacity-achieving.

Wolf et al. discussed the WOM-codes problem from its information-theoretic point of view [13]. The WOM model has been generalized for the multi-level case in [4]. Heegard studied the capacity of a WOM and a noisy WOM in [6], and Fu and Han-Vinck found the capacity of a non-binary WOM [3]. Error-correcting WOM-codes were first studied in [14], [15] and more constructions were recently given in [19]. Jiang discussed in [7] the generalization of error-correcting WOM-codes for the flash/floating codes model [8], [9].

Table I summarizes the best previously known WOM-code rates for $2 \leqslant t \leqslant 10$. The second column shows the rate of the best previously known construction, as well as a reference to the paper where it was first presented. The third column gives the general upper bound on the achievable rate, $\log_2(t+1)$, derived in [6] and [3]. The reference next to each rate indicates the paper where the code was presented. Note that a t-write WOM-code can serve also for a higher number of writes by simply writing no messages on the last writes. That explains the similar rates for four, five, and six writes.

978-1-4244-8216-0/10/$26.00 ©2010 IEEE

TABLE I
LOWER AND UPPER BOUNDS ON WOM-CODE RATES

Number of Writes	Previous Best Rate	Upper Bound
2	1.4928 [18]	$\log_2 3 = 1.5850$
3	1.530 [11]	$\log_2 4 = 2$
4	1.7524 [10]	$\log_2 5 = 2.3219$
5	1.7524 [10]	$\log_2 6 = 2.5850$
6	1.7524 [10]	$\log_2 7 = 2.8074$
7	1.8232 [10]	$\log_2 8 = 3$
8	1.8824 [17]	$\log_2 9 = 3.1699$
9	1.9535 [17]	$\log_2 10 = 3.3219$
10	2.0144 [17]	$\log_2 11 = 3.4594$

In this work, we present WOM-code constructions which reduce the gaps between the upper and lower bounds on the rates of WOM-codes for $3 \leqslant t \leqslant 10$. In Section III, we generalize the two-write WOM-code construction from [18] for non-binary cells. Then, in Section IV, we show how to use these non-binary two-write WOM-codes in order to construct binary multiple-write WOM-codes. We start with specific constructions for three and four writes, and then show a general code design approach that works for an arbitrary number of writes. In Section V, we introduce another general construction based upon concatenating WOM-codes. Finally, in Section VI we compare our codes to the best WOM-codes known previously.

II. PRELIMINARIES

In this section, we briefly repeat the definition of WOM-codes from [18]. The memory-state vectors are all the binary vectors of length n, $\{0,1\}^n$. For two memory-state vectors $\boldsymbol{c}, \boldsymbol{c}' \in \{0,1\}^n$, we say that $\boldsymbol{c} \geqslant \boldsymbol{c}'$ if and only if $\boldsymbol{c}_i \geqslant \boldsymbol{c}'_i$ for all $1 \leqslant i \leqslant n$.

Definition. *An $[n, M_1, \ldots, M_t, t]$* ***t-write WOM-Code*** *$\mathcal{C}$ is a coding scheme which consists of n binary cells and t pairs of encoding and decoding maps, denoted by $\mathcal{E}_i$ and $\mathcal{D}_i$ for $1 \leqslant i \leqslant t$. The t-write WOM-code $\mathcal{C}$ satisfies the following properties:*

1) $\mathcal{E}_1 : \{1, \ldots, M_1\} \to \{0,1\}^n$,
2) *For* $2 \leqslant i \leqslant t$,
$$\mathcal{E}_i : \{1, \ldots, M_i\} \times \{0,1\}^n \to \{0,1\}^n,$$
such that, for all $(m, \boldsymbol{c}) \in \{1, \ldots, M_i\} \times \{0,1\}^n$,
$$\mathcal{E}_i(m, \boldsymbol{c}) \geqslant \boldsymbol{c}.$$
3) *For* $1 \leqslant i \leqslant t$,
$$\mathcal{D}_i : \{0,1\}^n \to \{1, \ldots, M_i\},$$
such that $\mathcal{D}_1(\mathcal{E}_1(m)) = m$ *for all* $m \in \{1, \ldots, M_1\}$, *and for* $2 \leqslant i \leqslant t$, $\mathcal{D}_i(\mathcal{E}_i(m, \boldsymbol{c})) = m$ *for all* $(m, \boldsymbol{c}) \in \{1, \ldots, M_i\} \times \{0,1\}^n$.

The ***rate*** *of a t-write WOM-code $\mathcal{C}$ is defined to be*
$$\mathcal{R} = \frac{\sum_{i=1}^{t} \log_2 M_i}{n}.$$

As noted in [18], we assume that both the encoder and decoder know the write number when encoding new information or decoding the memory state vector. This knowledge does not affect the achievable rates of the codes.

The definition of WOM-codes can be generalized for non-binary cells, where each cell has q levels and on each programming operation the cell level can only increase its value. In the next section, we consider a special type of non-binary WOM-code where each cell is programmed at most once. Although these codes are not efficient WOM-codes for non-binary cells, they are useful in constructing efficient multiple-write WOM-codes for binary cells.

III. TWO-WRITE WOM-CODES

In [18], a construction of binary two-write WOM-codes based on linear codes was given. These codes are completely determined by the choice of a linear code $\mathcal{C}$ with parity check matrix $\mathcal{H}$. In this section, we extend the construction to non-binary WOM-codes. As the construction of the binary two-write WOM-code is a special case of the non-binary construction, we do not review the binary WOM-code construction from [18] and simply present the construction of the non-binary WOM-codes.

Suppose now that each cell has q levels, where q is a prime number or a power of a prime number. We start by choosing a linear code $\mathcal{C}[n,k]$ over $\mathrm{GF}(q)$ with a parity check matrix $\mathcal{H}$ of size $(n-k) \times n$. For a vector $\boldsymbol{v}$ of length n over $\mathrm{GF}(q)$, let $\mathcal{H}(\boldsymbol{v})$ be the matrix $\mathcal{H}$ with zero columns replacing the columns that correspond to the positions of the non-zero values in $\boldsymbol{v}$. Then we define
$$V_{\mathcal{C}}^{(q)} = \{\boldsymbol{v} \in (\mathrm{GF}(q))^n \mid \mathrm{rank}(\mathcal{H}(\boldsymbol{v})) = n-k\}. \qquad (1)$$

Next, we construct a non-binary two-write WOM-code $[n, |V_{\mathcal{C}}^{(q)}|, q^{n-k}, 2]$ in a similar manner to the construction in [18].

Theorem 1. *Let $\mathcal{C}[n,k]$ be a linear code with parity check matrix $\mathcal{H}$ over $\mathrm{GF}(q)$ and let $V_{\mathcal{C}}^{(q)}$ be the set defined in (1). Then there exists a q-ary $[n, |V_{\mathcal{C}}^{(q)}|, q^{n-k}, 2]$ two-write WOM-code of rate*
$$\frac{\log_2 |V_{\mathcal{C}}^{(q)}| + (n-k)\log_2 q}{n}.$$

Proof: The code is characterized by its encoding and decoding maps. We define $\{v_1, v_2, \ldots, v_{|V_{\mathcal{C}}^{(q)}|}\}$ to be an arbitrary ordering of the set $V_{\mathcal{C}}^{(q)}$.

1) On the first write, a symbol from an alphabet of size $|V_{\mathcal{C}}^{(q)}|$ is written. The encoding and decoding maps $\mathcal{E}_1, \mathcal{D}_1$ are defined as follows. For each $m \in \{1 \ldots, |V_{\mathcal{C}}^{(q)}|\}$,
$$\mathcal{E}_1(m) = v_m \text{ and } \mathcal{D}_1(v_m) = m.$$
2) On the second write, a vector $\boldsymbol{s}_2$ of length $n-k$ over $\mathrm{GF}(q)$ is written. Let $\boldsymbol{v}_1$ be the programmed vector on the first write and $\boldsymbol{s}_1 = \mathcal{H} \cdot \boldsymbol{v}_1$, then
$$\mathcal{E}_2(\boldsymbol{s}_2, \boldsymbol{v}_1) = \boldsymbol{v}_1 + \boldsymbol{v}_2,$$

where v_2 is a solution of the equation

$$\mathcal{H}(v_1) \cdot v_2 = -s_1 + s_2.$$

For the decoding map $\mathcal{D}_2$, if c is the vector of programmed cells, then the decoded value of the $n-k$ symbols over GF(q) is given by

$$\mathcal{D}_2(c) = \mathcal{H} \cdot c = \mathcal{H} \cdot v_1 + \mathcal{H} \cdot v_2 = s_1 - s_1 + s_2 = s_2.$$

The success of the second write results from the condition that for every vector $v \in V_{\mathcal{C}}$, rank$(\mathcal{H}(v)) = n-k$. ■

There is no restriction on the choice of the linear code $\mathcal{C}$ or the parity check matrix $\mathcal{H}$. Every such code/matrix generates a WOM-code. For a linear code $\mathcal{C}$ we define $\mathcal{R}_1(\mathcal{C}) = \frac{\log_2 |V_{\mathcal{C}}^{(q)}|}{n}$ and $\mathcal{R}_2(\mathcal{C}) = \frac{(n-k)\log_2 q}{n}$ so the rate of the generated WOM-code is $\mathcal{R}_1(\mathcal{C}) + \mathcal{R}_2(\mathcal{C})$. Next, we show that the capacity region of the achievable rates by this construction is

$$C_2 = \Big\{ (\mathcal{R}_1, \mathcal{R}_2) | \exists p \in [0, \frac{q-1}{q}], \mathcal{R}_1 \leqslant h(p) + p\log_2(q-1), \mathcal{R}_2 \leqslant (1-p)\log_2(q) \Big\}.$$

The proof is very similar to the one presented in [18] for the binary case and we repeat it here for our non-binary construction.

Theorem 2. *For any $(\mathcal{R}_1, \mathcal{R}_2) \in C_2$ and $\epsilon > 0$, there exists a linear code $\mathcal{C}$ satisfying $\mathcal{R}_1(\mathcal{C}) \geqslant \mathcal{R}_1 - \epsilon, \mathcal{R}_2(\mathcal{C}) \geqslant \mathcal{R}_2 - \epsilon$.*

Proof: Let $p \in [0, \frac{q-1}{q}]$ be such that $\mathcal{R}_1 \leqslant h(p) + p\log_2(q-1)$ and $\mathcal{R}_2 \leqslant (1-p)\log_2 q$. Let $k = \lceil np \rceil$ for n large enough and let us choose uniformly at random an $(n-k) \times n$ matrix $\mathcal{H}$ over GF(q). The matrix $\mathcal{H}$ is the parity check matrix of the linear code $\mathcal{C}$ used in our construction. For every vector $v \in (\text{GF}(q))^n$, define the indicator random variable X_v to be

$$X_v = \begin{cases} 1 & \text{if } v \in V_{\mathcal{C}}^{(q)} \\ 0 & \text{otherwise} \end{cases}$$

where $V_{\mathcal{C}}^{(q)}$ is defined in (1). The number of vectors in $V_{\mathcal{C}}^{(q)}$ is $X = \sum_{v \in (\text{GF}(q))^n} X_v$, and

$$E[X] = \sum_{v \in (\text{GF}(q))^n} E[X_v] = \sum_{v \in (\text{GF}(q))^n} \Pr\{X_v = 1\}. \quad (2)$$

The value of $\Pr\{X_v = 1\}$ depends on v only through its weight, $wt(v)$, and therefore (2) simplifies to

$$E[X] = \sum_{i=0}^{n} \binom{n}{i} (q-1)^i \Pr\{X_{v:wt(v)=i} = 1\} = \sum_{i=0}^{k} \binom{n}{i} (q-1)^i \Pr\{X_{v:wt(v)=i} = 1\},$$

since if $wt(v) \geqslant k+1$, then $X_v = 0$.

It remains to determine the value of $\Pr\{X_v = 1\}$ for a vector v such that $wt(v) = i, 0 \leqslant i \leqslant k$. By the definition of $V_{\mathcal{C}}^{(q)}$, we know that $v \in V_{\mathcal{C}}^{(q)}$ if and only if the sub-matrix of size $(n-k) \times (n - wt(v))$ is full rank. According to [1], the probability that an $(n-k) \times (n-i)$ uniformly random matrix is full rank is $\prod_{j=k-i+1}^{n-i}(1-q^{-j})$. Note that

$$\prod_{j=k-i+1}^{n-i} (1-q^{-j}) > \prod_{j=1}^{\infty} (1-q^{-j}) > (1-\frac{1}{q})(1-\sum_{j=2}^{\infty} q^{-j}) = \frac{q-1}{q} \cdot \frac{q^2-q-1}{q(q-1)} = \frac{q^2-q-1}{q^2} \geqslant \frac{1}{q^2},$$

where the last inequality holds for $q \geqslant 2$. Hence, for all v such that $0 \leqslant wt(v) \leqslant k$

$$\Pr\{X_v = 1\} = \prod_{j=k-i+1}^{n-i} (1-q^{-j}) > \frac{1}{q^2},$$

and we get

$$E[X] = \sum_{i=0}^{k} \binom{n}{i} (q-1)^i \prod_{j=k-i+1}^{n-i} (1-q^{-j}) > \frac{1}{q^2} \sum_{i=0}^{k} \binom{n}{i} (q-1)^i.$$

According to Lemma 4.8 in [12],

$$\sum_{i=0}^{k} \binom{n}{i} (q-1)^i \geqslant \frac{1}{n+1} 2^{nh(\frac{k}{n}) + n\frac{k}{n}\log_2(q-1)}$$

and therefore

$$E[X] > \frac{1}{q^2} \cdot \sum_{i=0}^{k} \binom{n}{i} (q-1)^i > \frac{2^{nh(\frac{k}{n}) + n\frac{k}{n}\log_2(q-1)}}{q^2(n+1)} = 2^{nh(\frac{k}{n}) + n\frac{k}{n}\log_2(q-1) - \log_2(q^2(n+1))}.$$

We conclude that there exists a parity check matrix $\mathcal{H}$ of a linear code $\mathcal{C}$, such that the size of the set $V_{\mathcal{C}}^{(q)}$ is at least $2^{nh(\frac{k}{n}) + n\frac{k}{n}\log_2(q-1) - \log_2(q^2(n+1))}$ and so

$$\mathcal{R}_1(\mathcal{C}) \geqslant h\left(\frac{k}{n}\right) + \frac{k}{n}\log_2(q-1) - \frac{\log_2(q^2(n+1))}{n} \geqslant h(p) + p\log_2(q-1) - \frac{\log_2(q^2(n+1))}{n} \geqslant \mathcal{R}_1 - \epsilon$$

$$\mathcal{R}_2(\mathcal{C}) = \frac{\log_2 q^{n-k}}{n} \geqslant (1-p-\frac{1}{n})\log_2 q \geqslant \mathcal{R}_2 - \epsilon$$

for n large enough. ■

The next Corollary provides the best achievable rate of the construction.

Corollary 3. *For any q-ary WOM-code generated using our construction, the best achievable rate is $\log_2(2q-1)$.*

Proof: First, note that

$$h(p) + p\log_2(q-1) + (1-p)\log_2 q = p\log_2\left(\frac{q-1}{p}\right) + (1-p)\log_2\left(\frac{q}{1-p}\right),$$

and since the function $f(x) = \log_2 x$ is a concave function

$$p \log_2 \frac{q-1}{p} + (1-p) \log_2 \frac{q}{1-p} \leqslant \log_2 \left(p \cdot \frac{q-1}{p} + (1-p) \frac{q}{1-p} \right) = \log_2(2q-1).$$

Also, for $p = \frac{q-1}{2q-1}$, the total achievable rate is $\log_2(2q-1)$. Therefore, there exists a WOM-code produced by our construction with achievable rate $\log_2(2q-1)$.

On the other hand, any WOM-code resulting from our construction satisfies the property that every cell is programmed at most once. This model was studied in [3] and the maximum achievable rate was proved to be $\log_2(2q-1)$. Therefore, our construction cannot produce a WOM-code with a rate that exceeds $\log_2(2q-1)$. ■

Remark 1. This construction does not achieve high rates for non-binary two-write WOM-codes in general. While the best achievable rate of the construction is $\log_2(2q-1)$, the upper bound on the rate is $\log_2 \binom{q+1}{2}$; see [3]. The decrease in the rate in our construction results from the fact that cells cannot be programmed twice. That is, if a cell was programmed on the first write, it cannot be reprogrammed on the second write even if it did not reach its highest level. In fact, it is possible to find non-binary two-write WOM-codes with better rates. However, our goal in this paper is not to find efficient non-binary WOM-codes. Rather, as shown in the next section, the non-binary codes that we have constructed can be used in the design of multiple-write binary WOM-codes.

The WOM-codes we use in the next section are WOM-codes over GF(3). We ran a computer search to find such a ternary WOM-code of rate 2.22, and we will use this code in order to construct specific multiple-write WOM-codes.

IV. Multiple-Write WOM-Codes

In this section, we look at a method for generating binary multiple-write WOM-codes. We start with three- and four-write WOM-codes before generalizing to t-write constructions.

A. Three-Write WOM-Codes

We start with a construction for binary three-write WOM-codes. The construction uses the codes we found in the previous section over GF(3).

Theorem 4. *Let $\mathcal{C}_3$ be an $[n, 2^{n\mathcal{R}_1}, 2^{n\mathcal{R}_2}, 2]$ two-write WOM-code over* GF(3) *constructed as in Section* III. *Then, there exists a $[2n, 2^{n\mathcal{R}_1}, 2^{n\mathcal{R}_2}, 2^n, 3]$ three-write WOM-code of rate $\frac{\mathcal{R}_1+\mathcal{R}_2+1}{2}$.*

Proof: We denote by $\mathcal{E}_{3,1}$ and $\mathcal{E}_{3,2}$ the encoding maps of the first and second writes, and by $\mathcal{D}_{3,1}$ and $\mathcal{D}_{3,2}$ the decoding maps of the first and second writes of the code $\mathcal{C}_3$, respectively. The $2n$ cells of the three-write WOM-code we construct are divided into n two-cell blocks, so the memory-state vector is of the form $((c_{1,1}, c_{1,2}), (c_{2,1}, c_{2,2}), \ldots, (c_{n,1}, c_{n,2}))$. In this construction we also use a map $\phi : \text{GF}(3) \mapsto (\text{GF}(2), \text{GF}(2))$ defined as follows:

$$\phi(0) = (0,0),$$
$$\phi(1) = (1,0),$$
$$\phi(2) = (0,1).$$

The map ϕ extends naturally to ternary vectors $\boldsymbol{v} = (v_1, \ldots, v_n) \in \text{GF}(3)^n$ using the rule

$$\phi(\boldsymbol{v}) = (\phi(v_1), \ldots, \phi(v_n)).$$

On the pairs (c, c') in the image of ϕ, we define $\phi^{-1}(c, c')$ to indicate the inverse function. The map ϕ^{-1} is extended similarly to work over vectors of such bit pairs. We are now ready to describe the encoding and decoding maps of the constructed three-write WOM-code.

1) On the first write, a message m from the set $\{1, \ldots, 2^{n\mathcal{R}_1}\}$ is written in the $2n$ cells:

$$\mathcal{E}_1(m) = \phi(\mathcal{E}_{3,1}(m)).$$

The decoding map is defined similarly, where $\boldsymbol{c}$ is the memory-state vector:

$$\mathcal{D}_1(\boldsymbol{c}) = \mathcal{D}_{3,1}(\phi^{-1}(\boldsymbol{c})).$$

2) On the second write, a message m from the set $\{1, \ldots, 2^{n\mathcal{R}_2}\}$ is written in the $2n$ cells as follows. Let $\boldsymbol{c}$ be the programmed vector on the first write. Then,

$$\mathcal{E}_2(m, \boldsymbol{c}) = \phi(\mathcal{E}_{3,2}(m, \phi^{-1}(\boldsymbol{c}))).$$

That is, first the memory-state vector $\boldsymbol{c}$ is converted to a ternary vector. Then, it is encoded using the encoding map $\mathcal{E}_{3,2}$ and the new message, producing a new ternary memory-state vector. Finally, the last vector is converted to a $2n$-bit vector. The decoding map is defined as on the first write:

$$\mathcal{D}_2(\boldsymbol{c}) = \mathcal{D}_{3,2}(\phi^{-1}(\boldsymbol{c})).$$

According to the construction of the code $\mathcal{C}_3$, no ternary cell is programmed twice and therefore each of the n pairs of bits is programmed at most once.

3) On the third write, an n-bit vector $\boldsymbol{v}$ is written. Let $\boldsymbol{c} = ((c_{1,1}, c_{1,2}), \ldots, (c_{n,1}, c_{n,2}))$ be the current memory-state vector. Then,

$$\mathcal{E}_3(\boldsymbol{v}, \boldsymbol{c}) = ((c'_{1,1}, c'_{1,2}), \ldots, (c'_{n,1}, c'_{n,2}))$$

is a vector, defined as follows. For $1 \leqslant i \leqslant n$, $(c'_{i,1}, c'_{i,2}) = (1,1)$ if $v_i = 1$ and otherwise $(c'_{i,1}, c'_{i,2}) = (c_{i,1}, c_{i,2})$. It is always possible to program the pair of bits to be $(1,1)$ since at most one cell in each pair was previously programmed. The decoding map $\mathcal{D}_2(\boldsymbol{c})$ is defined to be

$$\mathcal{D}_2(\boldsymbol{c}) = (c_{1,1} \cdot c_{1,2}, \ldots, c_{n,1} \cdot c_{n,2}).$$

That is, the decoded value of each pair of bits is one if and only if the value of both of them is one.

■

Corollary 5. *The best possible three-write WOM-code that is achievable using this construction is* $(\log_2 5 + 1)/2 \approx 1.66$.

Proof: Given a two-write WOM-code $\mathcal{C}_3$ over GF(3) with rates $(\mathcal{R}_1, \mathcal{R}_2)$, the constructed binary three-write WOM-code has rates $(\mathcal{R}_1/2, \mathcal{R}_2/2, 1/2)$ and its total rate is $\mathcal{R} = (\mathcal{R}_1 + \mathcal{R}_2 + 1)/2$. This rate is maximized when $\mathcal{R}_1 + \mathcal{R}_2$ is maximized. But $\mathcal{R}_1 + \mathcal{R}_2$ is the total rate of the two-write WOM-code over GF(3), which was proven in Corollary 3 to be maximized at $\log_2 5$. Then the maximum achievable rate of the constructed binary three-write WOM-code is

$$\frac{\log_2 5 + 1}{2} \approx 1.66.$$

■

Using the construction of WOM-codes over GF(3) presented in the previous section, we can construct a three-write WOM-code of rate $(2.22 + 1)/2 = 1.61$.

B. Four-Write WOM-Codes

We next present a construction for four-write binary WOM-codes.

Theorem 6. *Let* $\mathcal{C}_3$ *be an* $[n, 2^{n\mathcal{R}_{3,1}}, 2^{n\mathcal{R}_{3,2}}, 2]$ *two-write WOM-code over* GF(3) *constructed as in Section* III. *Let* $\mathcal{C}_2$ *be an* $[n, 2^{n\mathcal{R}_{2,1}}, 2^{n\mathcal{R}_{2,2}}, 2]$ *binary two-write WOM-code. Then, there exists a* $[2n, 2^{n\mathcal{R}_{3,1}}, 2^{n\mathcal{R}_{3,2}}, 2^{n\mathcal{R}_{2,1}}, 2^{n\mathcal{R}_{2,2}}, 4]$ *four-write WOM-code of rate* $\frac{\mathcal{R}_{3,1}+\mathcal{R}_{3,2}+\mathcal{R}_{2,1}+\mathcal{R}_{2,2}}{2}$.

Proof: The proof is very similar to the one used for three-write WOM-codes. We denote by $\mathcal{E}_{3,1}, \mathcal{E}_{3,2}$ the encoding maps of the first and second writes, and by $\mathcal{D}_{3,1}, \mathcal{D}_{3,2}$ the decoding maps of the first and second writes of the code $\mathcal{C}_3$, respectively. Similarly, the encoding and decoding maps of the code $\mathcal{C}_2$ for the first and second writes are denoted by $\mathcal{E}_{2,1}, \mathcal{E}_{2,2}$ and $\mathcal{D}_{2,1}, \mathcal{D}_{2,2}$, respectively. Using the encoding and decoding maps of $\mathcal{C}_3$, we define the first and second writes of our constructed four-write WOM-code as we did the first and second writes of the three-write WOM-codes. The third and fourth writes are defined in a similar way, as follows.

1) On the third write, a message m from the set $\{1, \ldots, 2^{n\mathcal{R}_{2,1}}\}$ is written. Let $\mathcal{E}_{2,1}(m) = \boldsymbol{v} = (v_1, \ldots, v_n)$ and let $\boldsymbol{c} = ((c_{1,1}, c_{1,2}), \ldots, (c_{n,1}, c_{n,2}))$ be the current memory-state vector. Then,
$$\mathcal{E}_3(m, \boldsymbol{c}) = ((c'_{1,1}, c'_{1,2}), \ldots, (c'_{n,1}, c'_{n,2})),$$
where for $1 \leqslant i \leqslant n$, $(c'_{i,1}, c'_{i,2}) = (1, 1)$ if $v_i = 1$ and, otherwise, $(c'_{i,1}, c'_{i,2}) = (c_{i,1}, c_{i,2})$. The decoding map $\mathcal{D}_3(\boldsymbol{c})$ is defined to be
$$\mathcal{D}_3(\boldsymbol{c}) = \mathcal{D}_{2,1}(c_{1,1} \cdot c_{1,2}, \ldots, c_{n,1} \cdot c_{n,2}).$$
2) On the fourth write, a message m from the set $\{1, \ldots, 2^{n\mathcal{R}_{2,2}}\}$ is written. Let
$$\mathcal{E}_{2,2}(m, (c_{1,1} \cdot c_{1,2}, \ldots, c_{n,1} \cdot c_{n,2})) = \boldsymbol{v} = (v_1, \ldots, v_n),$$
where $\boldsymbol{c} = ((c_{1,1}, c_{1,2}), \ldots, (c_{n,1}, c_{n,2}))$ is the current memory-state vector. Then,
$$\mathcal{E}_4(m, \boldsymbol{c}) = ((c'_{1,1}, c'_{1,2}), \ldots, (c'_{n,1}, c'_{n,2})),$$
where for $1 \leqslant i \leqslant n$, $(c'_{i,1}, c'_{i,2}) = (1, 1)$ if $v_i = 1$ and, otherwise, $(c'_{i,1}, c'_{i,2}) = (c_{i,1}, c_{i,2})$. The decoding map $\mathcal{D}_4(\boldsymbol{c})$ is defined, as before, by
$$\mathcal{D}_4(\boldsymbol{c}) = \mathcal{D}_{2,2}(c'_{1,1} \cdot c'_{1,2}, \ldots, c'_{n,1} \cdot c'_{n,2}).$$

■

Remark 2. The last theorem requires both the binary two-write and ternary two-write WOM-codes to have the same number of cells, n. However, we can construct a four-write binary WOM-code using any two such codes, even if they do not have the same number of cells. Suppose we have a WOM-code over GF(3) with n_1 cells and binary WOM-code with n_2 cells. Both codes can be extended to use $\mathrm{lcm}(n_1, n_2)$ cells. Then the construction above will give a four-write WOM-code.

Corollary 7. *The best possible four-write WOM-code that is achievable using this construction is* $(\log_2 5 + \log_2 3)/2 \approx 1.95$.

Proof: According to Corollary 3, the maximum value of $\mathcal{R}_{3,1} + \mathcal{R}_{3,2}$ is $\log_2 5$ and the maximum value of $\mathcal{R}_{2,1} + \mathcal{R}_{2,2}$ is $\log_2 3$. Therefore, the maximum rate of the constructed four-write WOM-codes is

$$\frac{\log_2(5) + \log_2(3)}{2} \approx 1.95.$$

■

If we use the WOM-code over GF(3) of rate 2.22 found in the previous subsection as the WOM-code $\mathcal{C}_3$ and the binary two-write WOM-code of rate 1.4928 found in [18] as the WOM-code $\mathcal{C}_2$, then there exists a four-write WOM-code of rate $(2.22 + 1.4928)/2 = 1.8564$.

C. Multiple-Write WOM-Codes

The construction for three and four writes can be easily generalized to an arbitrary number of writes. We state the following theorem and skip its proof since it is very similar to the proofs of the corresponding theorems for three- and four-write WOM-codes.

Theorem 8. *Let* $\mathcal{C}_3$ *be an* $[n, 2^{n\mathcal{R}_{3,1}}, 2^{n\mathcal{R}_{3,2}}, 2]$ *two-write WOM-code over* GF(3) *constructed as in Section* III. *Let* $\mathcal{C}_2$ *be an* $[n, 2^{n\mathcal{R}_{2,1}}, \ldots, 2^{n\mathcal{R}_{2,t-2}}, t-2]$ *binary* $(t-2)$*-write WOM-code. Then, there exists a*

$$[2n, 2^{n\mathcal{R}_{3,1}}, 2^{n\mathcal{R}_{3,2}}, 2^{n\mathcal{R}_{2,1}}, \ldots, 2^{n\mathcal{R}_{2,t-2}}, t]$$

t*-write WOM-code of rate*

$$\frac{\mathcal{R}_{3,1} + \mathcal{R}_{3,2} + \sum_{i=1}^{t-2} \mathcal{R}_{2,i}}{2}.$$

Theorem 8 implies that if there exists a $(t-2)$-write WOM-code of rate $\mathcal{R}_{t-2}$ then there exists a t-write WOM-code of rate

$$\mathcal{R}_t = \frac{\log_2 5 + \mathcal{R}_{t-2}}{2}.$$

The following corollary summarizes the possible achievable rates of t-write WOM-codes.

Corollary 9. *For $t \geqslant 3$, there exists a t-write WOM-code of rate*

$$\mathcal{R}_t = \begin{cases} \frac{(2^{\frac{t-1}{2}}-1)\cdot\log_2 5+1}{2^{\frac{t-1}{2}}} & , t \text{ odd} \\ \frac{(2^{\frac{t-2}{2}}-1)\cdot\log_2 5+\log_2 3}{2^{\frac{t-2}{2}}} & , t \text{ even.} \end{cases}$$

If we use again the two-write WOM-code over GF(3) of rate 2.22 and the binary two-write WOM-code of rate 1.4928 from [18], then for $t \geqslant 3$ we obtain a t-write WOM-code of rate $\mathcal{R}_t$, where

$$\mathcal{R}_t = \begin{cases} \frac{(2^{\frac{t-1}{2}}-1)\cdot 2.22+1}{2^{\frac{t-1}{2}}} & , t \text{ odd} \\ \frac{(2^{\frac{t-2}{2}}-1)\cdot 2.22+1.4928}{2^{\frac{t-2}{2}}} & , t \text{ even.} \end{cases}$$

V. Concatenated WOM-Codes

The construction presented in the previous section provides us with a family of WOM-codes for all $t \geqslant 3$. In this section, we will show a general scheme to construct more families of WOM-codes. In fact, the construction in the previous section is a special case of this general scheme.

Theorem 10. *Let $\mathcal{C}^*$ be an $[m, q_1, \ldots, q_{t/2}, t/2]$ binary $t/2$-write WOM-code where t is an even integer. For $1 \leqslant i \leqslant t/2$, let $\mathcal{C}_i$ be an $[n, 2^{n\mathcal{R}_{i,1}}, 2^{n\mathcal{R}_{i,2}}, 2]$ two-write WOM-code over $\mathrm{GF}(q_i)$, as constructed in Section III. Then, there exists an $[mn, 2^{n\mathcal{R}_{1,1}}, 2^{n\mathcal{R}_{1,2}}, \ldots, 2^{n\mathcal{R}_{t/2,1}}, 2^{n\mathcal{R}_{t/2,2}}, t]$ binary t-write WOM-code of rate*

$$\sum_{i=1}^{t/2} \frac{\mathcal{R}_{i,1} + \mathcal{R}_{i,2}}{m}.$$

Proof: For $1 \leqslant i \leqslant t/2$, let $\mathcal{E}_i^*, \mathcal{D}_i^*$ be the encoding, decoding maps on the i-th write of the WOM-code $\mathcal{C}^*$, respectively. The definition of $\mathcal{E}_i^*, \mathcal{D}_i^*$ for $1 \leqslant i \leqslant t/2$ extends naturally to vectors by simply invoking the maps on each entry in the vector. Similarly, for $1 \leqslant i \leqslant t/2$, let us denote by $\mathcal{E}_{i,1}$ and $\mathcal{E}_{i,2}$ the encoding maps of the first and second writes, and by $\mathcal{D}_{i,1}$ and $\mathcal{D}_{i,2}$ the decoding maps of the first and second writes of the WOM-code $\mathcal{C}_i$, respectively. We will present the specification of the encoding and decoding maps of the constructed t-write WOM-code.

In the following definitions of the encoding decoding maps, we consider the memory state vector $\boldsymbol{c}$ to have n symbols of m bits each, i.e. $\boldsymbol{c} \in (\mathrm{GF}(2^m))^n$. For $1 \leqslant i \leqslant t/2$, the $(2i-1)$-st write and $2i$-th write are implemented as follows.

1) On the $(2i-1)$-st write, a message $m_1 \in \{1, \ldots, 2^{n\mathcal{R}_{i,1}}\}$ is written to the memory-state vector $\boldsymbol{c}$ according to

$$\mathcal{E}_{2i-1}(m_1, \boldsymbol{c}) = \mathcal{E}_i^*(\mathcal{E}_{i,1}(m_1), \boldsymbol{c}).$$

The memory-state vector $\boldsymbol{c}$ is decoded according to

$$\mathcal{D}_{2i-1}(\boldsymbol{c}) = \mathcal{D}_{i,1}(\mathcal{D}_i^*(\boldsymbol{c})).$$

2) On the $2i$-th write, a message $m_2 \in \{1, \ldots, 2^{n\mathcal{R}_{i,2}}\}$ is written according to

$$\mathcal{E}_{2i}(m_2) = \mathcal{E}_i^*(\mathcal{E}_{i,2}(m_2, \mathcal{D}_i^*(\boldsymbol{c})), \boldsymbol{c})$$

and the memory-state vector $\boldsymbol{c}$ is decoded according to

$$\mathcal{D}_{2i}(\boldsymbol{c}) = \mathcal{D}_{i,2}(\mathcal{D}_i^*(\boldsymbol{c})).$$

■

We will demonstrate how this construction works in the following example.

Example 1. We choose a $[3,4,3,2,3]$ three-write WOM-code as the code $\mathcal{C}^*$. This code is described in Fig. 1 by its states diagram for all three writes. The three-bit vector in each state is the memory state and the number next to it is the decoded value. We need to find three more two-write WOM-codes over $\mathrm{GF}(4), \mathrm{GF}(3)$, and $\mathrm{GF}(2)$. For the code $\mathcal{C}_1$ over $\mathrm{GF}(4)$, we ran a computer search to find a two-write WOM-code over $\mathrm{GF}(4)$ of rate 2.6793. As for the code $\mathcal{C}_2$ over $\mathrm{GF}(3)$, we use the code of rate 2.22 which we found in Section III, and we use the binary two-write WOM-code of rate 1.49 for the code $\mathcal{C}_3$. Then, the rate of the six-write WOM-code is

$$\frac{2.6793 + 2.22 + 1.49}{3} = 2.1297.$$

It is possible to construct a five-write WOM-code by writing a vector of n bits in the last write so its rate is

$$\frac{2.6793 + 2.22 + 1}{3} = 1.9664.$$

Note that if one of the codes in the general construction is binary then we can actually use a WOM-code with more than two writes. That is, in this construction we can use any binary multiple-write WOM-code as the code $\mathcal{C}_3$. Therefore, we can generate another family of codes for $t \geqslant 5$. Their maximum achievable rates are given according to the following formula

$$\mathcal{R}_t = \frac{\log_2 7 + \log_2 5 + \mathcal{R}_{t-4}}{3},$$

where $\mathcal{R}_{t-4}$ is the maximum achievable rate for a $(t-4)$-write WOM-code. Similarly, the codes which we obtain using the codes found above have rates

$$\mathcal{R}'_t = \frac{2.6793 + 2.22 + \mathcal{R}'_{t-4}}{3},$$

where $\mathcal{R}'_{t-4}$ is the best rate of a $(t-4)$-write WOM-code which we can find. Table II summarizes these rates.

Note that the construction in Section IV is a special case of the generalized concatenated WOM-codes construction in which $\mathcal{C}^*$ is chosen to be $[2,3,2,2]$ binary two-write WOM-code.

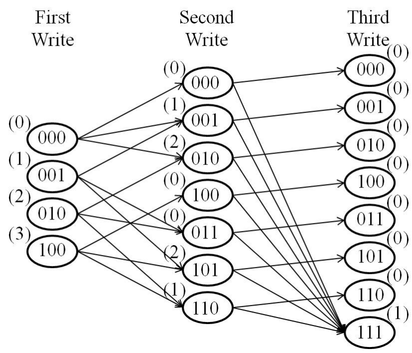

Fig. 1. A $[3, 4, 3, 2, 3]$ three-write WOM-code.

TABLE II
RATES OF CONCATENATED WOM-CODES

Number of Writes	Achieved New Rate	Maximum New Rate
5	1.9664	$\frac{\log_2 7+\log_2 5+1}{3} = 2.0431$
6	2.1297	$\frac{\log_2 7+\log_2 5+\log_2 3}{3} = 2.2381$
7	2.1697	$\frac{\log_2 7+\log_2 5+(\log_2 5+1)/2}{3} = 2.2634$
8	2.2519	$\frac{\log_2 7+\log_2 5+(\log_2 5+\log_2 3)/2}{3} = 2.3609$
9	2.2885	$\frac{\log_2 7+\log_2 5+(\log_2 7+\log_2 5+1)/3}{3} = 2.3908$
10	2.343	$\frac{\log_2 7+\log_2 5+(\log_2 7+\log_2 5+\log_2 3)/3}{3} = 2.4588$

VI. COMPARISON

Table III shows a comparison of the rates of the WOM-codes presented in this paper and the best known rates previously for $3 \leqslant t \leqslant 10$. The column labeled "Previous Best" is the highest rate achieved by a previously reported t-write WOM-code. The column "Achieved New Rate" gives the rates that we actually obtained through application of the new techniques. For three and four writes, we use the codes described in Section IV, and for $5 \leqslant t \leqslant 10$, we use the codes discussed in Section V. The column "Maximum New Rate" lists the maximum possible rate that can be obtained using our approach. Finally, the column "Upper Bound" gives the maximum possible rates for t-write WOM codes.

VII. CONCLUSION

We presented a new method for constructing multiple-write binary WOM-codes. The method makes use of two-write WOM-codes over GF(q), for which we describe a design technique. While the non-binary codes we construct are not capacity-achieving, they allow us to construct binary t-write WOM-codes for $t \geqslant 3$. We showed how to construct codes for three and four writes, and then showed that a recursive algorithm can be used to generate binary WOM-codes that support any number of writes. We also described a general concatenation scheme to construct other families of WOM-codes. Applying this scheme, we found another family of t-write WOM-codes that gives the best known rates for $5 \leqslant t \leqslant 10$. We showed that our codes outperform all previously known WOM-codes for $3 \leqslant t \leqslant 10$. It is possible to show that these codes also achieve the best known rates for $t > 10$ writes. We believe that it is possible to find other interesting families of codes using the concatenated WOM-codes construction.

TABLE III
COMPARISON WITH KNOWN WOM-CODES

Number of Writes	Previous Best	Achieved New Rate	Maximum New Rate	Upper Bound
3	1.53	**1.61**	1.6610	2
4	1.75	**1.8564**	1.9534	2.3219
5	1.75	**1.9664**	2.0431	2.5850
6	1.75	**2.1297**	2.2381	2.8074
7	1.82	**2.1697**	2.2634	3
8	1.88	**2.2519**	2.3609	3.1699
9	1.95	**2.2885**	2.3908	3.3219
10	2.01	**2.343**	2.4588	3.4594

VIII. ACKNOWLEDGEMENT

This work was supported in part by the University of California Lab Fees Research Program, Award No. 09-LR-06-118620-SIEP and the Center for Magnetic Recording Research at the University of California, San Diego.

REFERENCES

[1] R. Brent, S. Gao, and A. Lauder, "Random Krylov spaces over finite fields," *SIAM J. Discrete Math*, vol. 16, no. 2, pp. 276–287, February 2003.
[2] 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, September 1986.
[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 Trans. Inform. Theory*, vol. 45, no. 1, pp. 308–313, September 1999.
[4] A. Fiat and A. Shamir, "Generalized write-once memories," *IEEE Trans. Inform. Theory*, vol. 30, pp. 470–480, September 1984.
[5] P. Godlewski, "WOM-codes construits à partir des codes de Hamming," *Discrete Math.*, no. 65 pp. 237–243, 1987.
[6] C. Heegard, "On the capacity of permanent memory," *IEEE Trans. Inform. Theory*, vol. 31, pp. 34–42, January 1985.
[7] A. Jiang, "On the generalization of error-correcting WOM codes," *Proc. IEEE Int. Symp. Inform. Theory*, pp. 1391–1395, Nice, France, June 2007.
[8] A. Jiang and J. Bruck, "Joint coding for flash memory storage," *Proc. IEEE Int. Symp. Inform. Theory*, pp. 1741–1745, Toronto, Canada, July 2008.
[9] H. Mahdavifar, P.H. Siegel, A. Vardy, J.K. Wolf, and E. Yaakobi, "A nearly optimal construction of flash codes," *Proc. IEEE Int. Symp. Inform. Theory*, pp. 1239–1243, Seoul, Korea, July 2009.
[10] F. Merkx, "Womcodes constructed with projective geometries," *Traitement du signal*, vol. 1, no. 2-2, pp. 227–231, 1984.
[11] R.L. Rivest and A. Shamir, "How to reuse a write-once memory," *Information and Control*, vol. 55, nos. 1–3, pp. 1–19, December 1982.
[12] R.M. Roth, *Introduction to Coding Theory*, Cambridge University Press, 2005.
[13] J.K. Wolf, A.D. Wyner, J. Ziv, and J. Korner, "Coding for a write-once memory," *AT&T Bell Labs. Tech. J.*, vol. 63, no. 6, pp. 1089–1112, 1984.
[14] G. Zémor and G.D. Cohen, "Error-correcting WOM-codes," *IEEE Trans. Inform. Theory*, vol. 37, no. 3, pp. 730–734, May 1991.
[15] G. Zémor, "Problèmes combinatoires liés à l'écriture sur des mémoires," Ph.D. Dissertation, ENST, Paris, France, November 1989.
[16] Y. Wu, "Low complexity codes for writing write-once memory twice," *Proc. IEEE Int. Symp. Inform. Theory*, Austin, Texas, June 2010.
[17] Y. Wu and A. Jiang, "Position modulation code for rewriting write-once memories," submitted to *IEEE Trans. Inform. Theory*, September 2009. Preprint available at http://arxiv.org/abs/1001.0167.
[18] E. Yaakobi, S. Kayser, P.H. Siegel, A. Vardy, and J.K. Wolf, "Efficient two-write WOM-codes," *Proc. IEEE Information Theory Workshop*, Dublin, Ireland, August 2010.
[19] E. Yaakobi, P.H. Siegel, A. Vardy, and J.K. Wolf, "Multiple error-correcting WOM-codes," *Proc. IEEE Int. Symp. Inform. Theory*, Austin, Texas, June 2010.