Non-binary WOM-Codes for Multilevel Flash Memories

Ryan Gabrys*, Eitan Yaakobi†, Lara Dolecek*, Paul H. Siegel†, Alexander Vardy†, Jack K. Wolf†

*Electrical Engineering Department
University of California, Los Angeles
{*rgabrys, dolecek*}@*ee.ucla.edu*

†Electrical and Computer Engineering Department
University of California, San Diego
{*eyaakobi, psiegel, avardy, jwolf*}@*ucsd.edu*

***Abstract*—A Write-Once Memory (WOM)-code is a coding scheme that allows information to be written in a memory block multiple times, but in a way that the stored values are not decreased across writes. This work studies non-binary WOM-codes with applications to flash memory. We present two constructions of non-binary WOM-codes that leverage existing high sum-rate WOM-codes defined over smaller alphabets. In many instances, these constructions provide the highest known sum-rates of the non-binary WOM-codes. In addition, we introduce a new class of codes, called *level distance WOM-codes*, which mitigate the difficulty of programming a flash memory cell by eliminating all small-magnitude level increases. We show how to construct such codes and state an upper bound on their sum-rate.**

I. Introduction

A write-once memory (WOM) is a storage medium whose atomic memory elements are cells. The cells can be either binary or can take on $q > 2$ different values. The main property which distinguishes a WOM from other memories is that values written to the storage device are non-decreasing over a fixed number of writes. Codes for WOM systems were first introduced by Rivest and Shamir almost three decades ago [12]. These codes are known in the literature as *WOM-codes*.

Recently, there has been a renewed interest in these codes due to their relevance for the ubiquitous flash memories. Flash memories are composed of floating gate cells which are charged with electrons [6]. First generation flash memory cells have two levels (SLC) and thus store a single bit. In recently developed technologies, multilevel cells (MLC) can store 2 or more bits, and this number is expected to further increase in the future [13]. While it is relatively easy to increase a cell level, reducing its level is possible only if the entire block ($\sim 10^6$ cells) is first erased [6]. These erasure operations are not only time consuming but also degrade the lifetime of the flash memory. The endurance of a flash memory block is mostly affected by the number of times it is erased, where this number can be as low as 1000 erases before the memory block is deemed unusable [6]. Hence, the goal of a WOM-code is to maximize the total amount of information written across a prescribed number of writes (referred to as generations).

Since the pioneering work of Rivest and Shamir [12], the focus has been on studying binary WOM-codes with good properties, as in [1], [7] and more recently in [10], [14]. Despite the fact that Fiat and Shamir studied non-binary WOM-codes more than 20 years ago [2], only a few constructions for such codes exist. Notable work on this topic includes the results on the existence of some non-binary WOM-codes [3], and an expression for the capacity region for a fixed number of generations and levels [4]. In [10], a family of two-write non-binary WOM-codes was given. However, these codes do not achieve high information sum-rates since they allow each cell to be written only once, even though the cell may not have achieved its highest level. Huang *et al.* [9] used error-correcting codes to design non-binary WOM-codes. Recently, in [5] a two-write WOM-code was presented for $q = 3$ levels along with upper bounds on the sum-rates and sum-rate properties of non-binary WOM-codes.

One of the main challenges in designing flash memory cells with a large number of levels is to accurately program the cells to their target level [11]. In particular, small increases in the cell levels are hard to perform since the amount of charge that has to be injected into the cell is infinitesimal. Therefore, an interesting and relevant model, which circumvents these small increases, is the one where every increase in the cell level is no less than some specified positive value ℓ. We call these WOM-codes *level distance WOM-codes*. The capacity of such codes is derived using a generalized result from [4]. We also show how to design such WOM-codes suitable for this model.

This paper is organized as follows. In Section II, we define the model for a multilevel WOM-code and state two elementary constructions of such codes. In Section III, we present our non-binary WOM-code constructions. In Section IV, a new class of codes, referred to as level distance WOM-codes, is introduced and a construction of such codes is given. Section V concludes the paper.

II. Definitions And Simple Code Constructions

We assume in this work that the cells have q values: $\{0, 1, \ldots, q-1\}$. The initial state of each cell is zero. While it is possible to increase a cell level, it is not possible to decrease its value. We refer to the set $\{0, \ldots, q-1\}^n$ as the *cell-state vectors*. For two cell-state vectors $\boldsymbol{x} = (x_1, \ldots, x_n), \boldsymbol{y} = (y_1, \ldots, y_n) \in \{0, \ldots, q-1\}^n$, we denote $\boldsymbol{x} \geq \boldsymbol{y}$ if for all $1 \leq i \leq n$, $x_i \geq y_i$. Also, for $n \geq 1$, $\mathbf{1}_n$ is the all-one row vector of length n. All arithmetic operations described in the following sections are performed in the ring modulo q, where q is the number of levels in the flash memory device. We follow the definition of WOM-codes in [10], [14] and extend it to the non-binary setup.

Definition 1. *An* $[n, t; M_1, \ldots, M_t]_q$ *t-write non-binary WOM-code* $\mathcal{C}_q$ *is a coding scheme on n q-ary cells. The code* $\mathcal{C}_q$ *is specified by t pairs of encoding and decoding maps* $\mathcal{E}_j$ *and* $\mathcal{D}_j$*, for* $1 \leq j \leq t$*, satisfying the following properties:*

1) $\mathcal{E}_1 : \{1, \ldots, M_1\} \to \{0, \ldots, q-1\}^n$,
2) *For* $2 \leq j \leq t$,

$$\mathcal{E}_j : \{1, \ldots, M_j\} \times Im(\mathcal{E}_{j-1}),$$

978-1-4577-0437-6/11/$26.00

where $Im(\mathcal{E}_{j-1})$ denotes the image of the map $\mathcal{E}_{j-1}$ and for all $(m, \boldsymbol{c}) \in \{1, \ldots, M_j\} \times \{0, \ldots, q-1\}^n$,

$$\mathcal{E}_j(m, \boldsymbol{c}) \geq \boldsymbol{c}.$$

3) *For $1 \leq j \leq t$,*

$$\mathcal{D}_j : \{0, \ldots, q-1\}^n \to \{1, \ldots, M_j\},$$

so that $\mathcal{D}_1(\mathcal{E}_1(m)) = m$ for all $m \in \{1, \ldots, M_1\}$, and for $2 \leq j \leq t$, $\mathcal{D}_j(\mathcal{E}_j(m, \boldsymbol{c})) = m$ for all $(m, \boldsymbol{c}) \in \{1, \ldots, M_j\} \times \{0, \ldots, q-1\}^n$.

For the convenience of the presentation of the constructions we interpret the map $\mathcal{E}_1$ to be $\mathcal{E}_1 : \{1, \ldots, M_1\} \times \{0, \ldots, q-1\}^n \to \{0, \ldots, q-1\}^n$, where the second argument is ignored since the cell-state vector is always all zeros.

Definition 2. *The* rate *of a t-write WOM-code $\mathcal{C}_q$ on the j-th write, $1 \leq j \leq t$ is defined to be $\mathcal{R}_j(\mathcal{C}_q) = \frac{\log_2 M_j}{n}$, and the* sum-rate *of the WOM-code $\mathcal{C}_q$ is*

$$\mathcal{R}_{sum}(\mathcal{C}_q) = \sum_{j=1}^{t} \mathcal{R}_j(\mathcal{C}_q).$$

As argued in [14], it is assumed that the write number, j, is known both to the encoder and decoder as this information does not affect the achievable sum-rates by the constructions we present. Note that there are two different setups we can address when analyzing WOM-codes: we either require that on all writes the same number of messages is stored, or we allow the individual writes to use different numbers of messages. We call the former problem *fixed-rate WOM-codes* and the latter one *unrestricted-rate WOM-codes*. While this paper focuses on the unrestricted-rate WOM-code setup, it will be mentioned later how these constructions can be extended to fixed-rate WOM-codes as well.

We first present two simple ways to construct non-binary WOM-codes, and illustrate where these constructions outperform the best known sum-rates achieved. In Section III, it is shown how the sum-rates achieved by these simple constructions can in fact be improved using the codes proposed in this paper. The simple code constructions are as follows:

1) Assume that $t \geq q-1$. Suppose that a non-binary WOM-code uses the constituent binary WOM-code "layer-by-layer" as follows. Assume that $t = 2(q-1)$ and let $\mathcal{C}_2$ be a two-write binary WOM-code. A t-write q-ary WOM-code is constructed such that on the first two writes the WOM-code $\mathcal{C}_2$ is used on levels 0 and 1. On the following two writes again $\mathcal{C}_2$ is used on levels 1 and 2 and so on. If the sum-rate of the WOM-code $\mathcal{C}_2$ is $\mathcal{R}$ then the sum-rate of the non-binary WOM-code is $(q-1)\mathcal{R}$. It is possible to modify the construction such that on each level a different binary WOM-code with a different number of writes (even one write) is used. This modification will allow for a flexible number of writes. For example, a 3-write WOM-code can be constructed given 3 levels by using a binary 2-write code on the first two generations and simply adding a binary vector for the last write.
2) Assume that $t < q-1$. For example, assume that $t = 2$ and q is odd. On the first write only levels $0, 1, \ldots, \frac{q-1}{2}$ are used, and if there are n cells, it is possible to write $n \log_2\left(\frac{q+1}{2}\right)$ bits of information. On the second write, only levels $\frac{q-1}{2}, \frac{q+1}{2}, \ldots, q-1$ are used. Thus, it is possible to write the same number of bits. If $t > 2$, we simply choose $t-1$ values $q_1, \ldots, q_{t-1}$ such that $q_0 = 0 < q_1 < q_2 < \cdots < q_{t-1} < q_t = q-1$ and on the i-th write, $1 \leq i \leq t$ only levels $q_{i-1}, q_{i-1}+1, \ldots, q_i$ are used.

An idea similar to the first construction was proposed in [9]. In fact, it is possible to show that the proposed approach outperforms the results reported in [9]. For example, given 20 levels and using the best two-write code provided in [14] with the "level-by-level" approach, it is possible to obtain a sum-rate 26.1250 for a fixed-rate WOM-code and 28.3613 for an unrestricted-rate WOM-code which is higher than the sum-rate 25.3 stated in [9]. The second construction outperforms the code construction for $t = 2$ in [10] for $q \geq 5$. In fact, for $t = 2$ the sum-rate achieved by the second construction for odd q is $2\log_2\left(\frac{q+1}{2}\right)$ while the upper bound is $\log_2\left(\frac{q(q+1)}{2}\right)$. That is, the sum-rate of this construction is only within

$$\log_2\left(\frac{q(q+1)}{2}\right) - 2\log_2\left(\frac{q+1}{2}\right) = \log_2\left(\frac{2q}{q+1}\right) < 1$$

from the upper bound.

III. WOM-Code Constructions

In the following we present two constructions of non-binary WOM-codes that in many instances outperform the two simple constructions provided earlier. In general, the first construction provides higher sum-rate when q is smaller, but for large q the second construction has higher sum-rate. Both constructions build non-binary WOM-codes from WOM-codes over a smaller alphabet size. We refer to these smaller alphabet size WOM-codes as *base codes*. The sum-rates of the base codes dictate the achievable sum-rates for the resulting non-binary WOM-codes. In the next two subsections we present the code constructions, followed by the sum-rate analysis.

A. Construction A

Let q, m be two positive integers. The map

$$\phi_{q,m} : \{0, \ldots, q-1\}^m \to \{0, \ldots, q^m - 1\}$$

is the representation of a vector $\boldsymbol{x} \in \{0, \ldots, q-1\}^m$ as an integer according to base q (this is the inverse q-adic expansion). Let $\boldsymbol{x} = (x_1, \ldots, x_m)$. Then

$$\phi_{q,m}(\boldsymbol{x}) = \Sigma_{j=1}^{m} q^{m-j} x_j.$$

Since this map is one-to-one and onto, its inverse map $\phi_{q,m}^{-1}$ exists.

We now present the first construction. The general idea behind this construction is to independently encode k q-ary WOM-codes and, using ϕ, map these WOM-codes into an alphabet of size q^k.

Theorem 1. *Let $\mathcal{C}_q$ be an $[n, t : M_1, \ldots, M_t]_q$ t-write WOM-code. Then, there exists an $[n, t : M_1^k, \ldots, M_t^k]_{q^k}$ t-write WOM-code.*

Proof: Assume the t encoding and decoding maps of the WOM-code $\mathcal{C}_q$ are denoted by $\mathcal{E}_{q,j}, \mathcal{D}_{q,j}$, for $1 \leq j \leq t$. We construct an $[n, t : M_1^k, \ldots, M_t^k]_{q^k}$ t-write WOM-code which we denote by $\mathcal{C}_{q^k}$ with encoding and decoding maps $\mathcal{E}_{q^k,j}, \mathcal{D}_{q^k,j}$, for $1 \leq j \leq t$.

On the j-th write, $1 \leq j \leq t$, the input to the map $\mathcal{E}_{q^k,j}$ is an integer $w_j \in \{0, \ldots, M_j^k - 1\}$ and the cell-state vector

$\boldsymbol{c} = (c_1, \ldots, c_n) \in \{0, 1, \ldots, q^k - 1\}^n$. Let $\phi^{-1}_{M_j,k}(w_j) = (w_{j,1}, w_{j,2}, \ldots, w_{j,k})$ and for $1 \le \ell \le k$,

$$\boldsymbol{c}_\ell = (\phi^{-1}_{q,k}(c_1)_\ell, \ldots, \phi^{-1}_{q,k}(c_n)_\ell), \quad \boldsymbol{u}_\ell = \mathcal{E}_{q,j}(w_{j,\ell}, \boldsymbol{c}_\ell).$$

Then, the output of the encoding map is defined to be

$$\mathcal{E}_{q^k,j}(w_j, \boldsymbol{c}) = (\phi_{q,k}(u_{1,1}, \ldots, u_{k,1}), \ldots, \phi_{q,k}(u_{1,n}, \ldots, u_{k,n})).$$

Note that since $\mathcal{C}_q$ is a WOM-code, then for all $1 \le \ell \le k$, $\boldsymbol{u}_\ell \ge \boldsymbol{c}_\ell$ and hence $\mathcal{E}_{q^k,j}(w_j, \boldsymbol{c}) \ge \boldsymbol{c}$.

Similarly, we define for $1 \le j \le t$ the decoding map

$$\mathcal{D}_{q^k,j} : \{0, \ldots, q^k - 1\}^n \to \{0, 1, \ldots, M_j^k - 1\}$$

as follows. Let $\boldsymbol{c} = (c_1, \ldots, c_n) \in \{0, \ldots, q^k - 1\}^n$ be the cell-state vector. For $1 \le \ell \le k$, let

$$w_{j,\ell} = \mathcal{D}_{q,j}(\phi^{-1}_{q,k}(c_1)_\ell, \ldots, \phi^{-1}_{q,k}(c_n)_\ell),$$

then

$$\mathcal{D}_{q^k,j}(\boldsymbol{c}) = \phi_{M_j,k}(w_{j,1}, \ldots, w_{j,k}).$$ ■

Example 1. *Let $\mathcal{C}_2$ be the well-known Rivest-Shamir $[3, 2 : 4, 4]_2$ two-write WOM-code [12]. We use $\mathcal{C}_2$ as the base code in order to construct a $[3, 2 : 4^3, 4^3]_{2^3}$ two-write WOM-code over an alphabet with eight symbols $\{0, 1, \ldots, 7\}$. An example how to write the new WOM-code which is described as follows.*

Write number	Data bits	Encoding by the base-code $\mathcal{C}_2$	Encoded values in the 8-ary cell
1	(01,11,10)	(100,001,010)	(4,1,2)
2	(00,11,01)	(111,110,011)	(6,7,5)

On each write, six bits are written. Assume that the information bits on the first write are (01, 11, 10). Every pair of bits is encoded by the Rivest-Shamir code so the output is the three triplets of bits (100, 001, 010). Finally, the first bit from every triplet (shown in red) is stored in the first cell according to the mapping $\phi_{2,3}$. Similarly, the second, third bit (shown in black, blue) from every triplet is stored in the second, and third cell, respectively. The same principle applies to the second write.

Remark 1. *If $\mathcal{R}_A$ is the sum-rate of the base WOM-code C_q in Theorem 1, then the sum-rate of the new WOM-code is $k\mathcal{R}_A$.*

B. Construction B

Our second construction is presented in the next theorem. The key idea behind this construction is to divide up the q available levels into equally sized blocks of length k. At each generation we encode one of $\{0, 1, \ldots, k - 1\}$ values into a block where the block to write into is chosen independently according to a WOM-code.

Theorem 2. *Let $\mathcal{C}_q$ be an $[n, t : M_1, \ldots, M_t]_q$ t-write WOM-code and $k \ge 2$. Then, there exists an $[n, t : M_1 k^n, \ldots, M_t k^n]_{k(q+t-1)}$ t-write WOM-code.*

Proof: As in the proof of Theorem 1, we let $\mathcal{E}_{q,j}, \mathcal{D}_{q,j}$ for $1 \le j \le t$ be encoding and decoding maps of the WOM-code $\mathcal{C}_q$. We denote the t-write $[n, t : M_1 k^n, \ldots, M_t k^n]_{k(q+t-1)}$ WOM-code we construct by $\mathcal{C}_{q'}$, where $q' = k(q+t-1)$, and its encoding and decoding maps are $\mathcal{E}_{q',j}, \mathcal{D}_{q',j}$ for $1 \le j \le t$.

The input to the encoding map $\mathcal{E}_{q',j}$ on the j-th write, $1 \le j \le t$, is $w_j \in \{0, \ldots, M_j k^n - 1\}$ and a cell-state vector $\boldsymbol{c}$. Let $(w_{j,0}, w_{j,1}, \ldots, w_{j,n})$ be such that $w_{j,0} = w_j (\text{mod } M_j)$, and $(w_{j,1}, \ldots, w_{j,n}) = \phi^{-1}_{k,n}\left(\left\lfloor \frac{w_j}{M_j} \right\rfloor\right)$. Let

$$\boldsymbol{c}' = \left\lfloor \frac{\boldsymbol{c}}{k} \right\rfloor - (j-2) \cdot \mathbf{1}_n$$

and the output of $\mathcal{E}_{q,j}$ be

$$\boldsymbol{c}'' = \mathcal{E}_{q,j}(w_{j,0}, \boldsymbol{c}') + (j-1) \cdot \mathbf{1}_n.$$

The final encoded cell-state vector is

$$\boldsymbol{c} = k \cdot \boldsymbol{c}'' + (w_{j,1}, \ldots, w_{j,n}).$$

It is straightforward to verify that the cells cannot reduce their value but for abbreviation, we skip the details here.

The decoding map $\mathcal{D}_{q',j} : \{0, \ldots, q' - 1\}^n \to \{0, 1, \ldots, M_j k^n - 1\}$ for $1 \le j \le t$ is defined as follows. Let $\boldsymbol{c} \in \{0, \ldots, q' - 1\}^n$ be the cell-state vector and let $\boldsymbol{c}''' = \left\lfloor \frac{\boldsymbol{c}}{k} \right\rfloor - (j-1) \cdot \mathbf{1}_n$, then

$$w_{j,0} = \mathcal{D}_{q,j}(\boldsymbol{c}'''),$$

and for $1 \le i \le n$, $w_{j,i} \equiv c_i (\text{mod } k)$. Finally, we decode

$$w_{j,0} M_j + \phi_{k,n}(w_{j,1}, \ldots, w_{j,n}).$$ ■

Example 2. *Let $\mathcal{C}_2$ be the $[3, 2 : 4, 4]_2$ Rivest-Shamir two-write WOM-code with encoding maps $\mathcal{E}_{2,1}$ and $\mathcal{E}_{2,2}$. We use $\mathcal{C}_2$ as the base code to construct a $[3, 2 : 4 \cdot 3^3, 4 \cdot 3^3]_9$ two-write WOM-code over an alphabet with symbols $\{0, 1, \ldots, 8\}$.*

Suppose that on the first write we receive a message $(w_1, \boldsymbol{v}_1)$ such that $w_1 = (0, 1)$ and $\boldsymbol{v}_1 = (0, 1, 2)$. We have $\boldsymbol{c}_1 = \mathcal{E}_{2,1}(w_1) = (1, 0, 0)$. Then, on the first write the codeword is $\boldsymbol{c} = (3, 1, 2)$. Similarly, on the second write we receive a message $(w_2, \boldsymbol{v}_2)$ such that $w_2 = (0, 0)$ and $\boldsymbol{v}_2 = (2, 1, 2)$. Then we have that $\boldsymbol{c}_2 = \mathcal{E}_{2,2}(w_2, \lfloor \frac{\boldsymbol{c}}{3} \rfloor) = (1, 1, 1)$. The resulting second write codeword is $(8, 7, 8)$.

Remark 2. *If $\mathcal{R}_B$ is the sum-rate of the base WOM-code C_q in Theorem 2 then the sum-rate of the new WOM-code is $t \log_2 k + \mathcal{R}_B$.*

C. Sum-Rate Analysis

Assume that each cell has q levels. We seek to construct a t-write WOM-code according to Theorem 1 with a WOM-code $\mathcal{C}_{q'}$ of sum-rate $\mathcal{R}$, which uses q' levels such that $(q')^k = q$. Then, the sum-rate of the new WOM-code is $k \cdot \mathcal{R}$ and if $\mathcal{C}_{q'}$ is a capacity achieving WOM-code then the sum-rate is $\frac{\log_2 q}{\log_2 q'} \cdot \log_2 \binom{q'+t-1}{t-1}$. This value is maximized when q' is maximized and since $k \ge 2$, we get $q' \le \sqrt{q}$, so the best sum-rate one can achieve is $2 \cdot \log_2 \binom{\sqrt{q}+t-1}{t-1}$, where we assume for simplicity that $\sqrt{q}$ is an integer number.

Similarly, if a cell has q levels, then we construct a t-write WOM-code according to Theorem 2. Here, a WOM-code $\mathcal{C}_{q'}$ of sum-rate $\mathcal{R}$ uses q' levels for $q = k(q' + t - 1)$. The maximum sum-rate $t \log_2 \left(\frac{q}{q'+t-1}\right) + \log_2 \binom{q'+t-1}{t}$ is maximized when q' is maximized and since $k \ge 2$, we get $q' \le q/2 - (t-1)$ and thus the maximum achievable sum-rate is $t + \log_2 \binom{q/2}{t}$ where we assume that q is even.

Hence, for any given t, q the best sum-rate that is possible to achieve is the maximum between these two expressions. However, non-binary WOM-codes are not easy to find in general. A different approach to evaluate the sum-rate is to start with a base code and a prescribed number of writes. Assume we are given a WOM-code with q'-ary cells such that its sum-rate is $\mathcal{R}$. By Construction A, we can construct a WOM-code with cells of $q_A = (q')^{k_A}$ levels and its sum-rate is $k_A \cdot \mathcal{R}$. By Construction B, we can construct a WOM-code for $q_B = k_B(q' + t - 1)$ and its sum-rate is $t \log_2 k_B + \mathcal{R}$.

Comparing the resulting expressions for $\mathcal{R}_A$ and $\mathcal{R}_B$ provides the following result.

Lemma 1. *If $q_A = q_B = q$, then for fixed t Construction B yields a better sum-rate than Construction A if and only if*

$$\mathcal{R} \leq \frac{t\log_2\left(\frac{q}{q'+t-1}\right)}{\log_{q'}(q) - 1}.$$

In general, in order to evaluate the best approach to construct a multiple-write non-binary WOM-code, one should consider the two elementary constructions we stated in Section II as well as Constructions A and B. Since these constructions are based upon some base WOM-codes, the parameters of the base WOM-code as well as the desired non-binary WOM-code dictate the resulting sum-rate. There exist conditions under which one of the four constructions stated in this paper gives the highest sum-rate. We omit the details due to the lack of space and leave them to an extended version of this work.

Next, we state the highest sum-rates we found for some non-binary WOM-codes. Table I and Table II are the result of applying Construction A to the best known multiple write binary WOM-codes found in [14] and [10]. For $q = 16, 32, 64, 128$, the sum-rates listed in Table III are the result of applying the ternary two-write WOM-code in [5] to Construction B.

It is important to note that the tables below present the highest sum-rate known WOM-codes for the parameters shown.

TABLE I
ACHIEVED SUM-RATES FOR $q = 4$ BY CONSTRUCTION A

t	Achieved Sum-rate	Capacity
2	2.9856	3.3219
3	3.2200	4.3219
4	3.7128	5.1293
5	3.9328	5.8074
6	4.2594	6.3923
7	4.3394	6.9069

TABLE II
ACHIEVED SUM-RATES FOR $q = 8$ BY CONSTRUCTION A

t	Achieved Sum-rate	Capacity
2	4.4784	5.1699
3	4.8300	6.9069
4	5.5692	8.3663
5	5.8992	9.6294
6	6.3891	10.7448
7	6.5091	11.7448

TABLE III
ACHIEVED TWO-WRITE SUM-RATES BY CONSTRUCTION A AND B

q	Achieved Sum-rate	Capacity
4	2.9856	3.3219
8	4.4784	5.1699
16	6.3083	7.0875
32	4.4784	9.0444
64	10.3083	11.0224
128	12.3083	13.0112

The tables shown here are for unrestricted-rate WOM-codes, but the constructions provided in this paper can be applied to fixed-rate WOM-codes as well. The only requirement for a fixed-rate WOM-code created using Construction A or B is that the base WOM-code be fixed-rate as well. The results are left for an extended version of this paper.

IV. Level Distance WOM-Codes

In this section we introduce a new class of WOM-codes which we call *level distance WOM-codes*. A level distance WOM-code is a WOM-code with the additional property that any cell level increase is not less than ℓ, for some fixed $\ell > 0$.

Definition 3. *An $[n, t, \ell; M_1, \ldots, M_t]_q$ t-write ℓ-level distance WOM-code $\mathcal{C}_q$ is an $[n, t; M_1, \ldots, M_t]_q$ t-write WOM-code with encoding and decoding maps $\mathcal{E}_j, \mathcal{D}_j$, for $1 \leq j \leq t$ which satisfies the following additional constraint. For all $1 \leq j \leq t$ and $(w, \boldsymbol{c}) \in \{1, \ldots, M_j\} \times \{0, \ldots, q-1\}^n$, if $\mathcal{E}_j(w, \boldsymbol{c})_i > c_i$ for some $1 \leq i \leq n$, where i denotes the index of the cell-state vector, then*

$$\mathcal{E}_j(w, \boldsymbol{c})_i - c_i \geq \ell.$$

Given $q, \ell > 0$, the transition matrix of the constraint for the level distance WOM-codes is $\mathcal{A}_{\ell,q} = (a_{i,j})_{0\leq i,j\leq q-1}$, where for $0 \leq i, j \leq q-1$, $a_{i,j} = 1$ if $j - i \geq \ell$ or $i = j$. For all other values of i and j, $a_{i,j} = 0$. Hence, according to a general result proved in [4], the capacity for sum-rate of any t-write ℓ-level distance WOM-code with q-ary cells is

$$\log_2(\mathbf{1}_q \cdot (\mathcal{A}_{\ell,q})^{t-1} \cdot \mathbf{1}_q^T).$$

In order to construct level distance WOM-codes we use the construction of WOM-codes in Theorem 1 where the base WOM-code is binary. The main idea of the construction in Theorem 1 is that a cell with $q = 2^k$ levels stores k bits which belong to a binary WOM-code, and thus can only change irreversibly from a zero to a one. Therefore, the cell level, according to the map $\phi_{2,k}$, always increases when the bits change. In order to modify this WOM-code such that it will be a level distance WOM-code, first we need to eliminate all one level increases in the cells. In order to eliminate the increase from level zero to level one, and from level $q-2$ to level $q-1$, we add some artificial levels that will not be used to represent information but only to guarantee the level distance property. However, this constraint will not be enough. Other one level increases occur when the first bit changes its value. In order to eliminate these small increases we change the map $\phi_{2,k}$ as follows. The new map

$$\psi'_k : \{0, 1, \ldots, 2^k - 1\} \rightarrow \{0, 1\}^k$$

is defined by an ordering of all the binary vectors in $\{0, 1\}^k$. These binary vectors are ordered first by their weight and then all the vectors in the same weight-group are ordered according to their value by the map $\phi_{2,k}$. That is, the ordering of the vectors in $\{0, 1\}^k$ is defined to be $\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_{2^k-1}$:

1) $\boldsymbol{v}_0 = (0, 0, \ldots, 0)$, $\boldsymbol{v}_{2^k-1} = (1, 1, \ldots, 1)$.
2) For $1 \leq i \leq k-1$, $\boldsymbol{v}_{\sum_{j=0}^{i-1}\binom{k}{j}}, \ldots, \boldsymbol{v}_{\sum_{j=1}^{i}\binom{k}{j}}$ are all the weight-i binary vectors such that for $\sum_{j=0}^{i-1}\binom{k}{j} \leq j < h \leq \sum_{j=1}^{i}\binom{k}{j}$, $\phi_{2,k}(\boldsymbol{v}_j) < \phi_{2,k}(\boldsymbol{v}_h)$.

Next, we add $2(k-2)$ more artificial levels to construct the map

$$\psi_k : \{0, 1, \ldots, 2^k + 2(k-2) - 1\} \rightarrow \{0, 1\}^k \cup \{B\},$$

where the symbol B represents an unused level, as follows:

1) $\psi_k(0) = \psi'_k(0)$, $\psi_k(2^k + 2(k-2) - 1) = \psi'_k(2^k - 1)$.
2) For $1 \leq i \leq k-2$ and $2^k+(k-3) \leq i \leq 2^k+2(k-2)-2$, $\psi_k(i) = B$.
3) For $\ell \leq i \leq 2^k + (k-4)$, $\psi_k(i) = \psi'_k(i - (k-2))$.

An example of the map ψ_k for $k = 3$ is shown in Table IV. The level distance property is proved in the next lemma.

Lemma 2. *If a cell has $2^k + 2(k-2)$ levels and stores k bits according to the map ψ_k such that the bits do not decrease their value, then every cell level increase is not less than $k-1$.*

TABLE IV
ORDERING OF THE MAP ψ_3.

level	Third Bit	Second Bit	First Bit
0	0	0	0
1	B	B	B
2	0	0	1
3	0	1	0
4	1	0	0
5	0	1	1
6	1	0	1
7	1	1	0
8	B	B	B
9	1	1	1

Proof: Assume that the cell stores the vector $\boldsymbol{v}$ which is changed to be $\boldsymbol{u}$ and $\boldsymbol{u} \geq \boldsymbol{v}$. If $\boldsymbol{v} = (0, \ldots, 0)$ or $\boldsymbol{u} = (1, \ldots, 1)$ then it is clear that $\psi_k(\boldsymbol{u}) - \psi_k(\boldsymbol{v}) \geq k-1$ because of the $k-2$ artificial levels after the first level and before the last one. Assume that the weight of $\boldsymbol{v}$ is w where $1 \leq w \leq k-1$ and $\boldsymbol{v} = (v_k, \ldots, v_1)$. Consider the following cases:

1) Assume $w = 1$, and $v_i = 1$ for $1 \leq i \leq k$. For any cell level increase there are at least $k - i + \binom{i-1}{2} \geq k-2$ vectors before the new target level. $k-i$ in the weight-group of weight one and $\binom{i-1}{2}$ in the following weight-group.
2) Assume $2 \leq w \leq k-3$ and $v_k = 1$. Then, there are at least $\binom{k-1}{w+1} \geq k-1$ vectors that start with a zero in the weight-$(w+1)$ group of vectors and hence every cell level increase is greater than $k-1$.
3) Assume $w = k-2$ and $v_k = 1$. Assume further that the first m positions of $\boldsymbol{v}$ are equal to 1. Then there are at least $\binom{k-m-1}{w-m-1}$ vectors that are below v_k in the same weight-group if $m \neq w$ and 0 if m is equal to w. Furthermore, there are m vectors in the weight group $w+1 = k-1$ above v_k where the one of the first $m-1$ bits is a 0. It follows then that every cell increase is at least $k-1$.
4) Assume $2 \leq w \leq k-2$ and $v_k = 0$. Then, there are at least $\binom{k-1}{w-1} \geq k-1$ vectors that start with a one in the same weight-group and again every cell level increase is greater than $k-1$. ■

The next theorem summarizes this construction.

Theorem 3. *Let $\mathcal{C}_2$ be an $[n,t;M_1,\ldots,M_t]_2$ t-write binary WOM-code. Then, there exists an $[n,t,\ell;M_1^{\ell+1},\ldots,M_t^{\ell+1}]_{2^{\ell+1}+2(\ell-1)}$ t-write ℓ-level distance WOM-code.*

Proof: The proof follows from the proof of Lemma 2 and the proof of the construction of Theorem 1. ■

Example 3. *Let us consider the WOM-code construction in Example 1. The Rivest-Shamir $[3,2:4,4]_2$ two-write WOM-code is used to construct a $[3,2:4^3,4^3]_{2^3}$ two-write WOM-code over an alphabet with eight symbols. In order to modify this construction we use the map ψ_3 in Table IV instead of the map $\phi_{2,3}$. The example to write the bits is changed as follows.*

Write number	Data bits	Encoding by the base-code $\mathcal{C}_2$	Encoded values in the 10-ary cell
1	(01,11,10)	(100,001,010)	(4,2,3)
2	(00,11,01)	(111,110,011)	(7,9,6)

Note that the sum-rate of the level distance WOM-code is the same as the sum-rate of a code constructed using Theorem 1. In this construction we just had to use more levels in order to guarantee the level distance property. It is possible to apply a similar modification to the second construction in order to produce a level distance WOM-code. We save the details for an extended version of this paper.

V. Conclusion

In this work, we explored two basic constructions for non-binary WOM-codes that leverage existing high sum-rate binary WOM-codes. In general, the performance of the resulting non-binary WOM-code is dependent on the sum-rate of the underlying base code and a direct implication of this work is that high sum-rate WOM-codes over a smaller alphabet sizes can yield high sum-rate WOM-codes over larger alphabets.

In addition, we introduced a new class of codes, called level distance WOM-codes. These codes target the inherently difficult process of programming the flash memory cell to a specified threshold value by fixing a lower bound on the magnitudes by which each cell is increased between writes. The upper bound on the sum-rate of such codes is easily derived by the tools in [8], and we presented here a construction of such codes.

VI. Acknowledgement

This work was supported in part by the grant CCF-1029030 from NSF, the University of California Lab Fees Research Program, Award No. 09-LR-06-118620-SIEP, the Center for Magnetic Recording Research at the University of California, San Diego, the Intel Ph.D. Fellowship Program, and the SMART Scholarship Program.

References

[1] 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, October 1986.
[2] A. Fiat and A. Shamir, "Generalized "write-once" memories," *IEEE Trans. Inform. Theory*, vol. 30, no. 3, pp. 470–480, September 1984.
[3] H. Finucane, Z. Liu, and M. Mitzenmacher, "Designing floating codes for expected performance," *Proc. 46-th Allerton Conf. Communication, Control and Computing*, pp. 1389–1396, Monticello, IL, September 2008.
[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. Inform. Theory*, vol. 45, no. 1, pp. 308–313, September 1999.
[5] R. Gabrys and L. Dolecek, "Characterizing capacity achieving write once memory codes for multilevel flash memories," *Proc. IEEE Int. Symp. Inform. Theory*, pp. 2484–2488, St. Petersburg, Russia, August 2011.
[6] E. Gal and S. Toledo, "Algorithms and data structures for flash memories," *ACM Computing Surveys*, vol. 37, pp. 138–163, June 2005.
[7] P. Godlewski, "WOM-codes construits à partir des codes de Hamming," *Discrete Math.*, vol. 65, no. 3, pp. 237–243, July 1987.
[8] C. Heegard, "On the capacity of permanent memory," *IEEE Trans. Inform. Theory*, vol. 31, no. 1, pp. 34–42, January 1985.
[9] Q. Huang, S. Lin, and K. A. S. Abdel-Ghaffar, "Error-correcting codes for flash coding," accepted by *IEEE Trans. Inform. Theory*, April 2011.
[10] S. Kayser, E. Yaakobi, P. H. Siegel, A. Vardy, and J. K. Wolf, "Multiple-write WOM-codes," *Proc. 48-th Annual Allerton Conference on Communication, Control and Computing*, Monticello, IL, September 2010.
[11] H. T. Lue, T. H. Hsu, S. Y. Wang, E. K. Lai, K. Y. Hsieh, R. Liu, and C. Y. Lu, "Study of incremental step pulse programming (ISPP) and STI edge effect of BE-SONOS NAND flash," *Proc. IEEE Int. Symp. on Reliability Physics*, vol. 30, no. 11, pp. 693–694, May 2008.
[12] R .L. Rivest and A. Shamir, "How to reuse a write-once memory," *Inform. and Contr.*, vol. 55, no. 1–3, pp. 1–19, December 1982.
[13] N. Shibata *et al.*, "A 70 nm 16 Gb 16-level-cell NAND flash memory," *IEEE J. Solid-State Circuits*, vol. 43, pp. 929–937, April 2008.
[14] E. Yaakobi, S, Kayser, P. H. Siegel, A. Vardy, and J. K. Wolf, "Efficient two-write WOM-codes," *Proc. IEEE Inform. Theory Workshop*, Dublin, Ireland, August 2010.