Joint Rewriting and Error Correction in Flash Memories

Anxiao (Andrew) Jiang[†], Yue Li[†], Eyal En Gad[*], Michael Langberg[*‡], and Jehoshua Bruck[*]
[†]*Department of Computer Science and Engineering, Texas A&M University, College Station, TX 77843, USA*
[*]*Department of Electrical Engineering, California Institute of Technology, Pasadena, CA 91125, USA*
[‡]*Department of Mathematics and Computer Science, The Open University of Israel, Raanana 43107, Israel*
[†]*{ajiang, yli}@cse.tamu.edu* [*]*{eengad, mikel, bruck}@caltech.edu*

***Abstract*—The current NAND flash architecture requires block erasure to be triggered in order to decrease the level of a single cell inside a block. Block erasures degrade the quality of cells as well as the performance of flash memories. One solution is to model flash memories as write-once memories (WOM) where the level of a cell can only be increased. Various coding schemes for WOM can then be applied so that a block of cells can be rewritten multiple times without triggering a block erasure. Yet, due to the high storage density of flash memories, programming and reading memory cells bring nontrivial disturbance and interference, which introduce errors to the data. Therefore, both rewriting and error correction are important technologies for keeping flash memories efficient and reliable. However, coding schemes that combine them have been limited. This paper presents a new coding scheme that combines rewriting and error correction for the write-once memory model. Its construction is based on polar codes, and it supports any number of rewrites and corrects a substantial number of errors. The code is analyzed for the binary symmetric channel, and experimental results verify its performance. The results can be extended to multi-level cells and more general noise models.**

I. Introduction

The current NAND flash architecture requires block erasure to be triggered in order to decrease the level of a cell inside a block. Block erasures degrade the quality of cells as well as the performance of flash memories. Coding for rewriting is an important technology for mitigating such issues in flash memories. It has the potential to substantially increase their longevity, speed and power efficiency. Since its proposition in recent years [2], [7], lots of works have appeared in this area [14], [15], [16], [17]. The most basic model for rewriting is a write-once memory (WOM) model [10], where a set of binary cells are used to store data, and the cell levels can only increase when the data are rewritten. For flash memories, this constraint implies that the rewriting operation will delay the expensive block erasure, which leads to better preservation of cell quality and higher writing performance.

There have been many techniques for the design of WOM codes. They include linear code, tabular code, codes based on projective geometry, coset coding, etc. [4], [9], [10] Codes with substantially higher rates were discovered in recent years [14], [16]. In 2012, WOM codes that achieve capacity were discovered by Shpilka *et al.* [11], [12], [18] and Burshtein *et al.* [3]. The latter code used a very interesting construction based on polar coding. It should be noted that polar coding is now also used to construct rewriting codes for the rank modulation scheme [5].

Compared to the large amount of work on WOM codes, the work on WOM codes that also correct errors has been much more limited. Existing works are mainly on correcting a few errors (for example, 1, 2, or 3 errors [19], [20]). However, for rewriting to be widely used in flash memories, it is important to design WOM codes that can correct a substantial number of errors.

In this paper, we present a new coding scheme that combines rewriting with error correction. It supports any number of rewrites and can correct a substantial number of errors. The code construction uses polar coding. Our analytical technique is based on the frozen sets corresponding to the WOM channel and the error channel, respectively, including their common degrading and common upgrading channels. We present lower bounds to the sum-rate achieved by our code. The actual sum-rates are further computed for various parameters. The analysis focuses on the binary symmetric channel (BSC). An interesting observation is that in practice, for relatively small error probabilities, the frozen set for BSC is often contained in the frozen set for the WOM channel, which enables our code to have a nested structure. The code can be further extended to multi-level cells (MLC) and more general noise models.

The rest of the paper is organized as follows. In Section II, we present the basic model and notations. In Section III, we present the code construction. In Section IV, we analyze the code for BSC. In Section V, we discuss the code's extensions. In Section VI, we experimentally analyze the actual sum-rates achieved by our code. In Section VII, we present the concluding remarks.

II. Basic model

Let there be $N = 2^m$ cells that are used to store data. Every cell has two levels: 0 and 1. It can change only from level 0 to level 1, but not vice versa. That is called a WOM cell [10].

A sequence of t messages $M_1, M_2, \cdots, M_t$ will be written into the WOM cells, and when M_i is written, we do not need to remember the value of the previous messages. (Let $\mathcal{M}_j$ denote the number of bits in the message M_j, and let $M_j \in \{0,1\}^{\mathcal{M}_j}$.)

A shorter version of this paper has been published in the proceedings of 2013 IEEE International Symposium of Information Theory.

For simplicity, we assume the cells are all at level 0 before the first write happens.

After cells are programmed, noise will appear in the cell levels. For now, we consider noise to be a BSC with error probability p, denoted by $BSC(p)$. These errors are hard errors, namely, they physically change the cell levels from 0 to 1 or from 1 to 0. For flash memories, such errors can be caused by read/write disturbs, interference and charge leakage, and are quite common.

A. The model for rewriting

A code for rewriting and error correction consists of t encoding functions $\mathbf{E}_1, \mathbf{E}_2, \cdots, \mathbf{E}_t$ and t decoding functions $\mathbf{D}_1, \mathbf{D}_2, \cdots, \mathbf{D}_t$. For $i = 1, 2, \cdots, N$ and $j = 1, 2, \cdots, t$, let $s_{i,j} \in \{0,1\}$ and $s'_{i,j} \in \{0,1\}$ denote the level of the i-th cell right before and after the j-th write, respectively. We require $s'_{i,j} \geq s_{i,j}$. Let $c_{i,j} \in \{0,1\}$ denote the level of the i-th cell at any time after the j-th write and before the $(j+1)$-th write, when reading of the message M_j can happen. The error $c_{i,j} \oplus s'_{i,j} \in \{0,1\}$ is the error in the i-th cell caused by the noise channel $BSC(p)$. (Here $\oplus$ is an XOR function.) For $j = 1, 2, \cdots, t$, the encoding function

$$\mathbf{E}_j : \{0,1\}^N \times \{0,1\}^{\mathcal{M}_j} \to \{0,1\}^N$$

changes the cell levels from $\mathbf{s}_j = (s_{1,j}, s_{2,j}, \cdots, s_{N,j})$ to $\mathbf{s}'_j = (s'_{1,j}, s'_{2,j}, \cdots, s'_{N,j})$ given the initial cell state $\mathbf{s}_j$ and the message to store M_j. (Namely, $\mathbf{E}_j(\mathbf{s}_j, M_j) = \mathbf{s}'_j$.) When the reading of M_j happens, the decoding function

$$\mathbf{D}_j : \{0,1\}^N \to \{0,1\}^{\mathcal{M}_j}$$

recovers the message M_j given the noisy cell state $\mathbf{c}_j = (c_{1,j}, c_{2,j}, \cdots, c_{N,j})$. (Namely, $\mathbf{D}_j(\mathbf{c}_j) = M_j$.)

For $j = 1, \cdots, t$, $R_j = \frac{\mathcal{M}_j}{N}$ is called the rate of the j-th write. $R_{\text{sum}} = \sum_{j=1}^{t} R_j$ is called the sum-rate of the code. When there is no noise, the maximum sum-rate of WOM code is known to be $\log_2(t+1)$; however, for noisy WOM, the maximum sum-rate is still largely unknown [6].

B. Polar codes

We give a short introduction to polar codes due to its relevance to our code construction. A polar code is a linear block error correcting code proposed by Arıkan [1]. It is the first known code with an explicit construction that provably achieves the channel capacity of symmetric binary-input discrete memoryless channels (B-DMC). The encoder of a polar code transforms N input bits $\mathbf{u} = (u_1, u_2, \cdots, u_N)$ to N codeword bits $\mathbf{x} = (x_1, x_2, \cdots, x_N)$ through a linear transformation. (In [1], $\mathbf{x} = \mathbf{u}G_2^{\otimes m}$ where $G_2 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$, and $G_2^{\otimes m}$ is the m-th Kronecker product of G_2.) The N codeword bits $(x_1, x_2, \cdots, x_N)$ are transmitted through N independent copies of a B-DMC. For decoding, N transformed binary input channels $\{W_N^{(1)}, W_N^{(2)}, \cdots, W_N^{(N)}\}$ can be synthesized for $u_1, u_2, \cdots, u_N$, respectively. The channels are polarized such that for large N, the fraction of indices i for which $I(W_N^{(i)})$ is nearly 1 approaches the capacity of the B-DMC [1], while the values of $I(W_N^{(i)})$ for the remaining indices i are nearly 0. The latter set of indices are called the frozen set. For error correction, the u_i's with i in the frozen set take fixed values, and the other u_i's are used as information bits. A successive cancellation (SC) decoding algorithm achieves diminishing block error probability as N increases.

Polar code can also be used for optimal lossy source coding [8], which has various applications. In particular, in [3], the idea was used to build capacity achieving WOM codes.

Our code analysis uses the concept of upgrading and degrading channels, defined based on frozen sets. As in [13], a channel $W' : X \to Z$ is called "degraded with respect to a channel $W : X \to Y$" if an equivalent channel of W' can be constructed by concatenating W with an additional channel $Q : Y \to Z$, where the inputs of Q are linked with the outputs of W. That is,

$$W'(z|x) = \sum_{y \in Y} W(y|x)Q(z|y)$$

We denote it by $W' \preceq W$. Equivalently, the channel W is called "an upgrade with respect to W'", denoted by $W \succeq W'$.

III. Code construction

In this section, we introduce our code construction that combines rewriting with error correction.

A. Basic code construction with a nested structure

1) Basic concepts: First, let us consider a single rewrite step (namely, one of the t writes). Let $\mathbf{s} = (s_1, s_2, \cdots, s_N) \in \{0,1\}^N$ and $\mathbf{s}' = (s'_1, s'_2, \cdots, s'_N) \in \{0,1\}^N$ denote the cell levels right before and after this rewrite, respectively. Let $\mathbf{g} = (g_1, g_2, \cdots, g_n)$ be a pseudo-random bit sequence with i.i.d. bits that are uniformly distributed. The value of $\mathbf{g}$ is known to both the encoder and the decoder, and $\mathbf{g}$ is called a *dither*.

For $i = 1, 2, \cdots, N$, let $v_i = s_i \oplus g_i \in \{0,1\}$ and $v'_i = s'_i \oplus g_i \in \{0,1\}$ be the *value* of the i-th cell before and after the rewrite, respectively. As in [3], we build the WOM channel in Figure 1 for this rewrite, denoted by $WOM(\alpha, \epsilon)$. Here

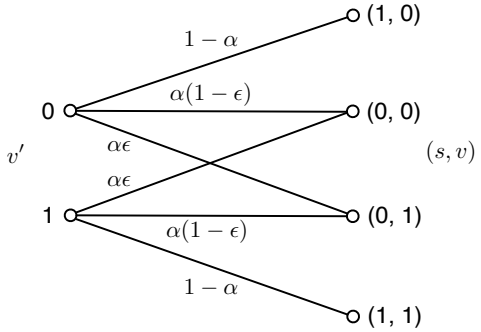

Fig. 1. The WOM channel $WOM(\alpha, \epsilon)$.

$\alpha \in [0,1]$ and $\epsilon \in [0, \frac{1}{2}]$ are given parameters, with $\alpha = 1 - \sum_{i=1}^{N} \frac{s_i}{N}$ representing the fraction of cells at level 0 before

the rewrite, and $\epsilon = \frac{\sum_{i=1}^{N} s'_i - s_i}{N - \sum_{i=1}^{N} s_i}$ representing the fraction of cells that are changed from level 0 to level 1 by the rewrite. Let $F_{\mathrm{WOM}(\alpha,\epsilon)} \subseteq \{1, 2, \cdots, N\}$ be the frozen set of the polar code corresponding to this channel $WOM(\alpha, \epsilon)$. It is known that $\lim_{N\to\infty} \frac{|F_{\mathrm{WOM}(\alpha,\epsilon)}|}{N} = \alpha\, \mathrm{H}(\epsilon)$. [3]

For the noise channel $BSC(p)$, let $F_{\mathrm{BSC}(p)} \subseteq \{1, 2, \cdots, N\}$ be the frozen set of the polar code corresponding to the channel $BSC(p)$. It is known that $\lim_{N\to\infty} \frac{|F_{\mathrm{BSC}(p)}|}{|N|} = \mathrm{H}(p)$.

In this subsection, we assume $F_{\mathrm{BSC}(p)} \subseteq F_{\mathrm{WOM}(\alpha,\epsilon)}$. It is as illustrated in Figure 2(a). In this case, the code has a nice nested structure: for any message $M \in \{0,1\}^{\mathcal{M}}$, the set of cell values $V_M \subseteq \{0,1\}^N$ that represent the message M is a linear subspace of a linear error correcting code (ECC) for the noise channel $BSC(p)$, and $\{V_M | M \in \{0,1\}^{\mathcal{M}}\}$ form a partition of the ECC. Later we will extend the code to general cases.

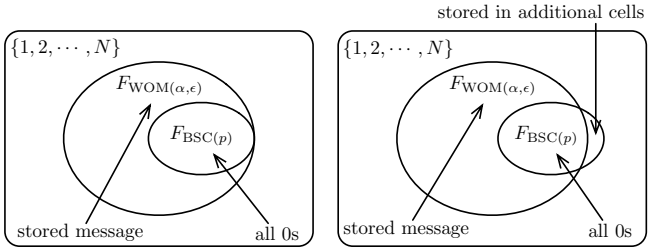

Fig. 2. (a) Nested code for $F_{\mathrm{BSC}(p)} \subseteq F_{\mathrm{WOM}(\alpha,\epsilon)}$. (b) General code.

2) The encoder: Let $\mathbf{E} : \{0,1\}^N \times \{0,1\}^{\mathcal{M}} \to \{0,1\}^N$ be the encoder for this rewrite. Namely, given the current cell state $\mathbf{s}$ and the message to write $M \in \{0,1\}^{\mathcal{M}}$, the encoder needs to find a new cell state $\mathbf{s}' = \mathbf{E}(\mathbf{s}, M)$ that represents M and is above $\mathbf{s}$ (that is, cell levels only increase).

The encoding process is similar to [3], but with some difference in how to assign bits to $F_{\mathrm{WOM}(\alpha,\epsilon)}$. For convenience of presentation, here we assume the polar code to be the original code designed by Arıkan [1]; however, note that it can be generalized to other polar codes as well. We present the encoding function in Algorithm 1. Here $\mathbf{y}$ and $\mathbf{u}$ are two vectors of length N; $\mathbf{u}_{F_{\mathrm{WOM}(\alpha,\epsilon)} - F_{\mathrm{BSC}(p)}} \triangleq \{u_i | i \in F_{\mathrm{WOM}(\alpha,\epsilon)} - F_{\mathrm{BSC}(p)}\}$ are all the bits u_i in the frozen set $F_{\mathrm{WOM}(\alpha,\epsilon)}$ but not $F_{\mathrm{BSC}(p)}$; $\mathbf{u}_{F_{\mathrm{BSC}(p)}} \triangleq \{u_i | i \in F_{\mathrm{BSC}(p)}\}$ are all the bits u_i in $F_{\mathrm{BSC}(p)}$; and $G_2^{\otimes m}$ is the m-th Kronecker product of $G_2 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$.

3) The decoder: We now present the decoder $\mathbf{D} : \{0,1\}^N \to \{0,1\}^{\mathcal{M}}$. Let $\mathbf{c} = (c_1, c_2, \cdots, c_N) \in \{0,1\}^N$ be the noisy cell levels after the message is written. Given $\mathbf{c}$, the decoder should recover the message as $\mathbf{D}(\mathbf{c}) = M$.

Our decoder works essentially the same way as a polar error correcting code. We present it as Algorithm 2.

By [1], it is easy to see that both the encoding and the decoding algorithms have time complexity $\mathcal{O}(N \log N)$.

4) Nested code for t writes: In the above, we have presented the encoder and the decoder for one rewrite. It can be naturally applied to a t-write error correcting WOM code as follows. For $j = 1, 2, \cdots, t$, for the j-th write, replace α, ϵ, $\mathbf{s}$, $\mathbf{s}'$, $\mathbf{v}$, $\mathbf{v}'$, M, $\mathcal{M}$, $\mathbf{E}$, $\mathbf{D}$, $\mathbf{c}$, $\hat{M}$, $\hat{\mathbf{v}}$ by α_{j-1}, ϵ_j, $\mathbf{s}_j$, $\mathbf{s}'_j$, $\mathbf{v}_j$, $\mathbf{v}'_j$, M_j, $\mathcal{M}_j$, $\mathbf{E}_j$, $\mathbf{D}_j$, $\mathbf{c}_j$, $\hat{M}_j$, $\hat{\mathbf{v}}_j$, respectively, and apply the above encoder and decoder.

Note that when $N \to \infty$, the values of $\alpha_1, \alpha_2, \cdots, \alpha_{t-1}$ can be computed using $\epsilon_1, \epsilon_2, \cdots, \epsilon_{t-1}$: for $BSC(p)$, $\alpha_j = \alpha_{j-1}(1-\epsilon_j)(1-p) + (1-\alpha_{j-1}(1-\epsilon_j))p$. Optimizing the code means to choose optimal values for $\epsilon_1, \epsilon_2, \cdots, \epsilon_t$ that maximize the sum-rate.

Algorithm 1 The encoding function $\mathbf{s}' = \mathbf{E}(\mathbf{s}, M)$

$\mathbf{y} \leftarrow ((s_1, v_1), (s_2, v_2), \cdots, (s_N, v_N))$.
Let $\mathbf{u}_{F_{\mathrm{WOM}(\alpha,\epsilon)} - F_{\mathrm{BSC}(p)}} \leftarrow M$.
Let $\mathbf{u}_{F_{\mathrm{BSC}(p)}} \leftarrow (0, 0, \cdots, 0)$.
for i from 1 to N **do**
  if $i \notin F_{\mathrm{WOM}(\alpha,\epsilon)}$ **then**
    $L_N^{(i)}(\mathbf{y}, (u1, u2, \cdots, u_{i-1})) \leftarrow \frac{W_N^{(i)}(\mathbf{y}, (u1, u2, \cdots, u_{i-1}) | u_i = 0)}{W_N^{(i)}(\mathbf{y}, (u1, u2, \cdots, u_{i-1}) | u_i = 1)}$.
    (Comment: Here $W_N^{(i)}(\mathbf{y}, (u1, u2, \cdots, u_{i-1}) | u_i = 0)$ and $W_N^{(i)}(\mathbf{y}, (u1, u2, \cdots, u_{i-1}) | u_i = 1)$ can be computed recursively using formulae (22), (23) in [1]).
    Let $u_i \leftarrow \begin{cases} 0 & \text{with probability } \frac{L_N^{(i)}}{1+L_N^{(i)}} \\ 1 & \text{with probability } \frac{1}{1+L_N^{(i)}} \end{cases}$.
Let $\mathbf{v}' \leftarrow \mathbf{u} G_2^{\otimes m}$.
Let $\mathbf{s}' \leftarrow \mathbf{v}' \oplus \mathbf{g}$.

Algorithm 2 The decoding function $\hat{M} = \mathbf{D}(\mathbf{c})$

View $\mathbf{c} \oplus \mathbf{g}$ as a noisy codeword, which is the output of a binary symmetric channel $BSC(p)$. Decode $\mathbf{c} \oplus \mathbf{g}$ using the decoding algorithm of the polar error-correcting code [1], where the bits in the frozen set $F_{\mathrm{BSC}(p)}$ are set to 0s. Let $\hat{\mathbf{v}} = (\hat{v}_1, \hat{v}_2, \cdots, \hat{v}_N)$ be the recovered codeword.
Let $\hat{M} \leftarrow \left(\hat{\mathbf{v}}(G_2^{\otimes m})^{-1}\right)_{F_{\mathrm{WOM}(\alpha,\epsilon)} - F_{\mathrm{BSC}(p)}}$, which denotes the elements of the vector $\hat{\mathbf{v}}(G_2^{\otimes m})^{-1}$ whose indices are in the set $F_{\mathrm{WOM}(\alpha,\epsilon)} - F_{\mathrm{BSC}(p)}$.

B. Extended code construction

We have introduced the code for the case $F_{\mathrm{BSC}(p)} \subseteq F_{\mathrm{WOM}(\alpha,\epsilon)}$ so far. Our experiments show that for relatively small p and typical values of $(\alpha_0, \epsilon_1), (\alpha_1, \epsilon_2), \cdots, (\alpha_{t-1}, \epsilon_t)$, the above condition holds. We now consider the general case where $F_{\mathrm{BSC}(p)}$ is not necessarily a subset of $F_{\mathrm{WOM}(\alpha,\epsilon)}$.

We first revise the encoder in Algorithm 1 as follows. After all the steps in the algorithm, we store the bits in $\mathbf{u}_{F_{\mathrm{BSC}(p)} - F_{\mathrm{WOM}(\alpha,\epsilon)}}$ using $N_{\mathrm{additional},j}$ cells (for the j-th write). (It is illustrated in Figure 2(b).) In this paper, for simplicity, we assume the bits in $\mathbf{u}_{F_{\mathrm{BSC}(p)} - F_{\mathrm{WOM}(\alpha,\epsilon)}}$ are stored using just an error correcting code designed for the noise channel $BSC(p)$. (It will not be hard to see that we can also store it using an error-correcting WOM code, such

as the one presented above, for higher rates. However, we skip the details for simplicity.) Therefore, we can have $\lim_{N\to\infty} \frac{N_{\text{additional},j}}{|F_{\text{BSC}(p)} - F_{\text{WOM}(\alpha_{j-1},\epsilon_j)}|} = \frac{1}{1-\text{H}(p)}$. And the sum-rate becomes $R_{\text{sum}} = \frac{\sum_{j=1}^{t} \mathcal{M}_j}{N + \sum_{j=1}^{t} N_{\text{additional},j}}$.

We now revise the decoder in Algorithm 2 as follows. First recover the bits in $\mathbf{u}_{F_{\text{BSC}(p)} - F_{\text{WOM}(\alpha,\epsilon)}}$ using the decoding algorithm of the ECC for the $N_{\text{additional},j}$ additional cells. Then carry out all the steps in Algorithm 2, except that the bits in $F_{\text{BSC}(p)} - F_{\text{WOM}(\alpha,\epsilon)}$ are known to the decoder as the above recovered values instead of 0s.

IV. Code analysis for BSC

In this section, we prove the correctness of the above code construction, and analyze its performance.

A. Correctness of the code

We first prove the correctness of our code. First, the encoder in Algorithm 1 works similarly to the WOM code encoder in [3], with an exception that the bits in $F_{\text{WOM}(\alpha,\epsilon)}$ are not all occupied by the message M; instead, the bits in its subset $F_{\text{WOM}(\alpha,\epsilon)} \cap F_{\text{BSC}(p)}$ are set to be constant values: all 0s. Therefore, it successfully rewrites data in the same way as the code in [3]. Next, the decoder in Algorithm 2 recovers the cell values from noise in the same way as the standard polar ECC. Then, the stored message M is extracted from it.

One important thing to note is that although the physical noise acts on the cell levels $\mathbf{s} = (s_1, s_2, \cdots, s_N)$, the error correcting code we use in our construction is actually for the cell values $\mathbf{v} = (v_1, v_2, \cdots, v_n) = (s_1 \oplus g_1, s_2 \oplus g_2, \cdots, s_N \oplus g_N)$. However, the pseudo-random dither $\mathbf{g}$ has independent and uniformly distributed elements; so when the noise channel for $\mathbf{s}$ is $BSC(p)$, the corresponding noise channel for $\mathbf{v}$ is also $BSC(p)$.

B. The size of $F_{\text{WOM}(\alpha,\epsilon)} \cap F_{\text{BSC}(p)}$

We have seen that if $F_{\text{BSC}(p)} \subseteq F_{\text{WOM}(\alpha,\epsilon)}$, the code has a very interesting nested structure. In general, it is also interesting to understand how large the intersection $F_{\text{WOM}(\alpha,\epsilon)} \cap F_{\text{BSC}(p)}$ can be. For convenience of presentation, we consider one rewrite as in Section III-A, where the parameters are α and ϵ (instead of α_{j-1}, ϵ_j).

Lemma 1. *When* $\text{H}(p) \le \alpha\,\text{H}(\epsilon)$, $\lim_{N\to\infty} \frac{|F_{\text{BSC}(p)}|}{N} \le \lim_{N\to\infty} \frac{|F_{\text{WOM}(\alpha,\epsilon)}|}{N}$.

Proof: $\lim_{N\to\infty} \frac{|F_{\text{BSC}(p)}|}{N} = \text{H}(p) \le \alpha\,\text{H}(\epsilon) = \lim_{N\to\infty} \frac{|F_{\text{WOM}(\alpha,\epsilon)}|}{N}$. ∎

Lemma 2. *When* $p \le \alpha\epsilon$,

$$F_{\text{WOM}(\alpha,\frac{p}{\alpha})} \subseteq \left(F_{\text{BSC}(p)} \cap F_{\text{WOM}(\alpha,\epsilon)}\right),$$

and

$$\left(F_{\text{WOM}(\alpha,\epsilon)} \cup F_{\text{BSC}(p)}\right) \subseteq F_{\text{BSC}(\alpha\epsilon)}.$$

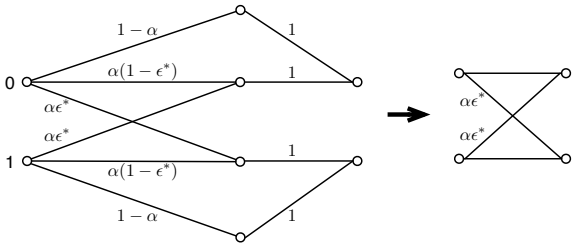

Fig. 3. Degrading the channel $WOM(\alpha, \epsilon^*)$ to $BSC(\alpha\epsilon^*)$. The two channels on the left and on the right are equivalent.

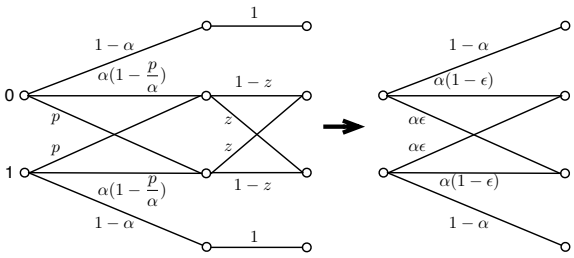

Fig. 4. Degrading channel $WOM(\alpha, \frac{p}{\alpha})$ to $WOM(\alpha, \epsilon)$. Here $z = \frac{\alpha\epsilon - p}{\alpha - 2p}$. The two channels on the left and on the right are equivalent.

Proof: (1) In Figure 3, by setting $\epsilon^* = \frac{p}{\alpha}$, we see that $BSC(p) \preceq WOM(\alpha, \frac{p}{\alpha})$. Therefore $F_{\text{WOM}(\alpha,\frac{p}{\alpha})} \subseteq F_{\text{BSC}(p)}$.

(2) In Figure 4, we can see that $WOM(\alpha,\epsilon) \preceq WOM(\alpha, \frac{p}{\alpha})$. Therefore, $F_{\text{WOM}(\alpha,\frac{p}{\alpha})} \subseteq F_{\text{WOM}(\alpha,\epsilon)}$.

(3) In Figure 3, by setting $\epsilon^* = \epsilon$, we see that $BSC(\alpha\epsilon) \preceq WOM(\alpha,\epsilon)$. Therefore $F_{\text{WOM}(\alpha,\epsilon)} \subseteq F_{\text{BSC}(\alpha\epsilon)}$.

(4) Since $p \le \alpha\epsilon$, clearly $BSC(\alpha\epsilon) \preceq BSC(p)$. Therefore $F_{\text{BSC}(p)} \subseteq F_{\text{BSC}(\alpha\epsilon)}$. ∎

We illustrate the meaning of Lemma 2 in Figure 5.

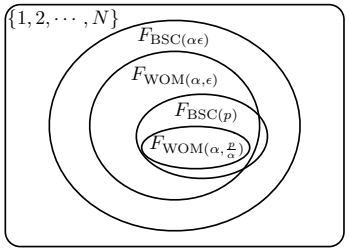

Fig. 5. The frozen sets for channels $BSC(p)$, $WOM(\alpha,\epsilon)$, $WOM(\alpha, \frac{p}{\alpha})$ and $BSC(\alpha\epsilon)$. Here $p \le \alpha\epsilon$.

Lemma 3. *When* $p \le \alpha\epsilon$, $\lim_{N\to\infty} \frac{|F_{\text{WOM}(\alpha,\epsilon)} \cap F_{\text{BSC}(p)}|}{N} \ge \lim_{N\to\infty} \frac{|F_{\text{WOM}(\alpha,\frac{p}{\alpha})}|}{N} = \alpha\,\text{H}(\frac{p}{\alpha})$.

Lemma 4. *When* $p \le \alpha\epsilon$, $\lim_{N\to\infty} \frac{|F_{\text{WOM}(\alpha,\epsilon)} \cap F_{\text{BSC}(p)}|}{N} \ge \lim_{N\to\infty} \frac{|F_{\text{WOM}(\alpha,\epsilon)}| + |F_{\text{BSC}(p)}| - |F_{\text{BSC}(\alpha\epsilon)}|}{N} = \alpha\,\text{H}(\epsilon) + \text{H}(p) - \text{H}(\alpha\epsilon)$.

Proof: $|F_{\mathrm{WOM}(\alpha,\epsilon)} \cap F_{\mathrm{BSC}(p)}| = |F_{\mathrm{WOM}(\alpha,\epsilon)}| + |F_{\mathrm{BSC}(p)}| - |F_{\mathrm{WOM}(\alpha,\epsilon)} \cup F_{\mathrm{BSC}(p)}| \geq |F_{\mathrm{WOM}(\alpha,\epsilon)}| + |F_{\mathrm{BSC}(p)}| - |F_{\mathrm{BSC}(\alpha\epsilon)}|$ (by Lemma 2). ■

C. Lower bound to sum-rate

We now analyze the sum-rate of our general code construction as $N \to \infty$. Let $x_j \triangleq \frac{|F_{\mathrm{WOM}(\alpha_{j-1},\epsilon_j)} \cap F_{\mathrm{BSC}(p)}|}{|F_{\mathrm{BSC}(p)}|} \leq 1$. For $j = 1, 2, \cdots, t$, the number of bits written in the j-th rewrite is

$$\begin{aligned}\mathcal{M}_j =& |F_{\mathrm{WOM}(\alpha_{j-1},\epsilon_j)}| - |F_{\mathrm{WOM}(\alpha_{j-1},\epsilon_j)} \cap F_{\mathrm{BSC}(p)}| \\ =& N\alpha_{j-1}\,\mathrm{H}(\epsilon_j) - x_j|F_{\mathrm{BSC}(p)}| \\ =& N(\alpha_{j-1}\,\mathrm{H}(\epsilon_j) - x_j\,\mathrm{H}(p))\end{aligned}$$

and the number of additional cells we use to store the bits in $F_{\mathrm{BSC}(p)} - F_{\mathrm{WOM}(\alpha_{j-1},\epsilon_j)}$ is

$$N_{\mathrm{additional},j} = \frac{N\,\mathrm{H}(p)(1-x_j)}{1-\mathrm{H}(p)}$$

Therefore, the sum-rate is $R_{\mathrm{sum}} \triangleq \frac{\sum_{j=1}^{t}\mathcal{M}_j}{N+\sum_{j=1}^{t}N_{\mathrm{additional},j}}$

$$\begin{aligned}=& \frac{\sum_{j=1}^{t}\alpha_{j-1}\,\mathrm{H}(\epsilon_j) - \mathrm{H}(p)\sum_{j=1}^{t}x_j}{1+\frac{\mathrm{H}(p)}{1-\mathrm{H}(p)}\sum_{j=1}^{t}(1-x_j)} \\ =& \frac{(1-\mathrm{H}(p))\sum_{j=1}^{t}\alpha_{j-1}\,\mathrm{H}(\epsilon_j) - \mathrm{H}(p)(1-\mathrm{H}(p))\sum_{j=1}^{t}x_j}{(1-\mathrm{H}(p)+\mathrm{H}(p)t) - \mathrm{H}(p)\sum_{j=1}^{t}x_j} \\ =& (1-\mathrm{H}(p))\cdot\frac{\frac{1}{\mathrm{H}(p)}\sum_{j=1}^{t}\alpha_{j-1}\,\mathrm{H}(\epsilon_j) - \sum_{j=1}^{t}x_j}{\frac{1-\mathrm{H}(p)+\mathrm{H}(p)t}{\mathrm{H}(p)} - \sum_{j=1}^{t}x_j}.\end{aligned}$$

Let $\gamma_j \triangleq \max\left\{\frac{\alpha_{j-1}\,\mathrm{H}(\frac{p}{\alpha_{j-1}})}{\mathrm{H}(p)}, \frac{\alpha_{j-1}\,\mathrm{H}(\epsilon_j)+\mathrm{H}(p)-\mathrm{H}(\alpha_{j-1}\epsilon_j)}{\mathrm{H}(p)}\right\}$.

Lemma 5. *Let $0 < p \leq \alpha_{j-1}\epsilon_j$. Then $x_j \geq \gamma_j$.*

Proof: By Lemma 3, we have

$$\begin{aligned}x_j =& \frac{|F_{\mathrm{WOM}(\alpha_{j-1},\epsilon_j)} \cap F_{\mathrm{BSC}(p)}|}{|F_{\mathrm{BSC}(p)}|} \\ \geq& \frac{|F_{\mathrm{WOM}(\alpha_{j-1},\frac{p}{\alpha_{j-1}})}|}{|F_{\mathrm{BSC}(p)}|} = \frac{\alpha_{j-1}\,\mathrm{H}(\frac{p}{\alpha_{j-1}})}{\mathrm{H}(p)}.\end{aligned}$$

By Lemma 4, we also have

$$\begin{aligned}x_j =& \frac{|F_{\mathrm{WOM}(\alpha_{j-1},\epsilon_j)} \cap F_{\mathrm{BSC}(p)}|}{|F_{\mathrm{BSC}(p)}|} \\ \geq& \frac{|F_{\mathrm{WOM}(\alpha_{j-1},\epsilon_j)}| + |F_{\mathrm{BSC}(p)}| - |F_{\mathrm{BSC}(\alpha_{j-1}\epsilon_j)}|}{|F_{\mathrm{BSC}(p)}|} \\ =& \frac{\alpha_{j-1}\,\mathrm{H}(\epsilon_j) + \mathrm{H}(p) - \mathrm{H}(\alpha_{j-1}\epsilon_j)}{\mathrm{H}(p)}.\end{aligned}$$

■

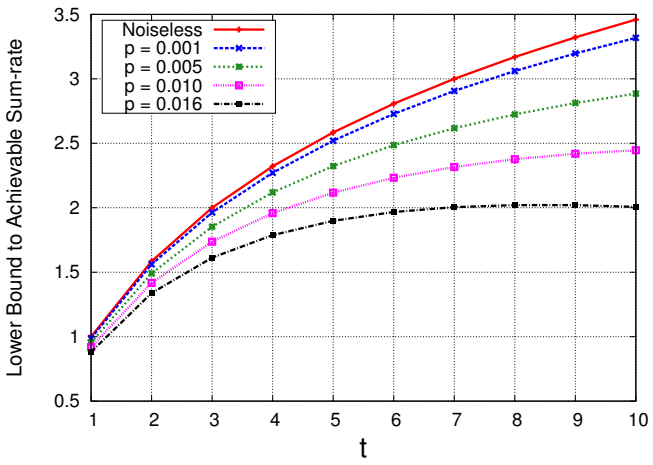

Fig. 6. Lower bound to achievable sum-rates for different error probability p.

Theorem 6 *Let $0 < p \leq \alpha_{j-1}\epsilon_j$ for $j = 1, 2, \cdots, t$. If $\sum_{j=1}^{t}\alpha_{j-1}\,\mathrm{H}(\epsilon_j) \geq 1 - \mathrm{H}(p) + \mathrm{H}(p)t$, then the sum-rate R_{sum} is lower bounded by*

$$(1-\mathrm{H}(p))\frac{\sum_{j=1}^{t}\left(\alpha_{j-1}\,\mathrm{H}(\epsilon_j) - \mathrm{H}(p)\gamma_j\right)}{1-\mathrm{H}(p)+\mathrm{H}(p)t - \mathrm{H}(p)\sum_{j=1}^{t}\gamma_j}.$$

If $\sum_{j=1}^{t}\alpha_{j-1}\,\mathrm{H}(\epsilon_j) < 1 - \mathrm{H}(p) + \mathrm{H}(p)t$, and $\mathrm{H}(p) \leq \alpha_{j-1}\,\mathrm{H}(\epsilon_j)$ for $j = 1, 2, \cdots, t$, then R_{sum} is lower bounded by

$$\left(\sum_{j=1}^{t}\alpha_{j-1}\,\mathrm{H}(\epsilon_j)\right) - \mathrm{H}(p)t.$$

Proof: If $\sum_{j=1}^{t}\alpha_{j-1}\,\mathrm{H}(\epsilon_j) \geq 1 - \mathrm{H}(p) + \mathrm{H}(p)t$, the sum-rate is minimized when x_j ($j = 1, 2, \cdots, t$) takes the minimum value, and we have $x_j \geq \gamma_j$. Otherwise, the sum-rate is minimized when x_j takes the maximum value 1. ■

We show some numerical results of the lower bound to sum-rate R_{sum} in Figure 6, where we let $\epsilon_i = \frac{1}{2+t-i}$. The curve for $p = 0$ is the optimal sum-rate for noiseless WOM code. The other four curves are the lower bounds for noisy WOM with $p = 0.001$, $p = 0.005$, $p = 0.010$ and $p = 0.016$, respectively, given by Theorem 6. Note that it is possible to further increase the lower bound values by optimizing ϵ_i. We also show in Figure 7 the lower bound to sum-rate when each step writes the same number of bits.

V. Extensions

We now consider more general noise models. For simplicity, we discuss it for an erasure channel. But it can be easily extended to other noise models. Let the noise be a BEC with erasure probability p, denoted by $BEC(p)$. After a rewrite, noise appears in some cell levels (both level 0 and level 1) and changes them to erasures. An erasure represents a noisy cell level between 0 and 1. We handle erasures this way: before a rewrite, we first increase all the erased cell levels to 1, and then perform rewriting as before.

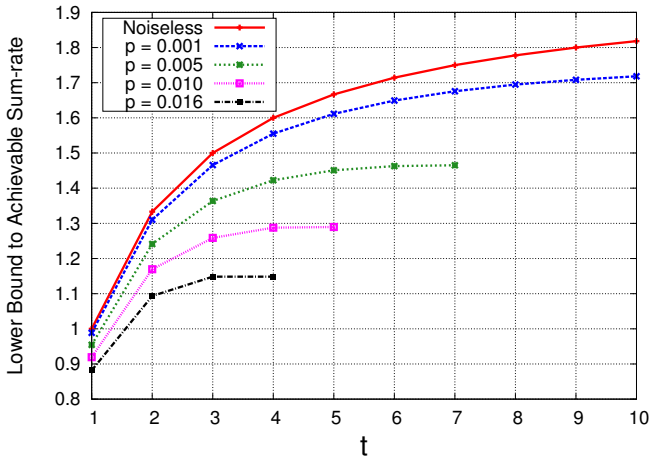

Fig. 7. Lower bound to achievable sum-rates for different error probability p. Here each rewriting step writes the same number of bits.

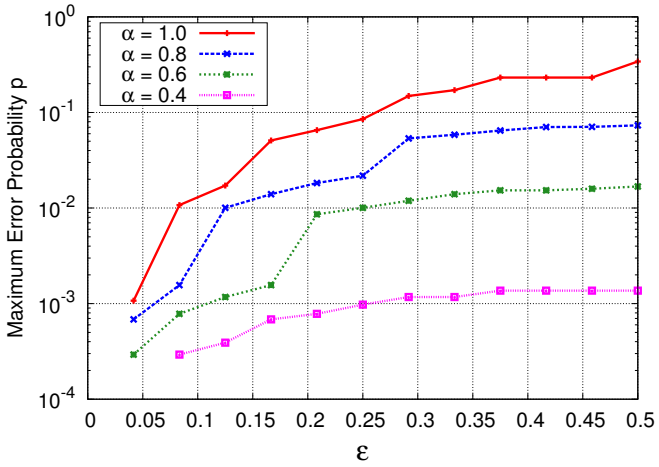

Fig. 8. The maximum value of p found for which $F_{\mathrm{BSC}(p)} \subseteq F_{\mathrm{WOM}(\alpha,\epsilon)}$.

Note that although the noise for cell levels is $BEC(p)$, when rewriting happens, the equivalent noise channel for the cell value $\mathbf{v} = \mathbf{s} \oplus \mathbf{g}$ is a $BSC(\frac{p}{2})$, because all the erased cell levels have been pushed to level 1, and dither has a uniform distribution. Therefore, the code construction and its performance analysis can be carried out the same way as before, except that we replace p by $\frac{p}{2}$.

The code can also be extended to multi-level cells (MLC), by using q-ary polar codes. We skip the details for simplicity.

VI. Experimental Results

In this section, we study the achievable rates of our error correcting WOM code, using polar codes of finite lengths. In the following, we assume the noise channel is $BSC(p)$, and search for good parameters $\epsilon_1, \epsilon_2, \cdots, \epsilon_t$ that achieve high sum-rate for rewriting. We also study when the code can have a nested structure, which simplifies the code construction.

A. Finding BSCs satisfying $F_{\mathrm{BSC}(p)} \subseteq F_{\mathrm{WOM}(\alpha,\epsilon)}$

The first question we endeavor to answer is when $BSC(p)$ satisfies the condition $F_{\mathrm{BSC}(p)} \subseteq F_{\mathrm{WOM}(\alpha,\epsilon)}$, which leads to an elegant nested code structure. We search for the answer experimentally. Let $N = 8192$. Let the polar codes be constructed using the method in [13]. To obtain the frozen sets, we let $|F_{\mathrm{WOM}(\alpha,\epsilon)}| = N(\alpha \mathrm{H}(\epsilon) - \Delta R)$, where $\Delta R = 0.025$ is a rate loss we considered for the polar code of the WOM channel [3]; and let $F_{\mathrm{BSC}(p)}$ be chosen with the target block error rate 10^{-5}.

The results are shown in Figure 8. The four curves correspond to $\alpha = 0.4$, 0.6, 0.8, and 1.0, respectively. The x-axis is ϵ, and the y-axis is the maximum value of p we found that satisfies $F_{\mathrm{BSC}(p)} \subseteq F_{\mathrm{WOM}(\alpha,\epsilon)}$. Clearly, the maximum value of p increases with both α and ϵ. And it has nontrivial values (namely, it is comparable to or higher than the typical error probabilities in memories).

B. Achievable sum-rates for nested code

We search for the achievable sum-rates of codes with a nested structure, namely, when the condition $F_{\mathrm{BSC}(p)} \subseteq F_{\mathrm{WOM}(\alpha_{j-1},\epsilon_j)}$ is satisfied for all $j = 1, 2, \cdots, t$. Given p, we search for $\epsilon_1, \epsilon_2, \cdots, \epsilon_t$ that maximize the sum-rate R_{sum}.

We show the results for t-write error-correcting WOM codes—for $t = 2, 3, 4, 5$—in Figure 9. (In the experiments, we let $N = 8192$, $\Delta R = 0.025$, and the target block error rate be 10^{-5}.) The x-axis is p, and the y-axis is the maximum sum-rate found in our algorithmic search. We see that the achievable sum-rate increases with the number of rewrites t.

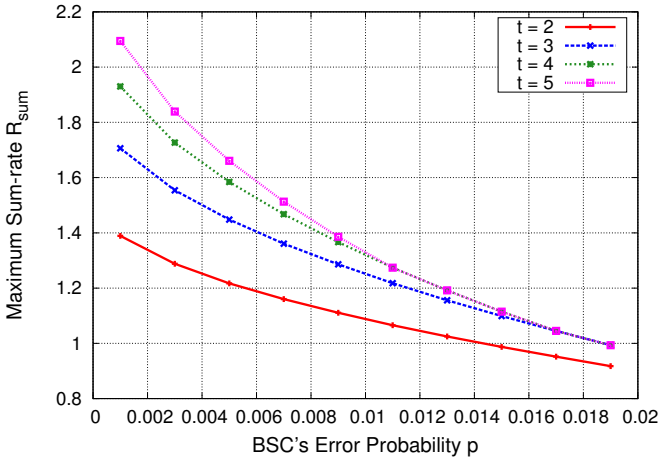

Fig. 9. Sum-rates for different t obtained in experimental search using code length $N = 8192$, when $F_{\mathrm{BSC}(p)} \subseteq F_{\mathrm{WOM}(\alpha,\epsilon)}$.

C. Achievable sum-rates for general code

We now search for the achievable sum-rates of the general code, when $F_{\mathrm{BSC}(p)}$ is not necessarily a subset of $F_{\mathrm{WOM}(\alpha_{j-1},\epsilon_j)}$. When p is given, the general code can search a larger solution space for $\epsilon_1, \epsilon_2, \cdots, \epsilon_t$ than the nested-code case, and therefore achieve higher sum-rates. However, for

relatively small p (*e.g.* $p < 0.016$), the gain in rate obtained in the experiments is quite small. This means the nested code is already performing well for this parameter range. For simplicity, we skip the details.

Note that the lower bound to sum-rate R_{sum} in Figure 6 is actually higher than the rates we have found through experiments by now. This is because the lower bound is for $N \to \infty$, while the codes in our experiments are still short so far and consider the rate loss ΔR. Better rates can be expected as we increase the code length and further improve our search algorithm due to the results indicated by the lower bound.

VII. Concluding remarks

This paper presents a new code construction for error-correcting WOM codes. It supports any number of rewrites and can correct a substantial number of errors. The construction is based on polar coding, and the results show that they achieve nice performance for both rewriting and error correction.

There are still a number of problems to be studied. For example, a stronger theoretical understanding is needed as to when the frozen set of one channel is contained in that of another channel. More generally, it is interesting to know how large the intersection of two frozen sets is. Those remain as our future research directions.

Acknowledgment

This work was supported in part by the NSF CAREER Award CCF-0747415, the NSF Grant CCF-1217944, a grant from Intellectual Ventures, the ISF grant 480/08 and BSF grant 2010075. This work was done while Michael Langberg was at the California Institute of Technology.

References

[1] E. Arıkan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. Inf. Theor.*, vol. 55, no. 7, pp. 3051–3073, July 2009.

[2] V. Bohossian, A. Jiang, and J. Bruck, "Buffer coding for asymmetric multi-level memory," in *Proc. IEEE International Symposium on Information Theory*, June 2007, pp. 1186 –1190.

[3] D. Burshtein and A. Strugatski, "Polar write once memory codes," in *Proc. IEEE International Symposium on Information Theory*, July 2012, pp. 1972 –1976.

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

[5] E. En Gad, E. Yaakobi, A. Jiang, and J. Bruck, "Rank-modulation rewriting codes for flash memories," submitted to IEEE International Symposium on Information Theory 2013.

[6] C. Heegard, "On the capacity of permanent memory," *IEEE Trans. Inf. Theor.*, vol. 31, no. 1, pp. 34–42, January 1985.

[7] A. Jiang, V. Bohossian, and J. Bruck, "Floating codes for joint information storage in write asymmetric memories," in *Proc. IEEE International Symposium on Information Theory*, June 2007, pp. 1166 –1170.

[8] S. Korada and R. Urbanke, "Polar codes are optimal for lossy source coding," *IEEE Trans. Inf. Theor.*, vol. 56, no. 4, pp. 1751–1768, April 2010.

[9] Merkx, "Womcodes constructed with projective geometries,," *Traitement du Signal*, vol. 1, no. 2-2, pp. 227–231, 1984.

[10] R. L. Rivest and A. Shamir, "How to reuse a write-once memory," *Information and Control*, vol. 55, no. 1-3, pp. 1–19, 1982.

[11] A. Shpilka, "Capacity achieving multiwrite wom codes," *CoRR*, vol. abs/1209.1128, 2012.

[12] ——, "Capacity achieving two-write wom codes," in *LATIN 2012: Theoretical Informatics*, ser. Lecture Notes in Computer Science, vol. 7256. Springer Berlin Heidelberg, 2012, pp. 631–642.

[13] I. Tal and A. Vardy, "How to construct polar codes," *CoRR*, vol. abs/1105.6164, 2011.

[14] Y. Wu, "Low complexity codes for writing a write-once memory twice," in *Proc. IEEE International Symposium on Information Theory*, June 2010, pp. 1928–1932.

[15] Y. Wu and A. Jiang, "Position modulation code for rewriting write-once memories," *IEEE Trans. Inf. Theor.*, vol. 57, no. 6, pp. 3692–3697, June 2011.

[16] E. Yaakobi, S. Kayser, P. H. Siegel, A. Vardy, and J. K. Wolf, "Codes for write-once memories," *IEEE Trans. Inf. Theor.*, vol. 58, no. 9, pp. 5985–5999, September 2012.

[17] E. Yaakobi, S. Kayser, P. Siegel, A. Vardy, and J. Wolf, "Efficient two-write wom-codes," in *Proc. IEEE Information Theory Workshop*, September 2010, pp. 1–5.

[18] E. Yaakobi and A. Shpilka, "High sum-rate three-write and non-binary wom codes," in *Proc. IEEE International Symposium on Information Theory*, July 2012, pp. 1386–1390.

[19] E. Yaakobi, P. Siegel, A. Vardy, and J. Wolf, "Multiple error-correcting wom-codes," *IEEE Trans. Inf. Theor.*, vol. 58, no. 4, pp. 2220–2230, April 2012.

[20] G. Zemor and G. D. Cohen, "Error-correcting wom-codes," *IEEE Trans. Inf. Theor.*, vol. 37, no. 3, pp. 730–734, May 1991.