Codes for Write-Once Memories

Eitan Yaakobi, *Student Member, IEEE*, Scott Kayser, Paul H. Siegel, *Fellow, IEEE*, Alexander Vardy, *Fellow, IEEE*, and Jack Keil Wolf, *Life Fellow, IEEE*

***Abstract*—A write-once memory (WOM) is a storage device that consists of *cells* that can take on q values, with the added constraint that rewrites can only increase a cell's value. A length-n, t-write WOM-code is a coding scheme that allows t messages to be stored in n cells. If on the ith write we write one of M_i messages, then the rate of this write is the ratio of the number of written bits to the total number of cells, i.e., $\log_2 M_i/n$. The *sum-rate* of the WOM-code is the sum of all individual rates on all writes. A WOM-code is called a *fixed-rate WOM-code* if the rates on all writes are the same, and otherwise, it is called a *variable-rate WOM-code*. We address two different problems when analyzing the sum-rate of WOM-codes. In the first one, called the fixed-rate WOM-code problem, the sum-rate is analyzed over all fixed-rate WOM-codes, and in the second problem, called the unrestricted-rate WOM-code problem, the sum-rate is analyzed over all fixed-rate and variable-rate WOM-codes. In this paper, we first present a family of two-write WOM-codes. The construction is inspired by the coset coding scheme, which was used to construct multiple-write WOM-codes by Cohen *et al.* and recently by Wu, in order to construct from each linear code a two-write WOM-code. This construction improves the best known sum-rates for the fixed- and unrestricted-rate WOM-code problems. We also show how to take advantage of two-write WOM-codes in order to construct codes for the Blackwell channel. The two-write construction is generalized for two-write WOM-codes with q levels per cell, which is used with ternary cells to construct three- and four-write binary WOM-codes. This construction is used recursively in order to generate a family of t-write WOM-codes for all t. A further generalization of these t-write WOM-codes yields additional families of efficient WOM-codes. Finally, we show a recursive method that uses the previously constructed WOM-codes in order to construct fixed-rate WOM-codes. We conclude and show that the WOM-codes constructed here outperform all previously known WOM-codes for $2 \leqslant t \leqslant 10$ for both the fixed- and unrestricted-rate WOM-code problems.**

***Index Terms*—Coding theory, flash memories, write-once memories (WOMs), WOM-codes.**

Manuscript received August 23, 2011; accepted February 15, 2012. Date of publication May 19, 2012; date of current version August 14, 2012. This work was supported in part by the University of California Lab Fees Research Program under Award 09-LR-06-118620-SIEP, in part by the National Science Foundation under Grant CCF-1116739, and in part by the Center for Magnetic Recording Research at the University of California, San Diego. This paper was presented in part at the 2010 IEEE Information Theory Workshop and in part at the 48th Annual Allerton Conference on Communications, Control, and Computing, Monticello, IL, September 29–October 3, 2010.

E. Yaakobi is with the Department of Electrical Engineering, California Institute of Technology, Pasadena, CA 91125 USA, and also with the Department of Electrical and Computer Engineering and the Center for Magnetic Recording Research, University of California, San Diego, La Jolla, CA 92093 USA (e-mail: eyaakobi@ucsd.edu).

S. Kayser and P. H. Siegel are with the Department of Electrical and Computer Engineering and the Center for Magnetic Recording Research, University of California, San Diego, La Jolla, CA 92093 USA (e-mail: skayser@ucsd.edu; psiegel@ucsd.edu).

A. Vardy is with the Department of Electrical and Computer Engineering, the Department of Computer Science and Engineering, and the Department of Mathematics, University of California, San Diego, La Jolla, CA 92093 USA (e-mail: avardy@ucsd.edu).

J. K. Wolf, deceased, was with the Department of Electrical and Computer Engineering and the Center for Magnetic Recording Research, University of California, San Diego, La Jolla, CA 92093 USA.

Communicated by G. Cohen, Associate Editor for Coding Theory.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIT.2012.2200291

TABLE I
WOM-CODE EXAMPLE

Data bits	First write	Second write (if data changes)
00	000	111
10	100	011
01	010	101
11	001	110

I. INTRODUCTION

WRITE-ONCE-MEMORY (WOM) codes were first introduced in 1982 by Rivest and Shamir [23]. They make it possible to record binary data more than once in a so-called write-once storage medium, such as a punch card or ablative optical disk. These media can be represented as a collection of write-once bit locations, each of which initially represents a bit value 0 that can be *irreversibly* overwritten with a bit value 1. A WOM-code allows the reuse of a write-once medium by introducing redundancy into the recorded bit sequence and, in subsequent write operations, observing the state of the medium before determining how to update the contents of the memory with a new bit sequence.

A simple example, presented in [23], enables the recording of two bits of information in three memory elements twice. The encoding and decoding rules for this WOM-code are described in a tabular form in Table I. It is easy to verify that after the first 2-bit data vector is encoded into a 3-bit codeword, if the second 2-bit data vector is different from the first, the 3-bit codeword into which it is encoded does not change any code bit 1 into a code bit 0, ensuring that it can be recorded in the write-once medium.

Flash memories impose constraints on recording that are similar to those associated with write-once memories. This connection was first brought in [3], [16], [17]. Flash memories contain floating gate cells. The cells are electrically charged with electrons and can represent multiple levels according to the number of electrons they contain [5]. The most conspicuous property of flash-storage technology is its inherent asymmetry between cell programming and cell erasing. While it is fast and simple to increase a cell level, reducing its level requires a long and cumbersome operation of first erasing its entire containing block ($\sim 10^6$ cells) and only then programming the cells [5]. Such block erasures are not only time consuming, but also degrade the lifetime of the memory. A typical block can generally tolerate at most 10^4–10^5 erasures. A WOM-code can be applied in this context to enable additional writes without first having to erase the entire block. The deferral of a block erasure is beneficial to the lifetime of the device. The cost associated with this increase in the endurance is the redundancy and the additional complexity associated with the encoding and decoding processes. For more details on the implementation of WOM-codes in flash memories, the reader is referred to [14] and [32].

TABLE II
PRIOR BEST-KNOWN SUM-RATES FOR THE UNRESTRICTED-RATE AND FIXED-RATE WOM-CODE PROBLEMS

Number of Writes	Best Prior (unrestricted)	Upper Bound (unrestricted)	Best Prior (fixed)	Upper Bound (fixed)
2	1.3707 [29]	1.585	1.343 [23]	1.546
3	1.5302 [23]	2	1.4348 [6]	1.9368
4	1.7524 [22]	2.3219	1.6042 [22], [23]	2.2436
5	1.7524 [22]	2.585	1.6279 [30]	2.4965
6	1.7524 [22]	2.8074	1.7143 [30]	2.712
7	1.8232 [22]	3	1.8232 [22]	2.9001
8	1.8824 [30]	3.1699	1.8824 [30]	3.0664
9	1.9535 [30]	3.3219	1.9535 [30]	3.2157
10	2.0144 [30]	3.4594	2.0144 [30]	3.352

The most fundamental problem in the WOM model is to maximize the total amount of information that can be written into n memory cells in t writes, while preserving the constraint that on each write one can only change cells in the zero state to the one state. We say that a binary $[n, t; M_1, M_2, \ldots, M_t]$ WOM-code can write t messages on n binary cells, where during the ith write, $1 \leqslant i \leqslant t$, one of M_i possible messages is written. The rate of the ith 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 *sum-rate* of the WOM-code is the sum of all the individual rates for each write

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

It is proved in [10] and [15] that the capacity region of a binary t-write WOM-code is

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

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 [23]. In this work, they also reported on more WOM-code constructions, including tabular WOM-codes and "linear" WOM-codes. Merkx constructed WOM-codes based on projective geometry [22]. In [6], using binary linear codes, Cohen *et al.* introduced a "coset-coding" technique that is used to construct WOM-codes, and in [13], an improvement to one of the constructions in [6] was given by Godlewski. Recently, position modulation codes have been introduced by Wu and Jiang in order to construct multiple-write WOM-codes [30]. Wu found WOM-codes for two writes in [29] which improved the best rate previously known.

Wolf *et al.* discussed the WOM-codes problem from its information-theoretic point of view [28]. In [9], the WOM model has been generalized for multi-level cells and studied information theory limits and code constructions for constrained sources. Heegard studied the capacity of a WOM and a noisy WOM in [15], and Fu and Han Vinck found the capacity of a nonbinary WOM [10]. Error-correcting WOM-codes were first studied in [34] and [35] and more constructions were recently given in [33]. Jiang discussed in [16] the generalization of error-correcting WOM-codes for the flash/floating codes model [17], [18], [21].

While there are different ways to analyze the efficiency of WOM-codes, we find that the appropriate figure of merit is to analyze the sum-rate under the assumption of a fixed number of writes. In general, the more writes the WOM-code can support, the better the sum-rate it can achieve. The goal is to give upper and lower bounds on the sum-rates of WOM-codes while fixing the number of writes t.

We also distinguish between two families of WOM-codes. If the rates on all writes of a WOM-code are all the same then it is called a *fixed-rate WOM-Code*, and otherwise it is called a *variable-rate WOM-code*. We also address two different problems when analyzing the sum-rate of WOM-codes. In the first one, called the fixed-rate WOM-code problem, the sum-rate is analyzed over all fixed-rate WOM-codes, and in the second problem, called the unrestricted-rate WOM-code problem, the sum-rate is analyzed over all fixed-rate and variable-rate WOM-codes.

Table II summarizes, for the two different problems, the best previously known sum-rates for each number of writes t, where $2 \leqslant t \leqslant 10$. The second column represents the best known sum-rates for the unrestricted-rate WOM-code problem and the third column gives the capacity, which is a tight upper bound on the achievable sum-rate, $\log_2(t+1)$, derived in [10] and [15]. Similarly, the fourth column represents the best known sum-rate for the fixed-rate WOM-code problem and the last column is the upper bound on the sum-rate, which was given in [15]. The citation next to each sum-rate corresponds to the reference in the bibliography where the WOM-code was first presented. Note that a t-write variable-rate WOM-code can also be used as a (degenerate) s-write variable-rate WOM-code, for s larger than t, simply by not writing messages on the last $s-t$ writes. This explains the equality of the entries for $t = 3, 4, 5$ writes in the unrestricted-rate problem.

In this paper, we present WOM-code constructions which reduce the gaps between the upper and lower bounds on sum-rates for the fixed- and unrestricted-rate WOM-code problems. In Section II, we formally define the WOM-codes problem. Section III reviews the previous works on WOM-codes that give the currently known lower and upper bounds on sum-rates for the fixed- and unrestricted-rate WOM-code problems. In

Section IV, we present a two-write WOM-code construction which improves the best known sum-rates for both cases and can achieve each point in the capacity region of two-write WOM-codes. Then, we discuss in this section the connection between the Blackwell channel and two-write WOM-codes and show how to take advantage of two-write WOM-codes in order to construct codes for the Blackwell channel. In Section V, we generalize the two-write WOM-code construction from Section IV for nonbinary cells. Then, it is shown how to use ternary-cell 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 approach that works for an arbitrary number of writes. We introduce another general construction based upon concatenating WOM-codes which provides us with more ways to construct families of WOM-codes. In Section VII, we show a recursive method to construct fixed-rate multiple-write WOM-codes. Finally, we summarize our findings in Section VIII and show that the constructions given in this paper outperform all previously known sum-rates for $2 \leqslant t \leqslant 10$ for both cases of the fixed- and unrestricted-rate WOM-code problems.

II. Preliminaries

In this paper, the memory elements, called ***cells***, have two states: 0 and 1. At the beginning, all the cells are in their 0 state. A cell can change its state from 0 to 1. This operation is irreversible in the sense that a cell cannot change its state from 1 to 0 unless the entire memory is erased. The *memory-state vectors* are all the binary vectors $\boldsymbol{c} = (c_1, c_2, \ldots, c_n)$ of length n, $\{0,1\}^n$. For two memory-state vectors $\boldsymbol{c}, \boldsymbol{c}' \in \{0,1\}^n$, we denote by $\boldsymbol{c} \geqslant \boldsymbol{c}'$, if and only if $c_i \geqslant c_i'$ for all $1 \leqslant i \leqslant n$ and say that $\boldsymbol{c}$ covers $\boldsymbol{c}'$.

Definition: An $[n, t; M_1, \ldots, M_t]$ t-write WOM-code $\mathcal{C}$ is a coding scheme which consists of n 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 *sum-rate* of a t-write WOM-code $\mathcal{C}$ is defined to be
$$\mathcal{R}_{\mathrm{sum}}(\mathcal{C}) = \frac{\sum_{i=1}^t \log_2 M_i}{n}.$$

Remark 1: We assume that the write number on each write is known. This knowledge does not affect the sum-rate. Indeed, assume that there exists an $[n, t; M_1, \ldots, M_t]$ t-write WOM-code $\mathcal{C}$ where the write number is known. Assume also that the sum-rate of $\mathcal{C}$ is $\mathcal{R}_{\mathrm{sum}}(\mathcal{C}) = \frac{\sum_{i=1}^t \log_2 M_i}{n}$. It is possible to change this WOM-code to an $[Nn+t, t; M_1^N, \ldots, M_t^N]$ t-write WOM-code $\mathcal{C}'$ by having N blocks of the t-write WOM-code $\mathcal{C}$ and t more cells indicating the write number. Then, the sum-rate of $\mathcal{C}'$ is
$$\begin{aligned}\mathcal{R}_{\mathrm{sum}}(\mathcal{C}') &= \frac{\sum_{i=1}^t \log_2 M_i^N}{Nn+t} = \frac{N\sum_{i=1}^t \log_2 M_i}{Nn+t} \\ &= \frac{N\sum_{i=1}^t \log_2 M_i}{Nn} \cdot \frac{Nn}{Nn+t} = \frac{\mathcal{R}_{\mathrm{sum}}(\mathcal{C})}{1+\frac{t}{Nn}}.\end{aligned}$$

Therefore, for N large enough, it is possible to achieve the sum-rate of the t-write WOM-code $\mathcal{C}$. For simplicity, it will be assumed in this paper that the write number is known in the encoding process.

III. Previous Work

It is proved in [10] and [15] that the capacity region of a binary t-write WOM-code is
$$\begin{aligned}C_t = \Big\{ &(\mathcal{R}_1, \ldots, \mathcal{R}_t) \mid \mathcal{R}_1 \leqslant h(p_1), \mathcal{R}_2 \leqslant (1-p_1)h(p_2), \\ &\ldots, \mathcal{R}_{t-1} \leqslant \Big(\prod_{i=1}^{t-2}(1-p_i)\Big)h(p_{t-1}), \mathcal{R}_t \leqslant \prod_{i=1}^{t-1}(1-p_i), \\ &\text{where } 0 \leqslant p_1, \ldots, p_{t-1} \leqslant 1/2\Big\}. \end{aligned} \tag{1}$$

It has been shown that all points in the capacity region can be achieved by random coding with either fixed-rate or variable-rate WOM-codes. The sum-rate of the WOM-code is given by
$$\mathcal{R} = \sum_{j=1}^t \mathcal{R}_j = h(p_1) + \sum_{j=2}^{t-1}\Big(\prod_{i=1}^{j-1}(1-p_i)h(p_j)\Big) + \prod_{i=1}^{t-1}(1-p_i).$$

It is proved in [15] that the sum-rate is maximized when
$$p_j = \frac{1}{2+t-j}$$
for $1 \leqslant j \leqslant t-1$, and the maximum sum-rate is $\log_2(t+1)$. For example, for $t=2$, the maximum sum-rate, $\log_2 3$, is achieved for $p_1 = 1/3$. Intuitively, this upper bound is plausible. During the course of the t writes, a particular cell can be programmed at some time $j \in \{1, \ldots, t\}$ or not programmed at all. Thus, there are $t+1$ possible scenarios, so the amount of information that can be stored in each cell is no greater that $\log_2(t+1)$. Of course, the result above indicates that this is a tight upper bound.

The case of fixed-rate WOM-codes was discussed in [15]. In this setting, we consider those points on the boundary of the capacity region C_t satisfying $\mathcal{R}_1 = \cdots = \mathcal{R}_t$. The maximum sum-rate, denoted by $\mathcal{R}_U^F(t)$, is given by the recursion in the following theorem [15].

Theorem 1: The values of $\mathcal{R}_U^F(t)$ for $t \geqslant 1$ satisfy the following recursive formula:
$$\begin{aligned}&\mathcal{R}_U^F(1) = 1 \\ &\mathcal{R}_U^F(t+1) = (t+1)\cdot \mathrm{root}\left\{h\left(\frac{z}{\mathcal{R}_U^F(t)/t}\right) - z\right\}\end{aligned}$$

TABLE III
UPPER BOUNDS ON THE SUM-RATE OF FIXED-RATE WOM-CODES

t	$\mathcal{R}_U^F(t)$	t	$\mathcal{R}_U^F(t)$
1	1	6	2.712
2	1.546	7	2.9001
3	1.9368	8	3.0664
4	2.2436	9	3.2157
5	2.4965	10	3.352

where $\mathrm{root}\{f(z)\}$ is the minimum positive value of z such that $f(z) = 0$.

As mentioned above, random coding achieves capacity and thus the upper bound $\mathcal{R}_U^F(t)$ is tight. Using the recursion in the theorem, the following results are obtained for $\mathcal{R}_U^F(t)$ in Table III.

The upper bounds presented previously on the sum-rates for the fixed- and unrestricted-rate WOM-codes problems have been shown to be achievable in theory. However, finding specific WOM-code constructions that achieve these maximum possible sum-rates remains an open problem. In the rest of this section, we give a brief summary of the highest known sum-rates that were achieved by previously published WOM-code constructions.

Rivest and Shamir, 1982 [23]: Rivest and Shamir constructed the first $[3, 2; 4, 4]$ WOM-code (sum-rate $\mathcal{R}_{\mathrm{sum}} = 1.3333$), that stores two bits twice using only three cells. They constructed other WOM-codes, including a $[7, 2; 26, 26]$ WOM-code ($\mathcal{R}_{\mathrm{sum}} = 1.343$) which has a slightly better sum-rate, a $[7, 3; 8, 8, 8]$ WOM-code ($\mathcal{R}_{\mathrm{sum}} = 1.2857$), and a $[7, 5; 4, 4, 4, 4, 4]$ WOM-code ($\mathcal{R}_{\mathrm{sum}} = 1.4286$). They also described construction methods for various classes of WOM-codes, including tabular WOM-codes and "linear" WOM-codes. In their paper, they also mentioned specific WOM-codes as well as some classes of WOM-codes designed by others, with the following parameters:

1) $[5, 3; 5, 5, 5]$ WOM-code ($\mathcal{R}_{\mathrm{sum}} = 1.3932$), by David Klaner.
2) $[7, 4; 7, 7, 7, 7]$ WOM-code ($\mathcal{R}_{\mathrm{sum}} = 1.6042$), by David Leavitt.
3) $[12, 3; 65, 81, 64]$ WOM-code ($\mathcal{R}_{\mathrm{sum}} = 1.5302$), by James B. Saxe.
4) $[n, n/2 - 1; n/2, n/2 - 1, n/2 - 2, \ldots, 2]$ WOM-code, n even ($\mathcal{R}_{\mathrm{sum}} \approx \frac{\log_2 n}{2}$ for n large enough), by James B. Saxe.

Merkx, 1984 [22]: Merkx constructed WOM-codes based on projective geometry codes. Parameters of some of his WOM-codes are as follows:

1) $[7, 4; 7, 7, 7, 7]$ WOM-code ($\mathcal{R}_{\mathrm{sum}} = 1.6042$).
2) $[31, 10; 31, 31, 31, 31, 31, 31, 31, 31, 31, 31]$ WOM-code ($\mathcal{R}_{\mathrm{sum}} = 1.5981$).
3) $[7, 4; 8, 7, 8, 8]$ WOM-code ($\mathcal{R}_{\mathrm{sum}} = 1.6868$).
4) $[7, 4; 8, 7, 11, 8]$ WOM-code ($\mathcal{R}_{\mathrm{sum}} = 1.7524$).
5) $[8, 4; 8, 14, 11, 8]$ WOM-code ($\mathcal{R}_{\mathrm{sum}} = 1.6583$).
6) $[16, 7; 16, 16, 16, 16, 16, 16, 16]$ WOM-code ($\mathcal{R}_{\mathrm{sum}} = 1.75$).
7) $[15, 7; 15, 15, 15, 15, 15, 15, 15]$ WOM-code ($\mathcal{R}_{\mathrm{sum}} = 1.8232$).

Cohen et al. 1986 [6]: Cohen *et al.* introduced the "coset-coding" technique, which uses binary linear codes in the construction of WOM-codes. This approach yielded WOM-codes with the following parameters:

1) $[23, 3; 2^{11}, 2^{11}, 2^{11}]$ WOM-code ($\mathcal{R}_{\mathrm{sum}} = 1.4348$).
2) $[3, 2; 4, 4]$ WOM-code ($\mathcal{R}_{\mathrm{sum}} = 1.3333$).
3) $[7, 3; 8, 8, 8]$ WOM-code ($\mathcal{R}_{\mathrm{sum}} = 1.2857$).
4) $[2^r - 1, 2^{r-2} + 2; 2^r, 2^r, \ldots, 2^r]$ WOM-code ($\mathcal{R}_{\mathrm{sum}} = \frac{r(2^{r-2}+2)}{2^r-1}$), for $r \geqslant 4$.

Godlewski 1987 [13]: Godlewski improved upon the last result in [6] by constructing WOM-codes with parameters:

1) $[2^r - 1, 2^{r-2} + 2^{r-4} + 2; 2^r, 2^r, \ldots, 2^r]$ WOM-code ($\mathcal{R}_{\mathrm{sum}} = \frac{r(2^{r-2}+2^{r-4}+2)}{2^r-1}$), for $r \geqslant 4$.

Wu and Jiang 2009 [30]: Recently, position modulation codes have been used by Wu and Jiang in order to construct multiple-write WOM-codes. Their construction can produce many WOM-codes, among them WOM-codes with the following parameters:

1) $[172, 5; 2^{56}, 2^{56}, 2^{56}, 2^{56}, 2^{56}]$ WOM-code ($\mathcal{R}_{\mathrm{sum}} = 1.6279$).
2) $[196, 6; 2^{56}, 2^{56}, 2^{56}, 2^{56}, 2^{56}, 2^{56}]$ WOM-code ($\mathcal{R}_{\mathrm{sum}} = 1.7143$).
3) $[216, 7; 2^{56}, 2^{56}, 2^{56}, 2^{56}, 2^{56}, 2^{56}, 2^{56}]$ WOM-code ($\mathcal{R}_{\mathrm{sum}} = 1.8148$).
4) $[238, 8; 2^{56}, 2^{56}, 2^{56}, 2^{56}, 2^{56}, 2^{56}, 2^{56}, 2^{56}]$ WOM-code ($\mathcal{R}_{\mathrm{sum}} = 1.8824$).
5) $[258, 9; 2^{56}, 2^{56}, 2^{56}, 2^{56}, 2^{56}, 2^{56}, 2^{56}, 2^{56}, 2^{56}]$ WOM-code ($\mathcal{R}_{\mathrm{sum}} = 1.9535$).
6) $[278, 10; 2^{56}, 2^{56}, 2^{56}, 2^{56}, 2^{56}, 2^{56}, 2^{56}, 2^{56}, 2^{56}, 2^{56}]$ WOM-code ($\mathcal{R}_{\mathrm{sum}} = 2.0144$).

Wu 2010 [29]: Wu designed two-write WOM-codes that had the highest sum-rates of any such WOM-codes known at the time. His best construction gave a $[10, 2; 176, 76]$ WOM-code ($\mathcal{R}_{\mathrm{sum}} = 1.3707$). He also presented a construction of "ϵ-error" two-write WOM-codes for which the second write is not guaranteed in the worst case, but is allowed with high probability.

The results of the best previously known sum-rates both for the fixed- and unrestricted-rate WOM-code problems as well as upper bounds for each case are summarized in Table II.

IV. TWO-WRITE WOM-CODES

In this section, we present a two-write WOM-codes construction that reduces the gap between the upper and lower bound on the sum-rates for both fixed- and unrestricted-rate WOM-code problems. The construction is inspired by the "coset-coding" scheme which was used in [6] and [13] and recently in [29]. In [6] and [13], multiple-write WOM-codes are constructed where on each write the "coset-coding" scheme is used. In [29], the "coset-coding" is used only on the second write in order to generate an ϵ-error two-write WOM-codes. In ϵ-error two-write WOM-codes the second write is not guaranteed in the worst case but is allowed with high probability. Here, it is shown how to generate from every linear code a two-write WOM-code. As in [29], we use the "coset-coding" scheme only on the second write, and the first write is modified such that the second write is guaranteed in the worst case. We show two specific examples of WOM-codes having better sum-rates than the previously best known ones. We also show that by choosing uniformly at random the parity-check matrix of the linear code, there exist WOM-codes that achieve all points in the capacity region of two-write WOM-codes. Finally, we discuss the connection between the Blackwell channel [1] and two-write WOM-codes. We show how to generate from each two-write WOM-code a code for the Blackwell channel.

A. Two-Write WOM-Codes Construction

Let $\mathcal{C}[n,k]$ be a linear code with parity-check matrix $\mathcal{H}$. For each $\boldsymbol{v} \in \{0,1\}^n$, the matrix $\mathcal{H}_{\boldsymbol{v}}$ is defined as follows. The ith column of $\mathcal{H}_{\boldsymbol{v}}$, $1 \leqslant i \leqslant n$, is the ith column of $\mathcal{H}$ if $\boldsymbol{v}_i = 0$ and otherwise it is the zeros column. The set $V_{\mathcal{C}}$ is defined to be

$$V_{\mathcal{C}} = \{\boldsymbol{v} \in \{0,1\}^n \mid \operatorname{rank}(\mathcal{H}_{\boldsymbol{v}}) = n-k\}. \tag{2}$$

We first note the following claim.

Claim 2: If a vector $\boldsymbol{v}$ belongs to $V_{\mathcal{C}}$, its weight is at most k.

The support of a binary vector $\boldsymbol{v}$, denoted by $\operatorname{supp}(\boldsymbol{v})$, is the set $\{i \mid \boldsymbol{v}_i = 1\}$. The dual of the code $\mathcal{C}$ is denoted by $\mathcal{C}^{\perp}$. The next lemma is a variation of a well known result (see, e.g., [6]).

Lemma 3: Let $\mathcal{C}[n,k]$ be a linear code with parity-check matrix $\mathcal{H}$. For each vector $\boldsymbol{v} \in \{0,1\}^n$, $\operatorname{rank}(\mathcal{H}_{\boldsymbol{v}}) = n-k$ if and only if $\boldsymbol{v}$ does not cover any nonzero codeword in $\mathcal{C}^{\perp}$.

Lemma 3 implies that if two matrices are parity-check matrices of the same linear code $\mathcal{C}$, then their corresponding sets $V_{\mathcal{C}}$ are identical, and so the set $V_{\mathcal{C}}$ is defined to be

$$V_{\mathcal{C}} = \{\boldsymbol{v} \in \{0,1\}^n \mid \boldsymbol{v} \text{ does not cover any nonzero } c \in \mathcal{C}^{\perp}\}.$$

The next theorem presents the two-write WOM-codes.

Theorem 4: Let $\mathcal{C}[n,k]$ be a linear code with parity-check matrix $\mathcal{H}$ and let $V_{\mathcal{C}}$ be the set defined in (2). Then there exists an $[n,2;|V_{\mathcal{C}}|,2^{n-k}]$ two-write WOM-code of sum-rate

$$\frac{\log_2|V_{\mathcal{C}}| + (n-k)}{n}.$$

Proof: We need to show the existence of the encoding and decoding maps on the first and second writes. First, let $\{v_1, v_2, \ldots, v_{|V_{\mathcal{C}}|}\}$ be an ordering of the set $V_{\mathcal{C}}$. The first and the second writes are implemented as follows.

1) On the first write, a symbol over an alphabet of size $|V_{\mathcal{C}}|$ is written. The encoding and decoding maps $\mathcal{E}_1$, $\mathcal{D}_1$ are defined as follows. For each $m \in \{1,\ldots,|V_{\mathcal{C}}|\}$, $\mathcal{E}_1(m) = v_m$ and $\mathcal{D}_1(v_m) = m$.
2) On the second write, a vector $\boldsymbol{s}_2$ of $n-k$ bits 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 $\boldsymbol{v}_2$ is a solution of the equation $\mathcal{H}_{\boldsymbol{v}_1}\cdot\boldsymbol{v}_2 = \boldsymbol{s}_1 + \boldsymbol{s}_2$. For the decoding map $\mathcal{D}_2$, if $\boldsymbol{c}$ is the vector of programmed cells, then the decoded value of the $n-k$ bits is given by

$$\mathcal{D}_2(\boldsymbol{c}) = \mathcal{H}\cdot\boldsymbol{c} = \mathcal{H}\cdot\boldsymbol{v}_1 + \mathcal{H}\cdot\boldsymbol{v}_2 = \boldsymbol{s}_1 + \boldsymbol{s}_1 + \boldsymbol{s}_2 = \boldsymbol{s}_2.$$

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

There is no condition on the code $\mathcal{C}$ and therefore we can use any linear code in this construction, though we seek to find codes that maximize the sum-rate $\frac{\log_2(|V_{\mathcal{C}}|)+n-k}{n}$. Next, we show two examples of two-write WOM-codes that achieve better sum-rates than the previously best known ones.

Example 1: Let us demonstrate how Theorem 4 works for the $[16,5,8]$ first-order Reed–Muller code. Its dual code is the $[16,11,4]$ second-order Reed-Muller, which is the extended Hamming code of length 16. Hence, we are interested in the size of the set

$$V_1 = \left\{\boldsymbol{v} \in \{0,1\}^{16} | \ \boldsymbol{v} \text{ does not cover any } c \in [16,11,4]\right\}.$$

According to Claim 2, the set V_1 does not contain vectors of weight greater than five. This extended Hamming code has 140 codewords of weight four and no codewords of weight five. The set V_1 consists of the following vector sets.

1) All vectors of weight at most three. There are $\sum_{i=0}^{3}\binom{16}{i} = 697$ such vectors.
2) All vectors of weight four that are not codewords. There are $\binom{16}{4} - 140 = 1680$ such vectors.
3) All vectors of weight five that do no cover any codeword of weight four. There are $\binom{16}{5} - 12\cdot 140 = 2688$ such vectors. Since the minimum distance of the code is four, a vector of weight five can cover at most one codeword of weight four.

Therefore, we get $|V_1| = 697 + 1680 + 2688 = 5065$ and the sum-rate is

$$(\log_2(5065) + 11)/16 = 1.4566.$$

It is possible to modify this WOM-code such that on the first write only 11 bits are written. Thus, we achieve a two-write fixed-rate WOM-code and its sum-rate is $22/16 = 1.375$, which is the best known fixed-rate WOM-code.

Example 2: In this example, we will use the $[23,11,8]$ Golay code. Its dual code is the $[23,12,7]$ Golay code so we are interested in the size of the set

$$V_2 = \left\{\boldsymbol{v} \in \{0,1\}^{23} \mid \boldsymbol{v} \text{ does not cover any } c \in [23,12,7]\right\}.$$

According to Claim 2, there are no vectors of weight greater than 11 in the set V_2. The $[23,12,7]$ Golay code has $A_7 = 253$ codewords of weight seven, $A_8 = 506$ codewords of weight eight, and $A_{11} = 1288$ codewords of weight 11. The set V_2 consists of the following vector sets.

1) All vectors of weight at most 6. This number of vectors is $\sum_{i=0}^{6}\binom{23}{i} = 145499$.
2) All vectors of weight between 7 and 10 besides those that cover a codeword of weight 7 or 8. Since the minimum distance of the code is 7 every vector can cover at most one codeword. Hence, this number of vectors is

$$\sum_{i=7}^{10}\binom{23}{i} - A_7\cdot\sum_{i=7}^{10}\binom{16}{i-7} - A_8\cdot\sum_{i=8}^{10}\binom{15}{i-8} = 2459160$$

3) All vectors of weight 11 that are not codewords and do not cover any codeword of weight either 7 or 8. This number was shown in [7] to be 695520.

Therefore, for the $[23, 11, 8]$ Golay code, we get

$$|V_2| = 145499 + 2459160 + 695520 = 3300179$$

and thus the sum-rate is

$$(\log_2(3300179) + 12)/23 = 1.4632.$$

B. Random Coding

The scheme we described in the previous section can work for any linear code $\mathcal{C}$. Given a linear code $\mathcal{C}[n, k]$ with parity-check matrix $\mathcal{H}_{\mathcal{C}}$, we denote $\mathcal{R}_1(\mathcal{C}) = \frac{\log_2 |V_{\mathcal{C}}|}{n}$, $\mathcal{R}_2(\mathcal{C}) = \frac{n-k}{n}$ so the sum-rate of the generated WOM-codes is

$$\mathcal{R}_1(\mathcal{C}) + \mathcal{R}_2(\mathcal{C}) = \frac{\log_2 |V_{\mathcal{C}}| + n - k}{n}.$$

Our goal in this section is to show that it is possible to achieve all points in the capacity region C_2 defined in (1), by choosing uniformly at random the parity-check matrix of the linear code $\mathcal{C}$. We prove that in the following theorem.

Theorem 5: 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 $p \in [0, 0.5]$ be such that $\mathcal{R}_1 \leqslant h(p)$ and $\mathcal{R}_2 \leqslant 1 - p$. Let $k = \lceil np \rceil$ for n large enough and let us choose uniformly at random an $(n-k) \times n$ matrix $\mathcal{H}$. The matrix $\mathcal{H}$ will be the parity-check matrix of the linear code $\mathcal{C}$ that will be used to construct the two-write WOM-code. For each vector $\boldsymbol{v} \in \{0,1\}^n$, let us define the indicator random variable $X_{\boldsymbol{v}}(\mathcal{H})$ on the space of all matrices as follows:

$$X_{\boldsymbol{v}}(\mathcal{H}) = \begin{cases} 1, & \text{if } \boldsymbol{v} \in V_{\mathcal{C}} \\ 0, & \text{otherwise} \end{cases}$$

where $V_{\mathcal{C}}$ is the set defined in (2). Note that choosing the matrix $\mathcal{H}$ uniformly at random induces a measure on the set $V_{\mathcal{C}}$ and thus a probability distribution on the random variable $X_{\boldsymbol{v}}(\mathcal{H})$. Then, the number of vectors in $V_{\mathcal{C}}$ is $X(\mathcal{H}) = \sum_{\boldsymbol{v} \in \{0,1\}^n} X_{\boldsymbol{v}}(H)$, and

$$E[X(\mathcal{H})] = \sum_{\boldsymbol{v} \in \{0,1\}^n} E[X_{\boldsymbol{v}}(\mathcal{H})] = \sum_{\boldsymbol{v} \in \{0,1\}^n} \Pr\{X_{\boldsymbol{v}}(\mathcal{H}) = 1\}. \tag{3}$$

We claim that $\Pr\{X_{\boldsymbol{v}}(\mathcal{H}) = 1\}$ depends on $\boldsymbol{v}$ only through its weight, $wt(\boldsymbol{v})$. In this case, (3) simplifies to

$$\begin{aligned} E[X(\mathcal{H})] &= \sum_{i=0}^{n} \binom{n}{i} \Pr\{X_{\boldsymbol{v}:wt(\boldsymbol{v})=i}(\mathcal{H}) = 1\} \\ &= \sum_{i=0}^{k} \binom{n}{i} \Pr\{X_{\boldsymbol{v}:wt(\boldsymbol{v})=i}(\mathcal{H}) = 1\} \end{aligned}$$

because if $wt(\boldsymbol{v}) \geqslant k+1$ then $X_{\boldsymbol{v}}(\mathcal{H}) = 0$ for all $\mathcal{H}$(Claim 2).

Now, let us determine the value of $\Pr\{X_{\boldsymbol{v}}(\mathcal{H}) = 1\}$ for a vector $\boldsymbol{v}$ of weight $0 \leqslant i \leqslant k$. Note that $\boldsymbol{v} \in V_{\mathcal{C}}$ if and only if the sub-matrix of size $(n-k) \times (n - wt(\boldsymbol{v}))$ induced by the zero entries of the vector $\boldsymbol{v}$ is full rank. It is well known, e.g., [4], that if we choose an $m \times n$ matrix, where $m \leqslant n$, uniformly at random then the probability that it is full rank is $\prod_{j=n-m+1}^{n} (1 - 2^{-j})$. Therefore, if we choose an $(n-k) \times (n-i)$ matrix uniformly at random then the probability that it is full rank is $\prod_{j=k-i+1}^{n-i} (1 - 2^{-j})$. Note that

$$\begin{aligned} \prod_{j=k-i+1}^{n-i} \left(1 - 2^{-j}\right) &> \prod_{j=1}^{\infty} \left(1 - 2^{-j}\right) \\ &> \left(1 - \frac{1}{2}\right) \left(1 - \sum_{j=2}^{\infty} 2^{-j}\right) = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4} \end{aligned}$$

and, hence, $\Pr\{X_{\boldsymbol{v}}(\mathcal{H}) = 1\} = \prod_{j=k-i+1}^{n-i} (1 - 2^{-j}) > 1/4$. According to [25, lemma 4.8]

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

and, therefore, we get

$$E[X(\mathcal{H})] = \sum_{i=0}^{k} \binom{n}{i} \prod_{j=k-i+1}^{n-i} (1 - 2^{-j}) > 2^{nh\left(\frac{k}{n}\right) - 2 - \log_2(n+1)}.$$

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

$$\begin{aligned} \mathcal{R}_1(\mathcal{C}) &\geqslant h\left(\frac{k}{n}\right) - \frac{2 + \log_2(n+1)}{n} \\ &\geqslant h(p) - \frac{2 + \log_2(n+1)}{n} \geqslant \mathcal{R}_1 - \epsilon \\ \mathcal{R}_2(\mathcal{C}) &= \frac{n-k}{n} \geqslant (1-p) - \frac{1}{n} \geqslant \mathcal{R}_2 - \epsilon \end{aligned}$$

for n large enough. ■

Random coding was proved to be capacity-achieving by constructing a partition code [10], [15]. However, the above random coding scheme has more structure that enables to look for WOM-codes with a relatively small block length. We ran a computer search to look for such WOM-codes. The parity-check matrix of the linear code $\mathcal{C}$ was chosen uniformly at random and then the size of the set $V_{\mathcal{C}}$ was computed. The results are shown in Fig. 1. Note that if $(\mathcal{R}_1, \mathcal{R}_2)$ and $(\mathcal{R}_3, \mathcal{R}_4)$ are two achievable rate points then for each $t \in \mathbb{Q}$ the point $(t\mathcal{R}_1 + (1-t)\mathcal{R}_2, t\mathcal{R}_3 + (1-t)\mathcal{R}_4)$ is an achievable rate point, too. This can simply be done by block sharing of a large number of blocks. Therefore, the achievable region is convex.

We ran a computer search to find more two-write WOM-codes with high sum-rates. For fixed-rate WOM-codes, the best construction achieved by a computer search has sum-rate $\frac{48}{33} \approx 1.4546$ and for variable-rate WOM-codes the best computer search construction achieved sum-rate 1.4928. The number of cells in these two constructions is 33.

Remark 2: The encoding and decoding maps of the second write are implemented by the parity-check matrix of the linear code $\mathcal{C}$ as described in the proof of Theorem 4. A naive scheme to implement the encoding and decoding maps of the first write is simply by a lookup table of the set $V_{\mathcal{C}}$. However, this can be done more efficiently using algorithms to encode and decode constant weight binary codes. There are several works which

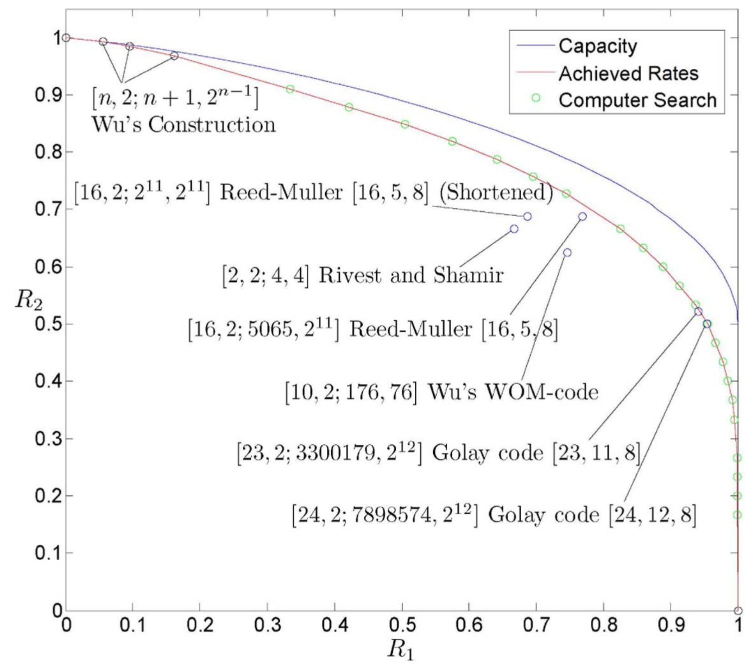

Fig. 1. Capacity region and achieved rates of two-write WOM-codes.

efficiently encode and decode all binary vectors of length n and weight k; see for example [2], [8], [20], [26], [27]. These works can be easily extended to construct efficient encoder and decoder maps to the set of all binary vectors of length n and weight at most k, denoted by

$$B(n,k) = \{\boldsymbol{v} \in \{0,1\}^n \mid \mathrm{supp}(\boldsymbol{v}) \leqslant k\}.$$

The set $V_{\mathcal{C}}$ is a subset of the set $B(n,k)$. Therefore, we can use these algorithms while constructing a smaller table, only for the vectors in the set $B(n,k) \setminus V_{\mathcal{C}}$ as follows. Assume that $f : \{1, \ldots, |B(n,k)|\} \to B(n,k)$ is a one-to-one and onto map such that the calculation of f and its inverse f^{-1} is practically feasible. List all the vectors in $B(n,k) \setminus V_{\mathcal{C}}$ according to a linear ranking of their corresponding values of $f^{-1}(\boldsymbol{v})$. Then, a mapping $g : \{1, \ldots, |V_{\mathcal{C}}|\} \to V_{\mathcal{C}}$ is constructed such that for all $x \in \{1, \ldots, |V_{\mathcal{C}}|\}$, $g(x) = f(x + a(x))$, where $a(x)$ is the number of vectors in $B(n,k) \setminus V_{\mathcal{C}}$ of value less than x. The time complexity to calculate $a(x)$ is $O(\log_2(|B(n,k) \setminus V_{\mathcal{C}}|))$ since this list is sorted. Similarly, for all $\boldsymbol{v} \in V_{\mathcal{C}}$, $g^{-1}(\boldsymbol{v}) = f^{-1}(\boldsymbol{v}) - a(f^{-1}(\boldsymbol{v}))$.

In many cases, the size of the set $B(k,n) \setminus V_{\mathcal{C}}$ will be significantly smaller than the size of $V_{\mathcal{C}}$. For example, for the Golay code $[23, 11, 8]$, the size of $V_{\mathcal{C}}$ is 3300179, while the size of $B(23, 11) \setminus V_{\mathcal{C}}$ is

$$\sum_{i=0}^{11} \binom{n}{i} - 3300179 = 894125.$$

Similarly, for the Reed–Muller code $[16, 5, 8]$, the size of the set $V_{\mathcal{C}}$ is 5065 while the size of the set $B(16, 5) \setminus V_{\mathcal{C}}$ is 1820.

C. Application to the Blackwell Channel

The Blackwell channel, introduced first by Blackwell [1], is one example of a deterministic broadcast channel. The channel

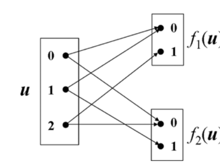

Fig. 2. Blackwell Channel.

is composed of one transmitter and two receivers. The input to the transmitter is ternary and the channel output to each receiver is a binary symbol. Let $\boldsymbol{u}$ be the ternary input vector to the transmitter of length n. For $1 \leqslant i \leqslant n$, $f(\boldsymbol{u}_i) = (f(\boldsymbol{u}_i)_1, f(\boldsymbol{u}_i)_2)$ is a binary vector of length two defined as follows (see Fig. 2):

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

The binary vectors $f_1(\boldsymbol{u})$, $f_2(\boldsymbol{u})$ are defined to be

$$f_1(\boldsymbol{u}) = (f(\boldsymbol{u}_1)_1, f(\boldsymbol{u}_2)_1, \ldots, f(\boldsymbol{u}_n)_1)$$
$$f_2(\boldsymbol{u}) = (f(\boldsymbol{u}_1)_2, f(\boldsymbol{u}_2)_2, \ldots, f(\boldsymbol{u}_n)_2)$$

and are the output vectors to the two receivers.

The capacity region of the Blackwell channel was found by Gel'fand [12] and consists of five subregions, given by their boundaries:

1) $\{(R_1, R_2) \mid 0 \leqslant R_1 \leqslant 1/2, R_2 = 1\}$.
2) $\{(R_1, R_2) \mid R_1 = 1 - p, R_2 = h(p), 1/3 \leqslant p \leqslant 1/2\}$.
3) $\{(R_1, R_2) \mid R_1 + R_2 = \log_2 3, \frac{2}{3} \leqslant R_1 \leqslant \log_2 3 - \frac{2}{3}\}$.
4) $\{(R_1, R_2) \mid R_1 = h(p), R_2 = 1 - p, 1/3 \leqslant p \leqslant 1/2\}$.
5) $\{(R_1, R_2) \mid R_1 = 1, 0 \leqslant R_2 \leqslant 1/2\}$.

The connection between the Blackwell channel and two-write WOM-codes was suggested by Roth [24]. The next theorem shows that from every two-write WOM-code of rate (R_1, R_2) it is possible to construct codes for the Blackwell channel of rates (R_1, R_2) and (R_2, R_1).

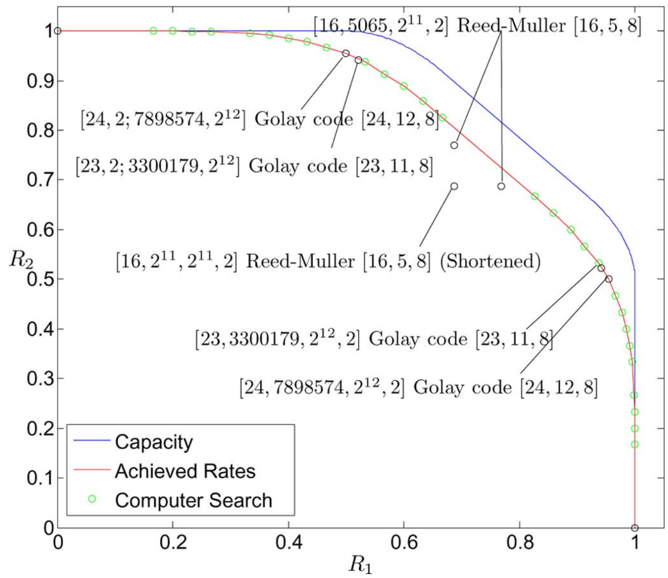

Fig. 3. Capacity region and the achieved rates for the Blackwell channel.

Theorem 6: If (R_1, R_2) is an achievable rate of a two-write WOM-code, then (R_1, R_2) and (R_2, R_1) are achievable rates for the Blackwell channel.

Proof: Assume that there exists a $[n, 2; 2^{nR_1}, 2^{nR_2}]$ two-write WOM-code and let $\mathcal{E}_1$, $\mathcal{E}_2$ and $\mathcal{D}_1$, $\mathcal{D}_2$ be its encoding and decoding maps. We claim that there exists a coding scheme for the Blackwell channel of rate (R_1, R_2). Let $(m_1, m_2) \in \{1, \ldots, 2^{nR_1}\} \times \{1, \ldots, 2^{nR_2}\}$ be two messages and let $\boldsymbol{v}_1 = \mathcal{E}_1(m_1)$ and $\boldsymbol{v}_2 = \mathcal{E}_2(m_2, \boldsymbol{v}_1)$. Let $\boldsymbol{u}$ be a ternary vector of length n defined as follows. For $1 \leqslant i \leqslant n$, $\boldsymbol{u}_i = f^{-1}(\boldsymbol{v}_{1,i}, \overline{\boldsymbol{v}_{2,i}})$. The vector $\boldsymbol{u}$ is well-defined since for all $1 \leqslant i \leqslant n$, $(\boldsymbol{v}_{1,i}, \boldsymbol{v}_{2,i}) \neq (1, 0)$ and hence $(\boldsymbol{v}_{1,i}, \overline{\boldsymbol{v}_{2,i}}) \neq (1, 1)$. The vector $\boldsymbol{u}$ is the input to the transmitter. Then, the vector $f_1(\boldsymbol{u})$ is transmitted to the first receiver and the vector $f_2(\boldsymbol{u})$ to the second receiver. Note that $f_1(\boldsymbol{u}) = \boldsymbol{v}_1$ and $f_2(\boldsymbol{u}) = \overline{\boldsymbol{v}_2}$. Therefore, the first receiver decodes its message according to $\mathcal{D}_1(f_1(\boldsymbol{u})) = \mathcal{D}_1(\boldsymbol{v}_1) = m_1$ and the second receiver decodes its message according to $\mathcal{D}_2(\overline{f_2(\boldsymbol{u})}) = \mathcal{D}_2(\boldsymbol{v}_2) = m_2$.

Similarly, it is possible to achieve the rate (R_2, R_1). Now we let $\boldsymbol{v}_2 = \mathcal{E}_2(m_2)$ and $\boldsymbol{v}_1 = \mathcal{E}_1(m_1, \boldsymbol{v}_2)$. The vector $\boldsymbol{u}$ is defined as $\boldsymbol{u}_i = f^{-1}(\overline{\boldsymbol{v}_{1,i}}, \boldsymbol{v}_{2,i})$ for $1 \leqslant i \leqslant n$. The decoded message by the first receiver is $\mathcal{D}_1(\overline{f_1(\boldsymbol{u})})$ and $\mathcal{D}_2(f_2(\boldsymbol{u}))$ is the decoded message by the second receiver. ■

Remark 3: It is possible to define the Blackwell channel differently such that the forbidden pair of bits is not $(1, 1)$ but another combination. Then, the construction of the codes can be adjusted accordingly.

Now, we can use the two-write WOM-codes in order to define codes for the Blackwell channel. By using time sharing, we see that the achievable region is convex. Fig. 3 shows the corresponding capacity region and achieved rates for the Blackwell channel.

V. Multiple-Write WOM-Codes

In this section, we present WOM-code constructions which reduce the gaps between the upper and lower bounds on the sum-rates of WOM-codes for $3 \leqslant t \leqslant 10$. First, we generalize the two-write WOM-code construction from Section IV for nonbinary cells. Then, we show how to use these nonbinary two-write WOM-codes in order to construct binary multiple-write WOM-codes. We start with a specific construction for three-write WOM-codes and the extension for four-write WOM-codes as well as arbitrary number of writes will appear in Appendix A.

A. Nonbinary Two-Write 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 nonzero values in $\boldsymbol{v}$. Then, we define

$$V_{\mathcal{C}}^{(q)} = \{\boldsymbol{v} \in (\mathrm{GF}(q))^n \mid \mathrm{rank}(\mathcal{H}(\boldsymbol{v})) = n - k\}. \quad (4)$$

Next, we construct a nonbinary two-write WOM-code $[n, 2; |V_{\mathcal{C}}^{(q)}|, q^{n-k}]$ in a similar manner to the construction in Section IV. Since the proof of the next theorem is very similar to the proof of Theorem 4, we omit it. A complete proof can be found in [19].

Theorem 7: 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 (4). Then, there exists a q-ary $[n, 2; |V_{\mathcal{C}}^{(q)}|, q^{n-k}]$ two-write WOM-code of sum-rate

$$\frac{\log_2 |V_{\mathcal{C}}^{(q)}| + (n-k)\log_2 q}{n}.$$

As was shown in the binary case, 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 sum-rate of the generated WOM-code is $\mathcal{R}_1(\mathcal{C}) + \mathcal{R}_2(\mathcal{C})$. The set of achievable rates by this construction is

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

The proof is also very similar to Theorem 5 in Section IV for the binary case, and thus, we omit it as the complete proof appears in [19].

Theorem 8: For any $(\mathcal{R}_1, \mathcal{R}_2) \in C_2^{(q)}$ and $\epsilon > 0$, there exists a linear code $\mathcal{C}$ satisfying $\mathcal{R}_1(\mathcal{C}) \geqslant \mathcal{R}_1 - \epsilon$, $\mathcal{R}_2(\mathcal{C}) \geqslant \mathcal{R}_2 - \epsilon$.

The next corollary provides the best achievable sum-rate of the construction.

Corollary 9: For any q-ary WOM-code generated using our construction, the highest achievable sum-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 achievable sum-rate is $\log_2(2q-1)$. Therefore, there exists a WOM-code produced by our construction with sum-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 [10] and the maximum achievable sum-rate was proved to be $\log_2(2q-1)$. Therefore, the construction cannot produce a WOM-code with a sum-rate that exceeds $\log_2(2q-1)$. ■

Remark 4: This construction does not achieve high sum-rates for nonbinary two-write WOM-codes in general. While the best achievable sum-rate of the construction is $\log_2(2q-1)$, the upper bound on the sum-rate is $\log_2\binom{q+1}{2}$; see [10]. The decrease in the sum-rate in this 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 nonbinary two-write WOM-codes with better sum-rates. However, the goal in this paper is not to find efficient nonbinary WOM-codes. Rather, as shown later, the nonbinary codes that we have constructed can be used in the design of binary multiple-write WOM-codes.

For the construction of binary multiple-write in Section V-B, we use WOM-codes over GF(3). We ran a computer search to find such a ternary two-write WOM-code of sum-rate 2.2205, and we will use this WOM-code in order to construct specific multiple-write WOM-codes.

B. Three-Write WOM-Codes

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

Theorem 10: Let $\mathcal{C}_3$ be an $[n, 2; 2^{n\mathcal{R}_1}, 2^{n\mathcal{R}_2}]$ two-write WOM-code over GF(3) constructed as in Section V-A. Then, there exists a $[2n, 3; 2^{n\mathcal{R}_1}, 2^{n\mathcal{R}_2}, 2^n]$ three-write WOM-code of sum-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 WOM-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 $\phi : \mathrm{GF}(3) \mapsto (\mathrm{GF}(2), \mathrm{GF}(2))$ defined as follows:

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

The map ϕ extends naturally to ternary vectors $\boldsymbol{v} = (v_1, \ldots, v_n) \in \mathrm{GF}(3)^n$ 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 for a 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 $\mathcal{E}_{3,2}$ 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 WOM-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 11: The best achievable sum-rate of a three-write WOM-code 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 sum-rate is $\mathcal{R} = (\mathcal{R}_1 +$

$\mathcal{R}_2+1)/2$. This sum-rate is maximized when $\mathcal{R}_1+\mathcal{R}_2$ is maximized. But $\mathcal{R}_1+\mathcal{R}_2$ is the sum-rate of the two-write WOM-code over $\mathrm{GF}(3)$, which was proven in Corollary 9 to be maximized at $\log_2 5$. Then the maximum achievable sum-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 $\mathrm{GF}(3)$ presented in the previous section, we can construct a three-write WOM-code of sum-rate $(2.2205+1)/2 = 1.6102$.

The extension of the last construction for four and multiple writes is similar and appears in Appendix A.

VI. Concatenated WOM-Codes

The construction presented in the previous section and Appendix A 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 and Appendix A is a special case of this general scheme.

Theorem 12: Let $\mathcal{C}^*$ be an $[m, t/2; q_1, \ldots, q_{t/2}]$ 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 $[n, 2; 2^{n\mathcal{R}_{i,1}}, 2^{n\mathcal{R}_{i,2}}]$ two-write WOM-code over $\mathrm{GF}(q_i)$, as constructed in Section V-A. 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 sum-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 ith 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 and 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

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

■

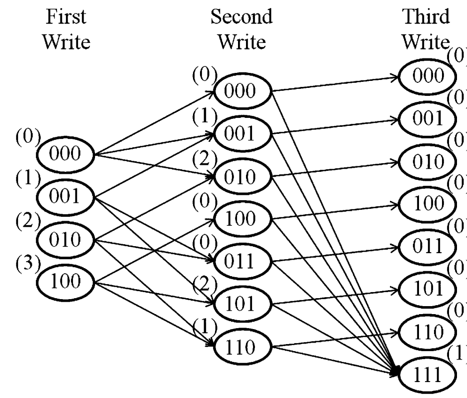

Fig. 4. $[3, 3; 4, 3, 2]$ three-write WOM-code.

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

Example 3: We choose a $[3, 3; 4, 3, 2]$ three-write WOM-code as the code $\mathcal{C}^*$. This code is depicted in Fig. 4 by a state diagram describing 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 sum-rate 2.6862. For the code $\mathcal{C}_2$ over $\mathrm{GF}(3)$, we use the code with sum-rate 2.22 which we found in Section V-A, and we use the binary two-write WOM-code of sum-rate 1.4928 for the code $\mathcal{C}_3$. Then, the sum-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 sum-rate is

$$\frac{2.6862+2.2205+1}{3} = 1.9689.$$

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

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

for $t \geqslant 5$ and $\mathcal{R}_{t-4}$ is the maximum achievable sum-rate for a $(t-4)$-write WOM-code. Similarly, the constructed WOM-codes which we obtain using the WOM-codes which we found before have sum-rates

$$\mathcal{R}'_t = \frac{2.6862+2.2205+\mathcal{R}'_{t-4}}{3}$$

TABLE IV
SUM-RATES OF CONCATENATED WOM-CODES

Number of Writes	Achieved New Sum-rate	Maximum New Sum-rate
5	1.9689	$\frac{\log_2 7+\log_2 5+1}{3} = 2.0431$
6	2.1331	$\frac{\log_2 7+\log_2 5+\log_2 3}{3} = 2.2381$
7	2.1723	$\frac{\log_2 7+\log_2 5+(\log_2 5+1)/2}{3} = 2.2634$
8	2.2544	$\frac{\log_2 7+\log_2 5+(\log_2 5+\log_2 3)/2}{3} = 2.3609$
9	2.2918	$\frac{\log_2 7+\log_2 5+(\log_2 7+\log_2 5+1)/3}{3} = 2.3908$
10	2.3466	$\frac{\log_2 7+\log_2 5+(\log_2 7+\log_2 5+\log_2 3)/3}{3} = 2.4588$

for $t \geqslant 5$, where $\mathcal{R}'_{t-4}$ is the best sum-rate of a constructed $(t-4)$-write WOM-code. Table IV summarizes these sum-rates.

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

The general scheme described in Theorem 12 provides many more families of WOM-codes. However, in order to construct WOM-codes with high sum-rates, the WOM-code $\mathcal{C}^*$ has to be chosen very carefully. In particular, it is important to choose such a WOM-code with as few cells as possible, since the sum of all sum-rates of the nonbinary two-write WOM-codes is averaged over the number of cells of the WOM-code $\mathcal{C}^*$. As the number of short WOM-codes is small, there are only a small number of possibilities to check. However, our search for better WOM-codes with between six and ten writes using WOM-codes with few cells did not lead to any better results.

VII. FIXED-RATE WOM-CODES

The WOM-code constructions for more than two writes improved the achieved sum-rates only in the case of unrestricted-rate WOM-code problem. In this section, we present a method to construct fixed-rate WOM-codes. The method is recursive and is based on the previously constructed WOM-codes.

Theorem 13: Let $\mathcal{C}$ be an $[n,t;2^{n\mathcal{R}_1},2^{n\mathcal{R}_2},\ldots,2^{n\mathcal{R}_t}]$ t-write WOM-code. Assume that for $1 \leqslant i \leqslant t-1$, there exists a fixed-rate WOM-code of sum-rate R_i. Let $\mathcal{R}'_1,\ldots,\mathcal{R}'_t$ be a permutation of $\mathcal{R}_1,\ldots,\mathcal{R}_t$ such that $\mathcal{R}'_1 \geqslant \cdots \geqslant \mathcal{R}'_t$. Then, there exists a fixed-rate t-write WOM-code of sum-rate

$$\frac{t\cdot\mathcal{R}'_1}{1+\sum_{i=1}^{t-1}\frac{i(\mathcal{R}'_{t-i}-\mathcal{R}'_{t-i+1})}{R_i}}.$$

Proof: For simplicity, let us assume that $\mathcal{R}_1 \geqslant \cdots \geqslant \mathcal{R}_t$ as it will be clear from the proof how to generalize to the arbitrary case. First, we add $(\mathcal{R}_{t-1}-\mathcal{R}_t)n$ more cells in order to write $(\mathcal{R}_{t-1}-\mathcal{R}_t)n$ bits on the last write. This guarantees that the rates on the last two writes are the same. Then, we add $2(\mathcal{R}_{t-2}-\mathcal{R}_{t-1})n/R_2$ more cells in order to write $(\mathcal{R}_{t-2}-\mathcal{R}_{t-1})n$ more bits on each of the last two writes. This part of the last two writes is invoked using the fixed-rate two-write WOM-code of sum-rate R_2, and therefore, the additional number of cells is $2(\mathcal{R}_{t-2}-\mathcal{R}_{t-1})n/R_2$. This addition of cells guarantees that the rates on the last three writes are all the same. In general, for $1 \leqslant i \leqslant t-1$ we add $i(\mathcal{R}_{t-i}-\mathcal{R}_{t-i+1})n/R_i$ more cells such that $(\mathcal{R}_{t-i}-\mathcal{R}_{t-i+1})n$ more bits are written on each of the last i writes and therefore the rates on the last $i+1$ writes are all the same. These bits are written using the fixed-rate i-write WOM-code which is assumed to exist.

With the addition of these cells, the number of bits written on the ith write for $1 \leqslant i \leqslant t$ is

$$\mathcal{R}_i n + \sum_{j=1}^{i-1}(\mathcal{R}_j-\mathcal{R}_{j+1})n = \mathcal{R}_1 n.$$

Thus, the rates on all writes are the same and the generated WOM-code is fixed-rate.

The total number of bits we add is

$$\sum_{i=1}^{t-1}\frac{i(\mathcal{R}_{t-i}-\mathcal{R}_{t-i+1})n}{R_i}$$

and thus the sum-rate is

$$\frac{t\cdot\mathcal{R}_1 n}{n+\sum_{i=1}^{t-1}\frac{i(\mathcal{R}_{t-i}-\mathcal{R}_{t-i+1})n}{R_i}} = \frac{t\cdot\mathcal{R}_1}{1+\sum_{i=1}^{t-1}\frac{i(\mathcal{R}_{t-i}-\mathcal{R}_{t-i+1})}{R_i}}.$$

■

Let us demonstrate how to apply the last theorem. We start with the three-write WOM-code we constructed in Section V-B. Its rates on the first, second, and third writes are 0.6291, 0.4811, and 0.5, respectively. We add $0.0189n$ more cells in order to guarantee that the rates on the last two writes are the same. Then we use the fixed-rate two-write WOM-code constructed in Section IV-A of sum-rate 1.4546. Hence, we add

$$\frac{2\cdot(0.6291-0.5)n}{1.4546} = 0.1775n$$

more cells, yielding a fixed-rate three-write WOM-code of sum-rate

$$\frac{3\cdot 0.6291}{1.1964} = 1.5775.$$

If we used the best fixed-rate two-write WOM-code of sum-rate 1.546 and the best three-write WOM-code of sum-rate 1.66, then we get a fixed-rate three-write WOM-code of sum-rate 1.6263.

Note that we could use a two-write WOM-code such that $0.0189n$ bits are written on its first write and $0.1291n$ bits are written on its second write. This will indeed add another small improvement to the sum-rate; however, this scheme is not easy to generalize. The goal here is to give a general scheme. We are aware that for each individual case it is possible to use other

TABLE V
SUM-RATES OF FIXED-RATE WOM-CODES

Number of Writes	Achieved New Sum-rate	Maximum New Sum-rate
3	1.5775	1.6263
4	1.7298	1.8249
5	1.8794	1.9302
6	1.9742	2.0570
7	1.991	2.0692
8	2.0375	2.1190
9	2.0951	2.1702
10	2.1327	2.2189

WOM-codes that will provide a WOM-code of the desired sum-rate with slightly fewer cells.

Now we move to the four-write WOM-code from Section A-A. Its component rates are 0.6291, 0.4811, 0.413, and 1/3. Three more groups of cells are added as follows:

1) $(0.413 - 1/3)n = 0.0797n$ more cells, so that the last two write have the same rate.
2) $2 \cdot (0.4811 - 0.413)n/1.4546 = 0.0936n$ more cells, so that the last three writes have the same rate.
3) $3 \cdot (0.6291 - 0.4811)n/1.5731 = 0.2822n$ more cells, so that the last four writes have the same rate.

Then, a fixed-rate four-write WOM-code is achieved with sum-rate

$$\frac{4 \cdot 0.6291}{1 + 0.0797 + 0.0936 + 0.2822} = 1.7298.$$

If we used the best fixed-rate two- and three-write WOM-codes and the best variable-rate four-write WOM-code, then we obtain a fixed-rate four-write WOM-code of sum-rate 1.8249. Fixed-rate t-write WOM-code for $t > 4$ can be similarly obtained. The results are summarized for the sum-rates that were actually found and the best ones we could find in this method in Table V.

VIII. SUMMARY AND COMPARISON

In this paper, we have presented several constructions for multiple-write WOM-codes. First, we showed a method to construct two-write WOM-codes. Using this method we found two-write WOM-codes with better sum-rates than the previously known codes. Then, we proved that it is possible to achieve each point in the capacity region of two-write WOM-codes using this scheme. Furthermore, we showed that each two-write WOM-code generates a code for the Blackwell channel.

We then presented another method for constructing binary multiple-write WOM-codes. The method made use of two-write WOM-codes over $\mathrm{GF}(q)$, for which we generalized the binary construction. While the nonbinary WOM-codes we constructed do not achieve high sum-rate, they allowed us to construct binary t-write WOM-codes for $t \geqslant 3$. We showed how to construct WOM-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 sum-rates for $5 \leqslant t \leqslant 10$ for the unrestricted-rate WOM-code problem. Lastly, we showed two methods to construct fixed-rate multiple-write WOM-codes.

TABLE VI
COMPARISON FOR THE UNRESTRICTED-RATE WOM-CODE PROBLEM

Number of Writes	Best Prior	Achieved New Sum-rate	Maximum New Sum-rate	Upper Bound
2	1.3707	**1.4928**	1.585	1.585
3	1.5302	**1.6102**	1.661	2
4	1.7524	**1.8566**	1.9534	2.3219
5	1.7524	**1.9689**	2.0431	2.585
6	1.7524	**2.1331**	2.2381	2.8074
7	1.8232	**2.1723**	2.2634	3
8	1.8824	**2.2544**	2.3609	3.1699
9	1.9535	**2.2918**	2.3908	3.3219
10	2.0144	**2.3466**	2.4588	3.4594

TABLE VII
COMPARISON FOR THE FIXED-RATE WOM-CODE PROBLEM

Number of Writes	Best Prior	Achieved New Sum-rate	Maximum New Sum-rate	Upper Bound
2	1.343	**1.4546**	1.546	1.546
3	1.4348	**1.5775**	1.6263	1.9366
4	1.6042	**1.7298**	1.8249	2.2436
5	1.6279	**1.8794**	1.9302	2.4965
6	1.7143	**1.9742**	2.0570	2.7120
7	1.8232	**1.991**	2.0692	2.9001
8	1.8824	**2.0375**	2.1190	3.0664
9	1.9535	**2.0951**	2.1702	3.2157
10	2.0144	**2.1327**	2.2189	3.3520

Table VI and VII show a comparison for $2 \leqslant t \leqslant 10$ between the sum-rates of the WOM-codes presented in this paper and the best previously known sum-rates for both the fixed- and unrestricted-rate WOM-code problems. The column labeled "Best Prior" is the highest sum-rate achieved by a previously reported t-write WOM-code. The column "Achieved New Sum-rate" gives the sum-rates that we actually obtained through application of the new techniques. The column "Maximum New Sum-rate" lists the maximum possible sum-rates that can be obtained using our approach. Finally, the column "Upper Bound" gives the maximum possible sum-rates for t-write WOM-codes.

For the unrestricted-rate two-write WOM-code problem, the results were found by the computer search method of Section IV. For three and four writes, we used the WOM-codes described in Section V, and for $5 \leqslant t \leqslant 10$, we used the WOM-codes discussed in Section VI. For the fixed-rate two-write WOM-code problem, we again used the computer search method of Section IV. The constructions for more than two writes were obtained by application of Theorem 13.

APPENDIX

In this appendix, the extension of the WOM-codes construction presented in Section V-B for four and multiple writes is presented.

A) Four-Write WOM-Codes: We next present a construction for four-write binary WOM-codes.

Theorem 14: Let $\mathcal{C}_3$ be an $[n, 2; 2^{n\mathcal{R}_{3,1}}, 2^{n\mathcal{R}_{3,2}}]$ two-write WOM-code over $\mathrm{GF}(3)$ constructed as in Section V-A. Let $\mathcal{C}_2$ be an $[n, 2; 2^{n\mathcal{R}_{2,1}}, 2^{n\mathcal{R}_{2,2}}]$ binary two-write WOM-code. Then, there exists a $[2n, 4; 2^{n\mathcal{R}_{3,1}}, 2^{n\mathcal{R}_{3,2}}, 2^{n\mathcal{R}_{2,1}}, 2^{n\mathcal{R}_{2,2}}]$ four-write WOM-code of sum-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 in Theorem 10. 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 WOM-code $\mathcal{C}_3$, respectively. Similarly, the encoding and decoding maps of the WOM-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 this constructed four-write WOM-code as we did for 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 5: 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 WOM-codes, even if they do not have the same number of cells. Suppose we have a WOM-code over $\mathrm{GF}(3)$ 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 15: The best achievable sum-rate of a four-write WOM-code using the construction in Theorem 14 is $(\log_2 5 + \log_2 3)/2 \approx 1.95$.

Proof: According to Corollary 9, 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 sum-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 $\mathrm{GF}(3)$ of sum-rate 2.2205 found in Section V-A as the WOM-code $\mathcal{C}_3$ and the binary two-write WOM-code of sum-rate 1.4928 found in Section IV as the WOM-code $\mathcal{C}_2$, then there exists a four-write WOM-code of sum-rate $(2.2205 + 1.4928)/2 = 1.8566$.

B) Multiple-Write WOM-Codes: The construction of three- and four-write WOM-codes 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 16: Let $\mathcal{C}_3$ be an $[n,2;2^{n\mathcal{R}_{3,1}},2^{n\mathcal{R}_{3,2}}]$ two-write WOM-code over $\mathrm{GF}(3)$ constructed as in Section V-A. Let $\mathcal{C}_2$ be an $[n,t-2;2^{n\mathcal{R}_{2,1}},\ldots,2^{n\mathcal{R}_{2,t-2}}]$ binary $(t-2)$-write WOM-code. Then, there exists a

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

t-write WOM-code of sum-rate

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

Theorem 16 implies that if there exists a $(t-2)$-write WOM-code of sum-rate $\mathcal{R}_{t-2}$, then there exists a t-write WOM-code of sum-rate

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

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

Corollary 17: For $t \geqslant 3$, there exists a t-write WOM-code of sum-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 $\mathrm{GF}(3)$ of sum-rate 2.2205 and the binary two-write WOM-code of sum-rate 1.4928 from Section IV, then for $t \geqslant 3$ we obtain a t-write WOM-code of sum-rate $\mathcal{R}_t$, where

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

ACKNOWLEDGMENT

The authors thank the anonymous reviewer for valuable comments and suggestions.

REFERENCES

[1] D. Blackwell, *Statistics 262*. Berkeley: Course taught at the University of California, Spring, 1963.

[2] T. Berger, F. Jelinek, and J. K. Wolf, "Permutation codes for sources," *IEEE Trans. Inf. Theory*, vol. IT-18, no. 1, pp. 160–168, Jan. 1972.

[3] V. Bohossian, A. Jiang, and J. Bruck, "Buffer coding for asymmetric multilevel memory," in *Proc. IEEE Int. Symp. Inf. Theory*, Nice, France, Jun. 2007, pp. 1186–1190.

[4] R. Brent, S. Gao, and A. Lauder, "Random Krylov spaces over finite fields," *SIAM J. Discrete Math*, vol. 16, no. 2, pp. 276–287, Feb. 2003.

[5] P. Cappelletti, C. Golla, P. Olivo, and E. Zanoni, *Flash Memories*. Boston, MA: Kluwer, 1999.

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

[7] J. H. Conway and N. J. Sloane, "Orbit and coset analysis of the Golay and related codes," *IEEE Trans. Inf. Theory*, vol. 36, no. 5, pp. 1038–1050, Sep. 1990.

[8] T. M. Cover, "Enumerative source encoding," *IEEE Trans. Inf. Theory*, vol. IT-19, no. 1, pp. 73–77, Jan. 1973.

[9] A. Fiat and A. Shamir, "Generalized "write-once" memories," *IEEE Trans. Inf. Theory*, vol. IT-30, no. 3, pp. 470–480, Sep. 1984.

[10] F. Fu and A. J. H. Vinck, "On the capacity of generalized write-once memory with state transitions described by an arbitrary directed acyclic graph," *IEEE Trans. Inf. Theory*, vol. 45, no. 1, pp. 308–313, Sep. 1999.

[11] E. Gal and S. Toledo, "Algorithms and data structures for flash memories," *ACM Comput. Surv.*, vol. 37, pp. 138–163, Jun. 2005.

[12] S. I. Gel'fand, "Capacity of one broadcast channel," *Problemy Peredachi Informatsii*, vol. 13, no. 3, pp. 106–108, 1977.

[13] P. Godlewski, "WOM-codes construits à partir des codes de Hamming," *Discrete Math.*, vol. 65, no. 3, pp. 237–243, Jul. 1987.

[14] L. Grupp, A. Caulfield, J. Coburn, S. Swanson, E. Yaakobi, P. H. Siegel, and J. K. Wolf, "Characterizing flash memory: Anomalies, observations, and applications," in *Proc. 42nd Annu. IEEE/ACM Int. Symp. Microarchit.*, Dec. 2009, pp. 24–33.

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

[16] A. Jiang, "On the generalization of error-correcting WOM codes," in *Proc. IEEE Int. Symp. Inf. Theory*, Nice, France, Jun. 2007, pp. 1391–1395.

[17] A. Jiang, V. Bohossian, and J. Bruck, "Floating codes for joint information storage in write asymmetric memories," in *Proc. IEEE Int. Symp. Inf. Theory*, Nice, France, Jun. 2007, pp. 1166–1170.

[18] A. Jiang and J. Bruck, "Joint coding for flash memory storage," in *Proc. IEEE Int. Symp. Inf. Theory*, Toronto, ON, Canada, Jul. 2008, pp. 1741–1745.

[19] S. Kayser, E. Yaakobi, P. H. Siegel, A. Vardy, and J. K. Wolf, "Multiple-write WOM-codes," presented at the presented at the 48th Annu. Allerton Conf. Commun., Control Comput., Monticello, IL, Sep. 2010.

[20] D. E. Knuth, "Efficient balanced codes," *IEEE Trans. Inf. Theory*, vol. IT-32, no. 1, pp. 51–53, Jan. 1986.

[21] H. Mahdavifar, P. H. Siegel, A. Vardy, J. K. Wolf, and E. Yaakobi, "A nearly optimal construction of flash codes," in *Proc. IEEE Int. Symp. Inf. Theory*, Seoul, Korea, Jul. 2009, pp. 1239–1243.

[22] F. Merkx, "Womcodes constructed with projective geometries," *Traitement Du Signal*, vol. 1, no. 2–2, pp. 227–231, 1984.

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

[24] R. M. Roth, Private Communication 2010.

[25] R. M. Roth, *Introduction to Coding Theory*. Cambridge, U.K.: Cambridge Univ. Press, 2005.

[26] J. P. M. Schalkwijk, "An algorithm for source coding," *IEEE Trans. Inf. Theory*, vol. IT-18, no. 3, pp. 395–399, May 1972.

[27] C. Tian, V. A. Vaishampayan, and N. J. A. Sloane, "Constant weight codes: A geometric approach based on dissections," *CiteSeerX—Sci. Literature Digital Library Search Engine*, 2010.

[28] 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.

[29] Y. Wu, "Low complexity codes for writing write-once memory twice," in *Proc. IEEE Int. Symp. Inf. Theory*, Austin, TX, Jun. 2010, pp. 1928–1932.

[30] Y. Wu and A. Jiang, "Position modulation code for rewriting write-once memories," *IEEE Trans. Inf. Theory*, vol. 57, no. 6, pp. 3692–3697, Jun. 2011, to be published.

[31] E. Yaakobi, S. Kayser, P. H. Siegel, A. Vardy, and J. K. Wolf, "Efficient two-write WOM-codes," presented at the presented at the IEEE Inf. Theory Workshop, Dublin, Ireland, Aug. 2010.

[32] E. Yaakobi, J. Ma, L. Grupp, P. H. Siegel, S. Swanson, and J. K. Wolf, "Error characterization and coding schemes for flash memories," presented at the presented at the Workshop Appl. Commun. Theory Emerg. Memory Technol., Miami, FL, Dec. 2010.

[33] E. Yaakobi, P. H. Siegel, A. Vardy, and J. K. Wolf, "Multiple error-correcting WOM-codes," in *Proc. IEEE Int. Symp. Inf. Theory*, Austin, TX, Jun. 2010, pp. 1933–1937.

[34] G. Zémor, "Problèmes combinatoires liés à l'écriture sur des mémoires," Ph.D. dissertation, ENST, Paris, France, Feb. 1989.

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

Eitan Yaakobi (S'07) received the B.A. degrees in computer science and mathematics, and the M.Sc. degree in computer science from the Technion—Israel Institute of Technology, Haifa, Israel, in 2005 and 2007, respectively, and the Ph.D. degree in electrical engineering from the University of California, San Diego, in 2011.

He is currently a joint postdoctorate researcher in electrical engineering at the California Institute of Technology, Pasadena, and in the University of California, San Diego, where he is associated with the Center for Magnetic Recording Research. His research interests include coding theory, algebraic error-correction coding, and their applications for digital data storage and in particular for nonvolatile memories.

Scott Kayser received his B.S. degree in Electrical Engineering from the University of California, San Diego in 2010.

He is currently a Ph.D. student at the University of California, San Diego. In the Fall of 2010, he joined Professor Paul Siegel's STAR (Signal Transmission and Recording) group, and is associated with the Center for Magnetic Recording and Research. His research interests include coding theory and, in particular, coding for flash memories.

Paul H. Siegel (M'82–SM'90–F'97) received the S.B. and Ph.D. degrees in mathematics from the Massachusetts Institute of Technology (MIT), Cambridge, in 1975 and 1979, respectively.

He held a Chaim Weizmann Postdoctoral Fellowship at the Courant Institute, New York University. He was with the IBM Research Division in San Jose, CA, from 1980 to 1995. He joined the faculty of the School of Engineering at the University of California, San Diego, in July 1995, where he is currently Professor of Electrical and Computer Engineering. He is affiliated with the California Institute of Telecommunications and Information Technology, the Center for Wireless Communications, and the Center for Magnetic Recording Research, where he holds an endowed chair and served as Director from 2000 to 2011. His primary research interests lie in the areas of information theory and communications, particularly coding and modulation techniques, with applications to digital data storage and transmission.

Prof. Siegel was a member of the Board of Governors of the IEEE Information Theory Society from 1991 to 1996 and was re-elected for a three-year term in 2009. He served as Co-Guest Editor of the May 1991 Special Issue on "Coding for Storage Devices" of the IEEE TRANSACTIONS ON INFORMATION THEORY. He served the same TRANSACTIONS as Associate Editor for Coding Techniques from 1992 to 1995, and as Editor-in-Chief from July 2001 to July 2004. He was also Co-Guest Editor of the May/September 2001 two-part issue on "The Turbo Principle: From Theory to Practice" of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS. He was corecipient, with R. Karabed, of the 1992 IEEE Information Theory Society Paper Award and shared the 1993 IEEE Communications Society Leonard G. Abraham Prize Paper Award with B. Marcus and J. K. Wolf. With J. B. Soriaga and H. D. Pfister, he received the 2007 Best Paper Award in Signal Processing and Coding for Data Storage from the Data Storage Technical Committee of the IEEE Communications Society. He holds several patents in the area of coding and detection, and was named a Master Inventor at IBM Research in 1994. He is a member of Phi Beta Kappa and the National Academy of Engineering.

Alexander Vardy (S'88–M'91–SM'94–F'98) was born in Moscow, U.S.S.R., in 1963. He earned his B.Sc. (summa cum laude) from the Technion, Israel, in 1985, and Ph.D. from the Tel-Aviv University, Israel, in 1991. During 1985–1990 he was with the Israeli Air Force, where he worked on electronic counter measures systems and algorithms. During the years 1992 and 1993 he was a Visiting Scientist at the IBM Almaden Research Center, in San Jose, CA. From 1993 to 1998, he was with the University of Illinois at Urbana-Champaign, first as an Assistant Professor then as an Associate Professor. He is now a Professor in the Department of Electrical Engineering, the Department of Computer Science, and the Department of Mathematics, all at the University of California San Diego (UCSD). While on sabbatical from UCSD, he has held long-term visiting appointments with CNRS, France, the EPFL, Switzerland, and the Technion, Israel.

His research interests include error-correcting codes, algebraic and iterative decoding algorithms, lattices and sphere packings, coding for digital media, cryptography and computational complexity theory, and fun math problems.

He received an IBM Invention Achievement Award in 1993, and NSF Research Initiation and CAREER awards in 1994 and 1995. In 1996, he was appointed Fellow in the Center for Advanced Study at the University of Illinois, and received the Xerox Award for faculty research. In the same year, he became a Fellow of the Packard Foundation. He received the IEEE Information Theory Society Paper Award (jointly with Ralf Koetter) for the year 2004. In 2005, he received the Fulbright Senior Scholar Fellowship, and the Best Paper Award at the IEEE Symposium on Foundations of Computer Science (FOCS). During 1995–1998, he was an Associate Editor for Coding Theory and during 1998–2001, he was the Editor-in-Chief of the IEEE TRANSACTIONS ON INFORMATION THEORY. He was also an Editor for the SIAM Journal on Discrete Mathematics. He has been a member of the Board of Governors of the IEEE Information Theory Society from 1998 to 2006.

Jack Keil Wolf (S'54–M'60–F'73–LF'97) received the B.S.E.E. degree from the University of Pennsylvania and the M.S.E., M.A., and Ph.D. degrees from Princeton University. He was the Stephen O. Rice Professor of Magnetics in the Center of Magnetic Recording Research at the University of California-San Diego and a Vice President, Technology, at Qualcomm, Incorporated. He was a member of the National Academy of Sciences, the National Academy of Engineering, and the American Academy of Arts and Sciences. He was the recipient (or co-recipient) of the following IEEE awards: the 1990 E. H. Armstrong Achievement Award, the 1975 IEEE Information Theory Group Prize Paper Award, the 1992 Leonard G. Abraham Prize Paper Award, the 1998 Koji Kobayashi Award, the 2001 Information Theory Society Shannon Award, the 2004 Hamming Medal, and the 2007 Aaron D. Wyner Award.