WOM with Retained Messages

Lele Wang*, Minghai Qin*, Eitan Yaakobi*†, Young-Han Kim*, Paul H. Siegel*
*University of California, San Diego
Electrical and Computer Engineering Department
†California Institute of Technology
Electrical Engineering Department

Abstract—**Write-once memory (WOM) is a binary storage medium in which each memory cell is initially in state 0 and can be irreversibly programmed to state 1. This paper studies the problem of writing multiple messages into a WOM. Instead of writing a new message (and obliterating old ones) as in the traditional setup, the user wishes to retain access to some of the previously written messages. The capacity region is studied and code constructions are proposed for three canonical cases.**

I. Introduction and Main Results

As part of the tremendous increase in coding research for the ubiquitous flash memories, a considerable attention has been given to rewriting codes. The motivation comes from the special physical properties of the flash memory floating-gate cells, the most conspicuous of which is the asymmetric programming behavior of the cells [1]. The memory cells can only increase their level by the injection of electrons to each cell. However, in order to decrease the level of even a single cell, its entire containing block ($\sim 10^6$ cells) has to be erased. This undesired property not only reduces the writing speed but also significantly affects the lifetime of flash memories, which is often specified in terms of a maximum number of block erasures [1]. As this number can be as low as a few hundreds or thousands, reducing the number of block erasures becomes critical in improving the lifetime of flash memories.

The idea of rewriting codes dates back to the pioneering work [10] by Rivest and Shamir on write-once memory (WOM) in 1982. The motivation came from storage media such as punch cards and ablative optical disks. These media are modeled as a collection of write-once binary cells, where each cell is initially in state 0 and can be irreversibly programmed to state 1. Figure 1 shows a typical model for rewriting t times on a binary WOM.

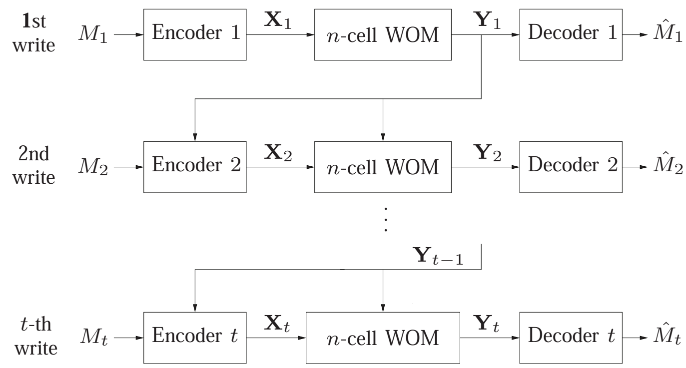

Fig. 1. A t-write WOM model

An $[n, t; 2^{nR_1}, \ldots, 2^{nR_t}]$ WOM code consists of

- t message sets $[1 : 2^{nR_1}], \ldots, [1 : 2^{nR_t}]$,
- t encoders, where encoder $i \in [1 : t]$ for the i-th write assigns a codeword $\mathbf{x}_i = \mathcal{E}_i(m_i, \mathbf{y}_{i-1}) \in \{0,1\}^n$ ($\mathbf{y}_0 = \emptyset$) to each message $m_i \in [1 : 2^{nR_i}]$ and the cell levels $\mathbf{y}_{i-1}$ from the previous write, and
- t decoders, where decoder $i \in [1 : t]$ assigns an estimate $\hat{m}_i = \mathcal{D}_i(\mathbf{y}_i)$ or an error e to the cell levels $\mathbf{y}_i$ from the i-th write.

The notation $[i : j]$ denotes the set $\{k \in \mathbb{Z} : i \leqslant k \leqslant j\}$. The average probability of error of the WOM code is defined as $P_e^{(n)} = \mathsf{P}\{(\hat{M}_1, \ldots, \hat{M}_t) \neq (M_1, \ldots, M_t)\}$. A rate tuple $(R_1, \ldots, R_t)$ is said to be *achievable* for the WOM if there exists a sequence of $[n, t; 2^{nR_1}, \ldots, 2^{nR_t}]$ WOM codes such that $\lim_{n\to\infty} P_e^{(n)} = 0$. The *capacity region* $\mathscr{C}_{\text{WOM}}(t)$ is the closure of the set of all achievable rate tuples $(R_1, \ldots, R_t)$. The *sum-capacity* $C_{\text{sum}}(t)$ of WOM is the maximum achievable *sum-rate* $\sum_{j=1}^t R_j$. A sequence of WOM codes is said to be *sum-rate optimal* if its sum-rate approaches the sum-capacity in the limit.

The capacity region as well as the sum-capacity for the WOM model is well studied in the literature [3], [5]; for example, it is known that the t-write sum-capacity is $C_{\text{sum}}(t) = \log_2(t+1)$ and the capacity region is

$$\begin{aligned}\mathscr{C}_{\text{WOM}}(t) = \Big\{&(R_1, \ldots, R_t) | R_1 \leqslant H(p_1),\\ &R_i \leqslant \left(\prod_{k=1}^{i-1} p_k\right) H(p_i),\ i \in [2 : t-1],\\ &R_t \leqslant \prod_{k=1}^{t-1} p_k \text{ for some } p_1, \ldots, p_{t-1} \in [\tfrac{1}{2}, 1]\Big\}.\end{aligned}$$

Furthermore, many code constructions have been found (e.g., [8], [12], [13]), the sum-rates of which are close to the sum-capacity.

Following the work by Rivest and Shamir on the binary WOM, many papers on WOM codes appeared during the 1980s and 1990s, (e.g., [2], [3], [5], [11]) as well as in the past few years (e.g., [4], [6], [8], [9], [12]). Among all of the existing models for rewriting on flash memories, one assumes that a *new* message is stored in the memory on each write, effectively overwriting previously written messages. This can be a drawback in some applications, however, if the user wishes to retain access to one or more previously written messages. For example, suppose that a police station keeps traffic surveillance videos for up to a certain amount of time, say 30 days. This requires that the most current video as well as the videos from the previous 29 days be retrievable at any time. If the entire set of 30 daily videos are treated as a completely new message to be written on top of the existing content of the memory cells, the writing efficiency will be low, because the same message is being written multiple times via different codewords.

This motivates the model of rewriting flash memories with retained messages. This model is related to the work on buffer codes and trajectory codes [6], [7], which are capable of remembering the most recent values stored in the memories. To make the problem simple, in this paper we consider rewriting on a binary WOM, where after each write the current message and some of the previously written messages need to be retrievable. We aim to characterize the optimal rate tradeoff and find code constructions focusing on three concrete problems motivated by different scenarios in real storage systems.

In Section II, we formulate the problem of retaining two days of video surveillance as follows.

Problem 1. Consecutive two-step WOM
On the i-th write, $i \in [1:t]$, encoder i stores (M_{i-1}, M_i) ($M_0 = \emptyset$) and decoder i has to recover both messages.

By ignoring the correlation between message pairs over multiple writes and treating (M_{i-1}, M_i) as a new message, one can achieve roughly $\frac{1}{2}\log_2(t+1)$ in sum-rate using a traditional WOM code, since every $M_i, i \in [1:t-1]$, is written twice. Is this optimal? We establish in Theorem 1 that the sum-capacity of the consecutive two-step WOM model is $\log_2\left(\left\lceil \frac{t}{2} \right\rceil + 1\right)$, which can be twice as large as $\frac{1}{2}\log_2(t+1)$ for large t. How can we fully exploit the correlation among messages? We propose in Construction 1 a very simple code, which turns out to be sum-rate optimal. The idea is to partition the set of n cells into two blocks and to update the new message alternately on the two blocks, as shown in Table I. An outer bound on the capacity region for general t is also derived.

TABLE I
WRITING ARRANGEMENT OF THE CONSECUTIVE 2-STEP WOM CODE

	block 1	block 2
1st write	M_1	
2nd write	M_1	M_2
3rd write	M_3	M_2
4th write	M_3	M_4
5th write	M_5	M_4

Now suppose the police station wishes to keep track of the video from the most recent day on which there was a traffic accident, as well as the traffic video for the current day. Since the traffic accident is unpredictable, we cannot tell which part of the video will be kept for the next day until the end of the current day. This is the situation in which the retrievable message for the next write can be an *arbitrary* one from the two messages currently written in the memory. To be more concrete, we formulate the problem as follows.

Problem 2. Arbitrary two-step WOM
On the first write, encoder 1 stores message M_1 and decoder 1 has to recover M_1. On the i-th write, $i \in [2:t]$, encoder i stores $(M_{s(i)}, M_i)$, where $M_{s(i)} \in \{M_{s(i-1)}, M_{i-1}\}$ is arbitrarily chosen from the two messages stored on the $(i-1)$-st write, and decoder i has to recover both messages.

For this problem, an idea arises naturally from the construction for the consecutive two-step WOM. With Table I in mind, we store M_1 and M_2 the same way as before for the first two writes. If (M_1, M_3) is stored on the third write, we update M_3 on block 2. If instead (M_2, M_3) is stored on the third write, we update M_3 on block 1. It can be shown that the sum-rate is roughly $\frac{1}{2}\log_2(t)$ in the worst case scenario. Can we do better than this? In Section III, we construct a code by enlarging the number of blocks, and we show that it can strictly outperform the above code. Moreover, it is shown to be asymptotically optimal in t. A simple outer bound on the capacity region and an upper bound on the sum-capacity are also presented.

Now we introduce the last problem. Suppose the surveillance videos are layered as high-fidelity and low-fidelity ones, e.g., encoded by the H.264 standard. On each day, all low-fidelity videos from previous days and the high-fidelity video from the current day should be stored. This motivates the following.

Problem 3. Incremental WOM
We rewrite each message M_i, $i \in [1:t]$, as two independent parts: the common message $M_i^c \in [1:2^{nR_i^c}]$ and the private message $M_i^p \in [1:2^{nR_i^p}]$, i.e., $M_i = (M_i^c, M_i^p)$. On the i-th write, encoder i stores all the previous common messages and its own full message, i.e., $(M_1^c, M_2^c, \ldots, M_i^c, M_i^p)$, and decoder i has to recover all of them.

One extreme special case of this problem is $M_i^c = \emptyset, i \in [1:t]$, i.e., there is no common message. Then we go back to the traditional t-write WOM. The other extreme special case is $M_i^p = \emptyset, i \in [1:t]$, i.e., there is no private message. Since all the previously written messages have to be recoverable by the current decoder, the performance is fundamentally limited by the last write. It can be shown that the capacity region for this extreme problem is $\sum_{i=1}^{t} R_i \leqslant 1$. Thus, an obvious choice to maximize the sum-rate is to set all the common-message rates to be zero and the sum-capacity is readily established as $\log_2(t+1)$. In Section IV, we establish the optimal trade-off between the common-message sum-rate $R_{\text{sum}}^c := \sum_{i=1}^{n} R_i^c$ and the private-message sum-rate $R_{\text{sum}}^p := \sum_{i=1}^{n} R_i^p$. Moreover, we investigate the *symmetric sum-capacity* $C_{\text{ssum}}^3(t)$, defined as the maximum achievable sum-rate when $R_1^c = R_1^p = R_2^c = R_2^p = \cdots = R_t^c = R_t^p = R$.

Since the problem formulation is apparently a combination of two completely solved extreme problems, one might think that a time-sharing strategy between the two optimal coding schemes would be optimal. Surprisingly, in Section IV, we construct a code that strictly outperforms the time-sharing code and is asymptotically optimal in t. The performance of this construction is illustrated in Figure 2.

II. Consecutive Two-Step WOM

In this section we establish the sum-capacity as well as an outer bound and an inner bound on the capacity region for the consecutive two-step WOM defined in Problem 1.

Proposition 1. *(Outer bound on the capacity region) If a rate tuple $(R_1, R_2, \ldots, R_t)$ is achievable for the t-write consecutive two-step WOM, it must satisfy $R_1 \leqslant H(Y_1), R_1 + R_2 \leqslant H(Y_2), R_2 + R_3 \leqslant H(Y_3|Y_1), R_3 + R_4 \leqslant H(Y_4|Y_2), \ldots, R_{t-1} + R_t \leqslant H(Y_t|Y_{t-2})$ for some pmf $p(x_1)p(x_2|y_1)\cdots p(x_t|y_{t-1})$.*

Proposition 2. *(Inner bound on the capacity region) For even t, let $s = t/2$. If two rate tuples $(R_1', \ldots, R_s')$ and $(R_1'', \ldots, R_s'')$ are achievable for the s-write WOM, then for* *all $\lambda \in [0,1]$, with $\bar{\lambda} = 1-\lambda$, the rate tuple $(R_1,\ldots,R_t) = (\lambda R_1', \bar{\lambda} R_1'', \lambda R_2', \bar{\lambda} R_2'', \ldots, \lambda R_s', \bar{\lambda} R_s'')$ is achievable for the t-write consecutive two-step WOM.*

For odd t, let $s = (t-1)/2$. If the rate tuple $(R_1',\ldots,R_{s+1}')$ is achievable for the $(s+1)$-write WOM and the rate tuple $(R_1'',\ldots,R_s'')$ is achievable for the s-write WOM, then for all $\lambda \in [0,1]$, the rate tuple $(R_1,\ldots,R_t) = (\lambda R_1', \bar{\lambda} R_1'', \lambda R_2', \bar{\lambda} R_2'', \ldots, \lambda R_s', \bar{\lambda} R_s'', \lambda R_{s+1}')$ is achievable for the t-write consecutive two-step WOM.

The above outer and inner bounds coincide at the sum-rate and establish the sum-capacity of the consecutive two-step WOM for every t.

Theorem 1. *The sum-capacity $C^1_{\text{sum}}(t)$ of the t-write consecutive two-step WOM is*

$$C^1_{\text{sum}}(t) = \log_2\left(\left\lceil \frac{t}{2} \right\rceil + 1\right).$$

Due to space limitations, we skip the proofs. In the following, we give a code construction[1] for even t, which is sum-rate optimal. It also serves as part of the proof for Proposition 2. Partition the set of all cells into two blocks and write odd messages to one block on odd writes and even messages to the other block on even writes, as shown in Table I. Thus, each block of cells can reliably store $t/2$ messages using a traditional $(t/2)$-write WOM code and decoder i can recover both messages (M_{i-1}, M_i) stored in the two blocks.

Construction 1 *Let t and n be positive integers, with t even, and let $\lambda \in [0,1]$ such that λn is an integer. Let $\bar{\lambda} = 1-\lambda$ and $s = t/2$. Suppose that the cell levels after the i-th write, $i \in [1:t]$, are $(\mathbf{y}_i', \mathbf{y}_i'')$, where $\mathbf{y}_i'$ and $\mathbf{y}_i''$ denote blocks of lengths λn and $(1-\lambda)n$, respectively. Let $\mathcal{C}_1$ be a $[\lambda n, s; 2^{\lambda n R_1'}, \ldots, 2^{\lambda n R_s'}]$ WOM code of length λn with encoder $\mathcal{E}_i'(m_i, \mathbf{y}_{i-1}')$, $m_i \in [1 : 2^{\lambda n R_i'}]$, on the i-th write, $i \in [1:s]$. Let $\mathcal{C}_2$ be a $[\bar{\lambda} n, s; 2^{\bar{\lambda} n R_1''}, \ldots, 2^{\bar{\lambda} n R_s''}]$ WOM code of length $\bar{\lambda} n$ with encoder $\mathcal{E}_i''(m_i, \mathbf{y}_{i-1}'')$, $m_i \in [1 : 2^{\bar{\lambda} n R_i''}]$, on the i-th write, $i \in [1:s]$. Let $R_{2i-1} = \lambda R_i'$ and $R_{2i} = \bar{\lambda} R_i''$, $\forall i \in [1:s]$. An $[n, t; 2^{nR_1}, \ldots, 2^{nR_t}]$ consecutive two-step WOM code $\mathcal{C}$ of length n is constructed as follows. The cells are partitioned into block 1 with length λn and block 2 with length $\bar{\lambda} n$. On the i-th write, the encoder i assigns the codeword $\mathbf{x}_i = (\mathbf{x}_i', \mathbf{x}_i'')$ as follows:*

1) *For odd $i = 2j-1$, write message $m_i \in [1 : 2^{nR_i}]$ to block 1 using the encoder on the j-th write from $\mathcal{C}_1$ and keep block 2 unchanged, i.e.,*
$$\mathbf{x}_i' = \mathcal{E}_j'(m_i, \mathbf{y}_{i-1}').$$
2) *For even $i = 2j$, write message $m_i \in [1 : 2^{nR_i}]$ to block 2 using the encoder on the j-th write from $\mathcal{C}_2$ and keep block 1 unchanged, i.e.,*
$$\mathbf{x}_i'' = \mathcal{E}_j''(m_i, \mathbf{y}_{i-1}''). \quad \blacksquare$$

It can be seen that $\mathcal{C}$ is a consecutive two-step WOM code. If $R_i' = R_i'', \forall i \in [1:s]$, the sum-rate of $\mathcal{C}$ in Construction 1 is $\sum_{i=1}^{t} R_t = \sum_{i=1}^{s} \lambda R_i' + \sum_{i=1}^{s} \bar{\lambda} R_i'' = \sum_{i=1}^{s} R_i'$. Therefore, if $\mathcal{C}_1$ and $\mathcal{C}_2$ are sum-rate optimal, then $\mathcal{C}$ achieves the sum-capacity $\log_2(\frac{t}{2}+1)$. For odd t, a consecutive two-step WOM code can be constructed similarly.

[1]In all the following constructions, the decoders of the WOM codes for Problems 1, 2, and 3 are similar to the decoders of the traditional WOM codes that are assumed to exist in each construction, and thus we omit the details of the decoders here.

III. Arbitrary Two-Step WOM

In this section we study the arbitrary two-step WOM defined in Problem 2. Note that if a WOM code $\mathcal{C}$ is a t-write arbitrary two-step WOM code, then we can construct from it a t-write consecutive two-step WOM code. Therefore, the sum-capacity $C^2_{\text{sum}}(t)$ of the arbitrary two-step WOM is upper bounded as $C^2_{\text{sum}}(t) \leqslant C^1_{\text{sum}}(t) = \log_2(\lceil \frac{t}{2} \rceil + 1)$.

Now we give a construction that strictly outperforms the construction in the introduction and achieves $2/3$ of $C^1_{\text{sum}}(t)$ of the consecutive two-step WOM, while keeping track of arbitrary messages as required. Partition the set of all cells into three blocks as illustrated in Table II. In the first two blocks, we write in the exactly same manner as for the consecutive two-step WOM. The third block is updated with $M_{s(i)}$ every other write to help retrieve the desired message of the arbitrary demand. This can be improved by further enlarging the number of blocks as given in Construction 2.

TABLE II
Writing arrangement of the arbitrary 2-step WOM code

	block 1	block 2	block 3
1st write	M_1		
2nd write	M_1	M_2	$M_{s(2)}$
3rd write	M_3	M_2	$M_{s(2)}$
4th write	M_3	M_4	$M_{s(4)}$
5th write	M_5	M_4	$M_{s(4)}$

Construction 2 *Let ℓ and t be positive integers such that t is a multiple of ℓ. The cells consist of $\ell+1$ blocks, each of size n'; thus $n = n'(l+1)$. After the i-th write, $i \in [1:t]$, the cell levels are $(\mathbf{y}_i^{(1)}, \mathbf{y}_i^{(2)}, \ldots, \mathbf{y}_i^{(\ell+1)})$, where $\mathbf{y}_i^{(j)}$, $j \in [1:\ell+1]$, denotes the j-th block of length-n' cells. Let $\mathcal{C}_W$ be an $[n', t/\ell; 2^{n'R_1'}, \ldots, 2^{n'R_{t/\ell}'}]$ WOM code of length n' with encoder $\mathcal{E}_i'(m_i, \mathbf{y}_{i-1})$, $m_i \in [1 : 2^{n'R_i'}]$, on the i-th write, $i \in [1 : t/\ell]$. An $[n, t; 2^{nR_1}, 2^{nR_2}, \ldots, 2^{nR_t}]$ arbitrary two-step WOM code $\mathcal{C}$ is constructed, where $R_i = R'_{\lceil i/\ell \rceil}, i \in [1:t]$. On the i-th write, $i \in [1:t]$, the encoder i assigns the codeword $\mathbf{x}_i = (\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)}, \ldots, \mathbf{x}_i^{(\ell+1)})$ using the following rules. Let $h = (i-1 \bmod \ell)$ and $j = \lceil \frac{i}{\ell} \rceil$.*

1) *Write message $m_i \in [1 : 2^{n'R_i}]$ to the $(h+1)$-st block, using the encoder on the j-th write from $\mathcal{C}_W$ and keep the rest of the first $n'\ell$ cells unchanged, i.e.,*
$$\mathbf{x}_i^{(h+1)} = \mathcal{E}_j'(m_i, \mathbf{y}_{i-1}^{(h+1)}).$$
2) *If $h = 0$ and $i \neq 1$, write message $m_{s(i)}$ to the $(\ell+1)$-st block, using the encoder on the $(j-1)$-st write from $\mathcal{C}_W$, i.e.,*
$$\mathbf{x}_i^{(\ell+1)} = \mathcal{E}_{j-1}'(m_{s(i)}, \mathbf{y}_{i-1}^{(\ell+1)}).$$
Otherwise, the last block is kept unchanged. $\blacksquare$

Proposition 3. *If the WOM code $\mathcal{C}_W$ is sum-rate optimal, then the code $\mathcal{C}$ in Construction 2 is an arbitrary two-step WOM code with sum-rate*

$$R^2_{\text{sum}}(t) = \frac{\ell}{\ell+1} \log_2\left(\frac{t}{\ell} + 1\right).$$

For large t, let $\ell = \log_2 t$. Then the asymptotic sum-rate is $R^2_{\text{sum}}(t) = \frac{\log_2 t}{\log_2 t+1} \log_2(\frac{t}{\log_2 t}+1) = \log_2 t - O(\log_2(\log_2 t))$. Since an upper bound on the sum-capacity is $\log_2\left(\lceil \frac{t}{2} \rceil + 1\right)$, this construction is asymptotically optimal in t.

If t is not a multiple of ℓ we slightly modify Construction 2. We use a $\lceil \frac{t}{\ell} \rceil$-write WOM code for the first $(t \bmod \ell)$ blocks and a $\lfloor \frac{t}{\ell} \rfloor$-write WOM code for the last $(\ell + 1 - (t \bmod \ell))$ blocks. The constructions yield the following corollary.

Corollary 1 *A lower bound of $C^2_{\text{sum}}(t)$ is given by*

$$\max_{\ell \in [1:t]} \frac{(t \bmod \ell)\log_2(\lceil \frac{t}{\ell} \rceil + 1) + (\ell - (t \bmod \ell))\log_2(\lfloor \frac{t}{\ell} \rfloor + 1)}{\ell + 1}.$$

IV. Incremental WOM

We study the incremental WOM model in Problem 3.

Theorem 2. *The sum-capacity of the t-write incremental WOM is*

$$C^3_{\text{sum}}(t) = \log_2(t+1).$$

The optimal trade-off between the common-message sum-rate $R^c_{\text{sum}} := \sum_{i=1}^n R^c_i$ and the private-message sum-rate $R^p_{\text{sum}} := \sum_{i=1}^n R^p_i$ is the set of rate pairs $(R^c_{\text{sum}}, R^p_{\text{sum}})$ such that

$$R^c_{\text{sum}} \leqslant \prod_{i=1}^{t-1} p_i,$$

$$R^c_{\text{sum}} + R^p_{\text{sum}} \leqslant \prod_{i=1}^{t-1} p_i + \sum_{i=1}^{t-1} \left(\prod_{k=1}^{i-1} p_k \right) H(p_i)$$

for some $p_1, p_2, \ldots, p_{t-1} \in [\frac{1}{2}, 1]$.

Theorem 2 follows by noting that $R^c_1 = R^c_2 = \cdots = R^c_{t-1} = 0$ is optimal for the sum-rate trade-off.

Now we focus on the symmetric sum-capacity $C^3_{\text{ssum}}(t)$, defined as the maximum achievable sum-rate when $R^c_1 = R^p_1 = R^c_2 = R^p_2 = \cdots = R^c_t = R^p_t = R$. We denoted by $[n, t; 2^{nR}]$ the t-write symmetric incremental WOM code. It can be proved that the symmetric sum-capacity of the t-write incremental WOM is upper bounded as

$$C^3_{\text{ssum}}(t) \leqslant 2 - \frac{2}{t+1} < 2.$$

In the following, we give a construction of t-write symmetric incremental WOM codes. To illustrate the basic idea, we show a construction for $t = 3$. Suppose that every private/common message represents $k = nR$ bits. Partition the n cells into three blocks. Write M^c_1 to the first block. Partition M^p_1 into two messages M^p_{11} with λk bits and M^p_{12} with $(1-\lambda)k$ bits. Write M^p_{11} and M^c_2 to the second block and $M^p_{12}, M^p_2, (M^c_3, M^p_3)$ to the third block. Thus, in the first block we use a one-write WOM code, in the second block we use a two-write WOM code and in the third block we use a three-write WOM code, as illustrated in Table III.

TABLE III
WRITING ARRANGEMENT OF THE 3-WRITE INCREMENTAL WOM CODE

	block 1	block 2	block 3
1st write	M^c_1	M^p_{11}	M^p_{12}
2nd write	M^c_1	M^c_2	M^p_2
3rd write	M^c_1	M^c_2	(M^c_3, M^p_3)

Suppose that the lengths of the first, second, and third blocks are n_1, n_2, and n_3, respectively. For the fixed k, the problem of maximizing the symmetric sum-rate is identical to minimizing the value of $n = n_1 + n_2 + n_3$ as a function of λ. Now we state the construction formally and then present the symmetric sum-rate analysis.

Construction 3 *Let k be a positive integer, $\lambda \in [0,1]$, and $n_1 = k$. Suppose that the cell levels after the i-th write are $(\mathbf{y}'_i, \mathbf{y}''_i, \mathbf{y}'''_i)$, where $\mathbf{y}'_i, \mathbf{y}''_i$, and $\mathbf{y}'''_i$ denote blocks of length n_1, n_2, and n_3, respectively. Let $\mathcal{C}_1$ be an $[n_1, 1; 2^k]$ WOM code with encoder $\mathcal{E}'_1(m_1)$ for the first write, $\mathcal{C}_2$ be an $[n_2, 2; 2^{\lambda k}, 2^k]$ WOM code with encoders $\mathcal{E}''_i(m_i, \mathbf{y}''_{i-1})$, $i \in [1:2]$, for the first two writes, and $\mathcal{C}_3$ be an $[n_3, 3; 2^{(1-\lambda)k}, 2^k, 2^{2k}]$ WOM code with encoder $\mathcal{E}'''_i(m_i, \mathbf{y}'''_{i-1})$, $i \in [1:3]$, for all three writes. An $[n, 3; (2^k, 2^k)]$ symmetric incremental WOM code $\mathcal{C}$ is constructed. On the i-th write, encoder i assigns the codeword $\mathbf{x}_i = (\mathbf{x}'_i, \mathbf{x}''_i, \mathbf{x}'''_i)$ using the following encoding rules:*

1) *If $i = 1$, then write message $m^c_1 \in [1:2^k]$ to block 1 using the encoder from $\mathcal{C}_1$, write message $m^p_{11} \in [1:2^{\lambda k}]$ to block 2 using the encoder for the first write from $\mathcal{C}_2$, and write message $m^p_{12} \in [1:2^{(1-\lambda)k}]$ to block 3 using the encoder for the first write from $\mathcal{C}_3$, i.e.,*
$$(\mathbf{x}'_1, \mathbf{x}''_1, \mathbf{x}'''_1) = \left(\mathcal{E}'_1(m^c_1), \mathcal{E}''_1(m^p_{11}), \mathcal{E}'''_1(m^p_{12})\right).$$
2) *If $i = 2$, then the first n_1 cells are unchanged, write message $m^c_2 \in [1:2^k]$ to block 2 using the encoder for the second write from $\mathcal{C}_2$, and write message $m^p_2 \in [1:2^k]$ to block 3 using the encoder for the second write from $\mathcal{C}_3$, i.e.,*
$$(\mathbf{x}''_2, \mathbf{x}'''_2) = \left(\mathcal{E}''_2(m^c_2, \mathbf{y}''_1), \mathcal{E}'''_2(m^p_2, \mathbf{y}'''_1)\right).$$
3) *If $i = 3$, then the first $n_1 + n_2$ cells are unchanged and write message $(m^c_3, m^p_3) \in [1:2^{2k}]$ to block 3, using the encoder for the third write from $\mathcal{C}_3$, i.e.,*
$$\mathbf{x}'''_3 = \mathcal{E}'''_3\left((m^c_3, m^p_3), \mathbf{y}'''_2\right). \qquad \blacksquare$$

The symmetric sum-rate of the code $\mathcal{C}$ is given by $6k/n$. As k is fixed, this value is maximized when n is minimized. We denote by $n_2(\lambda)$ the minimum length of an $[n_2, 2; 2^{\lambda k}, 2^k]$ WOM code and similarly $n_3(\lambda)$ is the minimum length of an $[n_3, 3; 2^{(1-\lambda)k}, 2^k, 2^{2k}]$ WOM code. Then, the problem is to find the value of $\min_{\lambda \in [0,1]}(n_2(\lambda) + n_3(\lambda))$.

Proposition 4. *The minimum value of n in Construction 3 is $n = 4.386k$, which is achieved by setting $\lambda = 0.3116$. The corresponding symmetric sum-rate is $R^3_{\text{ssum}}(3) = 1.3679$.*

Proof: Let us first find the value of $n_2(\lambda)$. That is, we find a WOM code of minimum length $n_2(\lambda)$ such that its rate on the first write is $R_1 = \lambda k / n_2(\lambda)$ and its rate on the second write is $R_2 = k/n_2(\lambda)$. Thus, we have $R_1/R_2 = \lambda$. Since the capacity region of the two-write WOM is given by $\{(R_1, R_2) | R_1 \leqslant h(p_1), R_2 \leqslant p_1, \text{ for some } p_1 \in [1/2, 1]\}$, and we find a WOM code of minimum length, the ratio of R_1 and R_2 satisfies

$$h(p_1)/p_1 = \lambda, \tag{1}$$

for some $p_1 \in [1/2, 1]$. Note that if λ is positive then equation (1) always has a solution, which we denote by $p_1(\lambda)$. Now, we deduce from $R_2 = k/n_2 = p_1$ that $n_2(\lambda) = k/p_1(\lambda)$.

Similarly, the capacity region of the three-write WOM is given by $\{(R_1, R_2, R_3) | R_1 \leqslant h(p_2), R_2 \leqslant p_2 h(p_3), R_3 \leqslant p_2 p_3,$ for some $p_2, p_3 \in [1/2, 1]\}$. Thus, it can be shown that the values of $p_2, p_3 \in [1/2, 1]$ that give the minimum code length for $n_3(\lambda)$ satisfy

$$h(p_2)/(p_2 h(p_3)) = 1 - \lambda, \tag{2}$$

$$h(p_3)/p_3 = 1/2. \tag{3}$$

The value of p_3 is independent of λ and is given by $p_3 = 0.9055$, and $p_2(\lambda)$ is the solution to equation (2). Hence, $n_3(\lambda)$ satisfies $n_3(\lambda) = 2k/((p_2(\lambda)p_3)$.

We are now left to solve the minimization problem

$$\text{minimize} \left(\frac{1}{p_1(\lambda)} + \frac{2}{p_2(\lambda)p_3} \right), \tag{4}$$

with $\lambda \in [0, 1]$, where $p_1(\lambda), p_2(\lambda)$, and p_3 satisfy equations (1), (2), and (3) respectively.

From Equation (3), p_3 was already calculated numerically. From Equation (1) and (2), we have

$$\frac{h(p_1(\lambda))}{p_1(\lambda)} + \frac{h(p_2(\lambda))}{p_2(\lambda)h(p_3)} = 1.$$

Therefore, we can formulate the minimization problem as

$$\text{minimize} \left(\frac{1}{p_1} + \frac{2}{p_2 p_3} \right)$$

with $p_1, p_2 \in [\frac{1}{2}, 1]$, subject to

$$\frac{h(p_1)}{p_1} + \frac{h(p_2)}{p_2 h(p_3)} = 1, \text{ where } p_3 = 0.9055.$$

It follows that $p_1 = p_2 = 0.9479$ and we get $\lambda = h(p_1)/p_1 = h(p_3)/(h(p_3) + 1) = 0.3116$. Therefore, $n = n_1 + n_2(\lambda) + n_3(\lambda) = 4.386k$ and $R^3_{\text{ssum}}(3)$ satisfies

$$R^3_{\text{ssum}}(3) = 6R = \frac{6k}{n} = \frac{6k}{n_1 + n_2(\lambda) + n_3(\lambda)} = 1.3679.$$

This completes the proof. ■

We are now ready to generalize the construction for an arbitrary number of writes t. Each of the messages $M_1^c, M_1^p, \ldots, M_t^c, M_t^p$ represents $k = nR$ bits, and the n cells are partitioned into t blocks. Message $M_i^p, i \in [1 : t-2]$, is partitioned into $t - i$ parts $(M_{i1}^p, M_{i2}^p, \ldots, M_{i,t-i}^p)$. The arrangement of these messages when written into the memory is depicted in Table IV.

TABLE IV
WRITING ARRANGEMENT OF THE t-WRITE INCREMENTAL WOM CODE

	block 1	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	block t
1	M_1^c	M_{11}^p	M_{12}^p	$\cdots$	$\cdots$	$M_{1,t-2}^p$	$M_{1,t-1}^p$
2	M_1^c	M_2^c	M_{21}^p	$\cdots$	$\cdots$	$M_{2,t-3}^p$	$M_{2,t-2}^p$
3	M_1^c	M_2^c	M_3^c	M_{31}^p	$\cdots$	$M_{3,t-4}^p$	$M_{3,t-3}^p$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
	M_1^c	M_2^c	M_3^c	M_4^c	$\cdots$	$M_{t-2,1}^p$	$M_{t-2,2}^p$
	M_1^c	M_2^c	M_3^c	M_4^c	$\cdots$	M_{t-1}^c	M_{t-1}^p
t	M_1^c	M_2^c	M_3^c	M_4^c	$\cdots$	M_{t-1}^c	(M_t^c, M_t^p)

According to this layout, the i-th block, for $i \in [1 : t]$, consists of n_i cells and is used to construct an i-write WOM code. Assume that message M_{ij}^p for $i \in [1 : t-2], j \in [1 : t-i]$ represents $\lambda_{i,j}k$ bits, where $\sum_{j=1}^{t-i} \lambda_{i,j} = 1$. Then for $i \in [2 : t-1]$, messages $(M_{1,i-1}^p, M_{2,i-2}^p, \ldots, M_{i-1,1}^p, M_i^c)$ will be written as an i-write WOM code of length n_i, and messages $\left(M_{1,t-1}^p, M_{2,t-2}^p, \ldots, M_{t-2,2}^p, M_{t-1}^p, (M_t^c, M_t^p)\right)$ will be written as a t-write WOM code of length n_t, where (M_t^c, M_t^p) represents $2k$ bits.

Figure 2 shows the achievable symmetric sum-rate of the time-sharing scheme described in the introduction and our construction based on the optimal partition strategy, $\lambda_{i,j}$, $i \in [1 : t-2], j \in [1 : t-i]$, that maximizes $R^3_{\text{ssum}}(t)$.

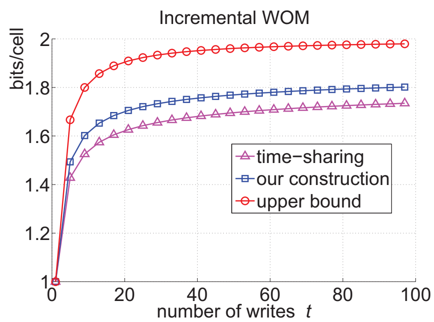

Fig. 2. Lower and upper bounds on $C^3_{\text{ssum}}(t)$

V. Acknowledgment

This research was supported in part by the ISEF Foundation, the Lester Deutsch Fellowship, the University of California Lab Fees Research Program, Award No. 09-LR-06-118620-SIEP, the National Science Foundation under Grants CCF-0747111 and CCF-1116739, and the Center for Magnetic Recording Research at the University of California, San Diego.

References

[1] P. Cappelletti, C. Golla, P. Olivo, and E. Zanoni, *Flash Memories*. Kluwer Academic Publishers, 1st Edition, 1999.

[2] F. Fu, "Maximum information bits stored in reusable memory," *Chinese Science Bulletin*, vol. 40, no. 15, pp. 1241–1244, August 1995.

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

[4] R. Gabrys and L. Dolecek, "Characterizing capacity achieving write once memory codes for multilevel flash memoriess," in *Proc. IEEE Int. Symp. Inform. Theory*, July-August 2011, pp. 2517–2521.

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

[6] A. Jiang, V. Bohossian, and J. Bruck, "Rewriting codes for joint information storage in flash memories," *IEEE Trans. Inform. Theory*, vol. 56, no. 10, pp. 5300–5313, October 2010.

[7] A. Jiang, M. Langberg, M. Schwartz, and J. Bruck, "Universal rewriting in constrained memories," in *Proc. IEEE Int. Symp. Inform. Theory*, Seoul, Korea, July 2009, pp. 1219–1223.

[8] S. Kayser, E. Yaakobi, P. H. Siegel, A. Vardy, and J. K. Wolf, "Multiple-write WOM-codes," in *Proc. 48-th Annual Allerton Conference on Communication, Control and Computing*, Monticello, IL, September 2010.

[9] L. Wang and Y.-H. Kim, "Sum-capacity of multiple-write noisy memory," in *Proc. IEEE Int. Symp. Inform. Theory*, Saint Petersburg, Russia, August 2011, pp. 2494–2498.

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

[11] 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.

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

[13] E. Yaakobi, S. Kayser, P. H. Siegel, A. Vardy, and J. K. Wolf, "Efficient two-write WOM-codes," in *Proc. IEEE Inform. Theory Workshop*, Dublin, Ireland, August 2010.