Constrained Codes that Mitigate Inter-Cell Interference in Read/Write Cycles for Flash Memories

Minghai Qin, *Student Member, IEEE,* Eitan Yaakobi, *Member, IEEE,* and Paul H. Siegel, *Fellow, IEEE*

***Abstract*—Inter-cell interference (ICI) is one of the main obstacles to precise programming (i.e., writing) of a flash memory. In the presence of ICI, the voltage level of a cell might increase unexpectedly if its neighboring cells are programmed to high levels. For q-ary cells, the most severe ICI arises when three consecutive cells are programmed to levels high - low - high, represented as $(q-1)\mathbf{0}(q-1)$, resulting in an unintended increase in the level of the middle cell and the possibility of decoding it incorrectly as a nonzero value. *ICI-free codes* are used to mitigate this phenomenon by preventing the programming of any three consecutive cells as $(q-1)\mathbf{0}(q-1)$.**

In this work, we extend ICI-free codes in two directions. First, we consider binary balanced ICI-free codes which, in addition to forbidding the 101 pattern, require the number of 0 symbols and 1 symbols to be the same. Using combinatorial methods, we determine the asymptotic information rate of these codes and show that the asymptotic rate loss due to the imposition of the balanced property is approximately 2%. Extensions to q-ary cells, for $q>2$ are also discussed. Next, we consider q-ary ICI-free write-once-memory (WOM) codes that support multiple writes of a WOM while mitigating ICI effects. These codes forbid the appearance of the $(q-1)\mathbf{0}(q-1)$ pattern in any codeword used in any writing step. Using properties of two-dimensional constrained codes and generalized WOMs, we characterize the maximum sum-rate of t-write ICI-free WOM codes or, equivalently, the t-write sum-capacity of an ICI-free WOM.

***Index Terms*—Constrained codes, flash memories, write-once memories**

I. Introduction

In RECENT years, non-volatile memories — in particular, flash memories — have attracted considerable attention due to their high data-transfer rate and low power consumption [1]. Flash memory cells consist of floating gate transistors, in which the amount of trapped charge determines the cell voltage, referred to as the *cell level*. A flash memory cell is written to, or "programmed," by applying a suitable voltage to the cell in order to inject the desired amount of charge to reach a certain cell level. Programming precision is an important factor governing the achievable capacity of flash memory storage. Another is inter-cell interference (ICI), caused by the parasitic capacitance between adjacent cells, as a result of which the voltage level of a so-called *victim* cell may be increased when a high voltage is applied to neighboring cells [2], [3].

As an example of ICI, consider a single-level cell (SLC) flash memory, meaning a memory whose cells supports only two levels. (The SLC designation is somewhat of a misnomer.) We denote the low level by the symbol 0 and the high level by the symbol 1. Now, if a group of three consecutive cells in a row are programmed with the 101 pattern, the level of the middle cell may be inadvertently increased due to the effect of ICI. During data recovery, the level of the victim cell may be erroneously interpreted as representing a programmed symbol 1. To combat this effect, the use of a constrained code that prevents the appearance of the ICI-prone symbol pattern 101 has been proposed. Similar ICI-mitigating constraints for multi-level flash memory cells, preventing the appearance of, say, the pattern $(q-1)0(q-1)$ in consecutive q-ary cells, have also been considered [4], [5]. We will refer to constrained codes that eliminate these ICI-inducing patterns as *ICI-free codes*.

In this paper, we investigate the application of ICI-free codes in two flash memory settings. First, we study information-theoretic limits on the efficiency of binary ICI-free codes that are also *balanced*, meaning that codewords have an equal number of 0's and 1's. These codes are of interest when a dynamic read threshold is used to reduce the number of errors caused by cell-level drift resulting from charge leakage. Specifically, we determine the asymptotic information rate of ICI-free balanced codes.

We then consider codes that allow for the efficient re-use of a binary ICI-free write-once-memory (WOM), that is, a binary WOM that does not support codewords containing the 101 pattern. The ICI-free WOM provides a model for a flash memory system that uses a multiple-write WOM code to extend the device lifetime, with the added constraint that none of the codewords contain the ICI-inducing 101 pattern. Our main result is a characterization of the t-write, sum-rate capacity of an ICI-free WOM, as well as an explicit numerical evaluation for $2 \leqslant t \leqslant 7$.

We now discuss these two applications in more detail.

Manuscript received May 15, 2013; revised October 1, 2013 and December 10, 2013. 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 Grant CCF-1116739, and the Center for Magnetic Recording Research at the University of California, San Diego.

M. Qin 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, U.S.A. (e-mail: {mqin,psiegel}@ucsd.edu).

E. Yaakobi is with the Department of Electrical Engineering, California Institute of Technology, Pasadena, CA 91125 USA (e-mail: eyaakobi@ucsd.edu). He is 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.

Digital Object Identifier 10.1109/JSAC.2014.140504.

A. ICI-free Balanced Codes for Dynamic Threshold Detection

Charge leakage, which results in a downward drift in cell voltage levels in a flash memory, can lead to errors in the data retrieval process when the read threshold(s) are fixed. In [6], a dynamic read threshold was proposed as a means to compensate for cell-level drift in SLC flash memory. The threshold adaptation is facilitated by the use of a balanced code. (See, for example, [7].) (Of necessity, the codeword length is even.)

Under the assumption that the cell-level drift is essentially uniform across all cells, the relative ranking of cell levels is largely preserved. Therefore, by adjusting the threshold value to the point where half of the cell levels are above the threshold and half fall below, the cells programmed to a 1 or a 0 may still be correctly identified with high probability. Since the asymptotic information rate of balanced binary codes is 1, the rate penalty associated with the use of a balanced code can be made negligibly small with proper code design.

The construction of efficient balanced codes has been extensively studied in [7]–[12], and extensions to non-binary and two-dimensional balanced codes have been considered in [13]–[16]. Codes that combine the balanced property with certain other constraints, such as runlength limitations, have also been addressed in, for example, [17].

We are specifically interested in binary codes that are both balanced and ICI-free, as defined above. These codes can be used with a dynamic threshold scheme, while also mitigating ICI effects.

B. Coding for an ICI-Free Write-Once-Memory (WOM)

One of the most conspicuous properties of flash memory cells is the asymmetry in the programming process. Cell levels can be easily increased by injecting additional charge into them [1]. In contrast, to decrease the level of even a single cell, the whole block of cells ($\sim 10^6$ cells) containing it has to be erased and then reprogrammed accordingly. These *block erasures* not only introduce significant latency into the writing process, but also degrade the floating gate cells, thereby shortening the usable lifetime of the device. Therefore, it is desirable to reduce the number of these block erasures in order to enhance the endurance of the flash memory and increase its lifetime storage capacity, which is the total amount of information that can be stored.

The original motivation for the use of rewriting codes came from storage media such as punch cards, optical disks, electronically programmable memories, and paper tapes, all of which consisted of "write-once" bits, or "wits," whose physical states could be changed only once. These technologies could be modeled as a write-once memory (WOM), that is, a binary storage medium consisting of cells supporting two states, designated as 0 and 1, such that during the recording process, a cell can remain in its existing state or, if it is in state 0, can be irreversibly changed to state 1.

Rivest and Shamir [18] showed that, through the use of properly designed codes, a WOM can store multiple generations of information much more efficiently than might have been expected.

Wolf et al. [19] studied binary WOMs from an information-theoretic perspective under various assumptions about state information available at the encoder and decoder. Heegard [20] determined the capacity region of achievable rates for binary WOMs with state information at the encoder, while also introducing several generalized models that allowed for noise, non-binary input and output alphabets, and different types of cell level transitions. The capacity region of non-binary WOMs with cell-state transitions described by an arbitrary directed acyclic graph was found by Fu and Han Vinck [5]. On the other hand, a WOM can be viewed as a special type of write-efficient memory (WEM) [21]. Within this framework, Fu and Yeung [22] derived the sum-capacity of deterministic WOMs described by a more general graph.

Several other works pertaining to WOMs and WOM codes have appeared, a number of them motivated by the relevance of the WOM model to flash memory devices [23]–[29].

In view of the distinct and complementary performance benefits offered by multiple-write WOM codes and ICI-mitigating codes, we investigate their combination in the framework of coding for an *input-constrained* WOM. Broadly speaking, an input-constrained WOM is a WOM that restricts the input words that it can store by forbidding the appearance of certain symbol patterns. In this framework, an *ICI-free WOM* is one that does not allow the pattern 1 0 1 to be written at any time.

C. Outline of the Paper

In Section II-A, we present the derivation of an exact expression for the generating function of the number of binary ICI-free balanced sequences, as well a combinatorial formula for its coefficients. From each of these, we deduce the asymptotic information rate of ICI-free balanced sequences. In Section II-B, we describe a heuristic, probabilistic argument that yields the same rate. Section III-A gives background on input-constrained WOMs and two-dimensional constraints. In Section III-B, we prove that the t-write sum-capacity of the input-constrained WOM is equal to the capacity of a corresponding two-dimensional constraint defined on an infinite t-row strip. Section III-C describes a construction of 2-write ICI-free WOM codes based upon covering subset partitions of bipartite graphs. Section IV concludes the paper.

II. ICI-free Balanced Codes

In this section, we study information-theoretic properties of ICI-free balanced codes. We derive the asymptotic information rate of ICI-free balanced codes over the binary alphabet using combinatorial properties of walks on the two-dimensional lattice $\mathbb{Z}^2$. We then present a heuristic, probabilistic argument that yields the same result.

We begin with some definitions.

Definition 1. *A length-$2n$ binary sequence $u \in \{0,1\}^{2n}$ is said to be* ***ICI-free balanced*** *if*

1) *it contains exactly n symbols that are 0 and n symbols that are 1;*
2) $u_{i-1}u_iu_{i+1} \neq 101$, *for all i such that $2 \leqslant i \leqslant 2n-1$.*

■

We refer to a set of ICI-free balanced sequences of the same length as an *ICI-free balanced code*.

Definition 2. *Let $\mathcal{C}_n$ be the set of all binary ICI-free balanced sequences of length $2n$. The* ***asymptotic information rate*** *of $\mathcal{C}_n$ is defined as*

$$C^{\langle 2\rangle} = \limsup_{n\to\infty} \frac{\log_2 |\mathcal{C}_n|}{2n}.$$

■

Remark 1. It can be deduced from Lemma 3 and Lemma 4 that $\lim_{n\to\infty} \frac{\log_2 |\mathcal{C}_n|}{2n}$ exists.

Referring to Definition 2, let $C(x)$ be the generating function of $|\mathcal{C}_n|$, that is,

$$C(x) = \sum_{n\geqslant 1} c_n x^n,$$

where $c_n = |\mathcal{C}_n|$ for $n \geqslant 1$. Our main contribution in this section is the following theorem, which gives a closed-form expression for $C(x)$ and determines precisely the asymptotic information rate $C^{\langle 2\rangle}$.

Theorem 1. *The generating function $C(x)$ is given by*

$$C(x) = \sqrt{\frac{1+x}{1-3x}} - 1,$$

and the asymptotic information rate of binary ICI-free balanced codes is

$$C^{\langle 2\rangle} = \frac{1}{2}\log_2 3.$$

■

A. Derivation of the Asymptotic Information Rate

This section is devoted to the proof of Theorem 1.

Let $\mathcal{S}_n$ denote the set of binary balanced sequences of length $2n$. We denote by $\mathcal{A}_n$ the set of all ICI-free balanced sequences of length $2n$ that start with a 1 and by $\mathcal{B}_n$ the set of all ICI-free balanced sequences of length $2n$ that start with a 0. Finally, the cardinalities of these subsets are denoted by $A_n = |\mathcal{A}_n|$ and $B_n = |\mathcal{B}_n|$, and the corresponding generating functions of the cardinalities are denoted by $A(x)$ and $B(x)$, respectively.

The derivation of $C(x)$ will make use of the close connection between ICI-free balanced sequences and paths in the integer lattice $\mathbb{Z}^2$. The following definition introduces several types of paths that will play a role in the analysis.

Definition 3. *A* ***path*** *of length n is an n-step walk on $\mathbb{Z}^2$, starting at $(0,0)$, such that every step is either an* ***upstep*** *obtained by adding $\mathsf{U} = (1,1)$ or a* ***downstep*** *obtained by adding $\mathsf{D} = (1,-1)$, respectively, to the current position. If the path is at (x_i, y_i) after i steps, the* ***height at step*** *i is defined to be y_i.*

A path of length n is called a ***symmetric path*** *if it ends in $(n, 0)$.*

A path is called a ***UDU-free path*** *if UDU is not a subsequence of the path.*

A symmetric path is called a ***Dyck path*** *if it never goes below the horizontal axis $y = 0$, i.e, the heights are non-negative after every step.* ■

Let $\mathcal{P}$ denote the set of all symmetric paths, including the empty path. Let $\mathcal{F}$ be the set of all UDU-free symmetric paths, including the empty path. Let $\mathcal{U} \subset \mathcal{F}$ be the set of non-empty paths that start with a U, let $\mathcal{D} \subset \mathcal{F}$ be the set of non-empty paths that start with a D, and let $\mathcal{H}$ be the set of UDU-free Dyck paths.

Now, let $\mathcal{P}_n \subset \mathcal{P}$ be the set of symmetric paths of length $2n$, for $n \geqslant 0$. Define $\mathcal{F}_n \subset \mathcal{F}$ to be the set of UDU-free paths of length $2n$. We use $F_n = |\mathcal{F}_n|$ to denote the cardinality of $\mathcal{F}_n$, and $F(x)$ to denote the corresponding generating function of F_n.

We define in an entirely analogous manner the sets $\mathcal{U}_n, \mathcal{D}_n$, and $\mathcal{H}_n$, their cardinalities U_n, D_n, and H_n, and their generating functions $U(x), D(x)$ and $H(x)$.

The evident connection between balanced sequences and symmetric paths, as well as between their subsets defined above, is stated formally in the following lemma.

Lemma 1. *There is a bijection between $\mathcal{S}_n$ and $\mathcal{P}_n$. The bijection maps $\mathcal{C}_n$ to $\mathcal{F}_n$ and, more specifically, $\mathcal{A}_n$ to $\mathcal{U}_n$ and $\mathcal{B}_n$ to $\mathcal{D}_n$.* ■

Proof: The bijective mapping between balanced sequences and symmetric paths is obtained by identifying the symbols 1 and 0 with steps U and D, respectively. It is clear that this mapping establishes the bijection between $\mathcal{C}_n$ and $\mathcal{F}_n$, as well as between $\mathcal{A}_n$ and $\mathcal{U}_n$ and between $\mathcal{B}_n$ and $\mathcal{D}_n$. □

The following lemma follows immediately from the definitions above and the properties of the bijection established in Lemma 1.

Lemma 2. *The generating functions $C(x)$ and $F(x)$ can be written as:*

$$C(x) = A(x) + B(x) = U(x) + D(x)$$

and

$$F(x) = U(x) + D(x) + 1.$$

■

For any $n \geqslant 1$, we will define a mapping from the subset $\mathcal{U}_n \subset \mathcal{F}_n$ of length-$2n$, UDU-free symmetric paths that begin with U to the subset $\mathcal{D}_n \subset \mathcal{F}_n$ of length-$2n$, UDU-free symmetric paths that begin with D. We will then show that this mapping is actually a bijection. In order to describe the mapping succinctly, we introduce the following terminology and notation.

Definition 4. *Let $u = [u_1, ..., u_{2n-1}, u_{2n}]$ be a path in $\mathcal{U}_n$. The k-left cyclic shift of u, denoted by $u^{(k)}$, is the path obtained by cyclically shifting u by k steps to the left. That is,*

$$u^{(k)} = [u_{k+1}, \ldots, u_{2n}, u_1, \ldots, u_{k-1}, u_k].$$

■

Note that all shifts of u are symmetric paths because the number of U steps and D steps remain equal. Define the mapping

$$\phi : \mathcal{U} \mapsto \mathcal{P}$$

as follows: Given a path $u \in \mathcal{U}$, let i be the index of the last symbol D in u such that the path falls from height 1 to height 0. Then $\phi(u) = u^{(i-1)}$, the $(i-1)$-left cyclic shift of u.

Proposition 1. *The restriction of the mapping ϕ to $\mathcal{U}_n$ is a bijection from $\mathcal{U}_n$ to $\mathcal{D}_n$, for all $n \geqslant 1$. Therefore, $|\mathcal{U}_n| = |\mathcal{D}_n|$, and $U(x) = D(x)$.*

Proof: We first show that $\phi(u) \in \mathcal{D}_n, \forall u \in \mathcal{U}_n$.

By construction, $\phi(u)$ is a symmetric path that starts with a D, so we need to show that $\phi(u)$ is UDU-free. Since $u \in \mathcal{U}_n$, this translates into showing that the UDU-free constraint is not violated when u_1 (which is a U) is cyclically shifted and concatenated with u_{2n}. Let i be the index of the last D that causes the path to fall from height 1 to height 0. If $u_{2n} = \mathsf{D}$, then $i = 2n$ and the first two steps of $\phi(u)$ are DU; otherwise, $u_{2n}u_1 = \mathsf{UU}$. In either case, we conclude that $\phi(u)$ is UDU-free. Hence, the image of ϕ lies in $\mathcal{D}_n$.

We now prove the injectivity of ϕ, that is, if $u, v \in \mathcal{U}_n$ and $u \neq v$, then $\phi(u) \neq \phi(v)$.

For any two distinct paths $u \in \mathcal{U}_n$ and $v \in \mathcal{U}_n$, let $\phi(u) = p$, where $p = [p_1, \ldots, p_{2n}]$ and $\phi(v) = q$, where $q = [q_1, \ldots, q_{2n}]$. Let i_u, resp. i_v, be the index of the last D such that u, resp. v, falls from height 1 to height 0. If $i_u = i_v$, then $p \neq q$ since, by definition of a left cyclic shift, distinct paths that are left-shifted by the same amount must yield distinct paths. On the other hand, suppose $i_u \neq i_v$ and, without loss of generality, assume $i_u < i_v$.

Then we claim that at least one of the following statements is true:

(a) There exists an index $j \in \{1, 2, \ldots, (2n - i_v + 1)\}$ such that $p_j \neq q_j$;

(b) $p_{(2n-i_v+2)} \neq q_{(2n-i_v+2)}$.

Each of these statements implies that $\phi(u) \neq \phi(v)$, as desired.

It suffices to prove that if (a) does not hold, then b) must. To see this, note that q is obtained by the $(i_v - 1)$-left cyclic shift of v, implying that $q_{(2n-i_v+1)} = v_{2n}$. Therefore, $q_{(2n-i_v+2)} = v_1 = U$. Meanwhile the height of q after $q_{(2n-i_v+1)}$ is -1. Now suppose (a) does not hold, i.e., $[p_1, \ldots, p_{(2n-i_v+1)}] = [q_1, \ldots, q_{(2n-i_v+1)}]$. Then, by construction, $p_{(2n-i_v+1)} = u_{(2n-i_v+i_u)}$ and the height of u after $u_{(2n-i_v+i_u)}$ is 0. Thus, $p_{(2n-i_v+2)} = u_{(2n-i_v+i_u+1)} = D$ since, otherwise, the height of u after $u_{(2n-i_v+i_u+1)}$ is 1 and u_{i_u} would not be the last D corresponding to a fall from height 1 to height 0 in u, contradicting the definition of i_u. Thus, $p_{(2n-i_v+2)} = \mathsf{U}$ and $q_{(2n-i_v+2)} = \mathsf{D}$, confirming that condition (b) holds.

This completes the proof that the mapping ϕ restricted to $\mathcal{U}_n$ is an injection into $\mathcal{D}_n$, and so $|\mathcal{U}_n| \leqslant |\mathcal{D}_n|$, for all $n \geqslant 1$.

In a similar manner, we define a mapping

$$\gamma : \mathcal{D}_n \to \mathcal{U}_n$$

as follows: Given a path $d \in \mathcal{D}_n$, let i be the label of the first U such that the path rises from height -1 to height 0. Then $\gamma(d) = d^{(i-1)}$, the $(i-1)$-left cyclic shift of d.

The proof that the restriction of the mapping γ to $\mathcal{D}_n$ is an injection into $\mathcal{U}_n$, for all $n \geqslant 1$, is similar to that of ϕ being an injection, so we omit the details. Consequently, $|\mathcal{D}_n| \leqslant |\mathcal{U}_n|$, for all $n \geqslant 1$.

In fact, one can see that γ and ϕ are inverse functions of one another, so they in fact define bijections between $\mathcal{U}_n$ and $\mathcal{D}_n$.

We can now conclude that $|\mathcal{U}_n| = |\mathcal{D}_n|$ for all $n \geqslant 1$, and, therefore, $U(x) = D(x)$. □

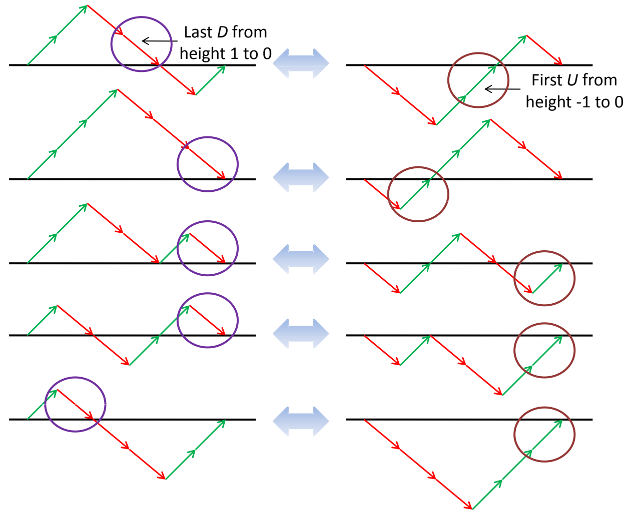

Fig. 1. Bijection between $\mathcal{U}_3$ and $\mathcal{D}_3$

Example 1. Fig. 1 illustrates the bijection between $\mathcal{U}_3$ and $\mathcal{D}_3$. There are 10 UDU-free paths of length 6 and the last D (or first U) that the path falls from height 1 to 0 (or rises from -1 to 0) is circled for each path.

We are now in a position to prove the main result.

Proof of Theorem 1: Any path in $\mathcal{U}$ can be written in one of the following two forms:

1) UDP, where P is empty or in $\mathcal{D}$; or,
2) UPDQ, where P is a non-empty path in $\mathcal{H}$, and Q is in $\mathcal{F}$.

(This is the so-called *first return decomposition* of a path in $\mathcal{U}$.)

This gives rise to the equation

$$U(x) = x(1 + D(x)) + x(H(x) - 1)F(x).$$

From Proposition 1, we have $D(x) = U(x)$. Together with Lemma 2, this implies that $F(x) = 2U(x) + 1$. Therefore,

$$U(x) = \frac{xH(x)}{1 + x - 2xH(x)}.$$

It was shown in [30] that

$$H(x) = \frac{1 + x - \sqrt{1 - 2x - 3x^2}}{2x}.$$

Therefore,

$$\begin{aligned} U(x) &= \frac{xH(x)}{1+x-2xH(x)} \\ &= \frac{1+x-\sqrt{1-2x-3x^2}}{2\sqrt{1-2x-3x^2}} \\ &= \frac{1}{2}\sqrt{\frac{1+x}{1-3x}} - \frac{1}{2}. \end{aligned}$$

We conclude that

$$C(x) = 2U(x) = \sqrt{\frac{1+x}{1-3x}} - 1.$$

If we treat $C(x)$ as a complex function, the Cauchy-Hadamard Theorem [31, p. 39], [32], [33] states that

$$\limsup_{n\to\infty} |c_n|^{1/n} = \frac{1}{\rho},$$

where ρ is the smallest modulus of a singularity of $C(x)$. From the expression for $C(x)$ shown above, we see that

$$\rho = \frac{1}{3}.$$

Recalling that $c_n = |\mathcal{C}_n|$, we conclude that

$$C^{\langle 2\rangle} = \lim_{n\to\infty} \frac{\log|\mathcal{C}_n|}{2n} = \frac{1}{2}\log 3.$$

This completes the proof. □

Note that the proof of Theorem 1 above does not involve explicit expressions for $|U_n|$ and $|D_n|$. However, Deutsch (A005773, [34]) and Callan [35] have shown that

$$U_n = D_n = \sum_{j=0}^{n-1} \binom{j}{\lfloor \frac{j}{2} \rfloor}\binom{n-1}{j} \tag{1}$$

and from this formula, one can obtain an alternative derivation of the asymptotic information rate $C^{\langle 2\rangle}$, as we now show.

From Callan's formula, we have

$$\begin{aligned} C^{\langle 2\rangle} &= \lim_{n\to\infty} \frac{\log_2(U_n+D_n)}{2n} \\ &= \lim_{n\to\infty} \frac{1}{2n}\left(1+\log_2 \sum_{j=0}^{n-1} \binom{j}{\lfloor \frac{j}{2} \rfloor}\binom{n-1}{j}\right) \\ &= \lim_{n\to\infty} \frac{1}{2n}\left(\log_2 \sum_{j=0}^{n-1} \binom{j}{\lfloor \frac{j}{2} \rfloor}\binom{n-1}{j}\right). \end{aligned}$$

The following lemma shows that replacing the summation with a maximization does not affect the value of the limit in the formula above.

Lemma 3.

$$C^{\langle 2\rangle} = \lim_{n\to\infty} \frac{1}{2n} \log_2 \max_{0\leqslant j\leqslant n-1} \binom{j}{\lfloor \frac{j}{2} \rfloor}\binom{n-1}{j}.$$

■

Proof: Since all terms in the summation above are positive, we have

$$\begin{aligned} &\lim_{n\to\infty} \frac{1}{2n}\left(\log_2 \sum_{j=0}^{n-1} \binom{j}{\lfloor \frac{j}{2} \rfloor}\binom{n-1}{j}\right) \\ &\geqslant \lim_{n\to\infty} \frac{1}{2n} \log_2 \max_{0\leqslant j\leqslant n-1} \binom{j}{\lfloor \frac{j}{2} \rfloor}\binom{n-1}{j}. \end{aligned}$$

On the other hand, by replacing the sum of terms with their maximum, we have

$$\begin{aligned} &\lim_{n\to\infty} \frac{1}{2n}\left(\log_2 \sum_{j=0}^{n-1} \binom{j}{\lfloor \frac{j}{2} \rfloor}\binom{n-1}{j}\right) \\ &\leqslant \lim_{n\to\infty} \frac{1}{2n} \log_2 \left(n \max_{0\leqslant j\leqslant n-1} \binom{j}{\lfloor \frac{j}{2} \rfloor}\binom{n-1}{j}\right) \\ &\leqslant \lim_{n\to\infty} \frac{\log_2 n}{2n} + \lim_{n\to\infty} \frac{1}{2n} \log_2 \max_{0\leqslant j\leqslant n-1} \binom{j}{\lfloor \frac{j}{2} \rfloor}\binom{n-1}{j} \\ &= \lim_{n\to\infty} \frac{1}{2n} \log_2 \max_{0\leqslant j\leqslant n-1} \binom{j}{\lfloor \frac{j}{2} \rfloor}\binom{n-1}{j} \end{aligned}$$

Therefore,

$$C^{\langle 2\rangle} = \lim_{n\to\infty} \frac{1}{2n} \log_2 \max_{0\leqslant j\leqslant n-1} \binom{j}{\lfloor \frac{j}{2} \rfloor}\binom{n-1}{j}.$$

□

The next lemma gives the limit of the normalized value of argument j that achieves the maximum in Lemma 3.

Lemma 4.

$$\lim_{n\to\infty} \frac{1}{n} \arg \max_{0\leqslant j\leqslant n-1} \binom{j}{\lfloor \frac{j}{2} \rfloor}\binom{n-1}{j} = \frac{2}{3}.$$

■

Proof: Note that

$$\binom{j}{\lfloor \frac{j}{2} \rfloor}\binom{n-1}{j} = \binom{n-1}{\lfloor \frac{j}{2} \rfloor, \lceil \frac{j}{2} \rceil, n-1-j}.$$

If $n-1$ is a multiple of 3, then this quantity is maximized when $\lfloor \frac{j}{2} \rfloor = \lceil \frac{j}{2} \rceil = n-1-j = \frac{n-1}{3}$, i.e., when $j = \frac{2(n-1)}{3}$. Similar reasoning shows that the maximizing values of j for all $n \geqslant 1$ satisfy $\lim_{n\to\infty} \frac{j}{n} = \frac{2}{3}$. □

From Lemma 3 and Lemma 4, we conclude that

$$C^{\langle 2\rangle} = \frac{1}{2}\log_2 3.$$

Remark 2. The capacity of the constraint that forbids 101 is approximately 0.8114. Thus, the value of $C^{\langle 2\rangle}$ shows that there is a 2% rate loss due to the additional balanced constraint. Balanced codes are special types of constant weight codes [36], where the weight is half of the code length $2n$. In general, constant weight codes can be used to adapt to the voltage drift during read cycles as well. It might be better to use constant weight codes with weight less than n if rate optimization is the only figure of merit; however, balanced codes have the advantage of easy encoding and decoding, while sacrificing only a small portion (2%) of the rate.

B. Heuristic Probabilistic Derivation

We now provide a heuristic probabilistic argument that yields the result derived in the previous section, namely $C^{\langle 2\rangle} = \frac{1}{2}\log_2 3$.

Let $\mathcal{S}$ be the set of all balanced sequences of length $2n$. It is evident that $|\mathcal{S}| = \binom{2n}{n}$. Now, let Z be a randomly chosen sequence in $\mathcal{S}$, and let Z_i be the i-th entry in Z, for $1 \leqslant i \leqslant 2n$. Clearly $\mathbb{P}(Z_i = 1) = \mathbb{P}(Z_i = 0) = \frac{1}{2}$ and, if n is sufficiently large, $\mathbb{P}(Z_{i-1} = Z_{i+1} = 1|Z_i = 0)$ is approximately $\frac{1}{4}$. Define a sequence of events $\mathtt{E}_i \overset{\text{def}}{=} \{(Z_{i-1}Z_iZ_{i+1}) \neq 101|Z_i = 0\}$, for $2 \leqslant i \leqslant 2n-1$. Then $\mathbb{P}(\mathtt{E}_i) \approx \frac{3}{4}$. The number of 0's in Z is n, so if we treat the events $\mathtt{E}_i$ as independent (though, in reality, they are not), then the probability that Z satisfies the ICI-free balanced constraint is approximately $\left(\frac{3}{4}\right)^n$. Thus, the number of ICI-free balanced sequences in $\mathcal{S}$ is approximately $\binom{2n}{n}\left(\frac{3}{4}\right)^n$. That is,

$$\frac{\log_2|\mathcal{C}_n|}{2n} \approx \frac{\log_2\left(\binom{2n}{n}\left(\frac{3}{4}\right)^n\right)}{2n} \xrightarrow{n\to\infty} \frac{1}{2}\log_2 3.$$

Although the independence assumption is not valid, this line of reasoning yieds the correct answer because the dependency of $\mathtt{E}_i$ and $\mathtt{E}_j$ decreases as $|i-j|$ increases.

Now, recall that, for q-level flash cells, ICI arises when three consecutive cells are programmed to the levels (c_1, c_2, c_3) such that c_1 and c_3 are much larger than c_2. It is expected that the most severe ICI will occur when three consecutive cells are programmed to the levels $((q-1), 0, (q-1))$. We now extend the definition of ICI-free balanced sequences to the q-ary case to avoid the most severe ICI pattern.

Definition 5. *A q-ary sequence $u \in \{0, 1, \ldots, q-1\}^{qn}$ is said to satisfy the q-ary ICI-free balanced constraint if*

1) $\forall j$ *such that* $0 \leqslant j \leqslant q-1$, *the number of* j*'s in* u *is* n*;*
2) $(u_{i-1}u_iu_{i+1}) \neq (q-1)0(q-1)$, $\forall i$ *such that* $2 \leqslant i \leqslant qn-1$.

■

Let $\mathcal{C}_n^{\langle q\rangle}$ be the set of all q-ary ICI-free-balanced sequences of length qn. The asymptotic information rate of $\mathcal{C}_n^{\langle q\rangle}$ is defined as

$$C^{\langle q\rangle} = \lim_{n\to\infty}\frac{\log_2|\mathcal{C}_n^{\langle q\rangle}|}{nq}.$$

By direct analogy to the heuristic argument used in the binary case, $q = 2$, one might conjecture that the asymptotic information rate is

$$\mathcal{C}_{\text{conj}}^{\langle q\rangle} = \log_2 q + \frac{1}{q}\log_2\left(\frac{q^2-1}{q^2}\right).$$

However, based on [37], $\mathcal{C}^{\langle 3\rangle} \approx 1.5258$ while the heuristic argument overestimates $\mathcal{C}^{\langle 3\rangle}$, yielding $\mathcal{C}_{\text{conj}}^{\langle 3\rangle} \approx 1.5283$.

III. ICI-free WOM Codes

In this section, we study the WOM model with certain input constraints. We first present the definition of an input-constrained WOM and then provide a derivation of the t-write sum-capacity. Finally, we give code constructions based on coverings of bipartite graphs.

A. Definitions

Suppose the number of cells is n and the number of rewriting cycles is t. The cell levels of a generalized q-level WOM after the i-th write are denoted by $y_{i,1}^n \in [0 : q-1]^n$, for $i \in [1 : t]$, where $[k_1 : k_2] \overset{\text{def}}{=} \{k \in \mathbb{Z}|k_1 \leqslant k \leqslant k_2\}$ and $y_{i,k_1}^{k_2} \overset{\text{def}}{=} (y_{i,k_1}, y_{i,k_1+1}, \ldots, y_{i,k_2})$. We will use $[n]$ as a shorthand for $[1 : n]$ when no confusion could occur. Furthermore, for vectors $x_1^n = (x_1, ..., x_n)$ and $y_1^n = (y_1, ..., y_n)$, we write $x_1^n \succeq y_1^n$ if and only if $\forall i \in [n], x_i \geqslant y_i$. We can describe the discrete memoryless generalized WOM by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of vertices and $\mathcal{E} \subseteq \mathcal{V}\times\mathcal{V}$ is the set of edges. For a q-level WOM, $\mathcal{V} = \{0, 1, \ldots, q-1\}$. If the level can be changed directly from s_1 to s_2, where $s_1, s_2 \in \mathcal{V}$, then there exists an edge from s_1 to s_2 and we denote it by $(s_1, s_2) \in \mathcal{E}$. For a state sequence $(s_0, s_1, \ldots, s_t)$, if $\forall i \in [0 : t-1], (s_i, s_{i+1}) \in \mathcal{E}$, then we say the path $(s_0, s_1, \ldots, s_t)$ exists and we denote it by $s_0 \to s_1 \to \cdots \to s_t$. For two vectors $(x_{1,1}^n, x_{2,1}^n) \in [0 : q-1]^n \times [0 : q-1]^n$, we write $x_{1,1}^n \Rightarrow x_{2,1}^n$ if and only if $\forall i \in [n]$, $(x_{1,i}, x_{2,i}) \in \mathcal{E}$. The transition matrix $A = (a_{i,j}) \in \{0, 1\}^{q\times q}$ is defined as follows. For $i, j \in [0 : q-1]$, $a_{ij} = 1$ if $(i, j) \in \mathcal{E}$; otherwise, $a_{ij} = 0$.

Definition 6. *Let $y_{i,1}^n$ denote the cell-state vector after the i-th write, for $i \in [t]$. An $[n, t; 2^{nR_1}, \ldots, 2^{nR_t}]$ q-ary WOM code $\mathcal{C}_{q,\mathcal{G}}$ described by the graph $\mathcal{G}$ is a coding scheme consisting of n cells and t pairs of encoders and decoders $(\mathcal{E}_i, \mathcal{D}_i)$, where $\forall i \in [t]$, the encoder is a mapping*

$$\mathcal{E}_i : [1 : 2^{nR_i}] \times \textit{Im}\{\mathcal{E}_{i-1}\} \to [0 : q-1]^n,$$

such that $\forall(m, y_{i-1,1}^n) \in [1 : 2^{nR_i}] \times \textit{Im}\{\mathcal{E}_{i-1}\}$,

$$y_{i-1,1}^n \Rightarrow y_{i,1}^n = \mathcal{E}_i(m, y_{i-1,1}^n),$$

where, by abuse of notation, we use $\textit{Im}\{\mathcal{E}_0\}$ to represent the initial cell-state vector $\{(0, ..., 0)\}$. The decoder is a mapping

$$\mathcal{D}_i : \textit{Im}\{\mathcal{E}_i\} \to [1 : 2^{nR_i}],$$

such that $\forall m \in [1 : 2^{nR_i}]$,

$$\mathcal{D}_i(\mathcal{E}_i(m, y_{i-1,1}^n)) = m.$$

Let w be a q-ary sequence. An **input-constrained WOM code** $\mathcal{C}_{S_w,q,\mathcal{G}}$ **avoiding** *w is a q-ary WOM code such that w is not a subsequence of any codeword in $\mathcal{C}_{S_w,q,\mathcal{G}}$.* ■

Definition 7. *A rate tuple $(R_1, \ldots, R_t)$ is said to be* achievable *if there exists a sequence of $[n, t; 2^{nR_1}, \ldots, 2^{nR_t}]$ WOM codes. The capacity region is defined as the closure of the set of all achievable rate tuples. The sum-capacity is defined as the supremum of achievable sum-rates $\sum_{i=1}^t R_i$.* ■

To mitigate ICI, the sequence is chosen such that $w \overset{\text{def}}{=} 101$ and the corresponding input-constrained WOM code is called the **binary ICI-free WOM code**. In this section, we are interested in the sum-capacity of the ICI-free binary WOM, i.e., the supremum of achievable sum-rates of $\mathcal{C}_{S_{101},2,\mathcal{G}}$, where $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $\mathcal{V} = \{0, 1\}$, $\mathcal{E} = \{(0, 0), (0, 1), (1, 1)\}$, and, therefore, $A = \left[\begin{smallmatrix}1 & 1\\ 0 & 1\end{smallmatrix}\right]$. We will first provide general results for an arbitrary constraint S and arbitrary number of levels q, and then apply them in the binary ICI-free WOM setting.

There is a connection between two-dimensional constrained codes and codes for input-constrained WOMs. Specifically, every t-write WOM code of length n can be expressed as a set of $t \times n$ arrays where the i-th row, $i \in [t]$, corresponds to the memory state after the i-th write. We will exploit this fact in our derivation of the t-write sum-capacity. Let us first recall the definition of the capacity of a two-dimensional constraint.

Definition 8. *Given a two-dimensional constraint S^{2D}, its capacity is defined to be*

$$C_{2D}(S^{2D}) = \lim_{m,n\to\infty} \frac{\log_2 N_S(m,n)}{mn},$$

*where $N_S(m,n)$ is the number of $m \times n$ arrays that satisfy the constraint S^{2D}. The t-**write column capacity** is defined to be*

$$C(t, S^{2D}) = \lim_{n\to\infty} \frac{\log_2 N_S(t,n)}{n}.$$

■

Remark 3. The exact capacity of most non-trivial two-dimensional constraints is not known. However, the t-write column capacity can be calculated numerically with the aid of the characteristic function of the adjacency matrix associated with the constraint S^{2D} when we fix one dimension of the 2-D array to be of size t [38].

There are a number of two-dimensional constraints that have been extensively studied, e.g., 2-D (d,k)-runlength-limited (RLL) [39], no isolated bits [40], [41], and checkerboard [42], [43]. For the input-constrained WOM codes $\mathcal{C}_{S_w,q,\mathcal{G}}$, where $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we define a constraint on two-dimensional arrays, denoted by S_w^{2D}, that is used to calculate the sum-capacity. Specifically, in a q-ary two dimensional array $B = \{b_{i,j}\}_{m\times n}$, we must have

$$(b_{i,j}, b_{i+1,j}) \in \mathcal{E}, \forall i \in [m-1], j \in [n]$$

and the pattern w must not be a subsequence in any row of B. We denote the two-dimensional constraint corresponding to the ICI-free WOM by S_{101}^{2D}.

B. Sum-Capacity

The following theorem characterizes the t-write sum-capacity of an input-constrained WOM in terms of the t-write column capacity of an associated two-dimensional constraint.

Theorem 2. *The t-write sum-capacity of the q-ary input-constrained WOM that does not allow w is $C(t, S_w^{2D})$.*

In particular, for the binary ICI-free WOM, the t-write sum-capacity is $C(t, S_{101}^{2D})$. ■

In order to prove Theorem 2, we will make use of the characterization of the sum-capacity of a WOM described by a general directed graph $\mathcal{G}$, as presented in [22, Prop. 2]. The derivation in [22] uses results about the sum-capacity of write-efficient memories [21]. Here we will present an alternative derivation based upon the Markov-chain WOM model from [20].

Lemma 5. *The sum-capacity $C_{\text{sum}}(t,\mathcal{G})$ of the generalized discrete memoryless q-level WOM described by graph $\mathcal{G}$ is the base-2 logarithm of the number of length-t paths that start from state 0, i.e.,*

$$C_{\text{sum}}(t,\mathcal{G}) = \log_2(\boldsymbol{\delta}_{0,q}^T \cdot A_{\mathcal{G}}^t \cdot \mathbf{1}_q),$$

where $A_{\mathcal{G}}$ is the transition matrix of the graph $\mathcal{G}$, and $\boldsymbol{\delta}_{i,q} = (0,\ldots,0,1,0,\ldots,0)^T, 0 \leqslant i \leqslant q-1$, is a column vector of length q such that the $(i+1)$-st entry is 1 and the remaining entries are 0's. ■

Proof: We adopt the notation used in [20]. According to Theorem 3 in [20], the sum-capacity equals

$$\begin{aligned} C_{\text{sum}} &= \sum_{i=1}^{t} R_i \\ &= \sum_{i=1}^{t} H(Y_i|Y_{i-1}) \\ &= H(Y_1, Y_2, \ldots, Y_t), \end{aligned}$$

where $Y_i, i \in [t]$ is a random variable representing the state of the WOM after the i-th write. The last equality follows from the fact that $Y_1 \to Y_2 \to \cdots \to Y_t$ form a Markov chain.

First we prove $C_{\text{sum}} \leqslant \log_2(\boldsymbol{\delta}_{0,q}^T \cdot A^t \cdot \mathbf{1}_q)$. The random vector $Y_1^t = (Y_1,\ldots,Y_t)$ corresponds to a path in $\mathcal{G}$; thus, the cardinality of Y_1^t is upper bounded by the number of length-t paths from state 0. Therefore, $H(Y_1^t) \leqslant \log_2(\boldsymbol{\delta}_{0,q}^T \cdot A^t \cdot \mathbf{1}_q)$.

Next we prove achievability. Let $p(y_1^t)$ be the joint probability mass function of Y_1^t. Let $p(y_1^t)$ be factored as $p_1(y_1), p_2(y_2|y_1), \ldots, p_t(y_t|y_{t-1})$, where $p_i(y_i|y_{i-1})$ is the conditional transition probability for the i-th write. We show that by appropriately choosing $p_1(y_1), p_2(y_2|y_1), \ldots, p_t(y_t|y_{t-1})$, $(Y_1,\ldots,Y_t)$ is uniformly distributed on its support. Let

$$p_1(y_1 = j) = \begin{cases} \frac{\boldsymbol{\delta}_{j,q}^T \cdot A^{t-1} \cdot \mathbf{1}_q}{\boldsymbol{\delta}_{0,q}^T \cdot A^t \cdot \mathbf{1}_q}, & \text{if } (0,j) \in \mathcal{E}; \\ 0, & \text{otherwise}, \end{cases}$$

for all $j \in [0:q-1]$. For $2 \leqslant i \leqslant t$, let

$$p_i(y_i = j|y_{i-1} = \ell) = \begin{cases} \frac{\boldsymbol{\delta}_{j,q}^T \cdot A^{t-i} \cdot \mathbf{1}_q}{\boldsymbol{\delta}_{\ell,q}^T \cdot A^{t-i+1} \cdot \mathbf{1}_q}, & \text{if } (\ell,j) \in \mathcal{E}, \{A^{i-1}\}_{0,\ell} = 1; \\ 0, & \text{otherwise}, \end{cases}$$

for all $j, \ell \in [0:q-1]$.

Then, for a state sequence $(s_1, s_2, \ldots, s_t)$, we have

$$\begin{aligned} &P(Y_1 = s_1, Y_2 = s_2, \ldots, Y_t = s_t) \\ =& p_1(y_1 = s_1) p_2(y_2 = s_2|y_1 = s_1) \\ &\quad \cdot p_3(y_3 = s_3|y_2 = s_2) \cdots p_t(y_t = s_t|y_{t-1} = s_{t-1}) \\ =& \frac{\boldsymbol{\delta}_{s_1,q}^T \cdot A^{t-1} \cdot \mathbf{1}_q}{\boldsymbol{\delta}_{0,q}^T \cdot A^t \cdot \mathbf{1}_q} \cdot \frac{\boldsymbol{\delta}_{s_2,q}^T \cdot A^{t-2} \cdot \mathbf{1}_q}{\boldsymbol{\delta}_{s_1,q}^T \cdot A^{t-1} \cdot \mathbf{1}_q} \\ &\quad \cdot \frac{\boldsymbol{\delta}_{s_3,q}^T \cdot A^{t-3} \cdot \mathbf{1}_q}{\boldsymbol{\delta}_{s_2,q}^T \cdot A^{t-2} \cdot \mathbf{1}_q} \cdots \frac{\boldsymbol{\delta}_{s_t,q}^T \cdot A^{0} \cdot \mathbf{1}_q}{\boldsymbol{\delta}_{s_{t-1},q}^T \cdot A \cdot \mathbf{1}_q} \\ =& \frac{1}{\boldsymbol{\delta}_{0,q}^T \cdot A^t \cdot \mathbf{1}_q} \end{aligned}$$

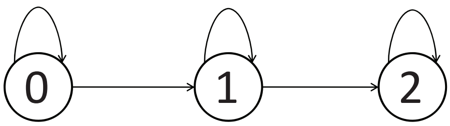

Fig. 2. Generalized WOM with state transition diagram

if the path $s_1 \to \cdots \to s_t$ exists; otherwise,

$$P(Y_1 = s_1, Y_2 = s_2, \ldots, Y_t = s_t) = 0.$$

This proves that $(Y_1, \ldots, Y_t)$ is uniformly distributed on its support set. Since the cardinality of the support set is $\boldsymbol{\delta}_{0,q}^T \cdot A^t \cdot \mathbf{1}_q$, then $H(Y_1, \ldots, Y_t) = \log_2(\boldsymbol{\delta}_{0,q}^T \cdot A^t \cdot \mathbf{1}_q)$. □

Example 2. For the state transition diagram in Fig. 2, suppose the number of writes is $t = 4$. We set the conditional probabilities as follows:

$$p_1(y_1 = 0) = \frac{7}{11}, p_1(y_1 = 1) = \frac{4}{11}.$$

$$p_2(y_2 = 0|y_1 = 0) = \frac{4}{7}, p_2(y_2 = 1|y_1 = 0) = \frac{3}{7}.$$

$$p_2(y_2 = 1|y_1 = 1) = \frac{3}{4}, p_2(y_2 = 2|y_1 = 1) = \frac{1}{4}.$$

$$p_3(y_3 = 0|y_2 = 0) = \frac{1}{2}, p_3(y_3 = 1|y_2 = 0) = \frac{1}{2}.$$

$$p_3(y_3 = 1|y_2 = 1) = \frac{2}{3}, p_3(y_3 = 2|y_2 = 1) = \frac{1}{3}.$$

$$p_3(y_3 = 2|y_2 = 2) = 1.$$

$$p_3(y_4 = 0|y_3 = 0) = \frac{1}{2}, p_3(y_4 = 1|y_3 = 0) = \frac{1}{2}.$$

$$p_3(y_4 = 1|y_3 = 1) = \frac{1}{2}, p_3(y_4 = 2|y_3 = 1) = \frac{1}{2}.$$

$$p_3(y_4 = 2|y_3 = 2) = 1.$$

Then, each possible state sequence has probability $\frac{1}{11}$, which means the 4-write sum-capacity is $\log_2 11$.

Now we are ready to prove Theorem 2. We give the proof for the case of a binary ICI-free WOM, i.e., the transition diagram $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is defined by $\mathcal{V} = \{0, 1\}$ and $\mathcal{E} = \{(0,0), (0,1), (1,1)\}$, and the input constraint is given by $w = 101$. The generalization to an arbitrary input-constrained WOM follows a similar line of reasoning.

Proof of Theorem 2:

Proof of achievability:

Let n and m be two positive integers such that n is a multiple of $(m+2)$, i.e., $n = \ell(m+2)$. The memory consists of n cells, denoted by $(c_1, \ldots, c_n)$, which are partitioned into ℓ blocks, each with $(m+2)$ cells. When the messages are written into the memory, within each block, the last 2 cells are kept at level 0, i.e., $c_{i(m+2)} = c_{i(m+2)-1} = 0, \forall i \in [\ell]$. In this way, it can be guaranteed that no 3 consecutive cells at the boundaries of adjacent blocks are 101. Each block of m cells constitutes the same t-write WOM code that avoids 101. To be more precise, we first introduce the following definitions.

Let $b : \mathbb{Z}_+ \mapsto \{0,1\}^m$ be the function that maps a non-negative integer $M \in [0, 2^m - 1]$ to its binary representation of length m, and let $b^{-1}(\boldsymbol{x})$ be the inverse function for $\boldsymbol{x} \in \{0,1\}^m$.

The following construction yields a sequence of binary ICI-free WOM codes with the claimed sum-rate efficiency.

Construction 1 *Let n, m and ℓ be positive integers such that $n = \ell(m+2)$. Suppose the cell-state vector is $c_{i,1}^n \in \{0,1\}^n$ after the i-th write, for $i \in [t]$. Let $y_{i,1}^\ell \in [0 : 2^m - 1]^\ell$ satisfy $y_{i,j} = b^{-1}(c_{i,(j-1)(m+2)+1}^{(j-1)(m+2)+m})$, for $j \in [\ell]$.*

A directed graph $\mathcal{G}_m = (\mathcal{V}, \mathcal{E})$ with 2^m vertices/states is defined as follows. The vertex set is $\mathcal{V} = [0 : 2^m - 1]$, and $\forall i, j \in \mathcal{V}$, $(i,j) \in \mathcal{E} \Leftrightarrow b(j) \succeq b(i)$ and 101 is not a subsequence of $b(j)$. Let A_m be the transition matrix for $\mathcal{G}_m$.

Let $\mathcal{C}_W$ be an $[\ell, t; 2^{\ell R_1}, 2^{\ell R_2}, \ldots, 2^{\ell R_t}]$ t-write 2^m-ary WOM code of length ℓ described by $\mathcal{G}_m$. Let $\mathcal{E}_i(m_i, y_{i-1,1}^\ell)$ be its encoder on the i-th write, for $i \in [t]$. An $[n, t; 2^{\ell R_1}, \ldots, 2^{\ell R_t}]$ binary ICI-free WOM code $\mathcal{C}_{ICI}$ of length n is constructed as follows. On the i-th write, the encoder uses the following rules:

1) *in each block of size $(m+2)$, the last two cells are kept as 0, i.e., $c_{i,(m+2)j-1}^{(m+2)j} = c_{i-1,(m+2)j-1}^{(m+2)j} = 0, \forall j \in [\ell]$.*
2) *write the message $M_i \in [1 : 2^{\ell R_i}]$ to the remaining $m\ell$ cells. Specifically, let $y_{i,1}^\ell = \mathcal{E}_i(M_i, y_{i-1,1}^\ell)$, and write the remaining $m\ell$ cells such that $c_{i,(j-1)(m+2)+1}^{(j-1)(m+2)+m} = b(y_{i,j}), \forall j \in [\ell]$.*

The decoder can be designed accordingly and we omit the details.

If $\mathcal{C}_W$ is sum-rate optimal, then the sum-rate of $\mathcal{C}_{\text{ICI}}$ is

$$\begin{aligned} R_{m,\text{ICI}}(t) &= \frac{\ell \sum_{i=1}^t R_i}{\ell(m+2)} \\ &= \frac{C_{\text{sum}}(t, \mathcal{G}_m)}{m+2} \\ &= \frac{\log_2(\boldsymbol{\delta}_{0,q}^T \cdot A_m^t \cdot \mathbf{1}_q)}{m+2} \\ &= \frac{\log_2(\boldsymbol{\delta}_{0,q}^T \cdot A_m^t \cdot \mathbf{1}_q)}{m} \cdot \frac{m}{m+2}. \end{aligned}$$

Note that $\boldsymbol{\delta}_{0,q}^T A_m^t \mathbf{1}_q$ counts the number of binary arrays $B = \{b_{i,j}\}_{t \times m}$ such that $b_{i+1,j} \geqslant b_{i,j}, \forall i \in [t-1], j \in [m]$, and the pattern (101) is not a subsequence of any row in B.

Letting m go to infinity, we see that there exists a sequence of t-write ICI-free WOM codes with rates

$$\begin{aligned} R_{\text{ICI}}(t) &= \lim_{m \to \infty} R_{m,\text{ICI}}(t) \\ &\geqslant \lim_{m \to \infty} \frac{\log_2(\boldsymbol{\delta}_{0,q}^T \cdot A_m^t \cdot \mathbf{1}_q)}{m} \cdot \frac{m}{m+2} \\ &= C(t, S_{101}^{2D}). \end{aligned}$$

The existence of the limit can be shown by the sub-additive property [44] of binary arrays B.

Proof of converse:

The converse can be easily proved by noting that if there exists a genie that, at decoding step $j \in [t]$, can provide all of the sequences written into the WOM from the first to the $(j-1)$-st write, then the sum-capacity equals the t-write column

TABLE I
SUM-CAPACITY OF ICI-FREE WOM

t	2	3	4	5	6	7
$C_{\text{ICI}}(t)$	1.264	1.584	1.831	2.035	2.207	2.356
$\log(t+1)$	1.585	2	2.322	2.585	2.807	3

capacity, which is $C(t, S_{101}^{2D})$. However, this genie does not exist, so for any t-write ICI-free WOM code with sum-rate $R_{\text{ICI}}(t)$, it follows that

$$R_{\text{ICI}}(t) \leqslant C(t, S_{101}^{2D}). \qquad \square$$

Example 3. Table I shows the t-write sum-capacity $C_{\text{ICI}}(t)$ of the ICI-free WOM, calculated using the techniques in [45]. Also shown is the t-write capacity of an unconstrained WOM, which is $\log(t+1)$, for $2 \leqslant t \leqslant 7$. An interesting observation is that $\frac{C_{\text{ICI}}(t)}{\log(t+1)}$ is close to 0.79 in general.

C. Code Constructions

We now proceed to the construction of some input-constrained WOM codes. In particular, we construct binary ICI-free WOM codes for $t = 2$ writes. The construction technique generalizes to q-ary alphabets with $q > 2$, $t > 2$ writes, and more general input constraints.

Let $\mathcal{C}$ be the set of binary vectors of length n that avoid 101. Let $\mathcal{C} = \mathcal{L} \cup \mathcal{R}$ be a partition of $\mathcal{C}$. For a pair of vectors $(\ell, r) \in \mathcal{L} \times \mathcal{R}$, we say r covers ℓ if $r \succeq \ell$. A bipartite graph $\mathcal{B} = (\mathcal{L} \cup \mathcal{R}, \mathcal{E})$ is defined where $\mathcal{L}$ and $\mathcal{R}$ are the sets of left and right nodes, respectively. An edge (ℓ, r) connecting $\ell \in \mathcal{L}$ and $r \in \mathcal{R}$ exists if r covers ℓ and we denote such an edge by $(\ell, r) \in \mathcal{E}$. For $\hat{\mathcal{R}} \subseteq \mathcal{R}$, the **covering** of $\hat{\mathcal{R}}$, denoted by $\text{CV}(\hat{\mathcal{R}})$, is defined as $\{\ell \in \mathcal{L} : \exists r \in \hat{\mathcal{R}}, r \text{ covers } \ell\}$ and the **covering cardinality** of $\hat{\mathcal{R}}$ is defined as $|\text{CV}(\hat{\mathcal{R}})|$. We say that $\widetilde{\mathcal{R}} \subseteq \mathcal{R}$ is a **covering subset** if $\text{CV}(\widetilde{\mathcal{R}}) = \mathcal{L}$. A partition $\mathcal{R} = \cup_{i=1}^{k} \mathcal{R}_i$ is called a **covering subset partition** of $\mathcal{R}$ if $\mathcal{R}_i$ is a covering subset for all $i \in [k]$.

Lemma 6. *Let $\mathcal{R} = \cup_{i=1}^{k} \mathcal{R}_i$ be a covering subset partition. Then there exists a 2-write ICI-free WOM code of length n with rate pair $(\frac{\log|\mathcal{L}|}{n}, \frac{\log k}{n})$.* ■

For a bipartite graph, finding the maximum number of subsets, k, in a covering subset partition is an interesting problem in its own right. In [46], a greedy algorithm is proposed to find covering subset partitions. We extend the greedy algorithm in [46] by adding another parameter g that controls the level of greediness. The algorithm in [46] would coincide with the following algorithm for $g = 1$.

Algorithm 1. FINDING COVERING SUBSETS OF A BIPARTITE GRAPH

Input:
a bipartite graph $\mathcal{B} = (\mathcal{L} \cup \mathcal{R}, \mathcal{E})$, where $\mathcal{L} = \{\ell_1, \ldots, \ell_n\}$ $\mathcal{R} = \{r_1, \ldots, r_m\}$;
a positive integer g that measure the extent of greediness in searching for a covering subset;
Output:
a partition of $\mathcal{R} = \cup_{i=1}^{k} \mathcal{R}_i$ such that $\mathcal{R}_i$ is a covering subset for all $i \in [k]$.

1: $k \leftarrow 0$;
2: $\mathcal{R}_{\text{unused}} \leftarrow \mathcal{R}$;
3: Mark all $\ell_i \in \mathcal{L}, i \in [n]$ as "uncovered";
4: $\mathcal{R}_{\text{temp}} \leftarrow \emptyset$;
5: **if** $\mathcal{R}_{\text{unused}} = \emptyset$
6: **return** $(\mathcal{R}_1, \ldots, \mathcal{R}_{k-1}, \mathcal{R}_k \cup \mathcal{R}_{\text{temp}})$;
7: **end if**
8: Choose $\hat{\mathcal{R}} \subseteq \mathcal{R}_{\text{unused}}$ such that $|\hat{\mathcal{R}}| \leqslant g$ and $\hat{\mathcal{R}}$ has the largest covering cardinality $|\text{CV}(\hat{\mathcal{R}})|$; /* In case of a tie, choose $\hat{\mathcal{R}}$ with minimum cardinality $|\hat{\mathcal{R}}|$; if there is still a tie, choose any.*/
9: $\mathcal{R}_{\text{temp}} \leftarrow \mathcal{R}_{\text{temp}} \cup \hat{\mathcal{R}}$;
10: $\mathcal{R}_{\text{unused}} \leftarrow \mathcal{R}_{\text{unused}} \setminus \hat{\mathcal{R}}$;
11: Mark $\ell_i \in \text{CV}(\hat{\mathcal{R}})$ as "covered";
12: **If** for all $i \in [n], \ell_i$ are covered,
13: $k \leftarrow k + 1$;
14: $\mathcal{R}_k \leftarrow \mathcal{R}_{\text{temp}}$;
15: Go to Step 3;
16: **else**
17: Go to Step 5;
18: **end if**

The following construction uses Algorithm 1 to construct a two-write ICI-free WOM code.

Construction 2 *Let m, n, n', ℓ be integers such that $m \leqslant n$ and $n' = (n+1)\ell$. Let $\mathcal{C}_n$ be the set of binary vectors that avoid 101 of length n, and let $\mathcal{C}_n = \mathcal{L} \cup \mathcal{R}$ be a partition of $\mathcal{C}_n$ such that $\mathcal{L} = \{\boldsymbol{x} \in \mathcal{C}_n : wt(\boldsymbol{x}) \leqslant m\}$ and $\mathcal{R} = \mathcal{C}_n \setminus \mathcal{L}$. Let $M_1 = |\mathcal{L}|$ and $f_1 : [0 : M_1 - 1] \to \mathcal{L}$ be an arbitrary bijective function. Let $\mathcal{R} = \cup_{i=0}^{k-1} \mathcal{R}_i$ be a covering subset partition of $\mathcal{R}$ obtained by running Algorithm 1. Suppose the cell-state vectors are $\boldsymbol{y}_{1,1}^{n'}$ and $\boldsymbol{y}_{2,1}^{n'}$ after the first and second write, respectively. A two-write ICI-free WOM code of length n' is constructed as follows:*

1) *On the first write, let $m \in [0 : M_1^{\ell} - 1]$ be the information message. Suppose $(m_1, m_2, \ldots, m_\ell)$ is the M_1-ary representation of m, i.e., $m = \sum_{i=1}^{\ell} m_i M_1^{\ell-i}$. Then for each $i \in [1 : \ell]$, write $\boldsymbol{y}_{1,(i-1)(n+1)+1}^{i(n+1)-1}$ according to the following rule,*
$$\boldsymbol{y}_{1,(i-1)(n+1)+1}^{i(n+1)-1} = f_1(m_i), \forall i \in [1 : \ell];$$
and write $y_{1,i(n+1)}$ according to the following rule
$$y_{1,i(n+1)} = \begin{cases} 1, & \text{if } y_{1,i(n+1)-1} = 1 \text{ and } y_{1,i(n+1)+1} = 1; \\ 0, & \text{otherwise}. \end{cases}$$

2) *On the second write, let $m \in [0 : k^{\ell} - 1]$ be the information message. Suppose $(m_1, m_2, \ldots, m_\ell)$ is the k-ary representation of m, i.e., $m = \sum_{i=1}^{\ell} m_i k^{\ell-i}$. Then for each $i \in [1 : \ell]$, write $\boldsymbol{y}_{2,(i-1)(n+1)+1}^{i(n+1)-1}$ according to the following rule,*
$$\boldsymbol{y}_{2,(i-1)(n+1)+1}^{i(n+1)-1} = \boldsymbol{x}_i \in \mathcal{R}_{m_i},$$
such that $\boldsymbol{x}_i$ covers $\boldsymbol{y}_{1,(i-1)(n+1)+1}^{i(n+1)-1}$;
and write $y_{2,i(n+1)}$ according to the following rule
$$y_{2,i(n+1)} = \begin{cases} 1, & \text{if } y_{2,i(n+1)-1} = 1 \text{ and } y_{2,i(n+1)+1} = 1; \\ 0, & \text{otherwise}. \end{cases}$$

Decoding is simply implemented by reversing the steps of the encoding procedure.

Remark 4. Construction 2 is a realization of Construction 1 for $t = 2$. The extension to $t > 2$ is straightforward. Only one "buffer" cell is used to avoid the ICI between adjacent blocks. Note that in Construction 1, it is possible to decrease the number of "buffer" cells from two to one. Two "buffer" cells are used to simplify the proof in Construction 1 since the number of "buffer" cells does not affect the asymptotic rate.

The following table shows the best rate we found using Algorithm 1 for selected values of n. From the table, we see that there exists a sequence of two-write ICI-free WOM codes of rate $R = 1.105 \times \frac{16}{17} \approx 1.04$, which represents 82% of the sum-capacity listed in Table I.

n	10	14	16
m	2	3	3
$\lvert\mathcal{L}\rvert$	48	336	513
k	46	139	1103
sum-rate	1.111	1.108	1.105

IV. Conclusions

ICI-free codes are used to mitigate the ICI during programming of flash memories. We extended ICI-free codes in two directions. First, we considered ICI-free balanced codes, which can be used with a dynamic read threshold to adapt to cell-level drift, and determined their asymptotic information rate. We then considered ICI-free WOM codes, which can be used to prolong the flash memory lifetime by reducing the number of block erasures. We calculated the sum-capacity of an ICI-free input-constrained WOM and provided simple code constructions that were used to design several codes with short block lengths. The derivation of the sum-capacity can also be generalized to WOMs with other input constraints.

V. Acknowledgment

The authors would like to thank David Callan for the proof of Equation (1) and would like to thank Aman Bhatia, Ryan Gabrys, and Anxiao Jiang for helpful discussions.

References

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

[2] J.-D. Lee, S.-H. Hur, and J.-D. Choi, "Effects of floating-gate interference on NAND flash memory cell operation," *IEEE Electron Device Lett.*, vol. 23, no. 5, pp. 264–266, May 2002.

[3] G. Dong, S. Li, and T. Zhang, "Using data postcompensation and predistortion to tolerate cell-to-cell interference in MLC NAND flash memory," *IEEE Trans. Circuits Syst.*, vol. 57, no. 10, pp. 2718–2728, October 2010.

[4] Q. Li, "WOM codes against inter-cell interference in NAND memories," in *Proc. 49-th Annual Allerton Conference on Communication, Control and Computing*, Monticello, IL, September 2011, pp. 1416–1423.

[5] A. Berman and Y. Birk, "Constrained flash memory programming," in *Proc. IEEE Int. Symp. Inform. Theory*, St. Petersburg, Russia, July - August 2011, pp. 2128–2132.

[6] H. Zhou, A. Jiang, and J. Bruck, "Error-correcting schemes with dynamic thresholds in nonvolatile memories," in *Proc. IEEE Int. Symp. Inform. Theory*, St. Petersburg, Russia, July–August 2011, pp. 2143–2147.

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

[8] P. Henry, "Zero disparity coding system," *U.S. Patent No. 4,309,694*, 1982.

[9] L. G. Tallini and B. Bose, "Balanced codes with parallel encoding and decoding," *IEEE Trans. Comput.*, vol. 48, no. 8, pp. 794–814, August 1999.

[10] L. G. Tallini, R. M. Capocelli, and B. Bose, "Design of some new efficient balanced codes," *IEEE Trans. Inf. Theory*, vol. 42, no. 3, pp. 790–802, May 1996.

[11] K. Immink and J. Weber, "Very efficient balanced codes," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 2, pp. 188–192, February 2010.

[12] J. Weber and K. Immink, "Knuth's balanced codes revisited," *IEEE Trans. Inf. Theory*, vol. 56, no. 4, pp. 1673–1679, April 2010.

[13] R. Mascella and L. G. Tallini, "On symbol permutation invariant balanced codes," in *Proc. IEEE Int. Symp. Inform. Theory*, Adelaide, Australia, September 2005, pp. 4–9.

[14] ——, "Efficient m-ary balanced codes which are invariant under symbol permutation," *IEEE Trans. Comput.*, vol. 55, no. 8, pp. 929–946, August 2006.

[15] T. G. Swart and J. H. Weber, "Efficient balancing of q-ary sequences with parallel decoding," in *Proc. IEEE Int. Symp. Inform. Theory*, Seoul, Korea, June - July 2009, pp. 1564–1568.

[16] E. Ordentlich and R. M. Roth, "Two-dimensinal weight-constrained codes through enumeration bounds," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1292–1301, July 2000.

[17] K. Immink, J. Weber, and H. Ferreira, "Balanced runlength limited codes using Knuth's algorithm," in *Proc. IEEE Int. Symp. Inform. Theory*, August 2011, pp. 317–320.

[18] 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.

[19] 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.

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

[21] R. Ahlswede and Z. Zhang, "Coding for write-efficient memory," *Inform. and Comput.*, vol. 83, no. 1, pp. 80–97, October 1989.

[22] F. Fu and R. Yeung, "On the capacity and error-correcting codes of write-efficient memories," *IEEE Trans. Inf. Theory*, vol. 46, no. 7, pp. 2299–2314, November 2000.

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

[24] 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.

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

[26] 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, pp. 1062–1068.

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

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

[29] L. Wang, M. Qin, E. Yaakobi, Y.-H. Kim, and P. H. Siegel, "WOM with retained messages," in *Proc. IEEE Int. Symp. Inform. Theory*, Cambridge, MA, USA, July 2012, pp. 1396–1400.

[30] Y. Sun, "The statistic "number of udu's" in Dyck paths," *Discrete Mathematics, 287*, pp. 177–186, July 2004.

[31] L. V. Ahlfors, *Complex Analysis*. New York: McGraw-Hill, 1966.

[32] A. Cauchy, *Analyse algébrique*, 1821.

[33] J. Hadamard, "Sur le rayon de convergence des séries ordonnées suivant les puissances d'une variable," *C. R. Acad. Sci. Paris 106*, pp. 259–262.

[34] N. J. A. Sloane, *The Online Encyclopedia of Integer Sequences*, https://oeis.org/.

[35] D. Callan, personal communication, October 2012.

[36] A. Brouwer, J. Shearer, N. Sloane, and W. Smith, "A new table of constant weight codes," *IEEE Trans. Inf. Theory*, vol. 36, no. 6, pp. 1334–1380, November 1990.

[37] R. Roth, personal communication, July 2013.

[38] J. Lee and V. K. Madisetti, "Constrained multitrack RLL codes for the storage channel," *IEEE Trans. Magn.*, vol. 31, no. 3, pp. 2355–2364, May 1995.

[39] A. Kato and K. Zeger, "On the capacity of two-dimensional run-length constrained channels," *IEEE Trans. Inf. Theory*, vol. 45, no. 5, pp. 1527–1540, July 1999.
[40] S. Forchhammer and T. V. Laursen, "A model for the two-dimensional no isolated bits constraint," in *Proc. IEEE Int. Symp. Inform. Theory*, Seattle, Washington, July 2006, pp. 1189–1193.
[41] S. Halevy, J. Chen, R. M. Roth, P. H. Siegel, and J. K. Wolf, "Improved bit-stuffing bounds on two-dimensional constraints," *IEEE Trans. Inf. Theory*, vol. 50, no. 5, pp. 824–838, May 2004.
[42] Z. Nagy and K. Zeger, "Asympototic capacity of two-dimensional channels with checkerboard constraints," *IEEE Trans. Inf. Theory*, vol. 49, no. 9, pp. 2115–2125, September 2003.
[43] R. M. Roth, P. H. Siegel, and J. K. Wolf, "Efficient coding schemes for the hard-square model," *IEEE Trans. Inf. Theory*, vol. 47, no. 3, pp. 1166–1176, March 2001.
[44] B. H. Marcus, R. M. Roth, and P. H. Siegel, *Constrained Systems and Coding for Recording Channels*. Handbook of Coding Theory (V. S. Pless and W. C. Huffman, eds.), ch. 20, Elsevier Science, 1998.
[45] W. Weeks and R. Blahut, "The capacity and coding gain of certain checkerboard codes," *IEEE Trans. Inf. Theory*, vol. 44, no. 3, pp. 1193–1203, May 1998.
[46] Y. Wu, "Low complexity codes for writing write-once memory twice," in *Proc. IEEE Int. Symp. Inform. Theory*, Austin, Texas, June 2010, pp. 1928–1932.

Minghai Qin (S'11) received the B.E. degree in electronic and electrical engineering from Tsinghua University, Beijing, China, in 2009. He is currently pursuing the Ph.D. degree in electrical engineering from the Department of Electrical and Computer Engineering at the University of California, San Diego, where he is associated with the Center for Magnetic Recording Research.

Eitan Yaakobi (S'07–M'12) 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 postdoctoral researcher in electrical engineering at the California Institute of Technology, Pasadena. His research interests include coding theory, algebraic error-correction coding, and their applications for digital data storage and in particular for non-volatile memories. Dr. Yaakobi received the Marconi Society Young Scholar in 2009 and the Intel Ph.D. Fellowship in 2010-2011.

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 at the University of California, San Diego in July 1995, where he is currently Professor of Electrical and Computer Engineering in the Jacobs School of Engineering. He is affiliated with 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 from 2009 to 2011. He was re-elected for another 3-year term in 2012. 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.

Prof. Siegel was co-recipient, 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.H. 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 an IEEE Fellow and a member of the National Academy of Engineering.