This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE Globecom 2011 proceedings.

Time-Space Constrained Codes for Phase-Change Memories

Minghai Qin, Eitan Yaakobi, and Paul H. Siegel
University of California, San Diego La Jolla, CA 92093 – 0401, USA
Emails: {mqin, eyaakobi, psiegel}@ucsd.edu

***Abstract*—Phase-change memory (PCM) is a promising non-volatile solid-state memory technology. A PCM cell stores data by using its amorphous and crystalline states. The cell changes between these two states using high temperature. However, since the cells are sensitive to high temperature, it is important, when programming cells, to balance the heat both in time and space.**

In this paper, we study the time-space constraint for PCM, which was recently proposed by Jiang et al. A code is called an (α, β, p)-*constrained code* if for any α consecutive rewrites and for any segment of β contiguous cells, the total rewrite cost of the β cells over those α rewrites is at most p. Here, the cells are binary and the rewrite cost is defined to be the Hamming distance between the current and next memory states. First, we show a general upper bound on the achievable rate of these codes which extends the results of Jiang et al. Then, we generalize their construction for $(\alpha \geqslant 1, \beta = 1, p = 1)$-constrained codes and show another construction for $(\alpha = 1, \beta \geqslant 1, p \geqslant 1)$-constrained codes. Finally, these two constructions are used to construct codes for all values of α, β, and p.

I. Introduction

Phase-change memory (PCM) devices are a promising technology for non-volatile memories. Like a flash memory, a PCM consists of cells that can be in distinct physical states. In the simplest case, the PCM cell has two possible states, an amorphous state and a crystalline state. Multiple-bit per cell PCMs can be implemented by using partially crystalline states [3].

While in a flash memory one can decrease a cell level only by erasing the entire block of about 10^6 cells that contains it, in a PCM one can independently decrease an individual cell level – but only to level zero. This operation is called a RESET operation. A SET operation can then be used to change the cell state to any valid level. Therefore, in order to decrease a cell level from one non-zero value to a smaller non-zero value, one needs to first RESET the cell to level zero, and then SET it to the new desired level [3]. Thus, as with flash memory programming, there is a significant asymmetry between the two operations of increasing and decreasing a cell level. Another possible technique to program the multilevel PCM cells is described in [2].

As in a flash memory, a PCM cell has a limited lifetime; the cells can tolerate only about $10^7 - 10^8$ RESET operations before beginning to degrade [4]. Therefore, it is still important when programming cells to minimize the number of RESET operations. Furthermore, a RESET operation can negatively affect the performance of a PCM in other ways. One of them is due to the phenomenon of thermal crosstalk. When a cell is RESET, the levels of its adjacent cells may inadvertently be increased due to heat diffusion associated with the operation [3], [10]. Another problem, called thermal accumulation, arises when a small area is subjected to a large number of program operations over a short period of time [3], [10]. The resulting accumulation of heat can significantly limit the write latency of a PCM, since the programming accuracy is sensitive to temperature. It is therefore desirable to balance the thermal accumulation over a local area of PCM cells in a fixed period of time. Coding schemes can help overcome the performance degradation resulting from these physical phenomena. Lastras et al. [7] studied the capacity of a Write-Efficient Memory (WEM) [1] for a cost function that is associated with the write model of a PCM described above.

Jiang et al. [6] have proposed codes to mitigate thermal cross-talk and heat accumulation effects in PCM. Under their thermal cross-talk model, when a cell is RESET, the levels of its immediately adjacent cells may also be increased. Hence, if these neighboring cells exceed their target level, they also will have to be RESET, and this effect can then propagate to many more cells. In [6], they considered a special case of this and proposed the use of constrained codes to limit the propagation effect. Capacity calculations for these codes were also presented.

The other problem addressed in [6] is that of heat accumulation. In this model, the rewrite cost is defined to be the number of programmed cells, i.e., the Hamming distance between the current and next cell-state vectors. A code is said to be (α, β, p)-*constrained* if for any α consecutive rewrites and for any segment of β contiguous cells, the total rewrite cost of the β cells over those α rewrites is at most p. A specific code construction was given for the $(\alpha \geqslant 1, \beta = 1, p = 1)$-constraint as well as an upper bound on the achievable rate of codes for this constraint. An upper bound on the achievable rate was also given for $(\alpha = 1, \beta \geqslant 1, p = 1)$-constrained codes.

Only a subset of the ranges of the parameters α, β, and p was considered in [6]. In this paper, we extend the code constructions and achievable-rate bounds to a larger portion of the parameter space. In Section II, we formally define the constrained-coding problem for PCM. In Section III, for any α, β, and p, we define a two-dimensional constraint whose capacity is an upper bound to that of the (α, β, p) constraint. If $\alpha = 1$ or $\beta = 1$, then the two-dimensional constraint effectively reduces to a one-dimensional constraint whose capacity can be computed. These results coincide with those in [6] for $(\alpha \geqslant 1, \beta = 1, p = 1)$ and $(\alpha = 1, \beta \geqslant 1, p = 1)$. We also derive upper bounds for some cases with parameters satisfying $(\alpha > 1, \beta > 1, p = 1)$ using known upper bounds on the capacity of certain two-dimensional constraints. In Section IV, we develop several lower bounds on capacity of time-space constraints. First, we describe an elementary code construction, applicable to any (α, β, p) constraint, that provides a general lower bound on capacity. We then give a more effi-

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE Globecom 2011 proceedings.

cient code construction for $(\alpha = 1, \beta \geqslant 1, p \geqslant 1)$ constraints and extend the construction for $(\alpha \geqslant 1, \beta = 1, p = 1)$ constraints in [6] to arbitrary p. Finally, we extend some of these code construction ideas to derive another lower bound valid for all values of α, β and p.

Due to page limitations, the proofs of some of the results in this paper are omitted.

II. Preliminaries

In this section, we give a formal definition of the constrained-coding problem. The number of cells is denoted by n and the memory cells are binary. The cell-state vectors are the binary vectors from $\{0,1\}^n$. Similar definitions and results can be derived for non-binary cell levels as an extension of this work in the future. If a cell state $\boldsymbol{u} = (u_1, \ldots, u_n) \in \{0,1\}^n$ is rewritten to another cell state $\boldsymbol{v} = (v_1, \ldots, v_n) \in \{0,1\}^n$, then the rewrite cost is defined to be the Hamming distance between $\boldsymbol{u}$ and $\boldsymbol{v}$, that is $d_H(\boldsymbol{u}, \boldsymbol{v}) = |\{i : u_i \neq v_i, 1 \leqslant i \leqslant n\}|$. The Hamming weight of a vector $\boldsymbol{u}$ is $wt(\boldsymbol{u}) = d_H(\boldsymbol{u}, \boldsymbol{0})$. The complement of a vector $\boldsymbol{u}$ is $\overline{\boldsymbol{u}} = (\overline{u_1}, \ldots, \overline{u_n})$.

Definition 1. *Let α, β, p be positive integers. A code $\mathcal{C}$ is called an* $\boldsymbol{(\alpha, \beta, p)}$***-constrained code*** *if for any α consecutive rewrites and for any segment of β contiguous cells, the total rewrite cost of those β cells over those α rewrites is at most p. That is, if $\boldsymbol{v}_i = (v_{i,1}, \ldots, v_{i,n})$, $i \geqslant 1$, is the cell-state vector on the i-th write, then, for all $i \geqslant 1$ and $1 \leqslant j \leqslant n - \beta + 1$,*

$$\left|\{(k,\ell) : v_{i+k,j+\ell} \neq v_{i+k+1,j+\ell}, 0 \leqslant k < \alpha, 0 \leqslant \ell < \beta\}\right| \leqslant p.$$

We will specify (α, β, p)-constrained codes by an explicit construction of their encoder and decoder maps. The encoder

$$\mathcal{E} : \{0,1\}^n \times \{1, \ldots, M\} \mapsto \{0,1\}^n$$

maps the current cell-state vector and the new information symbol to the next cell-state vector. The decoder

$$\mathcal{D} : \{0,1\}^n \mapsto \{1, \ldots, M\}$$

maps the cell-state vector to the represented information symbol. If the number of messages that can be written on each write has the same value M, we say that the ***rate R*** of the (α, β, p)-constrained code is

$$R = \frac{\log_2 M}{n}. \qquad (1)$$

Note that the size of the alphabet of messages for each write does not have to be the same. In this case, we denote by M_i the number of messages that can be written on the i-th write. The individual rate on the i-th write is $R_i = \frac{\log_2 M_i}{n}$, and the rate R of the (α, β, p)-constrained code is defined to be

$$R = \lim_{m \to \infty} \frac{\sum_{i=1}^{m} R_i}{m}. \qquad (2)$$

We assume the number of writes is large and in the constructions we present there will be a period of writes which repeats itself. Thus, it will be possible to change any (α, β, p)-constrained code $\mathcal{C}$ with varying individual rates to an (α, β, p)-constrained code $\mathcal{C}'$ with fixed individual rates such that the rates of the two constrained codes are the same. This can be achieved by multiple copies of the code $\mathcal{C}$ and in each copy of $\mathcal{C}$ to start writing from a different write within the period of writes. Therefore, we assume that there is no distinction between the two cases and the rate is as defined in (2), which is the amount of information written per cell per write.

The encoding and decoding maps can be either the same on all writes or can vary among the writes. In the latter case, we will need more cells in order to index the write number. However, arguing as in [14], it is possible to show that these extra cells do not reduce asymptotically the rate and therefore we assume here that the encoder and decoder know the write number.

A rate R is called an $\boldsymbol{(\alpha, \beta, p)}$***-achievable rate*** if there exists an (α, β, p)-constrained code $\mathcal{C}$ such that the rate of $\mathcal{C}$ is R. we denote by $C_n(\alpha, \beta, p)$ the supremum of all (α, β, p)-achievable rate while fixing the number of cells to be n. The $\boldsymbol{(\alpha, \beta, p)}$***-capacity*** of the (α, β, p)-constrained codes is denoted by $\boldsymbol{C(\alpha, \beta, p)}$ and is defined to be

$$C(\alpha, \beta, p) = \lim_{n \to \infty} C_n(\alpha, \beta, p).$$

Our goal in this paper is to give lower and upper bounds on the (α, β, p)-capacity, $C(\alpha, \beta, p)$, for all values of α, β, and p. Lower bounds will be given by specific constrained-code constructions while the upper bounds will be derived analytically using tools drawn from the theory of one and two-dimensional constrained codes.

III. Upper Bound on the Capacity

In this section, we will present upper bounds on the (α, β, p)-capacity obtained using techniques used in the analysis of two-dimensional constrained codes. There are a number of two-dimensional constraints that have been extensively studied, e.g., 2-D (d, k)-runlength-limited (RLL), no-isolated-bits, and checkerboard constraints. Given a constraint S, its capacity is defined to be

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

where $c_S(m,n)$ is the number of $m \times n$ arrays that satisfy the constraint S. The constraint of interest for us in this work is the one where in each rectangle of size $a \times b$, the number of ones is at most p.

Definition 2. *Let a, b, p be positive integers. An $(m \times n)$-array $A = (a_{i,j})_{1 \leqslant i \leqslant m, 1 \leqslant j \leqslant n}$ is called an (a, b, p)-array if in each sub-array of A of size $a \times b$, the number of 1's is at most p. That is, for all $1 \leqslant i \leqslant m - a + 1$, $1 \leqslant j \leqslant n - b + 1$,*

$$\left|\{(k,\ell) : 0 \leqslant k \leqslant a-1, 0 \leqslant \ell \leqslant b-1, a_{i+k,j+\ell} = 1\}\right| \leqslant p.$$

The capacity of the constraint is denoted by $C_{2D}(a, b, p)$.

Note that when $p = 1$, the $(a, a, 1)$ constraint coincides with the square checkerboard constraint of order $a - 1$ [13].

Theorem 1. *For all α, β, p, $C(\alpha, \beta, p) \leqslant C_{2D}(\alpha, \beta, p)$.*

Proof: Let $\mathcal{C}$ be an (α, β, p)-constrained code of length n. For any sequence of m writes, let us denote by $\boldsymbol{v}_i$, for $i \geqslant 0$, the cell-state vector on the i-th write, where $\boldsymbol{v}_0$ is the all-zero vector. The $m \times n$-array $A = (a_{i,j})$ is defined to be

$$a_{i,j} = v_{i,j} + v_{i-1,j},$$

where the addition is a modulo-2 sum. That is, $a_{i,j} = 1$ if and only if the j-th cell is changed on the i-th write. Since $\mathcal{C}$ is an (α, β, p)-constrained code, for all $1 \leqslant i \leqslant m - \alpha$ and $1 \leqslant j \leqslant n - \beta + 1$,

$$\left|\{(k,\ell) : v_{i+k,j+\ell} \neq v_{i+k+1,j+\ell}, 0 \leqslant k < \alpha, 0 \leqslant \ell < \beta\}\right| \leqslant p,$$

and therefore

$$\left|\{(k,\ell) : 0 \leqslant k \leqslant \alpha-1, 0 \leqslant \ell \leqslant \beta-1, a_{i+k,j+\ell} = 1\}\right| \leqslant p.$$

Thus, A is an (α, β, p)-array of size $m \times n$.

Every write sequence of the code $\mathcal{C}$ corresponds to an (α,β,p)-array and thus the number of write sequences of length m is at most the number of (α,β,p)-arrays, which is upper bounded by $2^{mnC_{2D}(\alpha,\beta,p)}$, for m,n large enough. Hence, the number of distinct write sequences is at most $2^{mnC_{2D}(\alpha,\beta,p)}$. However, if the individual rate on the i-th write is R_i, then the total number of distinct write sequences is $\prod_{i=1}^{m} 2^{nR_i}$. We conclude that

$$\prod_{i=1}^{m} 2^{nR_i} \leqslant 2^{mnC_{2D}(\alpha,\beta,p)}$$

and therefore, R, the rate of the (α,β,p)-constrained code satisfies

$$R = \frac{\sum_{i=1}^{m} R_i}{m} \leqslant C_{2D}(\alpha,\beta,p),$$

i.e., $C(\alpha,\beta,p) \leqslant C_{2D}(\alpha,\beta,p)$. ■

Theorem 1 provides a scheme to calculate an upper bound on the (α,β,p)-capacity from an upper bound on the capacity of a two-dimensional rectangular checkerboard constraint. Unfortunately, good upper bounds are known only for some special cases of the values of α,β,p, and in particular, when $p=1$. The checkerboard constraint has attracted considerable attention over the past 20 years and some lower and upper bounds on the capacity were given in [9], [12], [13]. For instance, [13] gives some upper bounds for square checkerboard constrained codes, from which we can conclude that $C(2,2,1) \leqslant 0.43431$ and $C(3,3,1) \leqslant 0.25681$.

In the rest of this section we discuss the cases where $\alpha = 1$ or $\beta = 1$, for which the two-dimenional constraints effectively reduce to one-dimensional constraints whose capacities can be computed. We first consider the case where $\alpha = 1$. If $p = 1$, we recover the connection to $(\beta-1,\infty)$-runlength-limited (RLL) constraints [15] observed in [6]. The lowest curve in Figure 1 shows the capacity of the $(\beta-1,\infty)$-RLL constraint as a function of β. For $p>1$, the corresponding one-dimensional constraint is defined below.

Definition 3. *Let β,p be two positive integers. A vector $\boldsymbol{u}$ satisfies the* ***(β,p)-window-weight-limited (WWL) constraint*** *if for any β consecutive cells there are at most p ones. We denote the capacity of the constraint by $C_{WWL}(\beta,p)$.*

Note that for $p=1$, the $(\beta,1)$-WWL constraint is the $(\beta-1,\infty)$-RLL constraint. According to Theorem 1, $C(1,\beta,p)$ is upper bounded by the capacity of the (β,p)-WWL constraint, $C_{WWL}(\beta,p)$. Thus, we are interested in finding the capacity of this constraint. The approach is similar to the one used in [13] in order to find an upper bound on the capacity of the checkerboard constraint.

Definition 4. *A* merge *of two vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ of the same length n is a function:*

$$f_n : \{0,1\}^n \times \{0,1\}^n \mapsto \{0,1\}^{n+1} \cup \{\boldsymbol{F}\}.$$

If the last $n-1$ bits of $\boldsymbol{u}$ are the same as the first $n-1$ bits of $\boldsymbol{v}$, the vector $f_n(\boldsymbol{u},\boldsymbol{v})$ is the vector $\boldsymbol{u}$ appended by the last bit of $\boldsymbol{v}$, otherwise $f_n(\boldsymbol{u},\boldsymbol{v}) = \boldsymbol{F}$.

Definition 5. *Let β,p be two positive integers. The set $S_{\beta,p}$ is the set of all vectors of length $\beta-1$ having at most p 1's. That is, $S_{\beta,p} = \{\boldsymbol{s} \in \{0,1\}^{\beta-1} : wt(\boldsymbol{s}) \leqslant p\}$, where $wt(\boldsymbol{s})$ denotes the Hamming weight of $\boldsymbol{s}$. The size of the set $S_{\beta,p}$ is $M = \sum_{i=0}^{p}\binom{\beta-1}{i}$. Let $\boldsymbol{s}_1,\boldsymbol{s}_2,\ldots,\boldsymbol{s}_M$ be an ordering of the vectors in $S_{\beta,p}$. The* transition matrix *for the (β,p)-WWL constraint, $A_{\beta,p} = (a_{i,j}) \in \{0,1\}^{M\times M}$ is defined as follows:*

$$a_{i,j} = \begin{cases} 1 & \text{if } f_{\beta-1}(\boldsymbol{s}_i,\boldsymbol{s}_j) \neq \boldsymbol{F} \text{ and } wt(f_{\beta-1}(\boldsymbol{s}_i,\boldsymbol{s}_j)) \leqslant p, \\ 0 & \text{otherwise.} \end{cases}$$

Example 1. The following illustrates the construction of the $A_{\beta=3,p=2}$ transition matrix. Note that $S_{3,2} = \{\boldsymbol{s}_1,\boldsymbol{s}_2,\boldsymbol{s}_3,\boldsymbol{s}_4\} = \{(0,0),(0,1),(1,0),(1,1)\}$. The merge of $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$ for $i,j=1,2,3,4$ determines the matrix $A_{3,2}$. For example, $f_2(\boldsymbol{s}_1,\boldsymbol{s}_1) = (0,0,0)$, $a_{1,1}=1$; $f_2(\boldsymbol{s}_2,\boldsymbol{s}_1) = \boldsymbol{F}$, $a_{2,1}=0$; $f_2(\boldsymbol{s}_1,\boldsymbol{s}_2) = (0,0,1)$, $a_{2,1} = 1 \neq a_{1,2}$. This shows that the matrix is not necessarily symmetric. Finally, $f_2(\boldsymbol{s}_3,\boldsymbol{s}_3) = (1,1,1)$, and $a_{3,3}=0$ since $(1,1,1)$ does not satisfy the (3,2)-WWL constraint.

$$A_{3,2} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

The next theorem is a special case of Theorem 3.9 in [8].

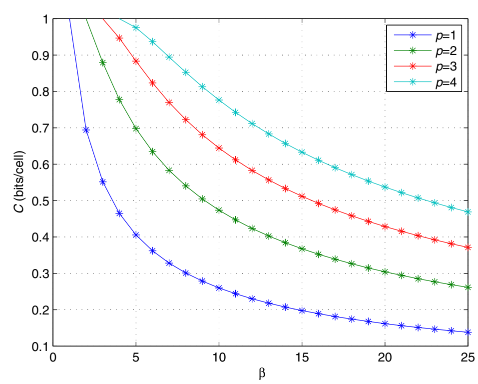
Theorem 2. *The capacity of the (β,p)-WWL constraint is*

$$C_{WWL}(\beta,p) = \log_2(\lambda_{max}),$$

where λ_{max} is the largest real eigenvalue of $A_{\beta,p}$.

Figure 1 shows the capacities of (β,p)-WWL constraints as a function of β, for $p=1,2,3,4$. These provide upper bounds on the corresponding capacities $C(1,\beta,p)$.

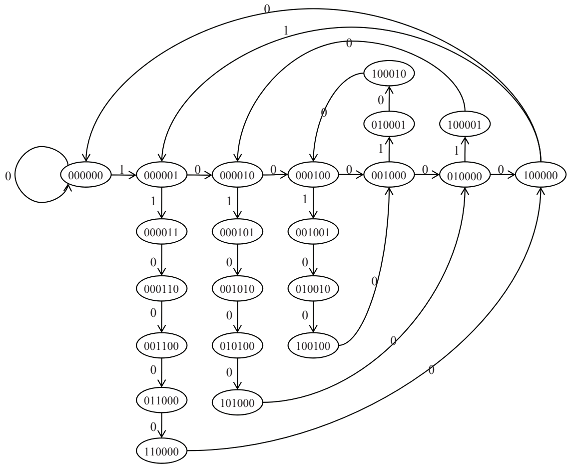
Fig. 1. Upper bound on $C(1,\beta,p)$



Remark 1. For $\beta = 1,2$, there is a convenient way to represent the (β,p)-WWL constraints using state-transition diagrams. The connection between transition matrices and state-transition diagrams is explained in detail in [8]. As an example, the state-transition diagram for the $(\beta=7,p=2)$-WWL constraint is shown in Fig. 2. The vectors in the ellipses represent the states and the edge labels represent the digits corresponding to the state transitions.

Remark 2. According to Theorem 1, the capacity $C(\alpha,1,p)$ is also upper bounded by the capacity of the (α,p)-WWL constraint, $C_{WWL}(\alpha,p)$. Jiang et al. [6] proposed an upper bound on the rate of an $(\alpha,1,1)$-constrained code with fixed block length n and multiple cell levels. By numerical experiment, we find that their upper bound converges to our bound for binary cells when $n\to\infty$.

Fig. 2. Transition diagram for (7,2)-WWL constraint



IV. Lower Bound on the Capacity

In this section, we give lower bounds on the capacity of (α, β, p)-constrained codes. Clearly, if $p \geqslant \alpha\beta$ then $C(\alpha, \beta, p) = 1$ so we assume here that $p < \alpha\beta$. The bounds will be given by specific code constructions. The first construction we give is an elementry one which achieves rate $\frac{p}{\alpha\beta}$. Then, we will show how to improve it for the cases $(1, \beta, p)$ and $(\alpha, 1, p)$. In this section, for all positive integers x, y, we represent the values $\{0, 1, \ldots, y-1\}$ of $x (\bmod y)$ as $\{y, 1, \ldots, y-1\}$.

Construction 1 *Let α, β, p be positive integers. We construct an (α, β, p)-constrained code $\mathcal{C}$ of length n as follows. For simplicity, we assume that $\beta | n$. Let $x = \left\lceil \frac{p}{\beta} \right\rceil, y = p (\bmod \beta)$. For all $i \geqslant 1$, on the i-th write, the encoder uses the following rules:*

- *If $1 \leqslant i (\bmod \alpha) < x$, n bits are written to the n cells.*
- *If $i (\bmod \alpha) = x$, yn/β bits are written in all cells c_j such that $1 \leqslant j (\bmod \beta) \leqslant y$.*
- *If $i (\bmod \alpha) > x$, no information is written to the cells.*

The decoder is implemented in a very similar way.

Theorem 3. *The code $\mathcal{C}$ constructed in Construction 1 is an (α, β, p)-constrained code and its rate is $R = \frac{p}{\alpha\beta}$.*

A. Space Constraint Improvement

In this subsection, we improve the lower bound on $C(1, \beta, p)$. Let $\mathcal{S}_n(\beta, p)$ be the set of all vectors of length n such that each vector satisfies the (β, p)-WWL constraint. We define a (β, p)**-WWL code** $\widetilde{\mathcal{C}}$ of length n as a subset of $\mathcal{S}_n(\beta, p)$. If the size of the code $\widetilde{\mathcal{C}}$ is M, then it is specified by an encoding map $\widetilde{\mathcal{E}} : \{1, \ldots, M\} \mapsto \widetilde{\mathcal{C}}$ and a decoding map $\widetilde{\mathcal{D}} : \widetilde{\mathcal{C}} \mapsto \{1, \ldots, M\}$, such that $\forall m \in \{1, \ldots, M\}$, $\widetilde{\mathcal{D}}(\widetilde{\mathcal{E}}(m)) = m$.

The problem of finding (β, p)-WWL codes that achieve the capacity $C_{WWL}(\beta, p)$ is of independent interest. We found encoding and decoding strategies with linear complexity that have rate approaching the capacity as the block length goes to infinity. Due to the page limitations, we defer the details to an extended version of this paper. In the sequel, we will simply assume that there exist such codes that achieve the capacity for all positive integers p and β. The next construction uses (β, p)-WWL codes to construct $(1, \beta, p)$-constrained codes. For a vector $\boldsymbol{x} = (x_1, \ldots, x_n)$, we define $\boldsymbol{x}_p^q = (x_p, x_{p+1}, \ldots, x_q)$, for all $1 \leqslant p \leqslant q \leqslant n$.

Construction 2 *Let β, p be positive integers such that $p \leqslant \beta$. Let $\widetilde{\mathcal{C}}$ be a (β, p)-WWL code of length n' and size M. Let $\widetilde{\mathcal{E}}$ and $\widetilde{\mathcal{D}}$ be its encoding and decoding maps. A $(1, \beta, p)$-constrained code $\mathcal{C}_{1,\beta,p}$ of length $n = 2n' + \beta - 1$ and its encoding map $\mathcal{E}$ and decoding map $\mathcal{D}$ are constructed as follows.*

1) *The encoding map $\mathcal{E} : \{1, \ldots, M\} \times \{0,1\}^n \to \{0,1\}^n$ is defined for all $(m, \boldsymbol{u}) \in \{1, \ldots, M\} \times \{0,1\}^n$ to be $\mathcal{E}((m, \boldsymbol{u})) = \boldsymbol{v}$, where*
   a) $\boldsymbol{v}_1^{n'} = \boldsymbol{u}_1^{n'} + \widetilde{\mathcal{E}}(m)$.
   b) $\boldsymbol{v}_{n'+1}^{n'+\beta-1} = \mathbf{0}$,
   c) $\boldsymbol{v}_{n'+\beta}^{n} = \boldsymbol{u}_1^{n'}$,
2) *The decoding map $\mathcal{D} : \{0,1\}^n \to \{1, \ldots, M\}$ is defined for all $\boldsymbol{u} \in \{0,1\}^n$ to be*
$$\mathcal{D}(\boldsymbol{u}) = \widetilde{\mathcal{D}}(\boldsymbol{v}_1^{n'} + \boldsymbol{v}_{n'+\beta}^{n}).$$

Theorem 4. *Let β, p be positive integers and let $\mathcal{C}_{1,\beta,p}$ be the code constructed in Construction 2 using the code $\widetilde{\mathcal{C}}$ of length n'. Then, the code $\mathcal{C}_{1,\beta,p}$ is a $(1, \beta, p)$-constrained code. If the rate of the code $\widetilde{\mathcal{C}}$ is $\widetilde{R}$, then the rate of the code $\mathcal{C}_{1,\beta,p}$ is $\frac{n'}{2n'+\beta-1} \cdot \widetilde{R}$.*

Proof: Let $\boldsymbol{u}$ be the cell-state vector in Construction 2.

1) For $\boldsymbol{u}_1^{n'}$, encoder step a) guarantees that the positions of rewritten satisfy (β, p)-WWL constraint. So there are at most p reprograms in any β consecutive cells in $\boldsymbol{u}_1^{n'}$.
2) For $\boldsymbol{u}_{n'+\beta}^{n}$, three consecutive writes should be examined. Let $\boldsymbol{w}, \boldsymbol{v}, \boldsymbol{u}$ be the cell-state vectors before the i-th, $(i+1)$-st, $(i+2)$-nd writes, $i \geqslant 1$. Encoder step a) means that $\boldsymbol{v}_1^{n'} = \boldsymbol{w}_1^{n'} + \widetilde{\mathcal{E}}(m_i)$, where $m_i \in \{1, \ldots, M\}$ is the message to encode on the i-th write. Since Encoder step c) guarantees that $\boldsymbol{v}_{n'+\beta}^{n} = \boldsymbol{w}_1^{n'}$ and $\boldsymbol{u}_{n'+\beta}^{n} = \boldsymbol{v}_1^{n'}$, we have $\boldsymbol{u}_{n'+\beta}^{n} = \boldsymbol{v}_{n'+\beta}^{n} + \widetilde{\mathcal{E}}(m_i)$. This proves that $\boldsymbol{u}_{n'+\beta}^{n}$ satisfies the $(1, \beta, p)$ constraint.
3) For $\boldsymbol{u}_{n'+1}^{n'+\beta-1}$, the cell levels are always set to be 0, which makes sure that no violation of constraint happens between $\boldsymbol{u}_1^{n'}$ and $\boldsymbol{u}_{n'+\beta}^{n}$.

On each write, one of M messages are encoded as a vector of length n. Hence, the rate is $\frac{\log_2 M}{n} = \left(\frac{\log_2 M}{n'} \frac{n'}{2n'+\beta-1} \right) = \frac{n'}{2n'+\beta-1} \cdot \widetilde{R}$. ■

Corollary 5. *Let β, p be two positive integers such that $p \leqslant \beta$, then*

$$C(1, \beta, p) \geqslant \max \left\{ \frac{C_{WWL}(\beta, p)}{2}, \frac{p}{\beta} \right\}.$$

Remark 3. For small $\frac{p}{\beta}$, $\frac{C_{WWL}(\beta,p)}{2}$ is greater than the trivial lower bound $\frac{p}{\beta}$. As an example of $p = 1$, when $\beta \geqslant 5$, $\frac{C_{WWL}(\beta,p)}{2} > \frac{p}{\beta}$.

B. Time Constraint Improvement

Jiang et al. constructed in [6] an $(\alpha,1,1)$-constrained code. Let us explain their construction as it serves as the basis for our construction. Their construction uses Write-Once Memory (WOM)-codes [11]. A WOM is a storage device consisting of cells. In the binary case, each cell can irreversibly change from state 0 to state 1. We denote $[n,t;M_1,\ldots,M_t]$ to be a t-write WOM-code $\mathcal{C}_W$ such that the number of messages that can be written to the memory on its i-th write is M_i, and the rate of the WOM-code is defined to be $R_W = \frac{\sum_{i=1}^{t}\log_2 M_i}{n}$. The code is specified by t pairs $(\mathcal{E}_i(m,v),\mathcal{D}_i(v))^1$ for $1 \leqslant i \leqslant t$, of encoding and decoding maps, where v stands for the cell-state vector and m is the new message to be encoded.

The constructed $(\alpha,1,1)$-constrained code uses a t-write WOM-code and has a period of $2(t+\alpha)$ writes repeating itself. On the first t writes of each period, the encoder simply writes the information using the encoding maps of the t-write WOM-code. Then, on the $(t+1)$-st write no information is written but all the cells are increased to level one. On the following $\alpha-1$ writes no information is written and the cells do not change their values; that completes half of the period. On the next t writes the same WOM-code is again used, however since now all the cells are in level one, the complement of the cell-state vector is written to the memory on each write. On the next write no information is written and the cells are reduced to level zero and in the last $\alpha-1$ writes no information is written and the cells do not change their values. We present this construction now in detail.

Construction 3 *Let α be a positive integer and let $\mathcal{C}_W$ be an $[n,t;M_1,\ldots,M_t]$ t-write WOM-code. For $1 \leqslant i \leqslant t$, let $\mathcal{E}_i$, $\mathcal{D}_i$ be its encoding, decoding map, respectively. An $(\alpha,1,1)$-constrained code $\mathcal{C}_{\alpha,1,1}$ is constructed as follows. For all $i \geqslant 1$, let $i' = i(\bmod(2(t+\alpha)))$, where $1 \leqslant i' \leqslant 2(t+\alpha)$. We denote by v the cell-state vector. On the i-th write, the encoder uses the following rules:*

- *If $1 \leqslant i' \leqslant t$, write one of $M_{i'}$ messages m using the encoding map $\mathcal{E}_{i'}(m,v)$.*
- *If $i' = t+1$, no information is written and the cell-state vector is changed to $(1,\ldots,1)$.*
- *If $t+2 \leqslant i' \leqslant t+\alpha$, no information is written and the cell-state vector is not changed.*
- *If $t+\alpha+1 \leqslant i' \leqslant 2t+\alpha$, write one of $M_{i'-t-\alpha}$ messages m using the encoding map $\overline{\mathcal{E}_{i'-t-\alpha}(m,\overline{v})}$.*
- *If $i' = 2t+\alpha+1$, no information is written and the cell-state vector is changed to $(0,\ldots,0)$.*
- *If $2t+\alpha+1 \leqslant i' \leqslant 2(t+\alpha)$, no information is written and the cell-state vector is not changed.*

Remark 4. Note that this construction is presented differently in [6]. This results from the constraint of having the same rate on each write which we can bypass in our work. Also, the WOM-codes used in [6] have to have the same rate on each write and in our case we can have varying rates and thus the code $\mathcal{C}_{\alpha,1,p}$ can achieve higher rate.

[1]For $i=1$, the encoding map is $\mathcal{E}_1(m)$, but for emphasis we refer to it as $\mathcal{E}_1(m,v)$ where v will be the all-zero vector.

Theorem 6. *The code $\mathcal{C}_{\alpha,1,1}$ constructed in Construction 3 is an $(\alpha,1,1)$-constrained code. If the rate of the t-write WOM-code is R_W, then the rate of $\mathcal{C}_{\alpha,1,1}$ is $\frac{R_W}{t+\alpha}$.*

Proof: In every period of $2(t+\alpha)$ writes, every cell is programmed at most twice. Once in the first $t+1$ writes and once in the first $t+1$ writes of the second part of the writes-period. After every such $t+1$ writes, the cell is not programmed for $\alpha-1$ writes. Therefore the rewrite of every cell among α consecutive rewrites is at most 1.

If the rate of the WOM-code is R_W then $2R_W n$ bits are written in every period of $2(t+\alpha)$ writes. Hence, the rate of $\mathcal{C}_{\alpha,1,1}$ is $\frac{2R_W n}{2(t+\alpha)n} = \frac{R_W}{t+\alpha}$. ■

Remark 5. It is shown in [5] that for a t-write WOM-code, R_W achieves $\log_2(t+1)$ asymptotically.

Next, we modify Construction 3 to design $(\alpha,1,p)$-constrained codes for all $p \geqslant 2$. For simplicity, we will assume that p is an even integer as the extension to odd values of p is straightforward. We assume that $\alpha \geqslant (p-1)t$ and the period of the code will be $\alpha+t$. On the first t writes of every such period, the encoder uses the encoding map of the t-write WOM-code. In the following t writes it uses again the same code but in the reverse direction, and so on, for p times. On the $(tp+1)$-st write, no new information is written and the cell-state vector is changed to the all-zero vector, and in the remaining $\alpha+t-(tp+1)$ writes, no information is written and the cell-state vector is not changed. That completes one period.

Construction 4 *Let α,p,t be positive integers such that $\alpha \geqslant (p-1)t$. Let $\mathcal{C}_W$ be an $[n,t;M_1,\ldots,M_t]$ t-write WOM-code. For $1 \leqslant i \leqslant t$, let $\mathcal{E}_i,\mathcal{D}_i$ be its encoding, decoding map, respectively. An $(\alpha,1,p)$-constrained code $\mathcal{C}_{\alpha,1,p}$ is constructed as follows. For all $i \geqslant 1$, let $i' = i(\bmod(\alpha+t))$, $i'' = i'(\bmod 2t)$ where $1 \leqslant i' \leqslant (\alpha+t), 1 \leqslant i'' \leqslant 2t$. On the i-th write, the encoder uses the following rules:*

- *If $1 \leqslant i' \leqslant pt$ and $1 \leqslant i'' \leqslant t$, write one of $M_{i''}$ messages m using the encoding map $\mathcal{E}_{i''}(m,\boldsymbol{c})$.*
- *If $1 \leqslant i' \leqslant pt$ and $t+1 \leqslant i'' \leqslant 2t$, write one of $M_{i''-t}$ messages m using the encoding map $\overline{\mathcal{E}_{i''-t}(m,\overline{\boldsymbol{c}})}$.*
- *If $i' = pt+1$, no information is written and the cell-state vector is changed to $(0,\ldots,0)$.*
- *If $pt+2 \leqslant i' \leqslant \alpha+t$, no information is written and the cell-state vector is not changed.*

Theorem 7. *The code $\mathcal{C}_{\alpha,1,p}$ constructed in Construction 4 is an $(\alpha,1,p)$-constrained code. If the rate of the t-write WOM-code $\mathcal{C}_W$ is R_W, then the rate of $\mathcal{C}$ is $\frac{pR_W}{t+\alpha}$.*

Proof: This is similar to the proof of Theorem 6, so we present here only a sketch of the proof. In every period of $(\alpha+t)$ writes, each cell is rewrite at most p times. In particular, the first rewritten happens before the $(t+1)$-st write. After that, the cell is rewritten at most $p-1$ times until the $(tp+1)$-st write and not programmed for $\alpha+t-(tp+1)$ writes. Therefore, each cell is rewritten at most p times on $\alpha+t-(tp+1)+(tp+1)-t = \alpha$ writes. This proves the validity of the code.

If the rate of the WOM code $\mathcal{C}_W$ is R_W then $R_W pn$ bits are written during each period of $\alpha + t$ writes since the WOM-code is used p times. Hence, the rate of $\mathcal{C}_{\alpha,1,p}$ is $\frac{2R_W pn}{2(t+\alpha)n} = \frac{pR_W}{t+\alpha}$. ■

Remark 6. In Construction 4 we required that $\alpha \geqslant (p-1)t$ and in particular $t \leqslant \left\lfloor \frac{\alpha}{p-1} \right\rfloor$. If $t \geqslant \left\lceil \frac{\alpha}{p-1} \right\rceil$, we can simply use Construction 4 while taking $\alpha = (p-1)t$, i.e., the period of writes is now pt and and we construct a $((p-1)t, 1, p)$-constrained code, which is in particular also an $(\alpha, 1, p)$-constrained code. The rate of the code is R_W/t, where R_W is the rate of the WOM-code $\mathcal{C}_W$.

The next corollary shows the lower bounds on $C(\alpha, 1, p)$.

Corollary 8. *Let α, p be positive integer such that $p \leqslant \alpha$.*

$$C(\alpha,1,p) \geqslant \max\left\{ \frac{p\log_2(t+1)}{\alpha+t}, \frac{\log_2(t^*+1)}{t^*}, \frac{p}{\alpha} \right\}$$
$$t \in \mathbf{Z}^+,\ 1 \leqslant t \leqslant \lfloor \frac{\alpha}{p-1} \rfloor, t^* = \lceil \frac{\alpha}{p-1} \rceil.$$

Figure 3 shows the rates of $(\alpha, \beta = 1, p)$-constrained codes obtained by selecting the best t for each pair of (α, p). Our code construction yields rates that are approximately double those obtained from the elementary Construction 1, shown as a dashed line. These constructive lower bounds achieve approximately 78% of the corresponding upper bounds on $C(\alpha, 1, p)$.

C. Time/Space Constraint Improvement

Finally, we state without proof another general lower bound on $C(\alpha, \beta, p)$, valid for all α, β, and p.

Theorem 9. *For all α, β, p positive integers,*

$$C(\alpha,\beta,p) \geqslant \max\left\{ \frac{C(\alpha,1,p)}{\beta}, \frac{C(1,\beta,p)}{\alpha} \right\}.$$

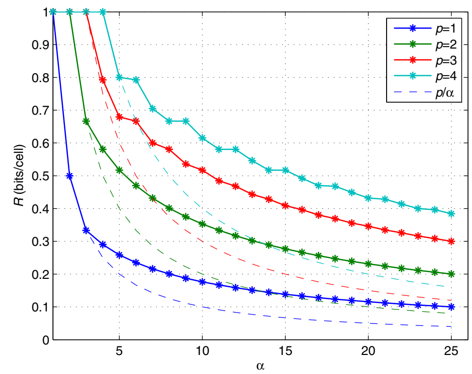
Proof: The (α, β, p)-constrained code can be constructed in two ways.

1) Let $\mathcal{C}$ be an $(1, \beta, p)$-constrained code of rate R and length n. We construct a new code $\mathcal{C}'$ with the same number of cells. New information is written to the memory on all i-th writes where $i \equiv 1 (\bmod \alpha)$, simply by using the $\left\lceil \frac{i}{\alpha} \right\rceil$-th write of the code $\mathcal{C}$. Then, the code $\mathcal{C}'$ is an (α, β, p)-constrained code and its rate is R/α. Therefore, we conclude that $C(\alpha, \beta, p) \geqslant \frac{C(1,\beta,p)}{\alpha}$.
2) Let $\mathcal{C}$ be an $(\alpha, 1, p)$-constrained code of rate R and length n. We construct a new code $\mathcal{C}'$ with $n\beta$ cells: $(c_1, c_2, \ldots, c_{n\beta})$. The code $\mathcal{C}'$ uses the same encoding and decoding maps of the code $\mathcal{C}$ while using only the n cells c_i such that $i \equiv 1 (\bmod \beta)$. Then, the code $\mathcal{C}'$ is an (α, β, p)-constrained code and its rate is R/β. Therefore, we conclude that $C(\alpha, \beta, p) \geqslant \frac{C(\alpha,1,p)}{\beta}$.

The capacity must be greater than or equal to the maximum of the two lower bounds. ■

Remark 7. For p small relative to α and β, the rate of the proposed construction is greater than the trivial one. As an example of $p = 2$, when $\alpha \geqslant 4$, $\frac{C(\alpha,1,p)}{\beta} > \frac{1}{\alpha\beta}$; when $\beta \geqslant 7$, $\frac{C(1,\beta,p)}{\alpha} > \frac{1}{\alpha\beta}$.

Fig. 3. Lower bound on $C(\alpha, 1, p)$



V. ACKNOWLEDGEMENT

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

REFERENCES

[1] R. Ahlswede and Z. Zhang, "Coding for write-efficient memory," *Inform. and Comput.*, vol. 83, no. 1, pp. 80–97, 1989.
[2] S. Braga, A. Sanasi, A. Cabrini, and G. Torelli, "Voltage-driven partial-reset multilevel programming in phase-change memories," *IEEE Trans. Electron Devices*, vol. 57, no. 10, pp. 2556–253, August 2010.
[3] G. Burr, M. Breitwisch, M. Franceschini, D. Garetto, K. Gopalakrishnan, B. Jackson, B. Kurdi, C. Lam, L. Lastras-Montaño, A. Padilla, B. Rajendran, S. Raoux, and R. Shenoy, "Phase change memory technology," *Journal of Vacuum Science and Technology*, vol. 28, no. 2, pp. 223–262, March 2010.
[4] F. Freitas and W. Wickle, "Storage-class memory: The next storage system technology," *IBM Journal of Research and Development*, vol. 52, no. 4/5, pp. 439–447, 2008.
[5] 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. Information Theory*, vol. 45, no. 1, pp. 308–313, September 1999.
[6] A. Jiang, J. Bruck, and H. Li, "Constrained codes for phase-change memories," in *Proc. IEEE Inform. Theory Workshop*, Dublin, Ireland, August-September 2010.
[7] L. Lastras-Montaño, M. Franceschini, T. Mittelholzer, J. Karidis, and M. Wegman, "On the lifetime of multilevel memories," in *Proc. IEEE Int. Symp. Inform. Theory*, Seoul, Korea, July 2009, pp. 1224–1228.
[8] B. Marcus, R. Roth, and P. 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.
[9] Z. Nagy and K. Zeger, "Asymptotic capacity of two-dimensional channels with checkerboard constraints," *IEEE Trans. Information Theory*, vol. 49, no. 9, pp. 2115–2125, September 2003.
[10] A. Pirovano, A. Redaelli, F. Pellizzer, F. Ottogalli, M. Tosi, D. Ielmini, A. Lacaita, and R. Bez, "Reliability study of phase-change nonvolatile memories," *IEEE Trans. Device and Materials Reliability*, vol. 4, no. 3, pp. 422–427, September 2004.
[11] 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.
[12] I. Tal and R. Roth, "Convex programming upper bounds on the capacity of 2-d constraints," *IEEE Trans. Information Theory*, vol. 57, no. 1, pp. 381–391, January 2011.
[13] W. Weeks and R. Blahut, "The capacity and coding gain of certain checkerboard codes," *IEEE Trans. Information Theory*, vol. 44, no. 3, pp. 1193–1203, May 1998.
[14] E. Yaakobi, S. Kayser, P. Siegel, A. Vardy, and J. Wolf, "Efficient two-write wom-codes," in *Proc. IEEE Inform. Theory Workshop*, Dublin, Ireland, August 2010.
[15] E. Zehavi and J. Wolf, "On runlength codes," *IEEE Trans. Information Theory*, vol. 34, no. 1, pp. 45–54, January 1988.