

On the Capacity of Bounded Rank Modulation for Flash Memories

Zhiying Wang

Electrical Engineering Department
California Institute of Technology
Pasadena, CA 91125, USA

Email: zhiying@paradise.caltech.edu

Anxiao (Andrew) Jiang

Computer Science Department
Texas A&M University
College Station, TX 77843, USA

Email: ajiang@cs.tamu.edu

Jehoshua Bruck

Electrical Engineering Department
California Institute of Technology
Pasadena, CA 91125, USA

Email: bruck@caltech.edu

Abstract—Rank modulation has been recently introduced as a new information representation scheme for flash memories. Given the charge levels of a group of flash cells, sorting is used to induce a permutation, which in turn represents data. Motivated by the lower sorting complexity of smaller cell groups, we consider bounded rank modulation, where a sequence of permutations of given sizes are used to represent data. We study the capacity of bounded rank modulation under the condition that permutations can overlap for higher capacity.

I. INTRODUCTION

Flash memory is an important non-volatile storage technology of wide applications. In flash memories, floating-gate cells use their charge-levels to store data [2]. For higher capacity, multi-level cells (MLCs) with an increasing number of levels are being developed. To increase a cell level, charge is injected into the cell by the Fowler-Nordheim tunneling mechanism or the hot-electron injection mechanism. This programming process is iterative to avoid over-injection. To lower any cell level, one must erase a whole cell block (typically 512K cells) and reprogram them starting at the lowest level. This asymmetric property caused by block erasure is a prominent feature of flash memories and presents a bottleneck of flash memories in terms of speed and reliability. There has been a number of recent works using the information theoretic approach to develop new storage schemes for flash memories. They include coding schemes for rewriting data [1] [4] [5] [6] [10], codes for correcting limited-magnitude errors [3], and the new rank modulation scheme for efficient and reliable cell programming and data storage [7] [8]. In this paper, we focus on and extend the rank modulation scheme.

Rank modulation is a new data representation scheme that uses the relative order of cell levels to represent data [7] [8]. Let (c_1, c_2, \dots, c_m) denote the charge levels of m cells, where each c_i (for $1 \leq i \leq m$) is an analog number and $\forall i \neq j, c_i \neq c_j$. Let $\mathcal{I}(c_1, c_2, \dots, c_m) = (a_1, a_2, \dots, a_m)$ be a function that induces from the charge levels a permutation, where for $i = 1, 2, \dots, m$, $a_i = |\{j | a_j \leq a_i, j = 1, 2, \dots, m\}|$. For example, if $m = 4$ and $(c_1, c_2, c_3, c_4) = (0.2, 0.3, 1.2, 0.5)$, then the induced permutation is $(a_1, a_2, a_3, a_4) = (1, 2, 4, 3)$. A

group of m cells can store $\log_2(m!)$ bits of information. Since rank modulation uses permutations to represent data, the charge levels can take analog values instead of discrete values, making the programming process much more robust to over-injection and the stored data more robust to asymmetric errors.

In this paper, we study the capacity of rank modulation with bounded permutation sizes. To induce a permutation from a group of cells, a sorting algorithm of complexity $O(m \log m)$ is needed. Reducing the sorting complexity is important for the efficient hardware implementation of rank modulation. To study the capacity under this constraint, we propose a discrete model. Normalize the gap between the minimum and maximum charge levels of the memory to 1, and let δ denote the minimum charge difference to distinguish two levels. Then the largest possible size for a permutation is $D = \lfloor \frac{1}{\delta} \rfloor + 1$. However, in practice the permutation size should be smaller than D not only to reduce the sorting complexity, but also to make cell programming efficiently implementable. In this paper, we let $m \leq D$ denote the given permutation size (which is also the number of cells in a group), and study the achievable capacity. Each cell level is denoted by an integer in the set $\{1, 2, \dots, D\}$. It should be noted that these D discrete numbers do not mean that in practice the charge levels are to be discrete instead of analog. They are used to derive the theoretical capacity under the considered constraints. When more constraints are introduced, the model can certainly be generalized.

An important observation is that by allowing cell groups to have overlaps (i.e., shared cells), the capacity can be improved. In this paper, we study this model, and explore the corresponding capacity. We present computational techniques and bounds for capacity and compare the capacities of different schemes.

Due to the limited space, we skipped some details in this paper. Interested readers please refer to [11].

II. BOUNDED RANK MODULATION

In this section, we define the basic concepts of bounded rank modulation. For convenience, \forall integers $a \leq b$, define $[a, b] = \{a, a + 1, \dots, b\}$.

Let m and D be integers such that $m \leq D$. A *block* is a set of m cells whose levels are from the set $[1, D]$ and are all distinct. Let (c_1, c_2, \dots, c_m) denote those m cell levels. Then by definition, $c_i \in [1, D]$ for $i \in [1, m]$ and $\forall i \neq j, c_i \neq c_j$. For convenience, we call (c_1, c_2, \dots, c_m) a *block*, too, and call $\mathcal{I}(c_1, c_2, \dots, c_m)$ the induced *permutation*. (\mathcal{I} is as defined in the previous section.) If a block B induces a permutation P , then B is called a *realization* of P . Note that a permutation may have multiple realizations. For example, if $m = 6$ and $P = (1, 4, 3, 2)$, then both $(1, 6, 4, 3)$ and $(2, 5, 4, 3)$ are realizations of P .

Let (c_1, c_2, \dots, c_n) be the levels of n cells. Let $v < m$ be an integer and for convenience, let $(n-v)/(m-v)$ be an integer as well. For $i = 1, 2, \dots, \frac{n-v}{m-v}$, let B_i denote the block $(c_{(i-1)(m-v)+1}, c_{(i-1)(m-v)+2}, \dots, c_{(i-1)(m-v)+m})$. Note that the last v cell levels of B_i are also the first v cell levels of B_{i+1} , so we say these two blocks overlap by v . We say (c_1, c_2, \dots, c_n) , or $(B_1, B_2, \dots, B_{(n-v)/(m-v)})$, is a *cell-level sequence* consisting of blocks that overlap by v . For $i = 1, 2, \dots, \frac{n-v}{m-v}$, let the m levels in B_i be all distinct and $P_i = \mathcal{I}(B_i)$. Then the sequence induces $\frac{n-v}{m-v}$ permutations $(P_1, P_2, \dots, P_{(n-v)/(m-v)})$, called the induced *permutation sequence*. And we call $(B_1, B_2, \dots, B_{(n-v)/(m-v)})$ its *realization*. Again, a permutation sequence may have multiple realizations.

Definition 1 (BOUNDED RANK MODULATION $\mathcal{C}(n, m, D, v)$) In a bounded rank modulation (BRM) code $\mathcal{C}(n, m, D, v)$, every codeword is a permutation sequence $(P_1, P_2, \dots, P_{(n-v)/(m-v)})$ that has at least one realization. Let $|\mathcal{C}(n, m, D, v)|$ denote the number of codewords in code \mathcal{C} . Then, the capacity of the code is

$$\text{cap}(\mathcal{C}) = \lim_{n \rightarrow \infty} \frac{\log |\mathcal{C}(n, m, D, v)|}{n}.$$

In general, allowing overlap between permutations can increase capacity. When there is no overlap (i.e., $v = 0$), the BRM code has capacity $\frac{\log m!}{m}$. When $v > 0$, the capacity may increase because every permutation consumes just $m - v$ cells on average.

III. BRM CODE WITH ONE OVERLAP AND CONSECUTIVE LEVELS

In this section, we study a special BRM code that allows efficient computation of its capacity. First, we present a computational method based on constrained systems. Detailed definitions are shown in [11].

Since $c_i \in [1, D]$ for $i \in [1, m]$, the BRM code is a *constrained system*. Let $G = (V, E, L)$ be a deterministic labeled graph representing $\mathcal{C}(n, m, D, v)$, where V, E and L are the state set, the edge set, and the edge labeling, respectively. $L(u, v) = l$ is denoted by $u \xrightarrow{l} v$, $l \in S_m$ (the symmetric group). If A_1, A_2, \dots, A_k are

the *adjacency matrices* of the irreducible components in G , then

$$\text{cap}(\mathcal{C}(n, m, D, v)) = \frac{\max_{1 \leq i \leq k} \log \lambda(A_i)}{m - v} \quad (1)$$

where $\lambda(A)$ is largest positive eigenvalue of A [9].

Example 2 A BRM code $\mathcal{C}(n, 2, 3, 1)$ can be represented by the deterministic graph G in Figure 1 (a). Each state represents the current cell level. $S_2 = \{12, 21\}$, $V = \{1, 2, 3\}$, and $E = \{(i, i+1) | i = 1, 2\} \cup \{(i, i-1) | i = 2, 3\}$. The labeling is $L(i, i+1) = 12, \forall i = 1, 2$ and $L(i, i-1) = 21, \forall i = 2, 3$. For example, the path along the states 1, 2, 3, and 2 is a realization of the permutation sequence $(12, 12, 21)$. The adjacency matrix of G is

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

By (1), the capacity is $\log(\lambda(A)) = 0.5$.

Notice in Example 2, every block $B_i = (c_i, c_{i+1})$ consists of two consecutive integers, i.e., $|c_i - c_{i+1}| = 1$. If we generalize this idea to arbitrary $D \geq 2$ but keep $m = 2$, and $v = 1$, we get the constrained system in Figure 1 (b), and the capacity is $\log(2 \cos(\frac{\pi}{D+1}))$ [9].

We now formally define this type of BRM code.

Definition 3 (BRM CODE WITH ONE OVERLAP AND CONSECUTIVE LEVELS $\mathcal{C}_I(n, m, D, 1)$) For the BRM code $\mathcal{C}_I(n, m, D, 1)$, every codeword $(P_1, P_2, \dots, P_{(n-1)/(m-1)})$ needs to satisfy the following additional constraint: the codeword has a realization $(B_1, B_2, \dots, B_{(n-1)/(m-1)})$ such that for $i = 1, 2, \dots, \frac{n-1}{m-1}$, the m cell levels in the block B_i form a set of m consecutive numbers. That is, if $B_i = (c'_1, c'_2, \dots, c'_m)$, then $\{c'_1, c'_2, \dots, c'_m\} = [\min_{j=1}^m c'_j, \max_{j=1}^m c'_j]$.

In a labeled graph for $\mathcal{C}_I(n, m, D, 1)$, each state corresponds to the charge level of an overlapped cell, so there are D states, $1, 2, \dots, D$. And each edge represents a permutation in a block (c'_1, \dots, c'_m) . The first (or last) digit in an edge labeling corresponds to the initial (or terminal) state of the edge. Let $(a_1, \dots, a_m) = \mathcal{I}(c'_1, \dots, c'_m)$, then $\forall k, l \in [1, m], c'_k - c'_l = a_k - a_l$. For example, the labeled graph for $\mathcal{C}_I(n, 3, 4, 1)$ is shown in Figure 1 (c).

The construction of the adjacency matrix for code $\mathcal{C}_I(n, m, D, 1)$ is presented in the following theorem.

Theorem 4 The adjacency matrix $A = (A_{ij})$ for $\mathcal{C}_I(n, m, D, 1)$ has

$$A_{ij} = (m-2)! \min\{m - |i - j|, i, j, D - i + 1, D - j + 1, D - m + 1\} \quad (2)$$

if $1 \leq |i - j| \leq m - 1$, and $A_{ij} = 0$ otherwise.

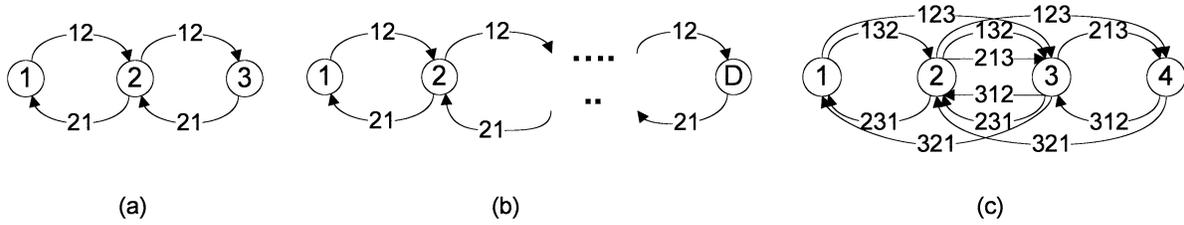


Fig. 1. Labeled graphs for \mathcal{C}_I . (a) $\mathcal{C}_I(n, 2, 3, 1)$; (b) $\mathcal{C}_I(n, 2, D, 1)$ and D is arbitrary; (c) $\mathcal{C}_I(n, 3, 4, 1)$.

Proof: A_{ij} indicates the number of permutations with $c'_1 = i, c'_m = j$. For fixed a_1 and a_m , there are $(m-2)!$ choices for (a_2, \dots, a_{m-1}) . Notice $i \rightarrow j$ only if $|a_1 - a_m| = |c'_1 - c'_m| \in [1, m-1]$. So $|\{(a_1, a_m)\}| \leq m - |i - j|$, if $|i - j| \in [1, m-1]$. And $|\{(a_1, a_m)\}| = 0$ otherwise. If $i \in [1, m]$, $\min_{1 \leq k \leq m} c'_k = c'_1 - (a_1 - 1) = i - a_1 + 1 \geq 1$, which implies $a_1 \in [1, i]$, or $|\{a_1\}| = i$. And we have similar results for other values of i . Hence, $|\{a_1\}| = \min\{i, D - i + 1, D - m + 1, m\}$. This argument also works for the terminal state j . Therefore, if $1 \leq |i - j| \leq m - 1$, then $A_{ij} = (m-2)! |\{(a_1, a_m)\}| = (m-2)! \min\{m - |i - j|, i, j, D - i + 1, D - j + 1, D - m + 1\}$. ■

The capacity of \mathcal{C}_I is $\text{cap}(\mathcal{C}_I) = \frac{\log \lambda(A)}{m-1}$. Some values of $\text{cap}(\mathcal{C}_I)$ and the capacity of the non-overlap code $\mathcal{C}(n, m, D, 0)$ (for comparison) are shown in Figure 2.

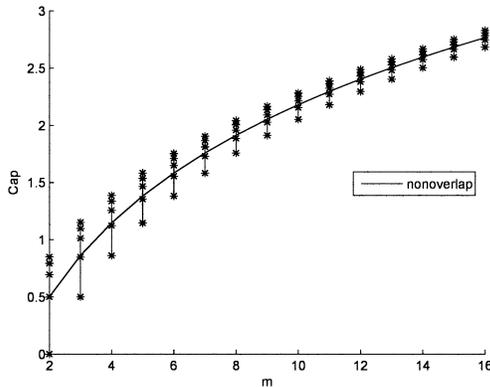


Fig. 2. Capacity for \mathcal{C}_I (stars) and for the non-overlap code (solid line). The stars in each vertical line correspond to the same permutation size m , and $D = m, m+1, \dots, m+4$ from bottom to top.

It is clear that the capacity of $\mathcal{C}_I(n, m, D, 1)$ increases with D . And if $D \rightarrow \infty$, $\text{cap}(\mathcal{C}_I(n, m, D, 1)) \rightarrow \frac{\log m!}{m-1}$, which is larger than the capacity of the non-overlap code. We now present a more general result.

Theorem 5 For any $m \geq 2$ and $D \geq m + 2$,

$$\text{cap}(\mathcal{C}_I(n, m, D, 1)) > \text{cap}(\mathcal{C}(n, m, D, 0))$$

Proof: Notice $\text{cap}(\mathcal{C}(n, m, D, 0)) = \log m! / m, \forall D \geq m$, so we need to prove $\text{cap}(\mathcal{C}_I(n, m, m+2, 1)) > \log m! / m$. When $m = 2, 3$, the theorem is trivial. When

$m \geq 4, D = m + 2$, by (2), A is

$$(m-2)! \begin{pmatrix} 0 & 1 & 1 & 1 & \dots & 1 & 0 & 0 \\ 1 & 0 & 2 & 2 & \dots & 2 & 1 & 0 \\ 1 & 2 & 0 & 3 & \dots & 3 & 2 & 1 \\ 1 & 2 & 3 & 0 & \dots & 3 & 2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 1 & 2 & 3 & 3 & \dots & 0 & 2 & 1 \\ 0 & 1 & 2 & 2 & \dots & 2 & 0 & 1 \\ 0 & 0 & 1 & 1 & \dots & 1 & 1 & 0 \end{pmatrix}_{(m+2) \times (m+2)}$$

Let $B = \frac{1}{(m-2)!} A$, I be the identity matrix and x be an indeterminate variable. $\det(B - xI) = 0$ implies $(-x - 3)^{m-3} (x^2 + x - 1) f(x) = 0$, where $f(x) = -x^3 + (3m - 8)x^2 + (7m - 10)x + 3m - 3$. Thus $\lambda(B)$ is the largest positive root of $f(x)$. It can be proven that $\lambda(B) > 3m - 6$, and $\lambda(A) > (3m - 6)(m - 2)!$. Now we are left to show $\frac{\log \lambda(A)}{m-1} > \frac{\log(3(m-2)(m-2)!)}{m-1} \geq \frac{\log m!}{m}$, which is equivalent to $\frac{3^m (m-2)^{m-1} (m-2)!}{m^{m-1} (m-1)^{m-1}} \geq 1$. Notice $(1 - \frac{1}{m})^m \geq \frac{1}{e}$, and Stirling's Approximation, thus $\frac{3^m (m-2)^m (m-2)!}{m^{m-1} (m-1)^{m-1}} \geq \frac{1}{2e} \left(\frac{3}{e}\right)^m \sqrt{2\pi(m-1)} \geq 1$. ■

IV. BRM CODE WITH ONE OVERLAP

We now consider the general BRM code with one overlap, $\mathcal{C}(n, m, D, 1)$, which does not have the additional constraint of code $\mathcal{C}_I(n, m, D, 1)$.

The cell levels of a block, $\{c'_1, \dots, c'_m\}$, can be any set Q such that $Q \subseteq \{1, 2, \dots, D\}$ and $|Q| = m$. The labeled graph H generated is not deterministic in general. However, we are able to find a deterministic graph G that is equivalent to H [9]. Here is an example.

Example 6 The labeled graph H of $\mathcal{C}(n, 2, 4, 1)$ is shown in Figure 3 (a). This is not deterministic since state 1 has 3 outgoing edges labeled 12. Let G be the deterministic representation of \mathcal{C} , then the states $V(G)$ are subsets of $V(H)$. And for $u, v \in V(G)$, $u \xrightarrow{l} v$ if $\forall j \in v, \exists i \in u$ and $i \xrightarrow{l} j$. The resulting graph G is shown in Figure 3 (b). States $\{2\}, \{3\}, \{1, 3\}$, etc., have only outgoing edges, so their capacities are 0. Therefore the irreducible component of G maximizing $\lambda(A_i)$ is as in Figure 3 (c). Hence by (1) $\text{cap}(\mathcal{C}(n, 2, 4, 1)) = \log \lambda(A_i) = 0.8791$.

In general, suppose the deterministic graph G represents $\mathcal{C}(n, 2, D, 1)$, and A_i is the adjacency matrix

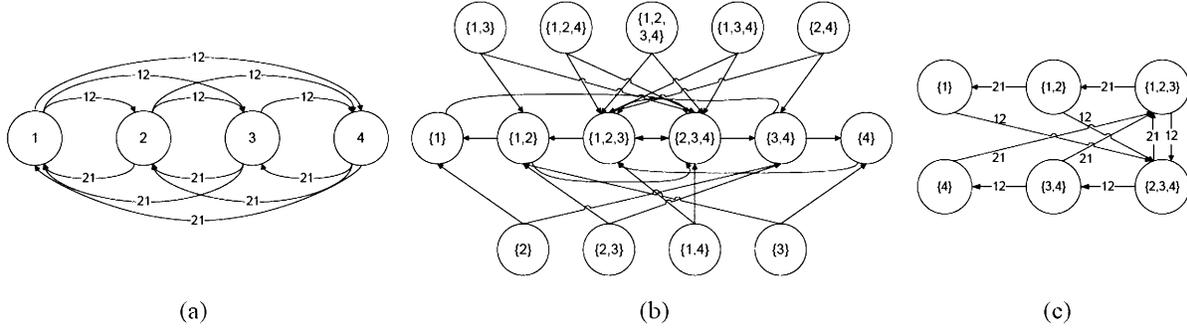


Fig. 3. Labeled graphs for $\mathcal{C}(n, 2, 4, 1)$. (a) Labeled graph; (b) deterministic graph; (c) irreducible graph.

for the irreducible component of G that has the largest eigenvalue. Then $\lambda(A_i)$ is the largest positive root of $-x^D + 2x^{D-1} - 1 = 0$. It can be seen that $\text{cap}(\mathcal{C})$ tends to 1 faster than $\text{cap}(\mathcal{C}_I)$ from the following table:

D	3	5	7	9	11
$\text{cap}(\mathcal{C}_I(n, 2, D, 1))$.5000	.7925	.8858	.9276	.9500
$\text{cap}(\mathcal{C}(n, 2, D, 1))$.6942	.9468	.9881	.9971	.9993

The construction in the above example can be naturally extended to the case $m > 2$.

Encoder/decoder for BRM codes can be constructed using sliding-block finite-state permutation encoder/decoder, cell-level encoder/decoder, and flash programming/reading. And the encoding rate can be arbitrarily close to the capacity. For example, a rate 3 : 4 block-decodable encoder can be constructed for $\mathcal{C}(n, 2, 4, 1)$. More details are shown in [11].

V. LOWER BOUND FOR CAPACITY

In this section, we present a lower bound to the capacity of the BRM code. To derive this, we first present a new form of rank modulation called the *star BRM*.

A. Star BRM

A Star BRM code uses $n + v$ cells. For convenience, let n be a multiple of $m - v$. v of these $n + v$ cells are called *anchors*, and we denote their cell levels by $(\ell_1, \ell_2, \dots, \ell_v)$. The other n cells are called *storage cells*, and we denote their cell levels by c_1, c_2, \dots, c_n . For $i = 1, 2, \dots, v$, $\ell_i \in [1, D]$; for $i = 1, 2, \dots, n$, $c_i \in [1, D]$. We call $(\ell_1, \ell_2, \dots, \ell_v, c_1, c_2, \dots, c_n)$ a *cell-level sequence*. For $i = 1, 2, \dots, \frac{n}{m-v}$, define block B_i as $(\ell_1, \ell_2, \dots, \ell_v, c_{(i-1)(m-v)+1}, c_{(i-1)(m-v)+2}, \dots, c_{i(m-v)})$. These $\frac{n}{m-v}$ blocks share the anchor cells. For $i = 1, 2, \dots, \frac{n}{m-v}$, we require that the m cell levels in B_i are all different, and let $P_i = \mathcal{I}(B_i)$. B_i is a *realization* of P_i . Again, a permutation sequence $(P_1, P_2, \dots, P_{n/(m-v)})$ may have multiple realizations.

Definition 7 (STAR BRM CODE $\mathcal{S}(n, m, D, v)$) In a Star BRM code $\mathcal{S}(n, m, D, v)$, every codeword is a permutation sequence $(P_1, P_2, \dots, P_{n/(m-v)})$ that has at

least one realization. Let $|\mathcal{S}(n, m, D, v)|$ denote the number of codewords in code \mathcal{S} . Then, the capacity is

$$\text{cap}(\mathcal{S}) = \lim_{n \rightarrow \infty} \frac{\log |\mathcal{S}(n, m, D, v)|}{n + v}.$$

To derive the capacity of Star BRM, we first show how the anchors $(\ell_1, \ell_2, \dots, \ell_v)$ affect the permutation sequences. For fixed $(\ell_1, \ell_2, \dots, \ell_v)$, define $Z(\ell_1, \ell_2, \dots, \ell_v)$ as the total number of permutations that can be induced by the cell levels $(\ell_1, \ell_2, \dots, \ell_v, c'_1, c'_2, \dots, c'_{m-v})$, where the m cell levels are all different and all belong to the set $[1, D]$. When we permute the v anchor levels, the value of $Z(\ell_1, \ell_2, \dots, \ell_v)$ remains the same. For example, when $v = 3$ and $D = 6$, $Z(2, 3, 6) = Z(3, 2, 6) = Z(6, 2, 3)$. So without loss of generality, assume $\ell_1 < \ell_2 < \dots < \ell_v$. Let $\beta(\ell_1, \ell_2, \dots, \ell_v)$ denote the number of solutions for the variables x_1, x_2, \dots, x_{v+1} such that (1) $\sum_{i=1}^{v+1} x_i = m - v$; (2) $x_1 \in [0, \ell_1 - 1]$, $x_i \in [0, \ell_i - \ell_{i-1} - 1]$ for $i \in [2, v]$, and $x_{v+1} \in [0, D - \ell_v]$.

Lemma 8. Given $D \geq m > v$, we have $Z(\ell_1, \ell_2, \dots, \ell_v) = (m - v)! \cdot \beta(\ell_1, \ell_2, \dots, \ell_v)$.

Sketch of the proof: A permutation induced by $(\ell_1, \ell_2, \dots, \ell_v, c'_1, c'_2, \dots, c'_{m-v})$ can be uniquely determined by the relative order of the $m - v$ cell levels $(c'_1, c'_2, \dots, c'_{m-v})$ and their relative values compared to $\ell_1, \ell_2, \dots, \ell_v$. ■

Lemma 9. $Z(\ell_1, \ell_2, \dots, \ell_v)$ is maximized when the numbers in the following set differ by at most one: $\{\ell_1 - 1, D - \ell_v\} \cup \{\ell_i - \ell_{i-1} - 1 \mid i = 2, 3, \dots, v\}$. (Every number in the above set is either $\lfloor \frac{D-v}{v+1} \rfloor$ or $\lceil \frac{D-v}{v+1} \rceil$.)

Please see [11] for detailed proofs of Lemma 8 and 9. Let $\ell_1^* < \ell_2^* < \dots < \ell_v^*$ be the v anchor levels that satisfy the condition in Lemma 9. and $Z^* = Z(\ell_1^*, \ell_2^*, \dots, \ell_v^*)$. Z^* can be computed using an algorithm of time complexity $O(D^2)$ (see [11]). The following theorem presents the capacity of the Star BRM.

Theorem 10. The capacity of $\mathcal{S}(n, m, D, v)$ is

$$\text{cap}(\mathcal{S}) = \frac{\log Z^*}{m - v}.$$

Sketch of the proof: $|\mathcal{S}(n, m, D, v)|$ is no less than the number of codewords induced by $(\ell_1^*, \ell_2^*, \dots, \ell_v^*, c_1, c_2, \dots, c_n)$, or $(Z^*)^{\frac{n}{m-v}}$. On the other hand, by Lemma 9, $|\mathcal{S}(n, m, D, v)|$ is no more than $(Z^*)^{\frac{n}{m-v}}$ times the number of choices for $(\ell_1, \ell_2, \dots, \ell_v)$, or $v! \binom{D}{v}$. Therefore, $\frac{\log Z^*}{m-v} \leq \text{cap}(\mathcal{S}) = \lim_{n \rightarrow \infty} \frac{\log |\mathcal{S}(n, m, D, v)|}{n+v} \leq \lim_{n \rightarrow \infty} \frac{\log(v! \binom{D}{v}) (Z^*)^{\frac{n}{m-v}}}{n} = \frac{\log Z^*}{m-v}$. So the theorem is proved. ■

The above proof leads to the following corollary.

Corollary 11 *The Star BRM code $\mathcal{S}(n, m, D, v)$ achieves its capacity even if the v anchor cell levels are fixed as $(\ell_1^*, \ell_2^*, \dots, \ell_v^*)$.*

The capacity of the Star BRM code $\mathcal{S}(n, m, D, v)$ is non-decreasing in D . However, when $D = (m - v + 1)v + (m - v)$, the capacity reaches its maximum value. Further increasing D will not increase the capacity. That is because when $D \geq (m - v + 1)v + (m - v)$, Z^* reaches its maximum value $m!/v!$.

B. Lower Bound for The Capacity of BRM

We now derive a lower bound for the capacity of the bounded rank modulation code $\mathcal{C}(n, m, D, v)$.

Theorem 12. *For the BRM code $\mathcal{C}(n, m, D, v)$, when $m \geq 2v$, its capacity*

$$\text{cap}(\mathcal{C}) \geq \frac{\log Z^* + \log v! + \log(m - 2v)!}{2(m - v)}.$$

Proof: Let $\mathcal{S}(n, m, D, v)$ be a Star BRM code such that every codeword has a realization in which the v anchors are $(\ell_1^*, \ell_2^*, \dots, \ell_v^*)$. By Corollary 11, \mathcal{S} achieves capacity.

For a codeword $s \in \mathcal{S}$, let $(\ell_1^*, \ell_2^*, \dots, \ell_v^*, c_1, c_2, \dots, c_n) = (B_1, B_2, \dots, B_{n/(m-v)})$ be its realization. For $i = 1, 2, \dots, n/(m - v)$, corresponding to block B_i , we build two blocks B_i' and B_i'' of length m as follows. Say $B_i = (\ell_1^*, \ell_2^*, \dots, \ell_v^*, c_1', c_2', \dots, c_{m-v}')$. The first v cell levels of B_i' take values from the set $\{\ell_1^*, \ell_2^*, \dots, \ell_v^*\}$ (we have $v!$ choices), and the next $m - v$ cell levels of B_i' are the same as $(c_1', c_2', \dots, c_{m-v}')$. The first v cell levels of B_i'' overlap the last v cell levels of B_i' . We pick $m - 2v \leq D - 2v$ values different from the first v and the last v cell levels of B_i' , and assign them to the next $m - 2v$ cell levels of B_i'' (we have $(m - 2v)!$ choices). The final v cell levels of B_i'' take values again from the set $\{\ell_1^*, \ell_2^*, \dots, \ell_v^*\}$. Then we construct a cell-level sequence $(B_1', B_1'', B_2', B_2'', \dots, B_{n/(m-v)}', B_{n/(m-v)}'')$, where every two adjacent blocks overlap by v . Corresponding to every codeword $s \in \mathcal{S}$, there are at least $(v!(m - 2v)!)^{\frac{n}{m-v}}$ such cell-level sequences, which we denote by Q_s . No two cell-level sequences in Q_s induce the same permutation sequence. And when $s \neq s'$, every pair of cell-level sequences

from Q_s and $Q_{s'}$, respectively, also induce different permutation sequences. Besides, every cell-level sequence constructed above induces a codeword in the code $\mathcal{C}(2n + v, m, D, v)$.

So corresponding to the $|\mathcal{S}(n, m, D, v)|$ codewords of the Star BRM code $\mathcal{S}(n, m, D, v)$, we can find at least $|\mathcal{S}(n, m, D, v)| (v!(m - 2v)!)^{\frac{n}{m-v}}$ codewords of the BRM code $\mathcal{C}(2n + v, m, D, v)$. So the capacity of code $\mathcal{C}(n, m, D, v)$ is $\text{cap}(\mathcal{C}) \geq \lim_{n \rightarrow \infty} \frac{\log |\mathcal{S}(n, m, D, v)| + (\log v! + \log(m - 2v)!) \cdot \frac{n}{m-v}}{2n+v} = \frac{\log Z^* + \log v! + \log(m - 2v)!}{2(m-v)}$. So the theorem is proved. ■

Corollary 13 *Let $\mathcal{C}(n, m, D, v)$ be a BRM code, and let $\mathcal{S}(n, m, D, v)$ be a Star BRM code. Then, when $m \geq 2v$,*

$$\text{cap}(\mathcal{C}) \geq \frac{1}{2} \cdot \text{cap}(\mathcal{S}).$$

In particular, if $v > 1$ or $m > 2v$, $\text{cap}(\mathcal{C}) > \frac{1}{2} \cdot \text{cap}(\mathcal{S})$.

Define $A_k^n = \binom{n}{k} k! = n! / (n - k)!$. Suppose $m < 2v$ and $v = k(m - v) + s$, where $k \in \mathbb{N}^+$ and $1 \leq s \leq m - v$. Let $r = m - v - s$. Define a constant $M = A_s^{m-v} (A_{m-v}^{2(m-v)-s})^{k-1} (m - v)!$. Similar to Theorem 12, we have the following lower bound for the BRM code when $m < 2v$ (see [11] for proof).

Theorem 14 *For the BRM code $\mathcal{C}(n, m, D, v)$, when $m < 2v$ and $D \geq m + r$, its capacity*

$$\text{cap}(\mathcal{C}) \geq \frac{\log(Z^* \cdot M \cdot r!)}{m + r}$$

ACKNOWLEDGMENT

This work was supported in part by the NSF CAREER Award CCF-0747415, the NSF grant ECCS-0802107, and the Caltech Lee Center for Advanced Networking.

REFERENCES

- [1] V. Bohossian, A. Jiang and J. Bruck, "Buffer codes for asymmetric multi-level memory," *Proc. IEEE ISIT*, 2007, pp. 1186-1190.
- [2] J. E. Brewer and M. Gill, *Nonvolatile memory technologies with emphasis on flash*, Chapter 12, Wiley-IEEE, 2007.
- [3] Y. Cassuto, M. Schwartz, V. Bohossian and J. Bruck, "Codes for asymmetric limited-magnitude errors with application to multi-level flash memories," *Proc. IEEE ISIT*, 2007.
- [4] H. Finucane, Z. Liu and M. Mitzenmacher, "Designing floating codes for expected performance," *Proc. of the 46th Annual Allerton Conference*, 2008.
- [5] A. Jiang, V. Bohossian and J. Bruck, "Floating codes for joint information storage in write asymmetric memories," *Proc. IEEE ISIT*, 2007, pp. 1166-1170.
- [6] A. Jiang and J. Bruck, "Joint coding for flash memory storage," *Proc. IEEE ISIT*, 2008, pp. 1741-1745.
- [7] A. Jiang, R. Mateescu, M. Schwartz and J. Bruck, "Rank modulation for flash memories," *Proc. IEEE ISIT*, 2008, pp. 1731-1735.
- [8] A. Jiang, M. Schwartz and J. Bruck, "Error-correcting codes for rank modulation," *Proc. IEEE ISIT*, 2008, pp. 1736-1740.
- [9] B. H. Marcus, R. M. Roth and P. H. Siegel, *An introduction to coding for constrained systems*, 5th Edition, 2001, available at <http://www.math.ubc.ca/~marcus/Handbook/index.html>.
- [10] E. Yaakobi, A. Vardy, P. H. Siegel and J. Wolf, "Multidimensional flash codes," *Proc. of the 46th Annual Allerton Conference*, 2008.
- [11] Z. Wang, A. Jiang, and J. Bruck, *On the capacity of bounded rank modulation for flash memories*, available at <http://www.paradise.caltech.edu/etr.html>