This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIT.2018.2859403, IEEE Transactions on Information Theory

Local Rank Modulation for Flash Memories

Michal Horovitz and Tuvi Etzion, Fellow, IEEE

***Abstract*—Local rank modulation scheme was suggested for representing information in flash memories in order to overcome drawbacks of rank modulation. For $0 < s \leq t \leq n$ with s dividing n, an (s,t,n)-LRM scheme is a local rank modulation scheme where the n cells are locally viewed cyclically through a sliding window of size t resulting in a sequence of small permutations which requires less comparisons and less distinct values. The gap between two such windows equals to s. In this work, encoding, decoding, and asymptotic enumeration of the $(1,t,n)$-LRM scheme is studied.**

***Index Terms*—Flash memory, local rank modulation.**

I. Introduction

Flash memory is a non-volatile technology that is both electrically programmable and electrically erasable. It incorporates a set of cells maintained at a set of charge levels to encode information. While raising the charge level of a cell is an easy operation, reducing the charge level requires the erasure of the whole block to which the cell belongs. For this reason charge is injected into the cell over several iterations. Such programming is slow and can cause errors since cells may be injected with extra unwanted charge. Other common errors in flash memory cells are due to charge leakage and reading disturbance that may cause charge to move from one cell to its adjacent cells. In order to overcome these problems, the novel framework of *rank modulation* was introduced in [8]. In this setup, the information is carried by the relative ranking of the cells' charge levels and not by the absolute values of the charge levels. Denote the charge level in the ith cell by c_i, $0 \leq i < n$, and hence $c = (c_0, c_1, \ldots, c_{n-1})$ is the sequence of the charge levels in the n cells. A codeword in this scheme is the permutation defined by the order of the charge levels, from the highest one to the lowest one, e.g. if $n = 5$ and $c = (3,5,2,7,10)$ then the permutation, i.e., the codeword in the rank modulation scheme, is $[5,4,2,1,3]$. This allows for more efficient programming of cells, and coding by the ranking of the cells' charge levels is more robust to charge leakage than coding by their actual values. The *push-to-the-top* operation is a basic minimal cost operation in the rank modulation scheme by which a single cell has its charge level increased such that it will be the highest of the set. Research on the rank modulation scheme since its introduction less than ten years ago has been developed in a few directions, such as error-correction [1], [2], [3], [9], [10], [13], [18], Gray codes [6], [7], [8], [16], [17], and capacity [14].

Two main metrics were studied in the literature. The first is the Kendall τ-metric [1], [2], [3], [10], [18] which corresponds to a case where the total difference in the charge levels can be bounded. The second is the infinity metric [13], [16] which models a different type of common errors, the limited-magnitude spike errors. A useful method for studying of the Kendall τ-metric is embedding the set of all permutations with the Kendall τ-metric into a different spaces and metrics, such as Lee metric and Hamming Distance [1], [10], [18]. This method is used also for constructing error-correcting codes for multi-permutations [3]. Many papers consider the single error case: design codes [1], [3], [10], explore bounds on the capacity [1], [2], and study the snake-in-the-box codes [6], [7], [15], [16], [17], which are Gray codes capable for correcting one error. In [10] the authors construct families of rank modulation codes that correct a number of errors that grows with the number of cells at varying rates.

A drawback of the rank modulation scheme is the need for a large number of comparisons when reading the induced permutation. Furthermore, n distinct charge levels are required for a group of n cells. The *local rank modulation* (LRM) scheme was suggested [5] in order to overcome these problems. In this scheme, the n cells are locally viewed through a sliding window, resulting in a sequence of permutations for a much smaller number of cells which requires fewer comparisons and fewer distinct values. For $0 < s \leq t \leq n$, where s divides n, the (s,t,n)-LRM scheme, defined in [5], [14], is a local rank modulation scheme over n physical cells, where t is the size (length) of each sliding window and s is the gap between two such windows. In this scheme the permutations are over $\{1,2,\ldots,t\}$, i.e., elements from S_t, and the push-to-the-top operation merely raises the charge level of the selected cell above those cells which are comparable with it. We say that a sequence with $\frac{n}{s}$ permutations, from S_t, is an (s,t,n)-LRM scheme *realizable* if it can be demodulated to a sequence of charges in n cells under the (s,t,n)-LRM scheme. Except for the degenerate case where $s = t = n$,

Michal Horovitz is with the Department of Computer Science, Tel-Hai College, and The Galilee Research Institute - Migal, Upper Galilee — Israel (email: horovitzmic@telhai.ac.il). This work is part of her Ph.D. thesis performed at the Department of Computer Science, Technion — Israel Institute of Technology.

Tuvi Etzion is with the Computer Science Department, Technion– Israel Institute of Technology, Haifa 3200003, Israel (e-mail: etzion@cs.technion.ac.il).

This work was supported in part by the United States — Israel Binational Science Foundation (BSF), Jerusalem, Israel, under Grant 2012016. This paper was presented in part in the Information Theory Workshop, Hobart, Tasmania, Australia, November 2–5, 2014

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIT.2018.2859403, IEEE Transactions on Information Theory

not every sequence is realizable.

In [14] bounded LRM codes were defined and studied. In this setup, the charge levels in each window are taken from $\{1, ..., D\}$ for some $D \geq t$. The authors mainly study the $(t-1, t, n)$-LRM.

The $(1, 2, n)$-LRM scheme was defined in [5] in order to get the simplest hardware implementation. All demodulated sequences of permutations in this scheme are realizable, except for the two sequences of permutations in which all permutations are the same. Hence, $2^n - 2$ sequences of permutations are realizable in this scheme. But, since only two permutations are used in this scheme, it follows that this scheme is relatively very weak, as the total number of possible codewords is relatively small. Therefore, we are interested in the $(1, t, n)$-LRM schemes for $t \geq 3$, and this is the motivation for this work.

In this paper we focus on the $(1, t, n)$-LRM schemes for $t \geq 3$, and suggest a demodulation method for these schemes. The $(1, t, n)$-LRM scheme is a local rank modulation scheme over n physical cells, where the size of each sliding window is t, and each cell starts a new window. Since the size of a sliding window is t, demodulated sequences of permutations in this scheme contain $t!$ permutations. Therefore, we need $t!$ symbols to represent the demodulated sequences of permutations.

Let $\theta = (\theta_1, \theta_2, \ldots, \theta_{t!})$ be an order of the $t!$ permutations from S_t, and $\Sigma = \{1, 2, \ldots, t!\}$ be an alphabet where i represents the permutation s_i. A sequence $\alpha = (\alpha_0, \alpha_1, \ldots, \alpha_{n-1})$ over the alphabet Σ is called a *base-word* in the $(1, t, n)$-LRM scheme, and it is *realizable*, if there exists a sequence of *charge levels* $c = (c_0, c_1, \ldots, c_{n-1})$, such that for each i, $0 \leq i < n$, α_i represents the permutation induced by $c_i, c_{i+1}, \ldots, c_{i+t-1}$, where indices are taken modulo n.

In this paper a mapping method, in which each base-word $\alpha = (\alpha_0, \alpha_1, \ldots, \alpha_{n-1})$ over the alphabet of size $t!$, is mapped to a *codeword* $g = (g_0, g_1, \ldots, g_{n-1})$ over an alphabet of size t, will be presented. A codeword is called *legal* if there exists a realizable base-word which is mapped to it. We have to make sure that two distinct realizable base-words are mapped into two distinct legal codewords. Note again, that the indices in the base-words, charge levels, and the codewords are taken modulo n.

Let M_t be the number of legal codewords in the $(1, t, n)$-LRM scheme. Since a symbol in a codeword is from an alphabet with t letters, it follows that $M_t \leq t^n$. But, this upper bound is not tight since there exist illegal codewords. We prove in this paper that this upper bound on M_t is asymptotically tight, i.e. $\lim_{n\to\infty} \frac{M_t}{t^n} = 1$.

Our setup assumes that the words are cyclic, i.e. there is wrap-around, a convention that was also assumed in [5]. A cyclic setup reduces the number of possible codewords in the sense that some base-words are not realizable. But, as it will be proved, asymptotically the number of codewords is not reduced, i.e. $\lim_{n\to\infty} \frac{M_t}{t^n} = 1$, when n cells with a window of size t are used. On the other hand, if we consider a noncyclic setup then with n cells there exist only $n - t + 1$ distinct windows of length t and hence a related code has at most t^{n-t+1} codewords. Therefore, the cyclic setup increases the number of codewords compared to the noncyclic setup in a factor of about t^{t-1}. This implies a considerable advantage (at least theoretically) for the cyclic setup on the noncyclic one. Moreover, the cyclic setup is more symmetric (with respect to the different cells) which makes it simple to handle (encoding/decoding), more appealing, and more interesting. The only advantage of the noncyclic setup is that all the codewords are legal. This make this setup very simple, but with a factor of about t^{t-1} less codewords.

Another possible drawback of the local rank modulation is a potential of too many charge levels. There are a few ways to overcome this problem. The most simple one is to have n not larger than the number of charge levels. It should be emphasis that by using this solution, the local rank modulation has no advantage on the rank modulation in the number of required distinct charge levels, but LRM is still better in sense of having less comparisons when the data is read. It should be noted also that the technology is improving all the time, and with the advancing time the number of possible charge level is increased. A large number of charge levels can be also achieved and solved by using a careful programming. Such a careful programming can reduce the gaps between consecutive charge levels. This is a natural topic for future research. Hence, advance in both hardware and software can achieve a large number of charge levels [12]. Moreover, it can be shown that the number of codewords with high charge levels is relatively small. Hence, the related codewords can be removed and be neglected, but this will cause a much more difficult analysis.

The rest of this paper is organized as follows. The encoding and decoding of the $(1, 3, n)$-LRM scheme is presented in Section II. Enumeration technique for the $(1, t, n)$-LRM scheme, $t \geq 3$, is given in Section III. In Section IV conclusion and problems for future research are presented.

II. The $(1, 3, n)$-LRM scheme

In the $(1, 3, n)$-LRM scheme the size of each sliding window is 3. Therefore, an alphabet of size 3! is required to represent the demodulated sequences of permutations.

$$\begin{array}{ll} \theta_1 = [1,2,3] & \theta_2 = [1,3,2] \\ \theta_3 = [2,1,3] & \theta_4 = [3,1,2] \\ \theta_5 = [2,3,1] & \theta_6 = [3,2,1] \end{array}$$

The alphabet of the base-words is $\Sigma = \{1, 2, \ldots, 6\}$, where the symbol ℓ represents the permutation θ_ℓ. Let $\alpha = (\alpha_0, \alpha_1, \ldots, \alpha_{n-1})$ be a base-word. Note that the last two cells which determine α_i, $0 \le i < n$, are the first two cells which determine α_{i+1}, i.e., the permutation related to α_{i+1} is obtained from α_i by the following way. The symbol 1 in the permutation related to α_i is omitted, the symbols $2, 3$ in the permutation are replaced with $1, 2$, respectively, and a new symbol 3 is inserted before $1, 2$, between them, or after both of them, depending on the new charge level c_{i+3} compared to c_{i+1} and c_{i+2}. Therefore, given α_i, there are exactly 3 options for α_{i+1}.

Let $\Sigma^1 = \{1, 3, 5\}$ and $\Sigma^2 = \{2, 4, 6\}$ be a partition of Σ into the even and the odd symbols, respectively. Note that for each Σ^i, $i \in \{1, 2\}$, the permutations related to the symbols in Σ^i agree on the order of cells 2 and 3. Therefore, they also agree on the three possibilities of their succeeding permutation. Denote the set of symbols of these succeeding permutations by $\tilde{\Sigma}^i$. It is readily verified that $\tilde{\Sigma}^1 = \{1, 2, 4\}$ and $\tilde{\Sigma}^2 = \{3, 5, 6\}$.

The base-word α is mapped to a codeword $g = (g_0, g_1, \ldots, g_{n-1})$ over the alphabet $\{0, 1, 2\}$. Given the charge levels c_i, c_{i+1}, c_{i+2}, the permutation α_i is uniquely determined. If we are given now also the charge level c_{i+3}, then its rank among c_{i+1}, c_{i+2} uniquely determines g_{i+1}. Therefore, α_{i+1} can be deduced from α_i and g_{i+1} instead of $c_{i+1}, c_{i+2}, c_{i+3}$. The relations between α_{i-1}, α_i, and g_i are presented in Table I. This table induces a mapping from the realizable base-words to the codewords. As mentioned before, given α_{i-1}, there are three options for α_i. In all these options the sub-permutation of $\{1, 2\}$ is the same, and the difference is the index of the symbol 3 in the permutation related to α_i. Thus, g_i represents the index of the symbol 3 in this permutation and it is equal to the number of symbols which are to the right of the symbol 3 in the permutation related to α_i. In other words, g_i represents the relation between c_{i+2}, the charge level in cell $i + 2$, and the charge levels in the two cells which proceed it, i.e., c_i and c_{i+1}.

$\alpha_{i-1} \in \Sigma^1$	$\alpha_i = 1$	$\alpha_i = 2$	$\alpha_i = 4$
$\alpha_{i-1} \in \Sigma^2$	$\alpha_i = 3$	$\alpha_i = 5$	$\alpha_i = 6$
	$g_i = 0$	$g_i = 1$	$g_i = 2$

TABLE I: The encoding key of the $(1, 3, n)$-LRM scheme.

Note that there might exist non-realizable base-words which are mapped to codewords by this method. A base-word α, which can be mapped to a codeword by this method, satisfies the dependencies between α_i and α_{i+1} for all i, but it can still be non-realizable. The n cells are viewed cyclically, i.e., the charge levels of the last two cells, c_{n-2} and c_{n-1}, are compared with the charge level in the first cell, c_0, to determine α_{n-2}. The same works for the three charge levels c_{n-1}, c_0, and c_1 to determine α_{n-1}. Therefore, there might exists a non-realizable dependency between the charge levels in the last two cells and the charge levels in the first two cells. Such a non-realizable base-word will be called a *cyclically non-realizable* base-word.

Example 1. *The following base-words are cyclically non-realizable. Recall, that a codeword is called legal if there exists a realizable base-word which is mapped to it.*

- $(\underbrace{6, 6, \ldots, 6}_{n\ times})$ *- the charge levels are increased cyclically, which is impossible. This base-word is mapped to the illegal codeword* $(\underbrace{2, 2, \ldots, 2}_{n\ times})$.
- $(\underbrace{2, 5, 2, 5 \ldots, 2, 5}_{n/2\ times})$ *where* n *is even - the charge level of each cell is between the charge levels of the two cells which proceed it, where the charge levels are taken cyclically. This base-word is mapped to the illegal codeword* $(\underbrace{1, 1, \ldots, 1}_{n\ times})$.
- $(\underbrace{1, 1, \ldots, 1}_{n-2\ times}, 2, 3)$ *- the prefix* $(\underbrace{1, 1, \ldots, 1}_{n-2\ times})$ *means that the charge levels always decrease, and therefore* $c_{n-1} < c_{n-2} < c_1 < c_0$. *Hence, in this case the only possible permutations for* α_{n-2} *and* α_{n-1} *are* $[3, 1, 2] = \theta_4$ *and* $[2, 3, 1] = \theta_5$*, respectively. That is, the only realizable base-word which completes the prefix* $(\underbrace{1, 1, \ldots, 1}_{n-2\ times})$ *is* $(\underbrace{1, 1, \ldots, 1}_{n-2\ times}, 4, 5)$. *Thus,* $(\underbrace{1, 1, \ldots, 1}_{n-2\ times}, 2, 3)$ *is a non-realizable base-word, which is mapped by Table I to the illegal codeword* $(\underbrace{0, 0, \ldots, 0}_{n-2\ times}, 1, 0)$. *The realizable base-word* $(\underbrace{1, 1, \ldots, 1}_{n-2\ times}, 4, 5)$ *is mapped to the legal codeword* $(\underbrace{0, 0, \ldots, 0}_{n-2\ times}, 2, 1)$.

Theorem 1. *Table I provides a one-to-one mapping between the realizable base-words and the legal codewords.*

Proof. Obviously, each base-word is mapped to exactly one codeword since the rules to determine a codeword are deterministic and unique. Now, we prove that the other direction is also true, i.e. given a legal codeword g, there is a unique base-word which is mapped to g. By Example 1, $(\underbrace{1, 1, \ldots, 1}_{n\ times})$ is an illegal codeword. Hence, given a legal codeword $g = (g_0, g_1, \ldots, g_{n-1})$, there exists $0 \le j < n$, such that $g_i \in \{0, 2\}$. If $g_j = 0$ then by Table I we have that $\alpha_j \in \{1, 3\}$, i.e., α_j is odd. Therefore, given g_{j+1}, the permutation α_{j+1} is determined by an entry in the first row of Table I, where the column is chosen by the value of g_{j+1}. Similarly, if $g_j = 2$ then $\alpha_j \in \{4, 6\}$, i.e., α_j is even. Hence, α_{j+1} is determined by an entry in the second row of Table I, where the column is chosen by the value of g_{j+1}. Now, it is easy to determine the symbols of the base-word

$\alpha_{j+2}, \alpha_{j+3}, \ldots, \alpha_{j+n-1}, \alpha_{j+n} = \alpha_j$ one by one from the rules given in Table I in this cyclic order. □

Theorem 1 implies a decoding algorithm for a codeword of length n in the $(1,3,n)$-LRM scheme. Given a codeword g it produces a base-word α of length n which implies the rankings between the n charge levels. Algorithm 1 presents the formal steps of the decoding.

Algorithm 1 : Decoding for the $(1,3,n)$-LRM scheme

Input: A codeword $g \in [3]^n$.
Output: A base-word $\alpha \in [6]^n$.
Let $T1(row, col)$, $row = 1, 2$ and $col = 0, 1, 2$, be the values for Σ^r and $g_i = col$ in Table I.
if $g = 1^n$ **then return** NIL $\{g$ is not legal$\}$
Let $j \in [n]$ such that $g_j \neq 1$.
if $g_j = 0$ **then** $start\_row \leftarrow 1$ **else** $start\_row \leftarrow 2$
$row_j \leftarrow start\_row$
for $k := 0 \ldots (n-1)$ **do**
  $col_{j+k} \leftarrow g_{j+k+1}$ {all the indices are taken modulo n}
  $\alpha_{j+k+1} \leftarrow T1(row_{j+k}, col_{j+k})$.
  {the rows are indicated by α_{j+k} and the columns by g_{j+k+1}}
  if $\alpha_{j+k+1} \in \Sigma^1$ **then** $row_{j+k+1} \leftarrow 1$ **else** $row_{j+k+1} \leftarrow 2$
end for
if $row_j \neq start\_row$ **then return** NIL $\{g$ is not legal$\}$
return $\alpha = (\alpha_0, \ldots, \alpha_{n-1})$

Note that decoding a given codeword g to a base-word α does not guarantee that g is legal. For some illegal codewords the decoding procedure fails, while for the others it succeeds without a notification about the illegality of the input g. Let j be the starting point of the decoding algorithm as described in the proof of Theorem 1 and in Algorithm 1. At the first step of the algorithm, α_i has two options ($\{1,3\}$ if $g_j = 0$ or $\{4,6\}$ if $g_j = 2$, as implied by Table I). At the last step, if α_j is not equal to one of these two optional initial values, which was chosen in the first step, then we conclude that the given codeword is illegal. However, the algorithm may decode some cyclically non-realizable base-words without realizing that it is an illegal codeword. For example, the procedure decodes the cyclically non-realizable base-word $\alpha = (\underbrace{1,1,\ldots,1}_{n\ times})$ from the illegal codeword $g = (\underbrace{0,0,\ldots,0}_{n\ times})$. Therefore, given such a codeword g, it would be interesting to decide efficiently whether it is legal or not. First, we apply the decoding algorithm to obtain a base-word α which corresponds to g. If the decoding algorithm fails, then g is an illegal codeword. However, the decoding algorithm might produce a cyclically non-realizable base-word α. Note, that by the decoding algorithm, the dependencies between α_i and α_{i+1} are preserved for all i. Thus, the only case in which α is non-realizable is related to the dependencies of the first two charge levels and the last two charge levels. These dependencies are implied by considering the consecutive permutations from α_0, α_1, α_2, and so on up to α_{n-4} and α_{n-3}. These dependencies can be inconsistent when we continue and consider the dependencies of the charge levels implied by the consecutive permutations α_{n-2} and α_{n-1}, i.e. α is cyclically non-realizable base-word. Thus, the question is how to indicate that a base-word is cyclically non-realizable. This question will be considered in the next section after a new concept of states will be defined. The formal steps to decide if a codeword g is legal by a decision if the related base-word α is cyclically realizable, are presented in Algorithm 3.

III. The $(1,t,n)$-LRM Scheme for $t \geq 3$

In this section we will consider the enumeration of the number of the legal codewords in the $(1,t,n)$-LRM Scheme, $t \geq 3$. The ideas will be described in details in this section, where the examples will be given for $t = 3$. It should be emphasized that for other concepts, some generalizations from the $(1,3,n)$-LRM scheme, are more complicated for $t = 4$ and become impractical as t increases. One of the concepts which are presented in this section are the states which also help to determine non-realizable base-words for the $(1,3,n)$-LRM scheme.

In the $(1,t,n)$-LRM scheme the size of each sliding window is t. Therefore, to present the demodulated sequences of permutations, the alphabet of the base-words is of size $t!$. The n charge levels form a sequence $c = (c_0, c_1, \ldots, c_{n-1})$. Given t consecutive charge levels, $c_i, c_{i+1}, \ldots, c_{i+t-1}$, the corresponding permutation α_i, from S_t, is uniquely determined by the order of these t consecutive charge levels. Therefore, the n charge levels define a sequence of permutations $\alpha = (\alpha_0, \alpha_1, \ldots, \alpha_{n-1})$. The position of the symbol t in the permutation α_i determines the value of g_i, i.e., $g_i = j$, $0 \leq j < t$, if t is in position $t - j$ in the permutation. In other words, g_i is the ranking of c_{i+t-1} among $c_i, c_{i+1}, \ldots, c_{i+t-1}$, i.e. $g_i = 0$ if c_{i+t-1} is the lowest charge level, $g_i = 1$ if only one charge level is below c_{i+t-1}, and so on, where finally $g_i = t-1$ if c_{i+t-1} is the highest charge level. The consecutive values g_0, g_1, etc. define the codeword $g = (g_0, g_1, \ldots, g_{n-1})$. This means that given the last $t-1$ charge levels $c_{i+1}, c_{t+2}, \ldots, c_{i+t-1}$, a new charge level c_{i+t} combined with these $t-1$ charge levels, define the permutation α_{i+1} and the new symbol g_{i+1} in the codeword g. Therefore, the base-word α defined by the charge levels' sequence c, uniquely determines the related codeword g. Clearly, given a permutation α_{i-1}, not all the $t!$ permutations of S_t can follow α_{i-1} to serve as α_i. Only t permutations can be used for α_i based on α_{i-1}, including α_0 which follows α_{n-1}. The formal steps to produce a legal codeword g from a realizable base-word α are presented in Algorithm 2.

Algorithm 2 : Encoding algorithm from a base-word to a codeword

Input: A realizable base-word $\alpha = (\alpha_0, \ldots, \alpha_{n-1}) \in [t!]^n$).
Output: A codeword $g \in [t]^n$.
for $k := 0 \ldots (n-1)$ **do**
$\pi \leftarrow \theta_{\alpha_k}$ ($\pi \in S_t$ is the permutation represented by α_k)
$m \leftarrow$ the position of t in π ($1 \leq m \leq t$)
$g_k \leftarrow t - m$.
end for
return $g = (g_0, \ldots, g_{n-1})$

Recall that a base-word α might not be realizable, even if it meets the dependencies between α_{i-1} and α_i. This might happen if there is no possible sequence of charge levels that can be demodulated from α due to the dependencies between the first $t-1$ charge levels and the last $t-1$ charge levels. If α_i can follow α_{i-1} for all i, then the base-word α can be mapped to a codeword g, but g might be illegal since the base-word α is not realizable by a sequence of charge levels. Given the suggested mapping between the base-words and the codewords, we are mainly interested in three related questions concerning the legal codewords of the $(1,t,n)$-LRM scheme:

1) Given a legal codeword g over the alphabet $\{0,1,\ldots,t-1\}$, present an efficient method to find the base-word α mapped to g.
2) Given a codeword g over the alphabet $\{0,1,\ldots,t-1\}$, present an efficient method to decide whether g is legal.
3) Find the number of legal codewords in the $(1,t,n)$-LRM scheme.

The rest of this section will be devoted to solve some of these questions.

To obtain the original base-word from the given codeword would be easy if for some i, α_i is given or known (in fact the permutation related to $t-1$ consecutive cells is sufficient to figure out the entire base-word from a known codeword, either legal or illegal). If no such permutation is known then the task becomes more complicated and we have to analyse the codeword based only on the mapping from the base-words to the codewords.

To enumerate the number of legal codewords in the $(1,t,n)$-LRM scheme, $t \geq 3$, we need another concept which describes the permutation defined by the current last $t-1$ charge levels $c_{i-t+2}, c_{i-t+3}, \ldots, c_i$ and the rank of each one of them among the first $t-1$ charge levels $c_0, c_1, \ldots, c_{t-2}$.

Given a prefix of a codeword $(g_0, g_1, \ldots, g_{i-t+1})$, $2t-3 \leq i < n$, obtained by the unknown charge levels $c_0, c_1, \ldots, c_i$, the ranking among the charge levels in the jth cell, c_j, $i-t+2 \leq j \leq i$, and the first $t-1$ cells, $c_0, c_1, \ldots, c_{t-2}$, might have a few options (at most t). These options will be denoted by 0, 1, up to $t-1$, where 0 represents that c_i is lower than $c_0, c_1, \ldots, c_{t-2}$, 1 represents that c_i is higher than exactly one of them, and so on. For each i, $2t-3 \leq i < n$, consider the following two properties regarding $c_{i-t+2}, c_{i-t+3}, \ldots, c_i$:

(Q.1) the permutation π_i induced by $c_{i-t+2}, c_{i-t+3}, \ldots, c_i$ ($(t-1)!$ possible permutations).
(Q.2) the set of all possible $(t-1)$-tuples of rankings of the charge level c_j, for each j, $i-t+2 \leq j \leq i$, among the charge levels $c_0, c_1, \ldots, c_{t-2}$.

The elements of the set defined in (Q.2) will be denoted by $(t-1)$-tuples $(x_{i-t+2}, x_{i-t+3}, \ldots, x_i)$, $x_j \in \{0,1,\ldots,t-1\}$, $i-t+2 \leq j \leq i$, where x_j represents the ranking of the charge level c_j among the charge levels $c_0, c_1, \ldots, c_{t-2}$. Note, that for a given permutation defined by (Q.1), not all the t^{t-1} possible $(t-1)$-tuples can be obtained. We call a pair defined by the permutation of (Q.1) and the set of $(t-1)$-tuples defined by (Q.2) a *state*. The state at index i (for $c_{i-t+2}, c_{i-t+3}, \ldots, c_i$) will be denoted by P_i. For the computation of the states, only the codeword g is known, while neither the charge levels nor the permutations defined by them, from which it was computed, are known. Lets denote by π_i, $t-2 \leq i < n$, the permutation defined by $c_{i-t+2}, \ldots c_{i-1}, c_i$.

Lemma 1. *A maximum of $\binom{2t-2}{t-1}$ possible $(t-1)$-tuples can be obtained in (Q.2) for a given state.*

Proof. A state P_i is first identified by the permutation π_i defined by $c_{i-t+2}, c_{i-t+3}, \ldots, c_i$ (see (Q.1)). The highest rank among $c_{i-t+2}, c_{i-t+3}, \ldots, c_i$ can be ranked in t different ways among $c_0, c_1, \ldots, c_{t-2}$. If it has rank ℓ then the next highest rank among $c_{i-t+2}, c_{i-t+3}, \ldots, c_i$ can be ranked in ℓ different ways among $c_0, c_1, \ldots, c_{t-2}$. If it has rank m then the next highest rank among $c_{i-t+2}, c_{i-t+3}, \ldots, c_i$ can be ranked in m different ways among $c_0, c_1, \ldots, c_{t-2}$. We continue in the same manner, and hence the maximum number of possibilities in (Q.2) is exactly the number of possible $(t-1)$-tuples $(b_1, b_2, \ldots, b_{t-1})$ over $\{1,2,\ldots,t\}$ such that $b_{j+1} \leq b_j$, $1 \leq j \leq t-2$. By using simple enumeration combinatorial arguments we have that the number of such $(t-1)$-tuples is $\binom{2t-2}{t-1}$. ∎

Example 2. *For the $(1,3,n)$-LRM scheme, assume that the prefix of the codeword is $g' = (g_0, g_1, \ldots, g_{n-3}) = \underbrace{(2,2,\ldots,2)}_{n-2 \text{ times}}$, i.e. the charge levels are increased between any two consecutive charge levels (with a possible exception between c_0 and c_1 and a cyclic exception since $c_{n-1} > c_0$), that is, c_0 or c_1 are the smallest charge levels, c_{n-2} and c_{n-1} are the largest, and $c_{n-2} < c_{n-1}$. Hence, $\pi_{n-1} = [2,1]$, and both c_{n-2} and c_{n-1} are larger*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIT.2018.2859403, IEEE Transactions on Information Theory

than c_0 and c_1. Therefore, $P_{n-1} = ([2,1], \{(2,2)\})$. There are two possible base-words depending on the ranking between c_0 and c_1. If $c_1 < c_0$, i.e. $\pi_1 = [1,2]$, then the base-word is $\alpha = (\underbrace{6,6,\ldots,6}_{n-2 \ times},3,2)$ and the codeword is $g = (\underbrace{2,2,\ldots,2}_{n-2 \ times},0,1)$. If $c_1 > c_0$, i.e. $\pi_1 = [2,1]$, then the base-word is $\alpha = (4,\underbrace{6,6,\ldots,6}_{n-3 \ times},3,1)$ and the codeword is $g = (\underbrace{2,2,\ldots,2}_{n-2 \ times},0,0)$.

Example 3. *For the $(1,3,n)$-LRM scheme, assume that the prefix of the codeword is $g' = (\underbrace{1,1,\ldots,1}_{i-3 \ times},0,2)$ which implies that for all j, $2 \le j \le i-2$, the charge level c_j is between c_{j-1} and c_{j-2}. Therefore, with a simple induction on j, we conclude that c_{i-3} and c_{i-2} are between c_0 and c_1. Thus, $g_{i-3} = 0$ implies that c_{i-1} might be between c_0 and c_1 or smaller than both of them, and $g_{i-2} = 2$ implies that c_i might be between c_0 and c_1 or larger than both of them. We conclude that $P_i = ([2,1], \{(0,1),(0,2),(1,1),(1,2)\})$.*

Recall that if the ranking between the charge levels $c_{i-t+1}, c_{i-t+2}, \ldots, c_i$ is known then g_{i-t+1} can be computed based on the ranking of the charge level c_i among the $t-1$ preceding charge levels $c_{i-t+1}, c_{i-t+2}, \ldots, c_{i-1}$. Recall also that the state P_i is defined by two properties (Q.1) and (Q.2). By (Q.1) we know the ranking between the charge levels $c_{i-t+2}, c_{i-t+3}, \ldots, c_i$ and by (Q.2) we know the ranking of each one of these last $t-1$ charge levels among the first $t-1$ charge levels. The state P_i is now determined based on these two properties.

Lemma 2. *If P_i and g_{i-t+2}, for some $2t-3 \le i < n-1$, are given, then P_{i+1} is uniquely determined.*

Proof. P_i is characterized by the permutation π_i in (Q.1) and the $(t-1)$-tuples in (Q.2). The permutation π_i is defined by the sequence of charge levels $c' = (c_{i-t+2}, c_{i-t+3}, \ldots, c_i)$ and g_{i-t+2} defines the ranking of c_{i+1} among the set of charge levels in c'. Hence, the permutation defined by $c'' = (c_{i-t+3}, c_{i-t+4}, \ldots, c_{i+1})$ is uniquely determined and property (Q.1) for P_{i+1} is well defined.

Let $y = (y_0, y_1, \ldots, y_{t-2})$ be a possible $(t-1)$-tuple in (Q.2) of P_{i+1}, that is, y represents a possible ranking of each charge level in c'' among $(c_0, c_1, \ldots, c_{t-2})$, where y_j, $0 \le j \le t-2$, represents the ranking of $c_{i-t+3+j}$ among the charge levels of the first $t-1$ cells. Then, there exists a possible $(t-1)$-tuple in (Q.2) of P_i, $x = (x_0, x_1, \ldots, x_{t-2})$, where $x_j = y_{j-1}$, $0 < j \le t-2$, since x_j represents a ranking of $c_{i-t+2+j}$ among the charge levels of the first $t-1$ cells in one possible $(t-1)$-tuple in (Q.2) of P_i.

Thus, to complete the proof, it is sufficient to show, that π_i, π_{i+1}, and x, determine all possibilities for y_{t-2}. Recall, that y_{t-2} relates to the ranking possibilities of c_{i+1} among the first $t-1$ charge levels.

The permutation π_{i+1} determines the ranking of c_{i+1} among $c' = (c_{i-t+3}, c_{i-t+4}, \ldots, c_{i+1})$. Denote by c_{j_1} and c_{j_2} the two charge levels in c' which are adjacent to c_{i+1} in their value, where $c_{j_1} < c_{i+1} < c_{j_2}$. (that is, $j_1-(i+1)+t-1$ and $j_2-(i+1)+t-1$ are adjacent to $t-1$ in π_{i+1}.) Note that if $t-1$ is the first or the last symbol in π_{i+1} then only one of j_1, j_2 exists. Then, $x_{j_1-i+t-2}$ and $x_{j_2-i+t-2}$ represent a possible ranking of c_{j_1} and c_{j_2} among the charge levels of the first $t-1$ cells. These possible rankings of c_{j_1} and c_{j_2}, with the only constraint on c_{i+1} to be between c_{j_1} and c_{j_2}, determine the possible rankings of c_{i+1} among the first $t-1$ cells, and therefore determine the possible values for y_{t-2}. □

Corollary 1. *If $P_i = P_j$ for some $2t-3 \le i < j < n-1$ and $g_{i-t+2} = g_{j-t+2}$ then $P_{i+1} = P_{j+1}$.*

A state which has all $\binom{2t-2}{t-1}$ possible $(t-1)$-tuples in property (Q.2) will be called a *complete state*. In other words, P_i is a complete state if each one of the ranks $c_{i-t+2}, c_{i-t+3}, \ldots, c_i$ is independent of the ranks $c_0, c_1, \ldots, c_{t-2}$. An immediate consequence is that

Lemma 3. *If P_i is a complete state then also P_{i+1} is a complete state.*

Lemma 4. *In the $(1,t,n)$-LRM there are $(t-1)!$ complete states.*

Proof. Each permutation on the ranks $c_{i-t+2}, c_{i-t+3}, \ldots, c_i$ induces $\binom{2t-2}{t-1}$ possible rankings among $c_0, c_1, \ldots, c_{t-2}$ as explained in the proof of Lemma 1. □

Example 4. *The two complete states in the $(1,3,n)$-LRM scheme are:*

1) $\mathbb{S}_1$: $([1,2], \{(0,0),(1,0),(1,1),(2,0),(2,1),(2,2)\})$.
2) $\mathbb{S}_2$: $([2,1], \{(0,0),(0,1),(0,2),(1,1),(1,2),(2,2)\})$.

We are only interested in complete states since non-complete states might lead to a relatively small number of legal codewords. The non-complete states and their related codewords will be omitted in the computations of the number of legal codewords which follows.

Given π_{t-2}, the permutation defined by the first $t-1$ charge levels and $g' = (g_0, g_1, \ldots, g_{n-4}, g_{n-t})$, we have to determine the sub-base-word $(\alpha_0, \alpha_1, \ldots, \alpha_{n-t})$, of a realizable base-word which corresponds to π_{t-2} and g'. This sub-base-word is determined unambiguously. But, there are a few possible assignments for $\alpha_{n-t+1}, \alpha_{n-t+2}, \ldots, \alpha_{n-1}$, which correspond to possible assignments for $g_{n-t+1}, g_{n-t+2}, \ldots, g_{n-1}$. These assignments are determined by the state P_{n-1} and the permutation π_{t-2}. Each assignment provides a distinct realizable base-word which is represented by the state P_{n-1} and the permutation π_{t-2}.

Example 5. *For the $(1,3,n)$-LRM scheme, given g_{i-1}, the succeeding state P_{i+1} of a state P_i which is a complete state, is given in Table II. Recall that if P_i is a complete state then also P_{i+1} is a complete state (see Lemma 3).*

P_i \ g_{i-1}	0	1	2
$\mathbb{S}_1$	$\mathbb{S}_1$	$\mathbb{S}_2$	$\mathbb{S}_2$
$\mathbb{S}_2$	$\mathbb{S}_1$	$\mathbb{S}_1$	$\mathbb{S}_2$

TABLE II: Succeeding states for complete states in the $(1,3,n)$-LRM scheme.

Recall, that only complete states will be considered in the computations. We generate a table $\mathcal{G}$ with $(t-1)!$ rows indexed by the number of complete states and $(t-1)!$ columns indexed by the possible assignments of π_{t-2}. In entry $\mathcal{G}(i,j)$, $1 \le i,j \le (t-1)!$, we have the number of realizable base-words that can be obtained from P_{n-1} which is the ith complete state when π_{t-2} is the jth permutation of S_{t-1}.

Example 6. *For the $(1,3,n)$-LRM scheme, Table III presents the number of possible pairs (g_{n-2}, g_{n-1}) which can complete a given prefix of a codeword $(g_0,\dots,g_{n-3})$ to a legal codeword, corresponds to a complete state P_{n-1} and to the permutation π_1 in the $(1,3,n)$-LRM scheme. The enumeration of the first row in the table is provided in the next two examples.*

P_{n-1} \ π_1	$[1,2]$	$[2,1]$
$\mathbb{S}_1$	5	4
$\mathbb{S}_2$	4	5

TABLE III: The number of possible pairs (g_{n-2}, g_{n-1}) which can complete a given prefix of a codeword $(g_0,\dots,g_{n-3})$ to a legal codeword, correspond to a complete state P_{n-1}, given π_1, in the $(1,3,n)$-LRM scheme.

Example 7. *For the $(1,3,n)$-LRM scheme, let $\pi_1 = [1,2]$ and $P_{n-1} = \mathbb{S}_1$, i.e. $c_1 < c_0$ and $\pi_{n-1} = [1,2]$, which implies that $c_{n-1} < c_{n-2}$. We distinguish now between the six possible pairs (x,y) of (Q.2) related to $\mathbb{S}_1$.*

1) *If $(x,y) = (0,0)$ then $c_{n-1} < c_{n-2} < c_1 < c_0$ which implies that $\alpha_{n-2} = [3,1,2] = \theta_4$, $\alpha_{n-1} = [2,3,1] = \theta_5$, $g_{n-2} = 2$, and $g_{n-1} = 1$.*
2) *If $(x,y) = (1,0)$ then $c_{n-1} < c_1 < c_{n-2} < c_0$ which implies that $\alpha_{n-2} = [3,1,2] = \theta_4$, $\alpha_{n-1} = [2,3,1] = \theta_5$, $g_{n-2} = 2$, and $g_{n-1} = 1$.*
3) *If $(x,y) = (1,1)$ then $c_1 < c_{n-1} < c_{n-2} < c_0$ which implies that $\alpha_{n-2} = [3,1,2] = \theta_4$, $\alpha_{n-1} = [2,1,3] = \theta_3$, $g_{n-2} = 2$, and $g_{n-1} = 0$.*
4) *If $(x,y) = (2,0)$ then $c_{n-1} < c_1 < c_0 < c_{n-2}$ which implies that $\alpha_{n-2} = [1,3,2] = \theta_2$, $\alpha_{n-1} = [2,3,1] = \theta_5$, $g_{n-2} = 1$, and $g_{n-1} = 1$.*
5) *If $(x,y) = (2,1)$ then $c_1 < c_{n-1} < c_0 < c_{n-2}$ which implies that $\alpha_{n-2} = [1,3,2] = \theta_2$, $\alpha_{n-1} = [2,1,3] = \theta_3$, $g_{n-2} = 1$, and $g_{n-1} = 0$.*
6) *If $(x,y) = (2,2)$ then $c_1 < c_0 < c_{n-1} < c_{n-2}$ which implies that $\alpha_{n-2} = [1,2,3] = \theta_1$, $\alpha_{n-1} = [1,2,3] = \theta_1$, $g_{n-2} = 0$, and $g_{n-1} = 0$.*

Thus, the 5 possible pairs $(\alpha_{n-2}, \alpha_{n-1})$ and (g_{n-2}, g_{n-1}) are given in the following table

(x,y)	$(\alpha_{n-2}, \alpha_{n-1})$	(g_{n-2}, g_{n-1})
$(0,0)$, $(1,0)$	$(4,5)$	$(2,1)$
$(1,1)$	$(4,3)$	$(2,0)$
$(2,0)$	$(2,5)$	$(1,1)$
$(2,1)$	$(2,3)$	$(1,0)$
$(2,2)$	$(1,1)$	$(0,0)$

Example 8. *For the $(1,3,n)$-LRM scheme, let $\pi_1 = [2,1]$ and $P_{n-1} = \mathbb{S}_1$, i.e. $c_0 < c_1$ and $\pi_{n-1} = [1,2]$, which implies that $c_{n-1} < c_{n-2}$. We distinguish now between the six possible pairs (x,y) of (Q.2) related to $\mathbb{S}_1$.*

1) *If $(x,y) = (0,0)$ then $c_{n-1} < c_{n-2} < c_0 < c_1$ which implies that $\alpha_{n-2} = [3,1,2] = \theta_4$, $\alpha_{n-1} = [3,2,1] = \theta_6$, $g_{n-2} = 2$, and $g_{n-1} = 2$.*
2) *If $(x,y) = (1,0)$ then $c_{n-1} < c_0 < c_{n-2} < c_1$ which implies that $\alpha_{n-2} = [1,3,2] = \theta_2$, $\alpha_{n-1} = [3,2,1] = \theta_6$, $g_{n-2} = 1$, and $g_{n-1} = 2$.*
3) *If $(x,y) = (1,1)$ then $c_0 < c_{n-1} < c_{n-2} < c_1$ which implies that $\alpha_{n-2} = [1,2,3] = \theta_1$, $\alpha_{n-1} = [3,1,2] = \theta_4$, $g_{n-2} = 0$, and $g_{n-1} = 2$.*
4) *If $(x,y) = (2,0)$ then $c_{n-1} < c_0 < c_1 < c_{n-2}$ which implies that $\alpha_{n-2} = [1,3,2] = \theta_2$, $\alpha_{n-1} = [3,2,1] = \theta_6$, $g_{n-2} = 1$, and $g_{n-1} = 2$.*
5) *If $(x,y) = (2,1)$ then $c_0 < c_{n-1} < c_1 < c_{n-2}$ which implies that $\alpha_{n-2} = [1,2,3] = \theta_1$, $\alpha_{n-1} = [3,1,2] = \theta_4$, $g_{n-2} = 0$, and $g_{n-1} = 2$.*
6) *If $(x,y) = (2,2)$ then $c_0 < c_1 < c_{n-1} < c_{n-2}$ which implies that $\alpha_{n-2} = [1,2,3] = \theta_1$, $\alpha_{n-1} = [1,3,2] = \theta_2$, $g_{n-2} = 0$, and $g_{n-1} = 1$.*

Thus, the 4 possible pairs $(\alpha_{n-2}, \alpha_{n-1})$ and (g_{n-2}, g_{n-1}) are given in the following table

(x,y)	$(\alpha_{n-2}, \alpha_{n-1})$	(g_{n-2}, g_{n-1})
$(0,0)$	$(4,6)$	$(2,2)$
$(1,0)$, $(2,0)$	$(2,6)$	$(1,2)$
$(1,1)$, $(2,1)$	$(1,4)$	$(0,2)$
$(2,2)$	$(1,2)$	$(0,1)$

Lemma 5. *The sum of elements in the ith row of $\mathcal{G}$ is t^{t-1}.*

Proof. Let P_{n-1} be the ith complete state. If there are no constraints then clearly the possible assignments for $g_{n-t+1}, g_{n-t+2}, \dots, g_{n-1}$ is at most t^{t-1} since there are t distinct assignments for each g_j. Given the permutation π_{n-1}, any assignment to $g_{n-t+1}, g_{n-t+2}, \dots, g_{n-1}$ yields a unique permutation π of the first $t-1$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIT.2018.2859403, IEEE Transactions on Information Theory

charge levels. Such an assignment is feasible since $c_{n-t+1}, c_{n-t+2}, \ldots, c_{n-1}$ and $c_0, c_1, \ldots, c_{t-2}$ are independent when P_{n-1} is a complete state. For this assignment we have $\pi_{t-2} = \pi$ (assume now that π is the jth permutation of S_{t-1}) and a contribution of *one* to $\mathcal{G}(i,j)$ and to the ith (which corresponds to π_{n-1}) row of $\mathcal{G}$. Thus, we have total contributions of t^{t-1} to the ith row of $\mathcal{G}$ and the lemma follows. $\square$

As a consequence we also have a conclusion which was illustrated in the last three examples.

Corollary 2. *Each complete state allows a legal codeword for arbitrary choices of* $g_{n-t+1}, g_{n-t+2}, \ldots, g_{n-1}$.

Theorem 2. *If* M_t *is the number of legal codewords in the* $(1,t,n)$*-LRM scheme then* $\lim_{n\to\infty} \frac{M_t}{t^n} = 1$.

Proof. Consider a prefix of a codeword $g' = (g_0, g_1, \ldots, g_{n-t})$ which contains the substring of length $2(t-1)$, $\beta = (g_{i-2t+3}, g_{i-2t+4}, \ldots, g_{i-1}, g_i) = (\underbrace{t-1, t-1, \ldots, t-1}_{t-1\ times}, \underbrace{0, 0, \ldots, 0}_{t-1\ times})$. We will prove now that for this subsequence the charge levels $c_{i+1}, c_{i+2}, \ldots, c_{i+t-1}$ are independent of the charge levels $c_0, c_1, \ldots, c_{t-2}$. This implies that P_{i+t-1} is one of the $(t-1)!$ complete state in the $(1,t,n)$-LRM scheme. For this, we have to prove that each one of the charge levels $c_{i+1}, c_{i+2}, \ldots, c_{i+t-1}$ can be lower than $c_0, c_1, \ldots, c_{t-2}$, between them ($t-2$ distinct possible options), or higher than all of them.

The substring β starts with $g_{i-2t+3} = t-1$ which imposes that c_{i-t+2} is higher than the $t-1$ charge levels before it, and implies that c_{i-t+2} might be higher than the first $t-1$ charge levels. The sub-codeword $(g_{i-2t+4}, \ldots, g_{i-t+1}) = (\underbrace{t-1, t-1, \ldots, t-1}_{t-2\ times})$ imposes that $c_{i-t+2} < c_{i-t+3} < \cdots < c_i$. Furthermore, $g_{i-t+2} = 0$ implies $c_{i+1} < c_{i-t+2}$, and $(g_{i-t+3}, \ldots, g_i) = (\underbrace{0, 0, \ldots, 0}_{t-2\ times})$ imposes the constraint $c_{i+1} > c_{i+2} > \cdots > c_{i+t-1}$. Hence, we have that

$$\begin{aligned} &c_{i+t-1} < c_{i+t-2} < \cdots < c_{i+2} < c_{i+1} \\ &\quad < c_{i-t+2} < c_{i-t+3} < \cdots < c_i, \end{aligned}$$

where c_{i-t+2} might be higher than the first $t-1$ charge levels, and there does not exist any additional constraint on the possible values of the charge levels $c_{i+1}, c_{i+2}, \ldots, c_{i+t-1}$ regarding their ranking among the previous charge levels. This implies that there is no dependency between the charge levels $c_{i+1}, c_{i+2}, \ldots, c_{i+t-1}$, and the first $t-1$ charge levels, i.e., P_{i+t-1} is a complete state. Moreover, the permutation π_{i+t-1} of the property (Q.1) for this state is $[1, 2, \ldots, t-1]$. As a consequence Lemma 3 implies that also P_{n-1} is a complete state.

Let B be the set of sequences of length $n-t+1$, over the alphabet $\{0, 1, \ldots, t-1\}$, which include β as a substring. We partition each sequence of length $2\ell(t-1)$ into ℓ consecutive disjoint segments of length $2(t-1)$ and one segment (the last) of length $n-t+1-2\ell(t-1)$. Let A be a subset of B which consists of the sequences which contain a segment equals to β. Let A^c be the complimentary set of A, i.e. the set of all sequences of length $n-t+1$, over the alphabet $\{0, 1, \ldots, t-1\}$, which do not contain a segment which equals to β. Now, we have

$$\begin{aligned} |B| \geq |A| &= t^{n-t+1} - |A^c| \\ &= t^{n-t+1} - (t^{2(t-1)} - 1)^\ell t^{n-t+1-2\ell(t-1)}. \end{aligned}$$

By Lemma 5 we have that $M_t \geq t^{t-1}|B|$ and therefore

$$\begin{aligned} &\lim_{n\to\infty} \frac{M_t}{t^n} \geq \lim_{n\to\infty} \frac{t^n - t^{n-2\ell(t-1)} \cdot (t^{2(t-1)} - 1)^\ell}{t^n} \\ &= \lim_{\ell\to\infty} \left(1 - \frac{(t^{2(t-1)} - 1)^\ell}{t^{2(t-1)\ell}}\right) = \lim_{\ell\to\infty} (1 - \mu^\ell) = 1, \end{aligned}$$

where clearly μ is a constant since t is a constant. $\square$

By using the Perron-Frobenius Theorem [4], [11] we can prove that for $t = 3$, $|B|$ tends to $3^{n-2} - 2.777^{n-2}$ when n tends to infinity. This improves the rate of convergence of $\lim_{n\to\infty} \frac{M_3}{3^n}$. Similar computation can be done for larger t, but the computations become more messy as t increases.

To end this section, we return to the question how to indicate that a base-word is cyclically non-realizable in the $(1,3,n)$-LRM scheme. To answer this question, we use the states defined in this section. Given a codeword $g = (g_0, g_1, \ldots, g_{n-1})$, and α_0 (which is computed by the decoding procedure), we can determine P_3. Then, by Lemma 2, we can compute P_{n-1}. Note, that α_0 determines the permutation of the first three cells and P_{n-1} (see (Q.1)) determines the permutation of the last two elements. Additionally, P_{n-1} (see (Q.2)) determines exactly all the possible rankings for the charge levels of each one of the last two cells among the first two cells. Thus, from α_0 and P_{n-1} we can determine whether α is cyclically realizable, i.e., if g is legal. The complexity of this procedure is $O(n)$. Algorithm 3 presents the formal steps of this procedure.

IV. Conclusions and Open Problems

In this paper, encoding, decoding, and enumeration of the $(1,t,n)$-LRM scheme are studied. A complete solution was given for the $(1,3,n)$-LRM scheme. A simple encoding for the $(1,t,n)$-LRM scheme for any $t \geq 3$ is presented. For the $(1,3,n)$-LRM scheme a related decoding was presented. We also proved that if M_t is the number of legal codewords in the $(1,t,n)$-LRM scheme then $\lim_{n\to\infty} \frac{M_t}{t^n} = 1$. We conclude with several problems for future research raised in our discussion.

1) Find an efficient algorithm to determine if a given codeword in the $(1,t,n)$-LRM scheme, for $t \geq 4$, is legal or not.

Algorithm 3 : Legality of a codeword and realizability of a base-word in the $(1, 3, n)$-LRM scheme

Input: A codeword $g \in [3]^n$.
Output: Is g legal?
  $\alpha \leftarrow$ Algorithm 1(g) {the decoding algorithm is applied on g}
  if α =NIL **then return** NO
  $\pi \leftarrow$ the permutation of $\{1, 2, \dots, (t-1)\}$ implied by α_2.
  (Q.2)$\leftarrow$ the possibilities of c_2 and c_3 regarding the first two charge levels (determined by α_0 and α_1)
  $P_3 \leftarrow \pi$,(Q.2)
  for $k := 4 \dots (n-1)$ **do**
    $P_k \leftarrow$ the kth state determined by P_{k-1} and g_{k-2} (see Lemma 2).
  end for
  if α_{n-2} **or** α_{n-1} can not be obtain by P_{n-1} and α_0 **then return** NO
  return YES

2) Prove that the encoding algorithm for the $(1, t, n)$-LRM scheme, $t \geq 4$, induces a bijection between the realizable base-words and the legal codewords.
3) Find an efficient decoding algorithm for the $(1, t, n)$-LRM scheme, $t \geq 4$.
4) Improve the lower bound on the asymptotic behavior and non-asymptotic behavior of the number of codewords in the $(1, t, n)$-LRM scheme, $t \geq 3$. Note, that one such improvement is done by using the Perron-Frobenius Theorem.
5) How would the results in the paper change if the largest values of the charge levels were bounded?

Acknowledgement

The authors would like to thank Ronny Roth for helpful discussions. The authors are also indebted to the anonymous reviewers for their important comments.

References

[1] A. Barg, A. Mazumdar, and G. Zémor, "Constructions of rank modulation codes," *IEEE Trans. on Inform. Theory*, vol. 59, pp. 1018–1029, February 2013.
[2] S. Buzaglo and T. Etzion, "Bounds on the Size of Permutation Codes with the Kendall's τ-Metric," *IEEE Trans. on Inform. Theory*, vol. 61, pp. 3241–3250, June 2015.
[3] S. Buzaglo, T. Etzion, E. Yaakobi, and J. Bruck, "Systematic Error-Correcting Codes for Permutations and Multi-permutations," *IEEE Trans. on Inform. Theory*, vol. 62, pp. 3113–3124, June 2016.
[4] G. Frobenius, "Uber Matrizen aus nicht negativen Elementen," S.-B. Preuss Acad. Wiss. Berlin pp. 456–477, 1912.
[5] E. E. Gad, M. Langberg, M. Schwartz, and J. Bruck, "Constant-weight Gray codes for local rank modulation," *IEEE Trans. on Inform. Theory*, vol. 57, pp. 7431–7442, November 2011.
[6] A. E. Holroyd, "Perfect snake-in-the-box codes for rank modulation," *IEEE Trans. on Inform. Theory*, vol. 63, pp. 104–110, January 2017.
[7] M. Horovitz and T. Etzion, "Constructions of snake-in-the-box codes for rank modulation," *IEEE Trans. on Inform. Theory*, vol. 60, pp. 7016–7025, November 2014.
[8] A. Jiang, R. Mateescu, M. Schwartz, and J. Bruck, "Rank modulation for flash memories," *IEEE Trans. on Inform. Theory*, vol. 55, pp. 2659–2673, June 2009.
[9] A. Jiang, M. Schwartz, and J. Bruck, "Correcting charge-constrained errors in the rank-modulation scheme," *IEEE Trans. on Inform. Theory*, vol. 56, pp. 2112–2120, May 2010.
[10] A. Mazumdar and A. Barg, "Coded in permutations and error correction for rank modulation," *IEEE Trans. on Inform. Theory*, vol. 56, pp. 3158–3165, July 2010.
[11] O. Perron, "Zur theorie der matrizen," *Math. Ann.* vol. 64, pp. 248–263, 1907.
[12] A. S. Spinelli, C. M. Compagnoni, and A. L. Lacaita, "Reliability of NAND Flash Memories: Planar Cells and Emerging Issues in 3D Devices," *Computers*, vol. 6, no. 2, pp. 16, 2017, [online] Available: http://www.mdpi.com/2073-431X/6/2/16.
[13] I. Tamo and M. Schwartz, "Correcting limited-magnitude errors in the rank-modulation scheme," *IEEE Trans. on Inform. Theory*, vol. 56, pp. 2551–2560, June 2010.
[14] Z. Wang, A. Jiang, and J. Bruck, "On the capacity of bounded rank modulation for flash memories," in *Proc. IEEE Int. Symp. on Inform. Theory*, Seoul, Korea, pp. 1234–1238, June 2009.
[15] Y. Yehezkeally and M. Schwartz, "Snake-in-the-box codes for rank modulation," *IEEE Trans. on Inform. Theory*, vol. 58, no. 8, pp. 5471–5483, August 2012.
[16] Y. Yehezkeally and M. Schwartz, "Limited-magnitude error-correcting Gray codes for rank modulation," *IEEE Trans. on Inform. Theory*, vol. 63, pp. 5774–5792, September 2017.
[17] Y. Zhang and G. Ge, "Snake-in-the-box codes for rank modulation under Kendalls τ-metric," *IEEE Trans. on Inform. Theory*, vol. 62, pp. 151–158, January 2016.
[18] H. Zhou, M. Schwartz, A. Jiang, and J. Bruck, "Systematic error-correction codes for rank modulation," *IEEE Trans. on Inform. Theory*, vol. 61, pp. 17–32, January 2015.

Michal Horovitz was born in Israel in 1987. She received the B.Sc. degree from the departments of Mathematics and Computer Science at the Open University, Ra'anana, Israel, in 2009, and the Ph.D. degree from the Computer Science Department at the Technion - Israel Institute of Technology, Haifa, Israel, in 2017. She is currently a Lecturer in the Department of Computer Science, Tel-Hai College, Israel and she is also a researcher in The Galilee Research Institute - Migal, Upper Galilee, Israel. Her research interests include coding theory with applications to non-volatile memories, information theory, and combinatorics.

Tuvi Etzion (M'89–SM'94–F'04) was born in Tel Aviv, Israel, in 1956. He received the B.A., M.Sc., and D.Sc. degrees from the Technion - Israel Institute of Technology, Haifa, Israel, in 1980, 1982, and 1984, respectively.

From 1984 he held a position in the Department of Computer Science at the Technion, where he now holds the Bernard Elkin Chair in Computer Science. During the years 1985-1987 he was Visiting Research Professor with the Department of Electrical Engineering - Systems at the University of Southern California, Los Angeles. During the summers of 1990 and 1991 he was visiting Bellcore in Morristown, New Jersey. During the years 1994-1996 he was a Visiting Research Fellow in the Computer Science Department at Royal Holloway University of London, Egham, England. He also had several visits to the Coordinated Science Laboratory at University of Illinois in Urbana-Champaign during the years 1995-1998, two visits to HP Bristol during the summers of 1996, 2000, a few visits to the Department of Electrical Engineering, University of California at San Diego during the years 2000-2017, and several visits to the Mathematics Department at Royal Holloway University of London, during the years 2007-2017.

His research interests include applications of discrete mathematics to problems in computer science and information theory, coding theory, network coding, and combinatorial designs.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIT.2018.2859403, IEEE Transactions on Information Theory

Dr. Etzion was an Associate Editor for Coding Theory for the IEEE Transactions on Information Theory from 2006 till 2009. From 2004 to 2009, he was an Editor for the Journal of Combinatorial Designs. From 2011 he is an Editor for Designs, Codes, and Cryptography. and from 2013 an Editor for Advances of Mathematics in Communications.