SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

October 2018, Vol. 61 100305:1–100305:16
https://doi.org/10.1007/s11432-018-9509-y

Special Focus on Distributed Storage Coding

Gray codes over certain run-length sequences for local rank modulation†

Xiang WANG[1] & Fang-Wei FU[2]

[1]*National Computer Network Emergency Response Technical Team, Beijing* 100029*, China;*
[2]*The Chern Institute of Mathematics and LPMC, Nankai University, Tianjin* 300071*, China*

Received 9 March 2018/Revised 9 May 2018/Accepted 28 June 2018/Published online 21 August 2018

Abstract In the local rank modulation (LRM) scheme, a sliding window produces a sequence of permutations by moving over a sequence of variables. LRM has been presented as a method of storing data in flash memory, which represents a natural generalization of the classical rank modulation scheme. In this paper, we present a study on Gray codes over certain run-length sequences for the $(1, 2, n)$-LRM scheme to simulate virtual multilevel flash memory cells while maintaining the advantages of LRM. Unlike previous studies on the LRM scheme, we present Gray codes over certain run-length sequences in the $(1, 2, n)$-LRM scheme. This class of Gray codes can overcome the drawback of the many distinct charge levels required in the rank modulation scheme and in certain Gray codes for LRM. Furthermore, we demonstrate that the proposed codes have an asymptotically optimal rate.

Keywords flash memory, local rank modulation, Gray codes, rank modulation, run-length sequences

Citation Wang X, Fu F-W. Gray codes over certain run-length sequences for local rank modulation. Sci China Inf Sci, 2018, 61(10): 100305, https://doi.org/10.1007/s11432-018-9509-y

1 Introduction

Flash memory is a nonvolatile type of memory that is electrically programmable and erasable. It is widely used for its high storage density and relatively low cost. For exactly and efficiently representing data in flash memory, the rank modulation scheme was presented and studied in a recent series of papers, specifically [1–4]. In the rank modulation scheme [1], a permutation induced by the relative charge levels of certain cells represents stored information. In this scheme, programming operation is restricted to the "push-to-the-top" operation. Specifically, in the rank modulation framework, a single cell is programmed by raising its charge level above that of all others. Therefore, in this scheme, over-programming (increasing the charge level of a cell above the desired amount) is not an issue. Additionally, this scheme can reduce corruption and speed up cell programming [1].

In the rank modulation scheme for flash memory, a set of n cells is utilized to simulate a single virtual multilevel flash cell with $n!$ levels. Jiang et al. [1] utilized a Gray code to traverse the $n!$ states, where the transition between two adjacent states in the Gray code was facilitated by utilizing a single "push-to-the-top" operation. However, the Gray code was first presented in [5] as a sequence of distinct binary vectors of fixed length, where adjacent vectors differed in only a single coordinate. In practice, they are widely used in many applications, such as storage and retrieval applications [6], processor allocation [7],

* Corresponding author (email: e-mail: xqwang@mail.nankai.edu.cn)
† Invited article

signal encoding [8], and data compression [9]. A thorough survey of Gray codes was presented in [10]. In the context of flash memory, Gray codes for the rank modulation scheme were studied in [1, 2]. Furthermore, various generalizations of Gray codes for rank modulation include Gray codes for local rank modulation [11, 12] and snake-in-the-box codes for rank modulation [13–16].

One drawback of the rank modulation scheme is the fact that a large number of comparisons are required to obtain a single permutation induced by a set of n charge levels. In order to acquire a permutation from the n charge levels, at least $\Omega(n \log n)$ comparisons are required, which requires significant storage space and computation time. To overcome this drawback, local rank modulation (LRM) was proposed in [4,11,12]. In the LRM scheme, only local comparisons are required. These comparisons yield a sequence of small permutations instead of a single large permutation. En Gad et al. [11, 12] presented Gray codes for the LRM scheme.

However, there is another drawback of the rank modulation scheme. Assume that δ is the minimum charge difference required to distinguish two distinct levels and D is the gap between the minimum and maximum charge levels of a flash memory cell. For a group of n flash memory cells, n distinct charge levels of these cells are required to induce one permutation over $\{1, 2, \ldots, n\}$ in the rank modulation scheme. Furthermore, two distinct charge levels represent two distinguishable levels, meaning the difference between the two distinct charge levels must be at least δ. In order to obtain n distinct charge levels, this value increases to $D \geqslant n\delta$. Therefore, the requirement for n distinct charge levels prohibits the use of large values of n.

However, in the LRM scheme, a small permutation induced by local charge levels only requires distinct charge levels for the corresponding local cells. En Gad et al. presented constant-weight Gray codes in [12] and generalized Gray codes for LRM in [11]. However, for n flash memory cells, there must be certain code words in these Gray codes that require at least $\sqrt{n}$ distinct charge levels.

The LRM scheme with a sequence of n permutations over two elements is called the $(1, 2, n)$-LRM scheme. To overcome the above drawbacks, in the $(1, 2, n)$-LRM scheme, we consider a set of sequences, denoted $\mathcal{R}(1, 2, n; d)$, in which each sequence requires at most d distinct charge levels, where d is a constant. In this paper, we propose a class of Gray codes over $\mathcal{R}(1, 2, n; d)$ by utilizing "push-to-the-top" operations. This class of Gray codes has an asymptotically optimal rate. Furthermore, this class of Gray codes only requires at most d distinct levels, which is much less than $\sqrt{n}$ (the Gray codes in [11,12] require at least $\sqrt{n}$ distinct charge levels). Therefore, the proposed Gray codes can maintain the advantages of LRM, overcome the drawback of requiring a large number of distinct levels, and have an asymptotically optimal rate.

The remainder of this paper is organized as follows. In Section 2, we define some basic concepts of the LRM scheme and provide some useful notations for readers. In Section 3, various properties of run-length sequences are presented. In Section 4, we present a construction of Gray codes over $\mathcal{R}(1, 2, n; d)$ for the $(1, 2, n)$-LRM scheme. Section 5 concludes this paper.

2 Preliminaries

In this section, we will introduce some notations and definitions for LRM and Gray codes. Some notations and definitions were presented in [11, 12].

2.1 Local rank modulation

Let S_n be the set of all permutations over $[n] = \{1, 2, \ldots, n\}$. Denote $\pi = [\pi(1), \pi(2), \ldots, \pi(n)]$ as a permutation over $[n]$ (i.e., $\pi \in S_n$). Given t flash memory cells, we number these cells as $1, 2, \ldots, t$. Let $\boldsymbol{c} = (c_1, c_2, \ldots, c_t) \in \mathbb{R}^t$ be a vector of t real-valued variables, where c_i is the charge level of the i-th cell for all $i \in [t]$. Assume that δ is the minimum charge difference required to distinguish two distinct levels and $c_i \neq c_j$ for all $i \neq j$ (i.e., $|c_i - c_j| \geqslant \delta$ for all $i \neq j$). For convenience, if not specified otherwise, we refer to the two distinguishable charge levels as two distinct levels (values). The t distinct

variables $c_1, \ldots, c_t$ (i.e., $\boldsymbol{c}$) induce one permutation, denoted $f_{\boldsymbol{c}} = [f_{\boldsymbol{c}}(1), f_{\boldsymbol{c}}(2), \ldots, f_{\boldsymbol{c}}(t)] \in S_t$, such that $c_{f_{\boldsymbol{c}}(1)} > c_{f_{\boldsymbol{c}}(2)} > \cdots > c_{f_{\boldsymbol{c}}(t)}$.

For a set of n flash memory cells, consider a vector of n real-valued variables $\boldsymbol{c} = (c_1, c_2, \ldots, c_n) \in \mathbb{R}^n$, where c_i is the level of the i-th cell for all $i \in [n]$. From a local perspective, we define a window of size t at position p to be

$$\boldsymbol{c}_{p,t} = (c_p, c_{p+1}, \ldots, c_{p+t-1}),$$

where the indices are taken modulo n, $1 \leqslant p \leqslant n$, and $1 \leqslant t \leqslant n$.

Let s, t, and n be positive integers such that $s \leqslant t \leqslant n$ and $s|n$. The (s, t, n)-LRM scheme is defined by utilizing the demodulation process from [11]. Given a vector of n distinct real-valued variables $\boldsymbol{c} = (c_1, c_2, \ldots, c_n)$, the demodulation maps $\boldsymbol{c}$ to a sequence of n/s permutations from S_t, denoted $\boldsymbol{f_c}$, where

$$\boldsymbol{f_c} = (f_{\boldsymbol{c}_{1,t}}, f_{\boldsymbol{c}_{1+s,t}}, \ldots, f_{\boldsymbol{c}_{n-s+1,t}}). \tag{1}$$

In other words, by utilizing windows of size t to scan the n variables, a sequence of n/s permutations from S_t is obtained in the (s, t, n)-LRM scheme. For example, if $s = 1, t = 2, n = 3$ and $\boldsymbol{c} = (0.5, 2.5, 1.5)$, then, by the definition of $\boldsymbol{f_c}$, $f_{\boldsymbol{c}_{1,2}} = [2, 1], f_{\boldsymbol{c}_{2,3}} = [1, 2], f_{\boldsymbol{c}_{3,1}} = [1, 2]$, and $\boldsymbol{f_c} = ([2, 1], [1, 2], [1, 2])$.

A sequence $\boldsymbol{g}$ of n/s permutations from S_t is (s, t, n)-LRM realizable if there exists a $\boldsymbol{c} \in \mathbb{R}^n$ such that $\boldsymbol{g} = \boldsymbol{f_c}$ (i.e., $\boldsymbol{g}$ is the demodulated sequence of $\boldsymbol{c}$ in the (s, t, n)-LRM scheme). Except for the degenerate case of $s = t$, not every sequence is realizable [12]. For the number of distinct levels of n flash memory cells, we define a function $\Psi : \mathbb{R}^n \to [n]$, where for every $\boldsymbol{c} = (c_1, \ldots, c_n) \in \mathbb{R}^n$, $\Psi(\boldsymbol{c})$ is the number of distinct components of $\boldsymbol{c}$. Similarly, a sequence $\boldsymbol{g}$ of n/s permutations over S_t is $(s, t, n; d)$-LRM realizable if there exists a $\boldsymbol{c} \in \mathbb{R}^n$ such that $\boldsymbol{g} = \boldsymbol{f_c}$ and $\Psi(\boldsymbol{c}) \leqslant d$. In other words, $\boldsymbol{g}$ is the demodulated sequence of $\boldsymbol{c}$ in the (s, t, n)-LRM scheme, where the number of distinct components of $\boldsymbol{c}$ is at most d.

For convenience, let $\mathcal{R}(s, t, n)$ be the set of all (s, t, n)-LRM realizable permutation sequences. Let $\mathcal{R}(s, t, n; d)$ be the set of all $(s, t, n; d)$-LRM realizable permutation sequences. It follows that $\mathcal{R}(s, t, n; d) \subseteq \mathcal{R}(s, t, n)$ and $\mathcal{R}(s, t, n; n) = \mathcal{R}(s, t, n)$. If $d_1 \leqslant d_2$ are positive integers, then $\mathcal{R}(s, t, n; d_1) \subseteq \mathcal{R}(s, t, n; d_2)$.

For each $\boldsymbol{f_c} = (f_{\boldsymbol{c}_{1,t}}, f_{\boldsymbol{c}_{1+s,t}}, \ldots, f_{\boldsymbol{c}_{n-s+1,t}}) \in \mathcal{R}(s, t, n)$, a compact representation of $\boldsymbol{f_c}$ may be derived by utilizing (mixed-radix) factoradic notation (see [1, 17]). Specifically, for any permutation $f = [f(1), \ldots, f(t)] \in S_t$, this permutation can be represented utilizing a sequence of integers $d_t, d_{t-1}, \ldots, d_1$, where d_{t+1-i} is the number of entries $f(j)$ for $j > i$ such that $f(j) < f(i)$. Specifically,

$$d_{t+1-i} = |\{j > i | f(j) < f(i)\}|.$$

Here, we refer to d_t as the most-significant digit and d_1 as the least-significant digit. Based on the overlap between adjacent local views, the local permutation $f_{\boldsymbol{c}_{i \cdot s+1,t}}$ can be represented utilizing only the s most-significant digits in its factoradic notation for every $0 \leqslant i \leqslant n/s - 1$. This representation is denoted $\bar{f}_{\boldsymbol{c}_{i \cdot s+1,t}}$ and referred to as the condensed factoradic representation. Accordingly, we define

$$\bar{\boldsymbol{f}}_{\boldsymbol{c}} = (\bar{f}_{\boldsymbol{c}_{1,t}}, \bar{f}_{\boldsymbol{c}_{1+s,t}}, \ldots, \bar{f}_{\boldsymbol{c}_{n-s+1,t}}),$$

and the set of all such presentations as $\bar{\mathcal{R}}(s, t, n)$. Similarly, let $\bar{\mathcal{R}}(s, t, n; d) = \{\bar{\boldsymbol{f}}_{\boldsymbol{c}} | \boldsymbol{f_c} \in \mathcal{R}(s, t, n; d)\}$. For example, if $s = 3, t = 5, n = 6$, and $\boldsymbol{c} = (2, 0.5, 1.5, 2.5, 1, 3)$, then, by the definition of $\boldsymbol{f_c}$ and $\bar{\boldsymbol{f}}_{\boldsymbol{c}}$, $\boldsymbol{f_c} = \big([4, 1, 3, 5, 2], [3, 1, 4, 2, 5]\big)$ and $\bar{\boldsymbol{f}}_{\boldsymbol{c}} = \big((3, 0, 1), (2, 0, 1)\big)$. Throughout the remainder of this paper, we will only consider the condensed factoradic representation and omit the term "condensed" for the sake of brevity. For convenience, we will make no distinction between $\bar{\boldsymbol{f}}_{\boldsymbol{c}}$ and $\boldsymbol{f_c}$.

When $s = t = n$, the (n, n, n)-LRM scheme degenerates into the rank modulation scheme, where we obtain a single permutation from S_n. The case $s = t < n$ was discussed by Ferreira et al. [18] in the context of permutation trellis codes, where a binary code word was transformed tuple-wise into a sequence of permutations with no overlap between adjacent tuples. Furthermore, Wang et al. [4] discussed the general case of $s \leqslant t < n$ (in this case, indices are not taken modulo n). Finally, En Gad et al. proposed Gray codes for the cases of $(1, 2, n)$-LRM [11] and (s, t, n)-LRM [12].

In the $(1, 2, n)$-LRM scheme, the demodulated sequences of permutations comprise the permutations $[1, 2]$ and $[2, 1]$. In the following explanation, we will represent the permutation $[1, 2]$ with the logical

value 1 and $[2,1]$ with 0, thereby forming a one-to-one mapping between binary sequences from $\bar{\mathcal{R}}(1,2,n)$ and permutation sequences from $\mathcal{R}(1,2,n)$. It can be easily verified that $\bar{\mathcal{R}}(1,2,n)$ includes all binary sequences of length n, excluding all-ones and all-zeros sequences (i.e., $\bar{\mathcal{R}}(1,2,n) = \{0,1\}^n - \{0^n, 1^n\}$). Therefore,

$$\bar{\mathcal{R}}(1,2,n;d) \subset \bar{\mathcal{R}}(1,2,n) \subset \{0,1\}^n.$$

For convenience, we will make no distinction between $\mathcal{R}(1,2,n)$ and $\bar{\mathcal{R}}(1,2,n)$. Similarly, we utilize the notation $\mathcal{R}(1,2,n;d)$ for $\bar{\mathcal{R}}(1,2,n;d)$ later.

2.2 Gray codes for LRM

Generally speaking, given a set $\mathcal{S}$ and subset of transformations $T \subset \{g | g : \mathcal{S} \to \mathcal{S}\}$, a Gray code over $\mathcal{S}$ of size M utilizing transformations from T is a sequence $C = (\boldsymbol{c}_0, \boldsymbol{c}_1, \ldots, \boldsymbol{c}_{M-1})$ of M distinct elements from $\mathcal{S}$, referred to as code words, such that for each $i \in [M-1]$, there exists some $\tilde{t}_i \in T$ for which $\boldsymbol{c}_i = \tilde{t}_i(\boldsymbol{c}_{i-1})$.

In the context of LRM for flash memory, consider $S = \mathcal{R}(1,2,n;d)$, which is the set of all realizable sequences in the $(1,2,n;d)$-LRM scheme.

In the rank modulation scheme, Jiang et al. [1] utilized the "push-to-the-top" operation to transform a permutation into an adjacent permutation. In other words, the charge level of a single cell was raised above that of all others. However, in the LRM scheme, the "push-to-the-top" operation merely raises the charge level of a single cell above that of the cells that are comparable to it [12].

We now prove a concrete definition of the "push-to-the-top" operation in the $(1,2,n;d)$-LRM scheme. Assume that $\boldsymbol{c} = (c_1, c_2, \ldots, c_n) \in \mathbb{R}^n$, where $c_1, c_2, \ldots, c_n$ are the charge levels of n flash memory cells. Let δ be the minimum charge difference required to distinguish two distinct levels. A "push-to-the-top" operation performed on the i-th cell changes the cell levels $\boldsymbol{c} = (c_1, c_2, \ldots, c_n)$ to the cell levels $\boldsymbol{c}' = (c'_1, c'_2, \ldots, c'_n) \in \mathbb{R}^n$, defined by

$$c'_j = \begin{cases} c_j, & \text{if } j \neq i, \\ \max\{c_{i-1}, c_{i+1}\} + \delta, & \text{if } j = i, \end{cases} \tag{2}$$

where $i-1, i+1$ should be taken modulo n and 0 is considered as n modulo n. Specifically, in the $(1,2,n)$-LRM scheme, the charge level of the i-th cell is pushed above the charge levels of the cells that are comparable to it.

Let $\boldsymbol{f_c} = (f_1, f_2, \ldots, f_n)$ and $\boldsymbol{f_{c'}} = (f'_1, f'_2, \ldots, f'_n)$ be the demodulated sequences of permutations for $\boldsymbol{c}$ and $\boldsymbol{c}'$, respectively, in the $(1,2,n)$-LRM scheme. Then, $\boldsymbol{f_c}, \boldsymbol{f_{c'}} \in \mathcal{R}(1,2,n)$. From (2), a single "push-to-the-top" operation is performed on the i-th cell to change $\boldsymbol{f_c}$ into $\boldsymbol{f_{c'}}$. Therefore, we define the set of allowed transitions over $\mathcal{R}(1,2,n)$ as $T = \{\tau_1, \tau_2, \ldots, \tau_n\}$, which is a set of functions $\tau_i : \mathcal{R}(1,2,n) \to \mathcal{R}(1,2,n)$, where τ_i represents a "push-to-the-top" operation performed on the i-th cell such that $\boldsymbol{f_{c'}} = \tau_i(\boldsymbol{f_c})$. Specifically,

$$f'_j = \begin{cases} 0, & \text{if } j \equiv i-1 \pmod{n}, \\ 1, & \text{if } j = i, \\ f_j, & \text{otherwise.} \end{cases}$$

For example, if $\boldsymbol{f_c} = (1,0,1,0,1,0)$, there exists one transition τ_4 performed on $\boldsymbol{f_c}$ to obtain another sequence $\boldsymbol{f_{c'}}$, where $\boldsymbol{f_{c'}} = (1,0,0,1,1,0)$. In this paper, we will present Gray codes over $\mathcal{R}(1,2,n;d)$ by utilizing "push-to-the-top" operations such that each code word is realized by at most d distinct levels.

Definition 1. A Gray code G over $\mathcal{R}(1,2,n;d)$ is a sequence of distinct length-n binary code words, denoted $G = (g_0, g_1, \ldots, g_{M-1})$, where $g_i \in \mathcal{R}(1,2,n;d)$ for all $0 \leqslant i \leqslant M-1$, such that for all $0 \leqslant i \leqslant M-2$, there exists some $j_i \in [n]$ for which $g_{i+1} = \tau_{j_i}(g_i)$. The Gray code G is cyclic if there exists some $l \in [n]$ such that $g_0 = \tau_l(g_{M-1})$ and optimal if $M = |\mathcal{R}(1,2,n;d)|$.

In the following explanation, if not specified otherwise, we simply refer to a cyclic Gray code as a Gray code.

Definition 2. Let $\{G_i\}_{i=1}^{\infty}$ be a family of Gray codes where G_i is a Gray code over $\mathcal{R}(1,2,n^{(i)};d^{(i)})$ of size M_i, and $n^{(i+1)} > n^{(i)}$ and $d^{(i+1)} \geqslant d^{(i)}$ are positive integers. We define the rate of the code G_i as $R(G_i) = \frac{\log_2 M_i}{n^{(i)}}$. Then, we state that $\{G_i\}_{i=1}^{\infty}$ is asymptotically rate-optimal if $\lim_{i\to\infty} R(G_i) = 1$.

3 Run-length sequences

In this section, we will discuss some properties of $\mathcal{R}(1,2,n;d)$. Note that each element of $\mathcal{R}(1,2,n;d)$ can be considered as a certain run-length sequence. Therefore, to define the asymptotic property of $|\mathcal{R}(1,2,n;d)|$, some notations and definitions for run-length binary sequences are introduced below.

Let $\boldsymbol{u} = (u_0, u_1, \ldots, u_{n-1}) \in \{0,1\}^n$ be a binary vector of length n. In this paper, u_0 and u_{n-1} are considered to be adjacent. Next, we provide definitions of a run, run length, and maximum run length.

Definition 3. Suppose $\boldsymbol{u}$ is defined as above. If $u_{i-1} \neq u_i = u_{i+1} = \cdots = u_{j-1} \neq u_j$, then $(u_i, \ldots, u_{j-1})$ is a run of $\boldsymbol{u}$, and $(j-i) \mod n$ is the run length of $(u_i, \ldots, u_{j-1})$, where the indices $i-1, i, \ldots, j-1, j$ should be taken modulo n.

Definition 4. Suppose $\boldsymbol{u}$ is defined as above. Denote $R_L(\boldsymbol{u})$ as the maximum run length of $\boldsymbol{u}$.

For example, if $\boldsymbol{u} = (1,0,0,0,1,1,0,0,1)$, then $\boldsymbol{u}$ has a run of zeros of length 3, a run of zeros of length 2, and two runs of ones of length 2. Therefore, $R_L(\boldsymbol{u}) = 3$. From Definition 4, we establish a relationship between certain run-length sequences and $\mathcal{R}(1,2,n;d)$ in the following lemma.

Lemma 1. For any $\boldsymbol{u} \in \mathcal{R}(1,2,n;d)$, we have that $R_L(\boldsymbol{u}) \leqslant d-1$. Furthermore, for any $\boldsymbol{u} \in \{0,1\}^n$, if $R_L(\boldsymbol{u}) \leqslant d-1$, then $\boldsymbol{u} \in \mathcal{R}(1,2,n;d)$. Therefore,

$$\mathcal{R}(1,2,n;d) = \{\boldsymbol{v} \in \{0,1\}^n | R_L(\boldsymbol{v}) \leqslant d-1\}.$$

Proof. First, we prove that if $\boldsymbol{u} = (u_0, u_1, \ldots, u_{n-1}) \in \mathcal{R}(1,2,n;d)$, then $R_L(\boldsymbol{u}) \leqslant d-1$. From the definition of $\mathcal{R}(1,2,n;d)$, if there exists a vector $\boldsymbol{c} \in \mathbb{R}^n$ such that $\Psi(\boldsymbol{c}) \leqslant d$ and $\boldsymbol{f_c} = \boldsymbol{u}$. If $R_L(\boldsymbol{u}) \geqslant d$, then there exists a run of $\boldsymbol{u}$, denoted $(u_i, \ldots, u_{j-1})$, such that the run length $(j-i) \mod n \geqslant d$. Because $u_i = \cdots = u_{j-1}$, it follows that the substring of $\boldsymbol{c}$, written $(c_i, c_{i+1}, \ldots, c_j)$, can be demodulated into a sequence of permutations

$$(\underbrace{[1,2], \ldots, [1,2]}_{(j-i) \mod n}) \quad \text{or} \quad (\underbrace{[2,1], \ldots, [2,1]}_{(j-i) \mod n}).$$

Then, $(c_i, c_{i+1}, \ldots, c_j)$ is an increasing or decreasing sequence. From the definition of $\Psi(\cdot)$, we have that

$$\Psi(\boldsymbol{c}) \geqslant \Psi((c_i, c_{i+1}, \ldots, c_j)) = ((j-i) \mod n) + 1. \tag{3}$$

From (3), $\Psi(\boldsymbol{c}) \geqslant d+1$, which results in a contradiction. Therefore, $R_L(\boldsymbol{u}) \leqslant d-1$.

Next, we prove that for any $\boldsymbol{u} \in \{0,1\}^n$, if $R_L(\boldsymbol{u}) \leqslant d-1$, $\boldsymbol{u} \in \mathcal{R}(1,2,n;d)$. Therefore, we only need to prove that there exists a vector $\boldsymbol{c} \in \mathbb{R}^n$ such that $\boldsymbol{f_c} = \boldsymbol{u}$ and $\Psi(\boldsymbol{c}) \leqslant d$. Without loss of generality, suppose

$$\boldsymbol{u} = (\underbrace{0,0,\ldots,0}_{d_1}, \underbrace{1,1,\ldots,1}_{d_2}, \ldots, \underbrace{0,0,\ldots,0}_{d_{2l-1}}, \underbrace{1,1,\ldots,1}_{d_{2l}}),$$

i.e., $\boldsymbol{u}$ has some runs of zeros of length $d_1, d_3, \ldots, d_{2l-1}$ and some runs of ones of length $d_2, d_4, \ldots, d_{2l}$, where l is a positive integer and $d_i \leqslant d-1$ for all $i \in [2l]$. Let $\boldsymbol{c} = (1, 1+\delta, \ldots, 1+(d_1-1)\delta, 1+\max\{d_1,d_2\}\cdot\delta, 1+(d_2-1)\delta, \ldots, 1+\delta, 1, \ldots, 1, 1+\delta, \ldots, 1+(d_{2l-1}-1)\delta, 1+\max\{d_{2l-1},d_{2l}\}\cdot\delta, 1+(d_{2l}-1)\delta, \ldots, 1+\delta)$, where δ is the minimum charge difference required to distinguish two distinct levels. That is to say, we denote the substring of $\boldsymbol{c}$ that generates a run of zeros of length d_{2i-1} as an increasing sequence $(1, 1+\delta, \ldots, 1+(d_{2i-1}-1)\delta)$ and denote the substring of $\boldsymbol{c}$ that generates a run of ones of length d_{2i} as a decreasing sequence $(1+\max\{d_{2i-1}, d_{2i}\}\cdot\delta, 1+(d_{2i}-1)\delta, \ldots, 1+\delta, 1)$ for all $i \in [l]$. From the construction of $\boldsymbol{c}$, $\Psi(\boldsymbol{c}) = \max_{i\in[2l]}\{d_i+1\}$ and $\boldsymbol{f_c} = \boldsymbol{u}$. Because $d_i \leqslant d-1$ for all $i \in [2l]$, we obtain $\Psi(\boldsymbol{c}) \leqslant d$. Therefore, $\boldsymbol{u} \in \mathcal{R}(1,2,n;d)$.

In [19], the author presented an enumeration of certain run-length sequences. Let $\boldsymbol{u} = (u_0, u_1, \ldots, u_{n-1}) \in \{0,1\}^n$ be a binary vector of length n. In the following definition, consider a class of runs of $\boldsymbol{u}$, $(u_i, \ldots, u_{j-1})$, where $0 \leqslant i \leqslant j \leqslant n$ (i.e., the indices $i-1, i, \ldots, j-1, j$ are not taken modulo n).

Definition 5. Suppose $\boldsymbol{u}$ is defined as above. Let $\mathcal{T}_{\boldsymbol{u}} = \{(u_i, \ldots, u_{j-1}) | (u_i, \ldots, u_{j-1})$ is a run of $\boldsymbol{u}$, and $i \leqslant j\}$. Denote $R_{\tilde{L}}(\boldsymbol{u})$ as the maximum run length of $\boldsymbol{u}$ in $\mathcal{T}_{\boldsymbol{u}}$.

For example, let $\boldsymbol{u} = (1,1,0,0,1,1)$ and $n = 6$. If the indices of runs are taken modulo n, then there exists a run of zeros of length 2 and a run of ones of length 4. Otherwise, we can obtain a run of zeros of length 2 and two runs of ones of length 2. From the definitions of $R_L(\boldsymbol{u})$ and $R_{\tilde{L}}(\boldsymbol{u})$, we have that $R_L(\boldsymbol{u}) = 4$ and $R_{\tilde{L}}(\boldsymbol{u}) = 2$. Therefore, $R_{\tilde{L}}(\boldsymbol{u}) \leqslant R_L(\boldsymbol{u})$.

For convenience, we denote $W(n,d) = \{\boldsymbol{u} \in \{0,1\}^n | R_{\tilde{L}}(\boldsymbol{u}) \leqslant d\}$ as the set of these run-length sequences. In [20], the author proposed a relationship between some compositions of n and sequences from $W(n,d)$. In the following explanation, the notation of composition will be introduced. A composition of the positive integer n is an ordered collection of positive integers, called parts, whose sum is n [19]. For convenience, let $C_d(n)$ be the number of compositions of n, no part of which is greater than d. Let $C_d^e(n)$ be the number of compositions of n with an even number of parts, no part of which is greater than d. Similarly, let $C_d^o(n)$ be the number of compositions of n with an odd number of parts, no part of which is greater than d.

Proposition 1 ([19])**.** Suppose that $C_d(n), C_d^e(n)$, and $C_d^o(n)$ are defined as above. Let $G_d(t), G_d^e(t)$, and $G_d^o(t)$ be the generating functions of these sequences, respectively (i.e., $G_d(t) = \sum_{n=1}^{\infty} C_d(n)t^n, G_d^e(t) = \sum_{n=1}^{\infty} C_d^e(n)t^n$, and $G_d^o(t) = \sum_{n=1}^{\infty} C_d^o(n)t^n$). Then, we have that

$$G_d(t) = \frac{t - t^{d+1}}{1 - 2t + t^{d+1}},\ G_d^e(t) = \frac{t^2(1-t^d)^2}{(1-t)^2 - t^2(1-t^d)^2},\ \text{and}\ G_d^o(t) = \frac{t(1-t)(1-t^d)}{(1-t)^2 - t^2(1-t^d)^2}. \tag{4}$$

For example, if $n = 5$ and $d = 2$, we can write all the compositions as follows: $(1,2,2), (2,1,2), (2,2,1), (2,1,1,1), (1,2,1,1), (1,1,2,1), (1,1,1,2), (1,1,1,1,1)$. Therefore, $C_2(5) = 8, C_2^e(5) = 4$, and $C_2^o(5) = 4$. From the relationship between certain compositions of n and sequences from $W(n,d)$, Kautz [20] demonstrated that $|W(n,d)| = 2C_d(n)$.

When $d = 2$, from (4), we have that

$$\begin{aligned}
G_2(t) &= \frac{t - t^3}{1 - 2t - t^3} = -1 + \frac{1}{1 - t - t^2} = -1 + \frac{1}{\left(1 - \frac{1-\sqrt{5}}{2}t\right)\left(1 - \frac{1+\sqrt{5}}{2}t\right)} \\
&= -1 + \frac{\sqrt{5}-1}{2\sqrt{5}} \cdot \frac{1}{1 - \frac{1-\sqrt{5}}{2}t} + \frac{\sqrt{5}+1}{2\sqrt{5}} \cdot \frac{1}{1 - \frac{1+\sqrt{5}}{2}t} \\
&\overset{(a)}{=} -1 + \frac{\sqrt{5}-1}{2\sqrt{5}} \cdot \sum_{i=0}^{\infty} \left(\frac{1-\sqrt{5}}{2}\right)^i t^i + \frac{\sqrt{5}+1}{2\sqrt{5}} \cdot \sum_{i=0}^{\infty} \left(\frac{1+\sqrt{5}}{2}\right)^i t^i \\
&= \sum_{i=1}^{\infty} \left(\frac{\sqrt{5}-1}{2\sqrt{5}} \cdot \left(\frac{1-\sqrt{5}}{2}\right)^i + \frac{\sqrt{5}+1}{2\sqrt{5}} \cdot \left(\frac{1+\sqrt{5}}{2}\right)^i \right) \cdot t^i \\
&= \frac{1}{\sqrt{5}} \cdot \sum_{i=1}^{\infty} \left(\left(\frac{1+\sqrt{5}}{2}\right)^{i+1} - \left(\frac{1-\sqrt{5}}{2}\right)^{i+1} \right) \cdot t^i
\end{aligned} \tag{5}$$

for all $|t| < \frac{\sqrt{5}-1}{2}$, where $\overset{(a)}{=}$ follows from the power-series expansion of $\frac{1}{1-x}$ for $x = \frac{1-\sqrt{5}}{2}t$ or $\frac{1+\sqrt{5}}{2}t$.

From (5), we have that

$$|W(n,2)| = 2C_2(n) = \frac{2}{\sqrt{5}} \left(\left(\frac{1+\sqrt{5}}{2}\right)^{n+1} - \left(\frac{1-\sqrt{5}}{2}\right)^{n+1} \right). \tag{6}$$

Therefore, from (6),

$$\lim_{n\to\infty} \frac{\log_2 |W(n,2)|}{n} = \log_2 \frac{1+\sqrt{5}}{2} \approx 0.6942.$$

For $d \geqslant 2$, Blake [19] derived the asymptotical property of $C_d(n)$ through the following lemma.

Lemma 2 ([19, Lemma])**.** Suppose that α_d is the largest root of the polynomial $f(t) = t^d - t^{d-1} - \cdots - t - 1$. Then, we have that

$$\lim_{n\to\infty} \frac{\log_2 |W(n,d)|}{n} = \log_2 \alpha_d.$$

The asymptotical property of α_d will be provided in the following lemma.

Lemma 3. Suppose that α_d is as defined in Lemma 2. Then, we have that $1 \leqslant \alpha_d \leqslant \alpha_{d+1} \leqslant 2$ for all $d \geqslant 2$ and $\lim_{d\to\infty} \alpha_d = 2$.

Proof. From the definition of $W(n,d)$, $W(n,d) \subseteq W(n,d+1)$. Therefore, $|W(n,d+1)| \geqslant |W(n,d)|$. From Lemma 2, $\alpha_d \leqslant \alpha_{d+1}$. From the definition of $f(t)$, we obtain that $f(t) = \frac{t^{d+1}-2t^d+1}{t-1}$ for all $t \neq 1$. Then, α_d is also the largest root of $t^{d+1} - 2t^d + 1$, denoted $g(t)$. From the definition of $g(t)$, $g(t) > 0$ for all $t \geqslant 2$ and $g(\frac{2d}{d+1}) < 0$. Then, we have that

$$\frac{2d}{d+1} < \alpha_d \leqslant 2. \tag{7}$$

Therefore, from (7), $1 \leqslant \alpha_d \leqslant \alpha_{d+1} \leqslant 2$ and $\lim_{d\to\infty} \alpha_d = 2$.

To discuss the asymptotical property of $|\mathcal{R}(1,2,n;d+1)|$, we present the following lemma.

Lemma 4. Suppose $\mathcal{R}(1,2,n;d+1)$ and $W(n,d)$ are defined as above. If $n \geqslant 2$, then

$$|W(n-d,d)| \leqslant |\mathcal{R}(1,2,n;d+1)| \leqslant |W(n,d)|.$$

Proof. First, we prove that $|\mathcal{R}(1,2,n;d+1)| \leqslant |W(n,d)|$. For any $\boldsymbol{u} \in \mathcal{R}(1,2,n;d+1)$, from Lemma 1, $R_L(\boldsymbol{u}) \leqslant d$. Then, $R_{\tilde{L}}(\boldsymbol{u}) \leqslant R_L(\boldsymbol{u}) \leqslant d$ and $\boldsymbol{u} \in W(n,d)$. Therefore, we obtain that $\mathcal{R}(1,2,n;d+1) \subseteq W(n,d)$ and $|\mathcal{R}(1,2,n;d+1)| \leqslant |W(n,d)|$.

Next, we prove that $|W(n-d,d)| \leqslant |\mathcal{R}(1,2,n;d+1)|$. Let us define the function $\phi : W(n-d,d) \to \mathcal{R}(1,2,n;d+1)$ in the following manner: for each $\boldsymbol{v} = (v_1, v_2, \ldots, v_{n-d}) \in W(n-d,d)$, we set $\boldsymbol{u} = (u_0, u_1, \ldots, u_{n-1}) = \phi(\boldsymbol{v})$ such that

$$u_j = \begin{cases} (1+v_1) \bmod 2, & \text{if } j = 0, \\ v_j, & \text{if } 1 \leqslant j \leqslant n-d, \\ (1+v_{n-d}) \bmod 2, & \text{if } n-d+1 \leqslant j \leqslant n-1. \end{cases} \tag{8}$$

From (8) and the definition of $\boldsymbol{v}$, it is easily verified that $\boldsymbol{u} \in \mathcal{R}(1,2,n;d+1)$ and ϕ is injective. Therefore, $|W(n-d,d)| \leqslant |\mathcal{R}(1,2,n;d+1)|$.

Based on Lemma 4, we will present one asymptotical property of $|\mathcal{R}(1,2,n;d+1)|$ in the following theorem.

Theorem 1. Suppose α_d is as defined in Lemma 2. Then, we obtain that

$$\lim_{n\to\infty} \frac{\log_2 |\mathcal{R}(1,2,n;d+1)|}{n} = \log_2 \alpha_d$$

for all $d \geqslant 2$.

Proof. From Lemmas 2 and 4,

$$\lim_{n\to\infty} \frac{\log_2 |\mathcal{R}(1,2,n;d+1)|}{n} \leqslant \lim_{n\to\infty} \frac{\log_2 |W(n,d)|}{n} = \log_2 \alpha_d$$

for all $d \geqslant 2$. Furthermore, we obtain that

$$\lim_{n\to\infty} \frac{\log_2 |\mathcal{R}(1,2,n;d+1)|}{n} \geqslant \lim_{n\to\infty} \frac{\log_2 |W(n-d,d)|}{n-d} \cdot \frac{n-d}{n} = \log_2 \alpha_d$$

for all $d \geqslant 2$. Therefore,

$$\lim_{n\to\infty} \frac{\log_2 |\mathcal{R}(1,2,n;d+1)|}{n} = \log_2 \alpha_d$$

for all $d \geqslant 2$.

4 Gray codes over certain run-length sequences

In this section, we will describe the construction of Gray codes for $(1, 2, n; d)$-LRM. We will propose Gray codes over certain run-length sequences $\mathcal{R}(1, 2, n; d)$ by utilizing "push-to-the-top" operations such that each code word is exactly realized by at most d distinct levels. Let $\boldsymbol{c} = (c_1, \ldots, c_n)$ be the charge levels of n cells and let $\boldsymbol{c}$ represent a vector $\boldsymbol{u}$ (i.e., $\boldsymbol{u} = \boldsymbol{f_c}$). Here, let the cell configuration of $\boldsymbol{u}$ be $\boldsymbol{c}$. Assume that δ is the minimum charge difference required to distinguish two distinct charge levels and that all charge levels are positive. For convenience, let $\delta = 1$. For each vector $\boldsymbol{u} \in \mathcal{R}(1, 2, n; d)$, we have the following property.

Lemma 5. For each vector $\boldsymbol{u} \in \mathcal{R}(1, 2, n; d)$, there must exist a vector $\boldsymbol{c} \in \mathbb{R}^n$ such that $\boldsymbol{u} = \boldsymbol{f_c}$ and $\Psi(\boldsymbol{c}) = n$.

Proof. Without loss of generality, let

$$\boldsymbol{u} = (\underbrace{0, \ldots, 0}_{d_1}, \underbrace{1, \ldots, 1}_{d_2}, \ldots, \underbrace{0, \ldots, 0}_{d_{2l-1}}, \underbrace{1, \ldots, 1}_{d_{2l}})$$

and $\sum_{i=1}^{2l} d_i = n$. We partition $[n]$ into $2l$ disjoint sets such that each set induces a run. Specifically, let

$$\boldsymbol{c} = \Bigg(\underbrace{1, 2, \ldots, d_1}_{d_1}, \underbrace{n, n-1, \ldots, n-d_2+1}_{d_2}, \underbrace{d_1+1, \ldots, d_1+d_2}_{d_2}, \ldots,$$

$$\underbrace{\sum_{i=1}^{l-1} d_{2i-1} + 1, \ldots, \sum_{i=1}^{l} d_{2i-1}}_{d_{2l-1}}, \underbrace{n - \sum_{i=1}^{l-1} d_{2i}, \ldots, n - \sum_{i=1}^{l} d_{2i} + 1}_{d_{2l}} \Bigg).$$

In other words, $\{1, 2, \ldots, \sum_{i=1}^{l} d_{2i-1}\}$ is divided into l disjoint sets such that each set can produce a run of 0s and $\{\sum_{i=1}^{l} d_{2i-1} + 1, \ldots, n\}$ is divided into l disjoint sets such that each set can produce a run of 1s. From the construction of $\boldsymbol{c}$, $\boldsymbol{u} = \boldsymbol{f_c}$ and $\Psi(\boldsymbol{c}) = n$.

For example, let $\boldsymbol{u} = (0, 0, 1, 0, 0, 1) \in \mathcal{R}(1, 2, 6; 3)$. Then, there exists a vector $\boldsymbol{c_0} = (1, 2, 6, 3, 4, 5)$ such that $\boldsymbol{u} = \boldsymbol{f_{c_0}}$ and $\Psi(\boldsymbol{c_0}) = 6$. Therefore, from Lemma 5, for any $\boldsymbol{u} \in \mathcal{R}(1, 2, n; d)$, $\boldsymbol{u}$ can be realized by n distinct levels. Let $\boldsymbol{v} = (0, 0, 1, 0, 1, 1) \in \mathcal{R}(1, 2, 6; 3)$ and let $\boldsymbol{c} = (1, 2, 3, 1, 2, 3)$ be the cell configuration of $\boldsymbol{u}$. Next, we can transform $\boldsymbol{u}$ into $\boldsymbol{v}$ by utilizing one "push-to-the-top" operation on the 5-th cell such that the cell configuration $\boldsymbol{c}$ is changed into another cell configuration $\hat{\boldsymbol{c}} = (1, 2, 3, 1, 4, 3)$. Then, $\hat{\boldsymbol{c}}$ is the cell configuration of $\boldsymbol{v}$ and $\Psi(\hat{\boldsymbol{c}}) > 3$. Therefore, for a Gray code over $\mathcal{R}(1, 2, n; d)$, the number of distinct components of the cell configuration of each code word may be greater than d. In order to construct a Gray code over $\mathcal{R}(1, 2, n; d)$ for $(1, 2, n; d)$-LRM, we must set a proper initial cell configuration and choose an appropriate set in $\mathcal{R}(1, 2, n; d)$, as well as a sequence of "push-to-the-top" operations. A rough description of one construction of a Gray code over $\mathcal{R}(1, 2, n; d)$ is provided below.

First, we consider $n = m \cdot n_1$ cells and partition the n cells into m blocks such that each block contains n_1 cells, where m and n_1 are positive integers. We denote the charge levels of the cells in block i as $\boldsymbol{c}_i$, where $\boldsymbol{c}_i = (c_{i,1}, \ldots, c_{i,n_1})$ for $i \in [m]$. Next, we will utilize $\boldsymbol{f_{c_i}}$ to represent the information for each block $\boldsymbol{c}_i$. Here, each block can be considered as an element of a binary vector set Σ of size M, denoted $\Sigma = \{\boldsymbol{u}_0, \ldots, \boldsymbol{u}_{M-1}\}$.

Now, consider any De-Bruijn sequence S of order $m-1$ over Σ. That is to say, S is a sequence of M^{m-1} elements $\boldsymbol{v}_{s_0}, \boldsymbol{v}_{s_1}, \ldots, \boldsymbol{v}_{s_{M^{m-1}-1}}$ over Σ such that the subsequences $\boldsymbol{v}_{s_i}, \boldsymbol{v}_{s_{i+1}}, \ldots, \boldsymbol{v}_{s_{i+m-2}}$ of S cover all the $(m-1)$-tuples of Σ exactly once, where the sub-indices of s are taken modulo M^{m-1}. In [21], Golomb proved that the sequence S exists.

In the following explanation, one construction of a Gray code G for $(1, 2, n; d)$-LRM is presented in two steps. First, we utilize the method proposed by En Gad et al. [11] to construct the anchor elements of G, defined as $\tilde{G} = \{\boldsymbol{g}_0, \boldsymbol{g}_1, \ldots, \boldsymbol{g}_{L-1}\}$, where $L = \mathrm{lcm}(m, M^{m-1})$. Furthermore, the elements of $\tilde{G}$ will

generate a Gray code over Σ^m. Specifically, let $\boldsymbol{g}_0$ be the first m elements of S. In the transition from $\boldsymbol{g}_i$ to $\boldsymbol{g}_{i+1}$, we transform $\underline{\boldsymbol{v}_{s_i}}$ into $\boldsymbol{v}_{s_{i+m}}$. A detailed description of the code $\tilde{G}$ is provided below.

$$\begin{aligned}
\boldsymbol{g}_0 &= \boldsymbol{v}_{s_{m-1}} \quad \boldsymbol{v}_{s_{m-2}} \quad \cdots \quad \boldsymbol{v}_{s_1} \quad \underline{\boldsymbol{v}_{s_0}},\\
\boldsymbol{g}_1 &= \boldsymbol{v}_{s_{m-1}} \quad \boldsymbol{v}_{s_{m-2}} \quad \cdots \quad \underline{\boldsymbol{v}_{s_1}} \quad \boldsymbol{v}_{s_m},\\
\boldsymbol{g}_2 &= \boldsymbol{v}_{s_{m-1}} \quad \boldsymbol{v}_{s_{m-2}} \quad \cdots \quad \boldsymbol{v}_{s_{m+1}} \quad \boldsymbol{v}_{s_m},\\
&\vdots\\
\boldsymbol{g}_{L-2} &= \boldsymbol{v}_{s_{L-1}} \quad \underline{\boldsymbol{v}_{s_{L-2}}} \quad \cdots \quad \boldsymbol{v}_{s_1} \quad \boldsymbol{v}_{s_0},\\
\boldsymbol{g}_{L-1} &= \underline{\boldsymbol{v}_{s_{L-1}}} \quad \boldsymbol{v}_{s_{m-2}} \quad \cdots \quad \boldsymbol{v}_{s_1} \quad \boldsymbol{v}_{s_0},
\end{aligned}$$

where the sub-indices of s are taken modulo M^{m-1}. The underlined block in each element is the block that is changed in the following element.

By utilizing the properties of the De-Bruijn Sequence S, En Gad et al. [11] proved that $\tilde{G}$ is a Gray code over Σ^m. However, $\tilde{G}$ is not a Gray code over $\mathcal{R}(1,2,n;d)$ because the transition between $\boldsymbol{g}_i$ and $\boldsymbol{g}_{i+1}$ may require many "push-to-the-top" operations.

Next, another code G can be obtained by adding different elements between each pair of adjacent elements from $\tilde{G}$ such that G is a Gray code over $\mathcal{R}(1,2,n;d)$.

The above two steps are a rough description of the construction of G. For each anchor element, it is crucial to identify which block is the underlined block. We then add two auxiliary bits to each block. Specifically, we add 00 to the underlined block and 01 to the other blocks.

In order to guarantee G for $(1,2,n;d)$-LRM, we consider the initial cell configuration of $\boldsymbol{g}_0$ and transitions between $\boldsymbol{g}_i$ and $\boldsymbol{g}_{i+1}$. Without loss of generality, we will only analyze the cell configuration of $\underline{\boldsymbol{v}_0}$ and the transitions from $\underline{\boldsymbol{v}_{s_0}}$ to $\boldsymbol{v}_{s_m}$. Let $\boldsymbol{c}^{(0)}$ and $\boldsymbol{c}^{(m)}$ be the cell configurations of $\underline{\boldsymbol{v}_{s_0}}$ and $\boldsymbol{v}_{s_m}$, respectively. Suppose $\boldsymbol{v}_{s_0}, \boldsymbol{v}_{s_m} \in \mathcal{R}(1,2,n_1;\hat{d})$, where $\hat{d} \geqslant 2$. Note that $\underline{\boldsymbol{v}_{s_0}}$ and $\boldsymbol{v}_{s_0}$ differ only in the final bit and that the last two bits of $\underline{\boldsymbol{v}_{s_0}}$ are 00. The last two bits 00 of $\underline{\boldsymbol{v}_{s_0}}$ and the first run of the right adjacent block of $\underline{\boldsymbol{v}_{s_0}}$ may generate a large run. To avoid this issue, let the first bit of each block be 1. Similarly, the first bit 1 of $\underline{\boldsymbol{v}_{s_0}}$ and the last run of the left adjacent block of $\underline{\boldsymbol{v}_{s_0}}$ may produce a large run. To avoid this issue, let the second bit of each block be 0.

For the initial cell configuration $\boldsymbol{c}^{(0)}$, we will set the cell configuration $\boldsymbol{c}^{(0)}$ based on all the runs of $\underline{\boldsymbol{v}_{s_0}}$. From the definition of $\boldsymbol{f}_{\boldsymbol{c}^{(0)}}$, a run of zeros (ones) corresponds to one increasing (decreasing) subsequence of $\boldsymbol{c}^{(0)}$. Therefore, the first 1 in a run of ones is induced by the locally highest cell and the first 0 in a run of zeros corresponds to the locally lowest cell charge level. In the following explanation, we will set the locally highest cell and locally lowest cell to be the highest cell and lowest cell among all cells, respectively. Then, let the second run of zeros in each block be the single longest run and let the first cell in each block be the second highest. Because the last two bits of each block except the underlined block are 01, the last cell among these blocks is the highest. Therefore, the last cell can yield 1 (the first cell in each block is the second highest). Here, because the last cell in the underlined block corresponds to 0, this cell must be lower than the first cell (the second highest). For example, when $n_1 = 8, \hat{d} = 4$, and $\underline{\boldsymbol{v}_{s_0}} = (1,0,0,0,1,1,0,0)$, we set the cell configuration $\boldsymbol{c}^{(0)} = (3,1,2,3,4,2,1,2)$. Based on the above explanation, we can choose an appropriate set Σ and set an initial cell configuration $\boldsymbol{c}^{(0)}$ utilizing Algorithm 1 for $(1,2,n;d)$-LRM.

We transform $\underline{\boldsymbol{v}_{s_0}}$ into $\boldsymbol{v}_{s_m}$ by utilizing a sequence of "push-to-the-top" operations such that the initial cell configuration $\boldsymbol{c}^{(0)}$ is changed into another cell configuration $\boldsymbol{c}^{(m)}$. In the following explanation, a rough description of the transitions from $\underline{\boldsymbol{v}_{s_0}}$ to $\boldsymbol{v}_{s_m}$ will be provided. In order to construct this sequence of operations, we first push all the locally lowest cells in $\boldsymbol{v}_{s_m}$ to the highest level in $\boldsymbol{c}^{(0)}$ by comparing them to the first cell in $\boldsymbol{c}^{(0)}$ (since the first cell in $\boldsymbol{c}^{(0)}$ is the second highest). Next, all other cells except the cells that are the highest in $\boldsymbol{v}_{s_m}$ and the first cell are pushed by utilizing runs of $\boldsymbol{v}_{s_m}$. We then push the first cell to the current highest level (i.e., it is the second highest in $\boldsymbol{c}^{(m)}$). Finally, all remaining cells are pushed to the highest level by comparing them to the first cell. Furthermore, Algorithm 2 is proposed to transform

Algorithm 1 Set the initial cell charge levels $\boldsymbol{c}_{(0,0)} = (c^1_{(0,0)}, c^2_{(0,0)}, \ldots, c^n_{(0,0)})$ such that $\boldsymbol{f}_{\boldsymbol{c}_{(0,0)}} = \boldsymbol{g}^0_0$ and $\boldsymbol{c}_{(0,0)} \in (0, \frac{d+1}{2}]^n$

Input: current cell configuration 0^n, the initial value $\boldsymbol{g}_0 = \boldsymbol{v}_{s_{m-1}} \boldsymbol{v}_{s_{m-2}} \cdots \boldsymbol{v}_{s_1} \underline{\boldsymbol{v}_{s_0}}$.

Output: new cell configuration $\boldsymbol{c}_{(0,0)} = (c^1_{(0,0)}, c^2_{(0,0)}, \ldots, c^n_{(0,0)})$.

$k \Leftarrow 0, T \Leftarrow \emptyset, i \Leftarrow 3, s \Leftarrow 0, t \Leftarrow 0, j \Leftarrow 0$

repeat

$c^{kn_1+2}_{(0,0)} \Leftarrow 1$ (let the lowest charge level be 1).

while $(3 \leqslant i \leqslant n_1)\{$

case 1: ($s = 0$ and $\boldsymbol{g}^0_0(kn_1 + i) = 0$)

$c^{kn_1+i}_{(0,0)} \Leftarrow c^{kn_1+i-1}_{(0,0)} + 1$ ($\delta = 1$ is the minimum charge difference required to distinguish two distinct charge levels and this cell corresponds the bit 0).

case 2: ($s = 0$ and $\boldsymbol{g}^0_0(kn_1 + i) = 1$)

$s \Leftarrow i, T \Leftarrow T \cup \{i\}$ (T is the set of cells with the highest level).

case 3: ($s \neq 0$ and $\boldsymbol{g}^0_0(kn_1 + i) = 0$)$\{$

$t \Leftarrow i, j \Leftarrow (t-1), c^{kn_1+i}_{(0,0)} \Leftarrow 1$ (this cell has the lowest level);

while $(s + 1 \leqslant j \leqslant t - 1)$

$c^{kn_1+j}_{(0,0)} \Leftarrow c^{kn_1+j+1}_{(0,0)} + 1, j \Leftarrow j - 1$ (this cell corresponds to the bit 1).

$s \Leftarrow 0.\}$

$i \Leftarrow i + 1.$

$\}$

$c^{kn_1+1}_{(0,0)} \Leftarrow c^{kn_1+1+d_1}_{(0,0)} + 1$ (push first cell in each block to the next-highest level);

while $(l \in T)\{$

$c^{kn_1+l}_{(0,0)} \Leftarrow \max\{c^{kn_1+1}_{(0,0)}, c^{kn_1+l-1}_{(0,0)}, c^{kn_1+l+1}_{(0,0)}\} + 1$ (push this cell to the highest level);

$T \Leftarrow T \setminus \{l\}\}$.

$i \Leftarrow 3, T \Leftarrow \emptyset, s \Leftarrow 0, k \Leftarrow k + 1$ (initially, set some parameters for block $k + 1$).

until $k = m - 1$.

$\underline{\boldsymbol{v}_{s_0}}$ into $\boldsymbol{v}_{s_m}$ as follows. Suppose that $n_1 = 8, \hat{d} = 4$, $\underline{\boldsymbol{v}_{s_0}} = (1,0,0,0,1,1,0,0)$, $\underline{\boldsymbol{v}_{s_m}} = (1,0,0,0,1,0,0,1)$, and $\boldsymbol{c}^{(0)} = (2,1,2,3,4,2,1,2)$. From the above explanation of the transformation from $\underline{\boldsymbol{v}_{s_0}}$ to $\boldsymbol{v}_{s_m}$, $\boldsymbol{c}^{(m)} = (6,4,5,6,7,5,6,7)$.

Because $\boldsymbol{v}_{s_0} \in \mathcal{R}(1,2,n_1;\hat{d})$, from the initial cell configuration $\boldsymbol{c}^{(0)}$, we have that $\boldsymbol{c}^{(0)} \in [1,\hat{d}]^{n_1}$ (i.e., the charge level of each cell is within $[1,\hat{d}]$). Through the transitions from $\underline{\boldsymbol{v}_{s_0}}$ to $\boldsymbol{v}_{s_m}$, we obtain that $\boldsymbol{c}^{(m)} \in [\hat{d}, 2\hat{d}-1]^{n_1}$. Therefore, to construct a Gray code G for $(1,2,n;d)$-LRM, let $d = 2\hat{d} - 1$. In the following explanation, one construction of a Gray code G over $\mathcal{R}(1,2,n;d)$ for $(1,2,n;d)$-LRM will be provided.

Construction. In the $(1,2,n;d)$-LRM scheme, let $n = m \cdot n_1$, $d = 2d_1 + 3$, $n_1 \geqslant 2d_1 + 4$, and $d_1 \geqslant 3$. Based on the above explanation, we choose an appropriate set Σ, where

$$\Sigma = \{(1, \underbrace{0, \ldots, 0}_{d_1+1}, 1, u_1, u_2, \ldots, u_{n_1-d_1-5}, 0, 1) | (u_1, u_2, \ldots, u_{n_1-d_1-5}) \in \mathcal{R}(1,2,n_1-d_1-5;d_1-1)\}.$$

Note that for each $\boldsymbol{v} \in \Sigma$, $\underline{\boldsymbol{v}}$ and $\boldsymbol{v}$ differ only in the final bit. Furthermore, the last two bits of $\underline{\boldsymbol{v}}$ are 00. From the construction of Σ, for each $\boldsymbol{v} \in \Sigma$, we obtain that $R_L(\boldsymbol{v}) = d_1 + 1$ and $R_L(\underline{\boldsymbol{v}}) = d_1 + 1$ (the second run of 0s of $\boldsymbol{v}$ or $\underline{\boldsymbol{v}}$ is the single longest run and its length is exactly $d_1 + 1$). Therefore, $|\Sigma| = |\mathcal{R}(1,2,n_1-d_1-5;d_1-1)|$. For convenience, let $|\mathcal{R}(1,2,n_1-d_1-5;d_1-1)| = M$, $\Sigma = \{\boldsymbol{u}_0, \boldsymbol{u}_1, \ldots, \boldsymbol{u}_{M-1}\}$, and $L = \mathrm{lcm}(m, M^{m-1})$.

Let S be a De-Bruijn sequence of order $m-1$ over Σ, $S = \boldsymbol{v}_{s_0}, \boldsymbol{v}_{s_1}, \ldots, \boldsymbol{v}_{s_{M^{m-1}-1}}$ (i.e., S is of length M^{m-1} and $\boldsymbol{v}_i \in \Sigma$). The Gray code $\tilde{G}$ of the anchor elements is a sequence of L vectors of length $n = mn_1$, denoted $\tilde{G} = \boldsymbol{g}_0, \boldsymbol{g}_1, \ldots, \boldsymbol{g}_{L-1}$. Specifically, let $\boldsymbol{g}_0$ be the first m elements of S, meaning

$$\boldsymbol{g}_0 = \boldsymbol{v}_{s_{m-1}} \quad \boldsymbol{v}_{s_{m-2}} \quad \cdots \quad \boldsymbol{v}_{s_1} \quad \underline{\boldsymbol{v}_{s_0}}.$$

Furthermore, in the transition from $\boldsymbol{g}_i$ to $\boldsymbol{g}_{i+1}$, we transform $\underline{\boldsymbol{v}_{s_i}}$ into $\boldsymbol{v}_{s_{i+m}}$. A detailed description of

Algorithm 2 Transform the cell configuration $\boldsymbol{c}_{s_i}$ into another cell configuration $\boldsymbol{c}_{s_{i+m}}$ such that $\boldsymbol{f}_{\boldsymbol{c}_{s_i}} = \boldsymbol{v}_{s_i}$ and $\boldsymbol{f}_{\boldsymbol{c}_{s_{i+m}}} = \boldsymbol{v}_{s_{i+m}}$

Input: Current cell configuration $\boldsymbol{c}_{s_i}$, the initial value $\boldsymbol{v}_{s_i}$, new block value $\boldsymbol{v}_{s_{i+m}}$.
Output: New cell configuration $\boldsymbol{c}_{s_{i+m}}$.
$k \Leftarrow 0, T \Leftarrow \emptyset, H_1 \Leftarrow \emptyset, H_2 \Leftarrow \emptyset, j \Leftarrow 3, s \Leftarrow 0, t \Leftarrow 0$
$\boldsymbol{c}_{s_{i+m}}(2) \Leftarrow \max\{\boldsymbol{c}_{s_i}(1), \boldsymbol{c}_{s_i}(3)\} + 1$ (push the 2nd cell to the highest level in $\boldsymbol{c}_{s_i}$).
while $(3 \leqslant j \leqslant n_1)\{$
  case 1: $(j \leqslant d_1 + 2)$
    $\boldsymbol{c}_{s_{i+m}}(j) \Leftarrow \max\{\boldsymbol{c}_{s_{i+m}}(j-1), \boldsymbol{c}_{s_i}(j+1)\} + 1$ (push the jth cell in a run of 0s).
  case 2: ($s = 0$ and $\boldsymbol{v}_{s_{i+m}}(j) = 1$)
    $s \Leftarrow j, T \Leftarrow T \cup \{j\}$ (this cell is the highest in $\boldsymbol{c}_{s_{i+m}}$).
    if $t \neq 0$ **then**
      $H_2 \Leftarrow H_2 \cup \{(t, s)\}$ (H_2 is a set of runs of 0s).
  case 3: ($s \neq 0$ and $\boldsymbol{v}_{s_{i+m}}(j) = 0$)$\{$
    $t \Leftarrow j$ (t represents the position of the lowest cell in $\boldsymbol{v}_{s_{i+m}}$).
    $H_1 \Leftarrow H_1 \cup \{(s, t)\}$ (H_1 is a set of runs of 1s).
    if $\boldsymbol{v}_{s_i}(j-1) = 0$ and $\boldsymbol{v}_{s_i}(j) = 1$ **then**
      $\boldsymbol{c}_{s_{i+m}}(j) \Leftarrow \boldsymbol{c}_{s_i}(j)$ (this is the highest cell in $\boldsymbol{c}_{s_i}$. Let the charge level of this cell remain unchanged);
    else
      $\boldsymbol{c}_{s_{i+m}}(j) \Leftarrow \max\{\boldsymbol{c}_{s_i}(1), \boldsymbol{c}_{s_i}(j-1), \boldsymbol{c}_{s_i}(j+1)\} + 1$.
    $s \Leftarrow 0\}$.
  $j \Leftarrow j + 1$.
$\}$
$\boldsymbol{c}_{s_{i+m}}(1) \Leftarrow \boldsymbol{c}_{s_{i+m}}(1 + d_1) + 1$ (push the first cell to next-highest level in the changed block);
while $\big(A = (s_1, t_1) \in H_1\big)\{$
  $k \Leftarrow (t_1 - 1)$
  if $(s_1 + 1 \leqslant k \leqslant t_1 - 1)$
    $\boldsymbol{c}_{s_{i+m}}(k) \Leftarrow \max\{\boldsymbol{c}_{s_i}(k-1), \boldsymbol{c}_{s_{i+m}}(k+1)\} + 1, k \Leftarrow (k-1)$; $H_1 \Leftarrow H_1 \setminus \{A\}\}$.
while $\big(B = (t_1, s_1) \in H_2\big)\{$
  $k \Leftarrow (t_1 + 1)$;
  if $(t_1 + 1 \leqslant k \leqslant s_1 - 1)$
    $\boldsymbol{c}_{s_{i+m}}(k) \Leftarrow \max\{\boldsymbol{c}_{s_{i+m}}(k-1), \boldsymbol{c}_{s_i}(k+1)\} + 1, k \Leftarrow (k+1)$; $H_2 \Leftarrow H_2 \setminus \{B\}\}$.
while $(l \in T)\{$
  $\boldsymbol{c}_{s_{i+m}}(l) \Leftarrow \max\{\boldsymbol{c}_{s_{i+m}}(1), \boldsymbol{c}_{s_{i+m}}(1-1), \boldsymbol{c}_{s_{i+m}}(1+1)\} + 1$ (let the charge level of this cell be the highest in the changed block); $T \Leftarrow T \setminus \{l\}\}$.

the code $\tilde{G}$ provided below.

$$
\begin{aligned}
\boldsymbol{g}_0 &= \boldsymbol{v}_{s_{m-1}} \quad \boldsymbol{v}_{s_{m-2}} \quad \cdots \quad \boldsymbol{v}_{s_1} \quad \underline{\boldsymbol{v}_{s_0}},\\
\boldsymbol{g}_1 &= \boldsymbol{v}_{s_{m-1}} \quad \boldsymbol{v}_{s_{m-2}} \quad \cdots \quad \underline{\boldsymbol{v}_{s_1}} \quad \boldsymbol{v}_{s_m},\\
&\vdots\\
\boldsymbol{g}_{L-2} &= \boldsymbol{v}_{s_{L-1}} \quad \underline{\boldsymbol{v}_{s_{L-2}}} \quad \cdots \quad \boldsymbol{v}_{s_1} \quad \boldsymbol{v}_{s_0},\\
\boldsymbol{g}_{L-1} &= \underline{\boldsymbol{v}_{s_{L-1}}} \quad \boldsymbol{v}_{s_{m-2}} \quad \cdots \quad \boldsymbol{v}_{s_1} \quad \boldsymbol{v}_{s_0},
\end{aligned}
$$

where the sub-indices of s are taken modulo M^{m-1}. The underlined block in each element is the block that is changed in the following element.

Next, a sequence of "push-to-the-top" operations are utilized to transform one anchor $\boldsymbol{g}_i$ into the next anchor $\boldsymbol{g}_{i+1}$. The sequence of "push-to-the-top" operations applied to the cells of the changing block will generate a sequence of auxiliary vectors, denoted $\boldsymbol{g}_i, \boldsymbol{g}_i^1, \boldsymbol{g}_i^2, \ldots, \boldsymbol{g}_i^{l_i}, i \in \{0, 1, 2, \ldots, L-1\}$. Therefore, the

entire Gray code G is constructed by the following sequence:

$$\begin{array}{llll} \boldsymbol{g}_0, & \boldsymbol{g}_0^1, & \boldsymbol{g}_0^2, & \cdots \quad \boldsymbol{g}_0^{l_0}, \\ \boldsymbol{g}_1, & \boldsymbol{g}_1^1, & \boldsymbol{g}_1^2, & \cdots \quad \boldsymbol{g}_1^{l_1}, \\ \vdots & & & \\ \boldsymbol{g}_{L-1}, & \boldsymbol{g}_{L-1}^1, & \boldsymbol{g}_{L-1}^2, & \cdots \quad \boldsymbol{g}_{L-1}^{l_{L-1}}. \end{array}$$

To complete the construction of G, we present Algorithm 1, which determines the initial cell configuration of $\boldsymbol{g}_0$. We also present Algorithm 2, which transforms $\underline{\boldsymbol{v}_{s_i}}$ into $\boldsymbol{v}_{s_{i+m}}$. During the process of this transformation, $\boldsymbol{v}_{s_{i+1}}$ (the left adjacent block of $\underline{\boldsymbol{v}_{s_i}}$) is transformed into $\underline{\boldsymbol{v}_{s_{i+1}}}$. In other words, Algorithm 2 specifies the sequence $\boldsymbol{g}_i^1, \boldsymbol{g}_i^2, \ldots, \boldsymbol{g}_i^{l_i}$ that allows us to move from one state $\boldsymbol{g}_i$ to another state $\boldsymbol{g}_{i+1}$.

In order to present Algorithms 1 and 2, some notations are provided. For convenience, let $\boldsymbol{g}_i^0 = \boldsymbol{g}_i$ for all $i \in [L-1]$. Let the cell configuration of $\boldsymbol{g}_i^j$ be $\boldsymbol{c}_{(i,j)} = (c_{(i,j)}^1, c_{(i,j)}^2, \ldots, c_{(i,j)}^n)$ for all $0 \leqslant i \leqslant L-1, 0 \leqslant j \leqslant l_i$.

For example, let $n = 20, n_1 = 10, d = 9$ and $\boldsymbol{g}_0 = (1,0,0,0,0,1,0,1,0,1,1,0,0,0,0,1,1,0,0,0)$. From Algorithm 1, $\boldsymbol{c}_{(0,0)} = (4,1,2,3,4,5,1,5,1,5,4,1,2,3,4,5,2,1,2,3)$. For convenience, let $\underline{\boldsymbol{v}_{s_i}} = (\boldsymbol{v}_{s_i}(1), \ldots, \boldsymbol{v}_{s_i}(n_1))$ and $\boldsymbol{v}_{s_{i+m}} = (\boldsymbol{v}_{s_{i+m}}(1), \ldots, \boldsymbol{v}_{s_{i+m}}(n_1))$. Let $\boldsymbol{c}_{s_i} = (\boldsymbol{c}_{s_i}(1), \ldots, \boldsymbol{c}_{s_i}(n_1))$ and $\boldsymbol{c}_{s_{i+m}} = (\boldsymbol{c}_{s_{i+m}}(1), \ldots, \boldsymbol{c}_{s_{i+m}}(n_1))$ be the cell configurations of $\underline{\boldsymbol{v}_{s_i}}$ and $\boldsymbol{v}_{s_{i+m}}$, respectively. Now, Algorithm 2 is defined as follows.

For example, let $n_1 = 10, d = 9, d_1 = 3, \boldsymbol{v}_{s_{i+m}} = (1,0,0,0,0,1,0,1,0,1), \underline{\boldsymbol{v}_{s_i}} = (1,0,0,0,0,1,1,0,0,0)$, and $\boldsymbol{c}_{s_i} = (4,1,2,3,4,5,2,1,2,3)$. From Algorithm 2, $T = \{6,8,10\}$, $H_1 = \{(6,7),(8,9)\}$, and $H_2 = \{(7,8),(9,10)\}$. Therefore, $\boldsymbol{c}_{s_{i+m}} = (8,5,6,7,8,9,6,9,5,9)$. In the following example, a Gray code over $\mathcal{R}(1,2,n;d)$ is derived by utilizing Algorithms 1 and 2.

Example 1. Let $d_1 = 3, d = 9, n_1 = 10$, and $m = 2$, then $n = 20$. From the definition of $\mathcal{R}(1,2,2;2)$, we have that $\mathcal{R}(1,2,2;2) = \{(0,1),(1,0)\}$. Therefore, $\Sigma = \{(1,0,0,0,0,1,0,1,0,1),(1,0,0,0,0,0,1,1,0,0,1)\}$. Then, we have that $M = |\Sigma| = 2$ and $L = \mathrm{lcm}(2,2^1) = 2$. For convenience, let $\boldsymbol{v}_0 = (1,0,0,0,0,0,1,1,0,0,1)$ and $\boldsymbol{v}_1 = (1,0,0,0,0,1,0,1,0,1)$. Therefore, $\Sigma = \{\boldsymbol{v}_i | 0 \leqslant i \leqslant 1\}$.

In order to construct $\tilde{G}_1$ over Σ, we utilize a De-Bruijn sequence of order 2 and an alphabet set of size 2, denoted S, where $S = \boldsymbol{v}_0, \boldsymbol{v}_1$. Then, the list of $\tilde{G}_1$ is $\boldsymbol{g}_0 = \boldsymbol{v}_1\ \underline{\boldsymbol{v}_0}$, $\boldsymbol{g}_1 = \underline{\boldsymbol{v}_1}\ \boldsymbol{v}_0$.

From the construction of $\boldsymbol{g}_0$. Then, $\boldsymbol{g}_0$ = 1000010101 $\underline{1000011000}$. Therefore, from Algorithm 1, $\boldsymbol{c}_{(0,0)} = (4,1,2,3,4,5,1,5,1,5,4,1,2,3,4,5,2,1,2,3)$. Furthermore, $\boldsymbol{g}_1 = \underline{1000010100}$ 1000011001. Then, we utilize Algorithm 2 to obtain that the transition from $\boldsymbol{g}_0$ to $\boldsymbol{g}_1$ is (changed digits are underlined):

$$\begin{aligned} \boldsymbol{g}_0^0 &= 1000010101\ \ 1000011000, \\ \boldsymbol{g}_0^1 &= 1000010101\ \ \underline{01}00011000, \\ \boldsymbol{g}_0^2 &= 1000010101\ \ 0\underline{01}0011000, \\ \boldsymbol{g}_0^3 &= 1000010101\ \ 00\underline{01}011000, \\ \boldsymbol{g}_0^4 &= 1000010101\ \ 000\underline{01}11000, \\ \boldsymbol{g}_0^5 &= 1000010101\ \ 0000\underline{11}0100, \\ \boldsymbol{g}_0^6 &= 1000010101\ \ 0000\underline{01}100, \\ \boldsymbol{g}_0^7 &= 1000010101\ \ 0000101\underline{01}0, \\ \boldsymbol{g}_0^8 &= 1000010100\ \ \underline{1}000101010, \\ \boldsymbol{g}_0^9 &= 1000010100\ \ 1000\underline{11}1010, \\ \boldsymbol{g}_1 &= 1000010100\ \ 1000011\underline{01}. \end{aligned}$$

Let $\boldsymbol{c}_1$ be the cell configuration of $\boldsymbol{g}_1$. From the above transitions and $\boldsymbol{c}_{(0,0)}$, $\boldsymbol{c}_1 = (4,1,2,3,4,5,1,5,1,5,8,5,6,7,8,9,6,5,6,9)$. Similarly, we can utilize Algorithm 2 to generate a sequence $\boldsymbol{g}_1^1, \boldsymbol{g}_1^2, \ldots, \boldsymbol{g}_1^{l_1}$ that allows us to transform from $\boldsymbol{g}_1$ into $\boldsymbol{g}_0$. Therefore, we obtain a Gray code $G_1 = (\boldsymbol{g}_0, \boldsymbol{g}_0^1, \ldots, \boldsymbol{g}_0^9, \boldsymbol{g}_1, \boldsymbol{g}_1^1, \ldots, \boldsymbol{g}_1^{l_1})$ over $\mathcal{R}(1,2,20;9)$ for $(1,2,20;9)$-LRM.

Now, we will prove that G is a Gray code over $\mathcal{R}(1,2,n;d)$ for $(1,2,n;d)$-LRM. In order to prove this result, we require the following lemmas.

Lemma 6. Algorithm 1 initially sets the cell configuration $\boldsymbol{c}_{(0,0)}$, which represents $\boldsymbol{g}_0$. Additionally, Algorithm 2 changes a block with the cell configuration $\boldsymbol{c}_{s_i}$ representing $\boldsymbol{v}_{s_i}$ into another block with the cell configuration $\boldsymbol{c}_{s_{i+m}}$ representing $\boldsymbol{v}_{s_{i+m}}$.

Proof. Let $\boldsymbol{c}_{(0,0)} = (c^1_{(0,0)}, c^2_{(0,0)}, \ldots, c^n_{(0,0)})$ and $\boldsymbol{g}_0 = (\boldsymbol{g}_0(1), \ldots, \boldsymbol{g}_0(n))$. When $\boldsymbol{g}_0(i) = 0$ and $\boldsymbol{g}_0(i+1) = 0$ for some $i \in [n]$, from Case 1 of Algorithm 1, we have $c^{i+1}_{(0,0)} > c^i_{(0,0)}$, which will represent 0 in the position i. If $\boldsymbol{g}_0(i) = 0$ and $\boldsymbol{g}_0(i+1) = 1$ for some $i \in [n]$, from Algorithm 1, the cell $i+1$ is pushed to the highest level. Then, we also have $c^{i+1}_{(0,0)} > c^i_{(0,0)}$, which will represent 0 in the position i. Similarly, if $\boldsymbol{g}_0(i) = 1$ for some $i \in [n]$, then $c^i_{(0,0)} > c^{i+1}_{(0,0)}$, which will represent 1 in the position i. Therefore, we utilize $\boldsymbol{c}_{(0,0)}$ to represent $\boldsymbol{g}_0$.

Similarly, Algorithm 2 can change the cell configuration $\boldsymbol{c}_{s_i}$ representing $\boldsymbol{v}_{s_i}$ into another cell configuration $\boldsymbol{c}_{s_{i+m}}$ representing $\boldsymbol{v}_{s_{i+m}}$.

Let the cell configuration of $\boldsymbol{g}_i$ be $(c^1_{(i,0)}, c^2_{(i,0)}, \ldots, c^n_{(i,0)})$, denoted $\boldsymbol{c}_{(i,0)}$, for all $0 \leqslant i \leqslant L-1$. Furthermore, let $i = km + t$, where $0 \leqslant k, 0 \leqslant t \leqslant m-1$.

Lemma 7. If $i = km$, then $\boldsymbol{c}_{(i,0)} \in (k\cdot\frac{d-1}{2}, k\cdot\frac{d-1}{2} + \frac{d+1}{2}]^n$. Furthermore, if $i = km+t$ and $0 < t \leqslant m-1$, then $\boldsymbol{c}_{(i,0)} \in (k\cdot\frac{d-1}{2}, k\cdot\frac{d-1}{2} + d]^n$.

Proof. Because $R_L(\boldsymbol{g}_0) = d_1 + 1$ (i.e., $\frac{d-1}{2}$), from Algorithm 1, we can obtain that the highest value of all the coordinates of $\boldsymbol{c}_{(0,0)}$ is $d_1 + 2$ and the lowest value of all the coordinates of $\boldsymbol{c}_{(0,0)}$ is 1. Therefore, $\boldsymbol{c}_{(0,0)} \in (0, \frac{d+1}{2}]^n$. To move from $\boldsymbol{g}_0$ to $\boldsymbol{g}_1$, we only need to change $\boldsymbol{v}_{s_0}$ into $\boldsymbol{v}_{s_m}$. For convenience, let $\boldsymbol{c}_{s_0} = (\boldsymbol{c}_{s_0}(1), \ldots, \boldsymbol{c}_{s_0}(n_1))$ and $\boldsymbol{c}_{s_m} = (\boldsymbol{c}_{s_m}(1), \ldots, \boldsymbol{c}_{s_m}(n_1))$ be the cell configurations of $\boldsymbol{v}_{s_0}$ and $\boldsymbol{v}_{s_m}$, respectively. From Algorithm 1, the first d_1+3 coordinates of $\boldsymbol{c}_{s_0}$ are $(1+d_1, 1, 2, \ldots, d_1+2)$. Furthermore, from Algorithm 2, we can change the cell configuration $\boldsymbol{c}_{s_0}$ into $\boldsymbol{c}_{s_m}$ such that the first d_1+3 coordinates of $\boldsymbol{c}_{s_m}$ are $(2d_1+2, d_1+2, d_1+3, \ldots, 2d_1+3)$. Note that Algorithm 2 pushes each cell of $\boldsymbol{c}_{s_0}$ to be higher than $\boldsymbol{c}_{s_0}(1)$ (i.e., d_1+1), except for the cells that are the highest level in $\boldsymbol{c}_{s_0}$, but which are the lowest level in $\boldsymbol{c}_{s_m}$. Therefore, the lowest value of the coordinates of $\boldsymbol{c}_{s_m}$ is $d_1 + 2$. Because $R_L(\boldsymbol{g}_1) = d_1 + 1$, the highest value of the coordinates of $\boldsymbol{c}_{s_m}$ is $2d_1 + 3$. Then, we have that $\boldsymbol{c}_{s_m} \in (d_1+1, 2d_1+3]^n$ (i.e., $(\frac{d-1}{2}, d]^n$). Therefore, $\boldsymbol{c}_{(1,0)} \in (0, d]^n$.

When $1 \leqslant i \leqslant m-1$, we can transform $\boldsymbol{g}_i$ into $\boldsymbol{g}_{i+1}$ by changing only $\boldsymbol{v}_{s_i}$ into $\boldsymbol{v}_{s_{i+m}}$. Similarly, $\boldsymbol{c}_{s_i} \in (0, \frac{d-1}{2}]^n$ and $\boldsymbol{c}_{s_{i+m}} \in (\frac{d-1}{2}, d]^n$. Then, we have that $\boldsymbol{c}_{(i,0)} \in (0, d]^n$ for all $1 \leqslant i \leqslant m-1$. Furthermore, $\boldsymbol{c}_{(m,0)} \in (\frac{d-1}{2}, d]^n$.

Similarly, by induction, we have that $\boldsymbol{c}_{(i,0)} \in (k\cdot\frac{d-1}{2}, k\cdot\frac{d-1}{2} + \frac{d+1}{2}]^n$ for all $i = km$ and $\boldsymbol{c}_{(i,0)} \in (k\cdot\frac{d-1}{2}, k\cdot\frac{d-1}{2} + d]^n$ for all $i = km + t$ and $0 < t \leqslant m-1$.

From Lemma 7, for each anchor $\boldsymbol{g}_i$, $0 \leqslant i \leqslant L-1$, the number of distinct coordinates of the cell configuration $\boldsymbol{c}_{(i,0)}$ is at most d. Between $\boldsymbol{g}_i$ and $\boldsymbol{g}_{i+1}$, the auxiliary vectors $\boldsymbol{g}^1_i, \boldsymbol{g}^1_i, \ldots, \boldsymbol{g}^{l_i}_i$ can be obtained by Algorithm 2. Let $\boldsymbol{c}_{(i,j)}$ be the cell configuration of $\boldsymbol{g}^j_i$ for all $1 \leqslant i \leqslant l_i$. Then, we can obtain $\Psi(\boldsymbol{c}_{(i,j)}) \leqslant d$ ($\Psi(\cdot)$ represents the number of distinct components of some vector) through the following corollary.

Corollary 1. Suppose $\boldsymbol{c}_{(i,j)}$ is as defined above for all $0 \leqslant i \leqslant L-1$ and $0 \leqslant j \leqslant l_i$. Then, we have that $\Psi(\boldsymbol{c}_{(i,j)}) \leqslant d$ for all $0 \leqslant i \leqslant L-1, 0 \leqslant j \leqslant l_i$.

Proof. If $i = km$, then, by Lemma 7, $\boldsymbol{c}_{(i,0)} \in (k\cdot\frac{d-1}{2}, k\cdot\frac{d-1}{2} + \frac{d+1}{2}]^n$ and $\boldsymbol{c}_{(i+1,0)} \in (k\cdot\frac{d-1}{2}, k\cdot\frac{d-1}{2} + d]^n$. We can change $\boldsymbol{c}_{(i,0)}$ into $\boldsymbol{c}_{(i+1,0)}$ by utilizing a sequence of $\boldsymbol{c}_{(i,1)}, \boldsymbol{c}_{(i,2)}, \ldots, \boldsymbol{c}_{(i,l_i)}$. Therefore, $\boldsymbol{c}_{(i,j)} \in (k\cdot\frac{d-1}{2}, k\cdot\frac{d-1}{2} + d]^n$ for all $1 \leqslant j \leqslant l_i$. Similarly, if $i = km + t$ and $1 \leqslant t \leqslant m-1$, then $\boldsymbol{c}_{(i,j)} \in (k\cdot\frac{d-1}{2}, k\cdot\frac{d-1}{2} + d]^n$ for all $1 \leqslant j \leqslant l_i$. Because $(k\cdot\frac{d-1}{2}, k\cdot\frac{d-1}{2} + d]$ represents at most d distinct values, we have that $N_d(\boldsymbol{c}_{(i,j)}) \leqslant d$ for all $0 \leqslant i \leqslant L-1$ and $0 \leqslant j \leqslant l_i$.

From Corollary 1, G is a code over $\mathcal{R}(1,2,n;d)$ for $(1,2,n;d)$-LRM. In the following proof, we will only prove that G is a Gray code.

Lemma 8. All the anchors of G are distinct.

Proof. Through the construction of all the anchors of G, we can determine the underlined block that

will be changed. Specifically, the last two bits of the underlined block are 00. From the properties of the De-Bruijn sequence S, all the anchors of G are distinct.

Lemma 9. All the code words in the code G are distinct.

Proof. For each non-anchor code, there exists a block with a first bit of 0 or two blocks with last bits of 10. However, for every anchor codes, there exists only a block with the last bits 00 and the first bit of each block is 1. Therefore, we can distinguish anchor codes from non-anchor codes. From the De-Bruijn sequence S, for all $0 \leqslant i_1, i_2 \leqslant L-1$, $0 \leqslant j_1 \leqslant l_{i_1}$, and $0 \leqslant j_2 \leqslant l_{i_2}$. If $i_1 \neq i_2$, then $\boldsymbol{g}_{i_1}^{j_1} \neq \boldsymbol{g}_{i_2}^{j_2}$, where $\boldsymbol{g}_i^0 = \boldsymbol{g}_i$ for all $0 \leqslant i \leqslant L-1$.

Now, we only need to prove that for all $0 \leqslant i \leqslant L-1$ and $1 \leqslant j_1 < j_2 \leqslant l_i$, we have $\boldsymbol{g}_i^{j_1} \neq \boldsymbol{g}_i^{j_2}$. By utilizing Algorithm 2, we transform $\boldsymbol{g}_i$ into $\boldsymbol{g}_{i+1}$ by transforming $\boldsymbol{v}_{s_i}$ into $\boldsymbol{v}_{s_{i+m}}$. For convenience, let $\boldsymbol{v}_i^j$ be the changing block of $\boldsymbol{g}_i^j$ for all $1 \leqslant j \leqslant l_i$. Assume that the k-th "push-to-the-top" operation is applied to the first cell of the changed block. If $1 \leqslant j < k$, then $\boldsymbol{v}_i^j(1) = 0$. Otherwise, if $k \leqslant j \leqslant l_i$, then $\boldsymbol{v}_i^j(1) = 1$. Therefore, if $j_1 < k \leqslant j_2$, then $\boldsymbol{g}_i^{j_1} \neq \boldsymbol{g}_i^{j_2}$.

Prior to the k-th operation (the "push-to-the-top" operation), we push each cell that is the lowest level in $\boldsymbol{v}_{s_{i+m}}$, except for cells whose positions are between 3 and d_1+2 (here, the lowest level cell is lower than its two adjacent cells). For all of the first $k-1$ elements, given a cell configuration, let j be the next cell that will be pushed, where c_j is the current charge level of cell j. Because cell j will be pushed, there exists one adjacent cell $\hat{j}$ such that $c_{\hat{j}} \geqslant c_j$, where $c_{\hat{j}}$ is the charge level of cell $\hat{j}$. Following the "push-to-the-top" operation, $\tilde{c}_j > c_{\hat{j}}$, where $\tilde{c}_j$ is the changed level of cell j. Because there exists at least one cell between the two lowest levels in $\boldsymbol{v}_{s_{i+m}}$, cell $\hat{j}$ will not be pushed in the first $k-1$ operations. Therefore, $\boldsymbol{g}_i^{j_1} \neq \boldsymbol{g}_i^{j_2}$ for all $1 \leqslant j_1 < j_2 \leqslant k-1$.

After all of the lowest cells in $\boldsymbol{v}_{s_{i+m}}$ are pushed, if cell j is pushed, then there exists one cell $\hat{j}$ that is higher than cell j and cell $\hat{j}$ will not be pushed. Similarly, we also have that $\boldsymbol{g}_i^{j_1} \neq \boldsymbol{g}_i^{j_2}$ for all $k \leqslant j_1 < j_2 \leqslant l_i$. Therefore, $\boldsymbol{g}_i^{j_1} \neq \boldsymbol{g}_i^{j_2}$ for all $1 \leqslant j_1 < j_2 \leqslant l_i$, meaning all the code words in the code G are distinct.

By combining Corollary 1 and Lemma 9, we present the following theorem.

Theorem 2. The code G is a Gray code of size N for $(1, 2, n; d)$-LRM, where $L \leqslant N \leqslant n_1 L$.

Proof. First, by combining Corollary 1 and Lemma 9, the code G is a Gray code of size at least L for $(1, 2, n; d)$-LRM. Furthermore, we can transform one anchor into another anchor by changing a block and each cell in the changed block is pushed at most once. Because the length of each block is n_1, the number of auxiliary vectors between any two adjacent anchors is at most n_1. Therefore, $L \leqslant N \leqslant n_1 L$.

Next, we consider the asymptotical property of the Gray code G in the following theorem. For convenience, let d_1 and m be integers such that $d_1 \geqslant 3$ and $m \geqslant 2$, and let $d = 2d_1 + 3$. Suppose that $\{n_1^{(i)}\}_{i=1}^{\infty}$ and $\{n^{(i)}\}_{i=1}^{\infty}$ are infinite integer sequences such that $n_1^{(i+1)} > n_1^{(i)}$, $n_1^{(1)} \geqslant 2d_1 + 4$, and $n^{(i)} = m \cdot n_1^{(i)}$. Furthermore, α_{d_1-2} in Theorem 3 is defined in Lemma 2.

Theorem 3. There exists a family of codes, denoted $\{G_i\}_{i=1}^{\infty}$, where G_i is a Gray code over $\mathcal{R}(1, 2, n^{(i)}; d)$ of size N_i for $(1, 2, n; d)$-LRM. Therefore, we have that

$$\lim_{i\to\infty} R(G_i) = \lim_{i\to\infty} \frac{\log_2 N_i}{n^{(i)}} = \frac{m-1}{m} \log_2 \alpha_{d_1-2}, \tag{9}$$

where $R(G_i)$ is the rate of G_i defined in Definition 2.

Proof. From the construction of L and M in the Gray code G, L_i and M_i are defined in the construction of G_i. Then, $N_i \geqslant L_i = \mathrm{lcm}(m, M_i^{m-1})$ and $M_i = |\mathcal{R}(1, 2, n_1^{(i)} - d_1 - 5; d_1 - 1)|$, and we obtain that

$$\begin{aligned}
\lim_{i\to\infty} \frac{\log_2 N_i}{n^{(i)}} &\geqslant \lim_{i\to\infty} \frac{(m-1)\log_2 |\mathcal{R}(1, 2, n_1^{(i)} - d_1 - 5; d_1 - 1)|}{m n_1^{(i)}} \\
&= \lim_{i\to\infty} \frac{(m-1)\cdot(n_1^{(i)} - d_1 - 5)}{m n_1^{(i)}} \cdot \frac{\log_2 |\mathcal{R}(1, 2, n_1^{(i)} - d_1 - 5; d_1 - 1)|}{n_1^{(i)} - d_1 - 5} \\
&\overset{\text{(b)}}{=} \frac{m-1}{m} \log_2 \alpha_{d_1-2},
\end{aligned}$$

Table 1 Parameters of G, G_1, and G_2

Gray code	n	N	$\Psi(\cdot)$	Parameter constraint
G_1 [12]	m^2	$\mathrm{lcm}(\binom{m}{w}^{m-2}, m)\cdot(w+2)\cdot m$	$\geqslant m$	$m \geqslant w, w$ is a positive integer
G_2 [11]	m^2	$O(2^{(m-1)(m-3)})$	$\geqslant m$	$m \geqslant 4$
G	m^2	$O(2^{m^2\cdot\log_2\alpha_d})$	d	$d = 2d_1+3, d_1 \geqslant 3, m \geqslant 2d_1+4$, α_d is defined in Lemma 2

where $\overset{\text{(b)}}{=}$ follows from Theorem 1. Similarly, $\lim_{i\to\infty}\frac{\log_2 N_i}{n^{(i)}} \leqslant \lim_{i\to\infty}\frac{\log_2(n_1 L_i)}{n^{(i)}} = \frac{m-1}{m}\log_2\alpha_{d_1-2}$. Therefore, $\lim_{i\to\infty} R(G_i) = \frac{m-1}{m}\log_2\alpha_{d_1-2}$.

Furthermore, suppose $\{m_i\}_{i=1}^{\infty}$, $\{d_1^{(i)}\}_{i=1}^{\infty}$, and $\{d^{(i)}\}_{i=1}^{\infty}$ are integer sequences such that $m_{i+1} > m_i$, $d_1^{(i+1)} > d_1^{(i)}$, $n_1^{(i)} \geqslant 2d_1^{(i)}+4$, and $d^{(i)} = 2d_1^{(i)}+3$. Furthermore, let $n^{(i)} = m_i \cdot n_1^{(i)}$ and

$$\lim_{i\to\infty}\frac{d_1^{(i)}}{n_1^{(i)}} = 0.$$

Corollary 2. Suppose $\{G_i\}_{i=1}^{\infty}$ is as defined in Theorem 3, where G_i is a Gray code over $\mathcal{R}(1,2,n^{(i)};d^{(i)})$ of size N_i for $(1,2,n^{(i)};d^{(i)})$-LRM. Then, we have that $\lim_{i\to\infty} R(G_i) = 1$, where $R(G_i)$ is the rate of G_i defined in Definition 2.

Proof. From Theorem 3 and Lemma 3, we can obtain that $\lim_{i\to\infty} R(G_i) = 1$.

Next, we compare our results to those in [11, 12]. En Gad et al. [12] presented constant-weight Gray codes for the $(1,2,n)$-LRM scheme and [11] proposed generalized Gray codes for the (s,t,n)-LRM scheme (here, consider $s=1$ and $t=2$). When $n=m^2$, we partition the n cells into m blocks such that each block contains m cells. In this case, for the above two classes of Gray codes, there always exists an anchor code, denoted $\boldsymbol{v} = \boldsymbol{v}_m\ \boldsymbol{v}_{m-1}\ \cdots\ \boldsymbol{v}_1$, such that

$$\boldsymbol{v}_k = (\underbrace{0,\ldots,0}_{m})$$

for some $k \in [m]$. Therefore, $R_L(\boldsymbol{v}) \geqslant \sqrt{n}$, where $R_L(\boldsymbol{v})$ represents the maximum run length of $\boldsymbol{v}$. Furthermore, let $\boldsymbol{c_v} \in \mathbb{R}^n$ be the cell configuration of $\boldsymbol{v}$. That is to say, $\boldsymbol{f}_{\boldsymbol{c_v}} = \boldsymbol{v}$. Because $R_L(\boldsymbol{v}) \geqslant \sqrt{n}$, from Lemma 1, we can obtain $\Psi(\boldsymbol{c_v}) \geqslant \sqrt{n}$, where $\Psi(\boldsymbol{c_v})$ is the number of distinct components of $\boldsymbol{c_v}$. Therefore, to construct the two classes of Gray codes proposed by En Gad et al. [11, 12], there exists some code word that requires a large number of distinct charge levels.

However, in the $(1,2,n;d)$-LRM scheme, we can construct a Gray code G such that $\Psi(\boldsymbol{c_u}) \leqslant d$ for each code word $\boldsymbol{u} \in G$, where $\boldsymbol{c_u}$ is the cell configuration of $\boldsymbol{u}$. In other words, for each code word of G, at most d distinct charge levels are required.

Furthermore, En Gad et al. [12] chose a set Σ_1 to construct a Gray code G_1 with length $n=m^2$ of size N_1, where Σ_1 is a set of constant-weight code words of length m and weight w, and $N_1 = \mathrm{lcm}(\binom{m}{w}^{m-2}, m)\cdot(w+2)\cdot m$. En Gad et al. [11] also utilized another set Σ_2 to construct another Gray code G_2 with length $n=m^2$ of size N_2, where $\Sigma_2 = \mathcal{R}(1,2,m-3;m-3)\bigcup\{0^{m-3},1^{m-3}\}$ and $\mathrm{lcm}\big(2^{(m-3)(m-1)},m\big) \leqslant N_2 \leqslant m\cdot\mathrm{lcm}\big(2^{(m-3)(m-1)},m\big)$. In this study, we chose a set Σ to construct the Gray code G with length $n=m^2$ of size N_0, where $\mathrm{lcm}\big(|\mathcal{R}(1,2,m-d_1-5;d_1-1)|^{m-1},m\big) \leqslant N_0 \leqslant m\cdot\mathrm{lcm}\big(|\mathcal{R}(1,2,m-d_1-5;d_1-1)|^{m-1},m\big)$. Because $|\mathcal{R}(1,2,m-d_1-5;d_1-1)| < 2^{m-d_1-5}$ and $m^2 < 2^{2(m-1)}$, we obtain that $N_0 \leqslant m^2\cdot|\mathcal{R}(1,2,m-d_1-5;d_1-1)|^{m-1} < 2^{(m-3)(m-1)} \leqslant N_2$. Therefore, the size of G_2 is larger than the size of G. Table 1 summarizes some parameters of our Gray code G, as well as the Gray codes G_1 [12] and G_2 [11], where n is the length of a code, N is the size of a code, and $\Psi(\cdot)$ is the number of distinct levels that a code requires.

From the definition of α_d in Lemma 3, we obtain $\log_2\alpha_2 = 0.6942$, $\log_2\alpha_3 = 0.8543$, $\log_2\alpha_4 = 0.9468$, and $\log_2\alpha_5 = 0.9752$. When $d_1 = 7$ and $d = 2d_1+3 = 17$, from Theorem 3, there exists a family of $\{G_i\}_{i=1}^{\infty}$ such that its asymptotical rate is at least 0.9752, where G_i is a Gray code over $\mathcal{R}(1,2,n^{(i)};17)$ for the $(1,2,n^{(i)};17)$-LRM. Therefore, when $d=17$ (i.e, d is very small), there exists a family of Gray codes such that its asymptotical rate is close to 1. If $d_1^{(i)} = \lceil\log_2 n^{(i)}\rceil$ and $d^{(i)} = 2\lceil\log_2 n^{(i)}\rceil + 3$, from Corollary 2, there exists a family of $\{G_i\}_{i=1}^{\infty}$ such that its asymptotical rate is 1, where G_i is a Gray code

Table 2 Rates of our Gray code G with respect to d

d	$R = \frac{\log_2 N}{n}$
13	$\log_2 \alpha_3 = 0.8543$
15	$\log_2 \alpha_4 = 0.9468$
17	$\log_2 \alpha_5 = 0.9752$
$d \to \infty$	1

over $\mathcal{R}(1, 2, n^{(i)}; d^{(i)})$ for the $(1, 2, n^{(i)}; d^{(i)})$-LRM. Therefore, when d is very large, there exists a family of Gray codes such that its asymptotical rate is 1. Table 2 summarizes the rates of our Gray code G with respect to d, where $n \to \infty$.

5 Conclusion

For the $(1, 2, n)$-LRM scheme, we studied Gray codes over certain run-length sequences such that each code word is exactly realized by the bounded number of distinct charge levels. In this paper, we established a relationship between certain run-length sequences and $(1, 2, n; d)$-LRM realizable permutation sequences, and proposed an asymptotical property of $|\mathcal{R}(1, 2, n; d)|$. Furthermore, we presented Gray codes over $\mathcal{R}(1, 2, n; d)$ for $(1, 2, n; d)$-LRM by utilizing a De-Bruijn sequence. Finally, we constructed a family of Gray codes such that its asymptotical rate is 1.

Acknowledgements This work was supported by National Basic Research Program of China (973) (Grant No. 2013CB834204) and National Natural Science Foundation of China (Grant No. 61571243).

References

1 Jiang A X, Mateescu R, Schwartz M, et al. Rank modulation for flash memories. IEEE Trans Inform Theor, 2009, 55: 2659–2673
2 Jiang A X, Schwartz M, Bruck J. Correcting charge-constrained errors in the rank-modulation scheme. IEEE Trans Inform Theor, 2010, 56: 2112–2120
3 Tamo I, Schwartz M. Correcting limited-magnitude errors in the rank-modulation scheme. IEEE Trans Inform Theor, 2010, 56: 2551–2560
4 Wang Z Y, Jiang A X, Bruck J. On the capacity of bounded rank modulation for flash memories. In: Proceedings of IEEE International Symposium Information Theory, Seoul, 2009. 1234–1238
5 Gray F. Pulse Code Communication. U. S. Patent, 2632058, 1953, 17
6 Chang C C, Chen H Y, Chen C Y. Symbolic gray code as a data allocation scheme for two-disc systems. Comput J, 1992, 35: 299–305
7 Chen M S, Shin K G. Subcube allocation and task migration in hypercube multiprocessors. IEEE Trans Comput, 1990, 39: 1146–1155
8 Ludman J. Gray code generation for MPSK signals. IEEE Trans Commun, 1981, 29: 1519–1522
9 Richards D. Data compression and Gray-code sorting. Inf Process Lett, 1986, 22: 201–205
10 Savage C D. A survey of combinatorial Gray codes. SIAM Rev, 1997, 39: 605–629
11 En Gad E, Langberg M, Schwartz M, et al. Generalized Gray codes for local rank modulation. IEEE Trans Inform Theor, 2013, 59: 6664–6673
12 En Gad E, Langberg M, Schwartz M, et al. Constant-weight Gray codes for local rank modulation. IEEE Trans Inform Theor, 2011, 57: 7431–7442
13 Farnoud F, Skachek V, Milenkovic O. Error-correction in flash memories via codes in the Ulam metric. IEEE Trans Inform Theor, 2013, 59: 3003–3020
14 Horovitz M, Etzion T. Constructions of snake-in-the-box codes for rank modulation. IEEE Trans Inform Theor, 2014, 60: 7016–7025
15 Ge G N, Zhang Y W. Snake-in-the-box codes for rank modulation under Kendall's τ -metric. IEEE Trans Inform Theor, 2016, 62: 151–158
16 Yehezkeally Y, Schwartz M. Snake-in-the-box codes for rank modulation. IEEE Trans Inform Theor, 2012, 58: 5471–5483
17 Laisant C A. Sur la numération factorielle, application aux permutations. Bull de la Société Mathématiq. de France, 1888, 16: 176–183
18 Ferreira H C, Vinck A J H, Swart T G, et al. Permutation trellis codes. IEEE Trans Commun, 2005, 53: 1782–1789
19 Blake I F. The enumeration of certain run length sequences. Inf Control, 1982, 55: 222–237
20 Kautz W H. Fibonacci codes for synchronization control. IEEE Trans Inform Theor, 1965, 11: 284–292
21 Golomb S W. Shift Register Sequences. San Francisco: Holden-Day, 1967