# Low-Power Cooling Codes With Efficient Encoding and Decoding

Yeow Meng Chee, Tuvi Etzion, *Fellow, IEEE*, Han Mao Kiah, *Member, IEEE*, Alexander Vardy, *Fellow, IEEE*, and Hengjia Wei

***Abstract*— In a bus with $n$ wires, each wire has two states, '0' or '1', representing one bit of information. Whenever the state transitions from '0' to '1', or '1' to '0', joule heating causes the temperature to rise, and high temperatures have adverse effects on on-chip bus performance. Recently, the class of low-power cooling (LPC) codes was proposed to control such state transitions during each transmission. As suggested in earlier work, LPC codes may be used to control simultaneously both the peak temperature and the average power consumption of on-chip buses. Specifically, an $(n, t, w)$-LPC code is a coding scheme over $n$ wires that (i) avoids state transitions on the $t$ hottest wires (thus preventing the peak temperature from rising); and (ii) allows at most $w$ state transitions in each transmission (thus reducing average power consumption). In this paper, for any fixed value of $w$, several constructions are presented for large LPC codes that can be encoded and decoded in time $O(n \log^2(n/w))$ along with the corresponding encoding/decoding schemes. In particular, we construct LPC codes of size $(n/w)^{w-1}$, which are asymptotically optimal. We then modify these LPC codes to also correct errors in time $O(n^3)$. For the case where $w$ is proportional to $n$, we further present a different construction of large LPC codes, based on a mapping from cooling codes to LPC codes. Using this construction, we obtain two families of LPC codes whose encoding and decoding complexities are $O(n^3)$.**

***Index Terms*— Cooling codes, low-power cooling (LPC) codes, thermal-management coding.**

## I. Introduction

POWER and heat dissipation have emerged as first-order design constraints for chips, whether targeted for battery-powered devices or for high-end systems. High temperatures have dramatic negative effects on bus performance. Power-aware design alone is insufficient to address the thermal challenges, since it does not directly target the spatial and temporal behavior of the operating environment. For this reason, thermally-aware approaches have emerged as one of the most important domains of research in chip design today. Numerous techniques have been proposed to reduce the overall power consumption of on-chip buses (see [3] which uses coding techniques and the references therein using non-coding techniques). However, all the non-coding techniques do not directly address peak temperature minimization.

In an $n$-bit bus, each of the $n$ wires is charged to one of two voltages, representing the two *states* '0' and '1'. When the state is switched from '0' to '1', or '1' to '0', joule heating causes temperature to rise regardless of the direction of current flow (see [21] for an analysis). In other words, the temperature of a wire increases whenever the wire undergoes a state transition; conversely, in the absence of state transitions, the temperature will gradually decrease.

*Remark 1:* In the literature, there are also differing thermal models (see [16], [17]), wherein the heating rate and energy consumption of a '$0 \rightarrow 1$' transition differ from that of a '$1 \rightarrow 0$' transition. However, analysis of such models is beyond the scope of this paper, and is deferred to future work.

Recently, a new class of codes, called *cooling codes*, was introduced in [3] to directly control the peak temperature of a bus by cooling its hottest wires. This is achieved by avoiding state transitions on the hottest wires for as long as necessary until their temperature drops off. Cooling codes are based on differential encoding. Specifically, if the current state of the wires is $(s_1, s_2, \ldots, s_n)$, i.e., wire $i$ is in state $s_i$, and we want to transmit the binary vector $(x_1, x_2, \ldots, x_n)$, we set the wires to the state $(s'_1, s'_2, \ldots, s'_n)$, where $s'_i$ is equal to $s_i + x_i$ modulo 2 for $1 \leq i \leq n$. Therefore, there is a state transition on the wire $i$ if and only if $x_i = 1$ (see Figure 1).

Manuscript received August 9, 2018; revised October 30, 2019; accepted February 15, 2020. Date of publication March 2, 2020; date of current version July 14, 2020. The work of Yeow Meng Chee was supported in part by the Singapore Ministry of Education under Grant MOE2017-T3-1-007 and Grant MOE2015-T2-2-086. The work of Tuvi Etzion was supported in part by the Binational Science Foundation-National Science Foundation (BSF-NSF) under Grant 2016692 and in part by NSF under Grant CCF-1719139. The work of Han Mao Kiah and Hengjia Wei was supported by the Singapore Ministry of Education under Grant MOE2015-T2-2-086. The work of Alexander Vardy was supported in part by the National Science Foundation under Grant CCF-1405119 and Grant CCF-1719139, in part by the Binational Science Foundation-National Science Foundation (BSF-NSF) under Grant 2016692, and in part by NSF under Grant CCF-1719139. This article was presented at the 2018 IEEE International Symposium on Information Theory. *(Corresponding author: Hengjia Wei.)*

Yeow Meng Chee is with the Department of Industrial Systems Engineering and Management, National University of Singapore, Singapore 117576 (e-mail: pvocym@nus.edu.sg).

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

Han Mao Kiah is with the School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore 637371 (e-mail: hmkiah@ntu.edu.sg).

Alexander Vardy is with the Department of Electrical and Computer Engineering, University of California at San Diego, San Diego, CA 92093 USA, and also with the Department of Computer Science and Engineering, University of California at San Diego, San Diego, CA 92093 USA (e-mail: avardy@ucsd.edu).

Hengjia Wei is with the Department of Electrical and Computer Engineering, Ben-Gurion University of the Negev, Be'er Sheva 84105, Israel (e-mail: hjwei05@gmail.com).

Communicated by P. Sadeghi, Associate Editor for Coding Techniques.

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIT.2020.2977871

0018-9448 © 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

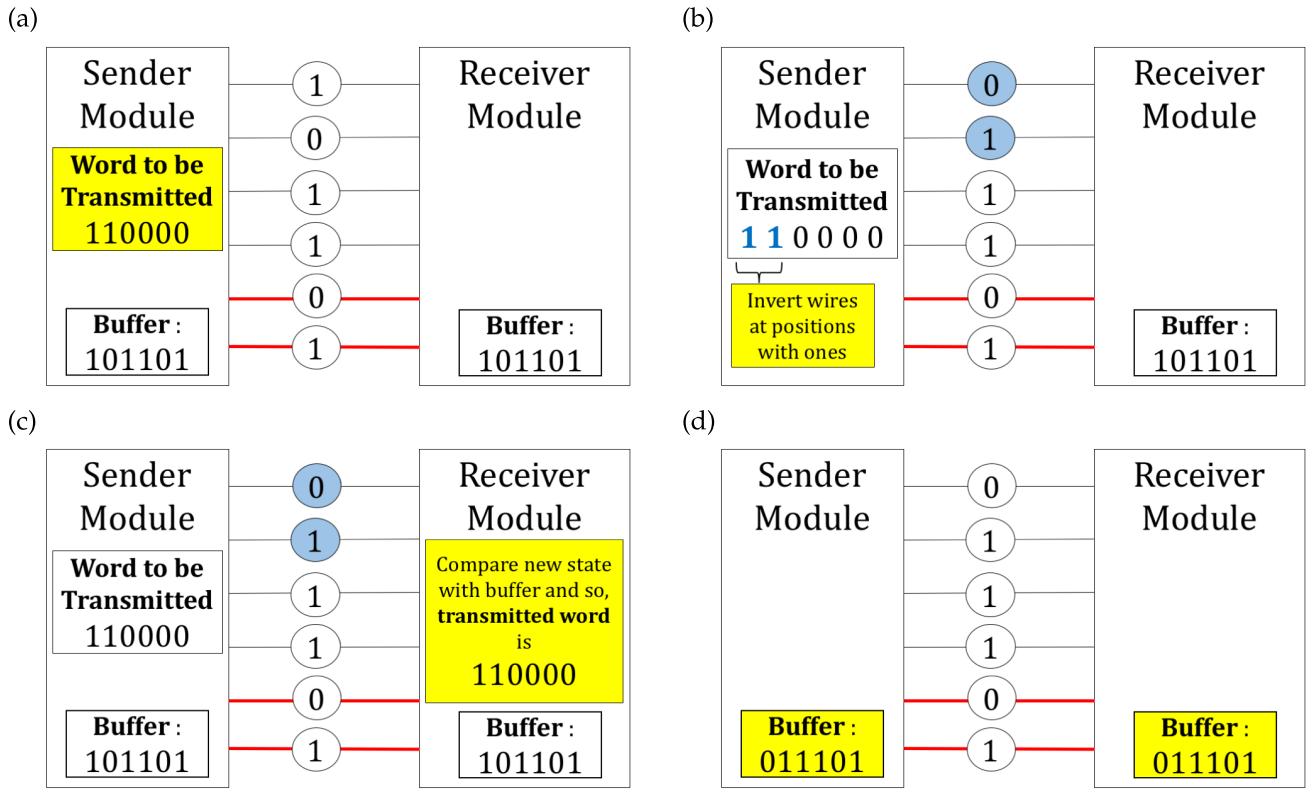


Fig. 1. (a) The current state of the wires is 101101, which is stored in the buffers of both the sender and receiver modules. Suppose that the fifth and sixth wires are the hottest wires and we choose a word 110000 to transmit. (b) To do so, since the word 110000 has ones on the first two positions, we invert the state of the first two wires, thus changing the state to 011101. Observe that the vector 110000 is chosen so as to avoid state transitions on the hottest wires (here, the hottest wires are the last two wires). Hence, only the first two wires may heat up, while the temperature of the remaining wires cool down. (c) The receiver module sees that the wires' states have changed. Comparing the new state 011101 with the buffer word 101101, the receiver module computes the received word to be 110000. (d) Finally, both sender and receiver modules update their buffer to reflect the new state of the wires.

Given the $t$ hottest wires, an $(n,t)$-cooling code allows one to encode data into a vector of length $n$ that has zero entries in the corresponding $t$ coordinates, thereby avoiding state transitions on these $t$ wires. A formal definition and an example are given in Section II.

In this paper, we are particularly interested in *low-power cooling (LPC) codes* as they control both the peak temperature and the average power consumption simultaneously. Specifically, an $(n,t,w)$-LPC code is an $(n,t)$-cooling code in which every codeword has Hamming weight at most $w$. Such a coding scheme has the following two features:

(i) none of the transmissions cause state transitions on the $t$ hottest wires;

(ii) the number of state transitions on all the wires is at most $w$ in every transmission.

Using *partial spreads*, we constructed LPC codes with efficient encoding and decoding schemes in [3]. When $t \leq 0.687n$ and $w \geq (n-t)/2$, these codes achieve optimal asymptotic rates. However, when $w$ is small, i.e., low-power coding is used, the code rates are small and we proposed another construction based on *the decomposition of the complete hypergraph* into perfect matchings. While the construction results in LPC codes of large size, efficient encoding and decoding algorithms are generally not known.

In this work, we focus on the latter regime (i.e., $w$ small) and construct LPC codes with efficient encoding and decoding schemes. Specifically, our contributions are as follows:

(i) We propose a method that takes a linear erasure code as input and constructs an LPC code. This method is applicable whenever $n/w$ is a prime power such that $n/w \geq 2w-2$ and $t < n/w$. Using this method, we construct a family of $(n,t,w)$-LPC codes of size $(n/w)^{w-1}$, which attains the asymptotic upper bound $O(n^{w-1})$ when $w$ is fixed. We also use this method to construct a class of LPC codes of size $(n/w)^{w-e-1}$ which is able to correct $e$ transmission errors.

(ii) We propose efficient encoding/decoding schemes for the above family of LPC codes. In particular, we demonstrate encoding with $O(n)$ multiplications over $\mathbb{F}_q$ and decoding with $O(w^3)$ multiplications over $\mathbb{F}_q$, where $q = n/w$. Furthermore, we present a decoding algorithm for the related class of LPC codes that corrects $e$ errors with complexity $O(n^3)$.

(iii) We propose a new family of low-power cooling codes, called *constant-power cooling* (CPC in short) codes, which have the same weight for all the codewords. All our previous constructions can be applied to obtain such codes.

(iv) We provide a recursive construction of a class of $(nq,tq,w)$-CPC codes (and also $(nq,tq,w)$-LPC codes) from $(n,t,w)$-CPC codes (and a special type of $(n,t,w)$-LPC codes).

Compared with previous constructions, this recursive construction admits a larger range of parameters. Specifically, this method can produce CPC codes of high cooling capability, $t \geq n/w$.

(v) We construct a class of $(n, t, w)$-LPC codes based on $(m, t)$-cooling codes. In an $(m, t)$-cooling code, each codeword is a binary vector of length $m$. Our construction takes an $(m, t)$-cooling code $\mathbb{C}$ as input, where $2^m \leq \sum_{i=0}^{w} \binom{n}{i}$, and uses a mapping $\varphi$ to send each codeword $\boldsymbol{x}$ of $\mathbb{C}$ into a binary vector of length $n$ and weight at most $w$ such that each coordinate of the resulting vector $\varphi(\boldsymbol{x})$ is "dominated" by a coordinate of the codeword $\boldsymbol{x}$. This property of being "dominated", which is explained in Section VI, guarantees that the cooling property of the $(m, t)$-cooling code is preserved.

Our emphasis in this paper is on cooling codes in the context controlling temperature of buses. We note however that such codes have other applications too. One such application is in the design of WOM (Write Once Memory) codes which are very important in coding for flash memories (see [4] and references therein). This application of codesets for the construction of WOM codes was given in detail in [4] and is described in short as follows. In a WOM code, we write binary information messages into a memory of length $n$, and the information can only be written in positions where there are *zeroes*. The goal is to write as many rounds as possible until there is no way to distinguish between some of the written words. A coding solution is to identify each information message with a codeset, and choose a codeword $\boldsymbol{x}$ from the appropriate codeset that has *ones* on all positions where the memory has *ones*. In other words, the *ones* in complement $\overline{\boldsymbol{x}}$ of $\boldsymbol{x}$ should have empty intersection with the *ones* of the memory. If we set $S$ to be the set of ones of the memory, then the above property can rewritten as: $\mathrm{supp}(\overline{\boldsymbol{x}}) \cap S = \varnothing$, which is analogous to Definition 1. We refer the interested reader to [4] for more details. We believe that other applications will arise in the future.

The rest of this paper is organized as follows. In Section II, we present some necessary definitions for our exposition, some of the known results, and new upper bounds on the sizes of low-power cooling codes and constant-power cooling codes. Then, the known constructions are presented and we motivate our first construction based on disjoint Turán systems. Section III suggests a construction for CPC codes based on non-binary linear codes in general and on MDS codes in particular. For these codes efficient encoding and decoding algorithms are derived. We continue in Section IV and add error-correction capabilities for such codes and provide efficient algorithms also in this case. The construction that was used in Section IV is modified in Section V to provide a recursive construction for $(nq, tq, w)$-CPC codes (and related $(nq, tq, w)$-LPC codes) from $(n, t, w)$-CPC codes (and some special $(n, t, w)$-LPC codes). While in Section III the constructions are for $t \leq n/w - 1$, the construction in Section V admits larger values of $t$. In Section VI a method to transfer an $(m, t)$-cooling code to an $(n, t, w)$-LPC code is given. This method is based on a special injection from the set of all binary words of length $m$ into binary words of length $n$ and weight at most $w$. A product construction using this method implies codes with efficient encoding and decoding algorithms. We further analyze and compare between this construction and constructions in previous works.

## II. Preliminaries

Given a positive integer $n$, the set $\{1, 2, \ldots, n\}$ is abbreviated as $[n]$. The *Hamming weight* of a vector $\boldsymbol{x} \in \mathbb{F}_q^n$, denoted $\mathrm{wt}(\boldsymbol{x})$, is the number of nonzero positions in $\boldsymbol{x}$, while the *support* of $\boldsymbol{x}$ is defined as $\mathrm{supp}(\boldsymbol{x}) \triangleq \{i \in [n] : x_i \neq 0\}$. A $q$-ary *code* $\mathcal{C}$ *of length* $n$ is a subset of $\mathbb{F}_q^n$, while the *minimum distance* of $\mathcal{C}$ is the smallest Hamming distance between any two of its codewords. The *code size* of $\mathcal{C}$ is its cardinality $|\mathcal{C}|$, while its *rate* is given by $\log_q |C|/n$.

Let $S \subseteq [n]$ and suppose $S$ represents a set of hot wires. As described in Section I, in order to avoid state transitions on these wires, we require the transmitted codeword to have zeroes at the coordinates in $S$. However, as the choice of $S$ is arbitrary, instead of encoding each message to exactly one codeword, we associate each message to a *codeset* of possible codewords. Specifically, we have the following definition.

*Definition 1:* For $n$ and $t$, an $(n, t)$*-cooling code* $\mathbb{C}$ of size $M$ is defined as a collection of codesets $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_M$, where $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_M$ are disjoint subsets of $\{0, 1\}^n$ satisfying the following property: for any set $S \subseteq [n]$ of size $|S| = t$ and for $i \in [M]$, there exists a vector $\boldsymbol{x} \in \mathcal{C}_i$ such that $\mathrm{supp}(\boldsymbol{x}) \cap S = \varnothing$. We refer to the vectors in $\bigcup_{i=1}^{M} \mathcal{C}_i$ as *codewords*.

*Example 1:* Consider the communication over a bus consisting of 6 wires as depicted in Figure 1. We associate each of the following five messages $\boldsymbol{m}_i$ with a codeset $\mathcal{C}_i$:

$$\begin{aligned}
\boldsymbol{m}_1 &: \ \mathcal{C}_1 = \{110000,\ 001100,\ 000011\},\\
\boldsymbol{m}_2 &: \ \mathcal{C}_2 = \{100001,\ 011000,\ 000110\},\\
\boldsymbol{m}_3 &: \ \mathcal{C}_3 = \{100010,\ 010100,\ 001001\},\\
\boldsymbol{m}_4 &: \ \mathcal{C}_4 = \{100100,\ 010001,\ 001010\},\\
\boldsymbol{m}_5 &: \ \mathcal{C}_5 = \{101000,\ 010010,\ 000101\}.
\end{aligned}$$

Suppose the last two wires are hottest wires. For each message $\boldsymbol{m}_i$, one can choose the following vector from the corresponding codeset to transmit. Observe that the last two bits in the chosen vector are always zero.

$$\begin{aligned}
&\boldsymbol{m}_1 \mapsto 110000, \quad \boldsymbol{m}_2 \mapsto 011000, \quad \boldsymbol{m}_3 \mapsto 010100,\\
&\boldsymbol{m}_4 \mapsto 100100, \quad \boldsymbol{m}_5 \mapsto \ 101000.
\end{aligned}$$

In particular, for Figure 1, the sender module is sending message $\boldsymbol{m}_1$ and chose to transmit the word 110000 to avoid state transitions on the last two wires. Since the codesets are mutually disjoint, we can always uniquely decode the message from the transmitted vector. □

To limit the number of state transitions in each transmission, i.e., limit the average power consumption, we introduce the notion of *low-power cooling codes*. As discussed in Section I, this corresponds to bounding the weight of each codeword.

*Definition 2:* For $n$, $t$ and $w$ with $t + w \leq n$, an $(n, t, w)$-*low-power cooling (LPC) code* is an $(n, t)$-cooling code in which every codeword has Hamming weight at most $w$.

In this paper, we focus on a class of $(n, t, w)$-LPC codes where every transmission results in exactly $w$ state transitions. We call such codes $(n, t, w)$-*constant-power cooling (CPC) codes*. In particular, let $J(n, w) \triangleq \{\boldsymbol{x} \in \{0,1\}^n : \mathrm{wt}(\boldsymbol{x}) = w\}$. Then an $(n, t, w)$-CPC code is an $(n, t, w)$-LPC code such that $\mathcal{C}_i \subseteq J(n, w)$ for each $i \in [M]$.

For given values of $n$, $t$, and $w$, our objective is to construct $(n, t, w)$-LPC codes and $(n, t, w)$-CPC codes with the largest possible code sizes, and therefore, the highest code rates. In particular, Construction 1 yields a family of codes whose code sizes are asymptotically optimal.

In addition to large code sizes, we also aim to equip the codes with efficient encoding and decoding schemes that have polynomial time complexity. Specifically, for an $(n, t, w)$-LPC code $\mathbb{C}$ of size $M$, we define encoding and decoding as follows.

- *Encoding* refers to a function ENC that maps a message $i \in [M]$ and a $t$-subset $S$ of $[n]$ to a codeword $\boldsymbol{x} \in \mathcal{C}_i$. Here, $S$ represents a set of $t$ hottest wires and we require $\mathrm{supp}(\boldsymbol{x}) \cap S = \varnothing$ so that we avoid state transitions on the wires corresponding to $S$.
  In Section III, we present a class of $(n, t, w)$-LPC codes $\mathbb{C}$ of size $M = (n/w)^{w-1}$. A naive encoding method stores *all* $(n/w)^{w-1}$ codesets in a codebook. Given a message $i \in [M]$ and a $t$-subset $S$ of $[n]$, the naive encoder then finds $\mathcal{C}_i$ in $O(\log M)$ time and determines $\boldsymbol{x}$ in $O(n)$ time. In contrast, for our LPC code, we demonstrate that it suffices to store a matrix $\boldsymbol{G}$ with $O(nw)$ entries over $\mathbb{F}_q$, where $q = n/w$. For this matrix $\boldsymbol{G}$, we provide a corresponding encoder that computes the codeword $\boldsymbol{x}$ using $O(n)$ arithmetic operations over $\mathbb{F}_q$.
- *Decoding* refers to a function DEC that maps a codeword $\boldsymbol{x} \in C_i$ back to the message $i$. Unless otherwise stated, we assume that the word $\boldsymbol{x}$ is *read without errors*. For the codes constructed in Section III, given the $\mathbb{F}_q$-matrix $\boldsymbol{G}$, we provide a decoder that computes the message $i$ in $O(w^3)$ time. In the presence of errors, we modify our decoder to correct the errors and compute the message $i$ in $O(n^3)$ time.

### A. Set Systems

For a finite set $X$ of size $n$, $2^X$ denotes the collection of all subsets of $X$, i.e., $2^X \triangleq \{A : A \subseteq X\}$. A *set system* of order $n$ is a pair $(X, \mathcal{B})$, where $X$ is a finite set of $n$ *points*, $\mathcal{B} \subseteq 2^X$, and the elements of $\mathcal{B}$ are called *blocks*. Two set systems $(X, \mathcal{B}_1)$ and $(X, \mathcal{B}_2)$ with the same point set are called *disjoint* if $\mathcal{B}_1 \cap \mathcal{B}_2 = \varnothing$, i.e., they do not have any block in common.

A *partial parallel class* of a set system $(X, \mathcal{B})$ is a collection of pairwise disjoint blocks. If a partial parallel class partitions the point set $X$, it is called *parallel class*.

There is a canonical one-to-one correspondence between the set of all codes of length $n$ and the set of all set systems of order $n$: the coordinates of vectors in $\{0,1\}^n$ correspond to the points in $[n]$, and each vector $\boldsymbol{x} \in \{0,1\}^n$ corresponds to the block defined by $\mathrm{supp}(\boldsymbol{x})$. Thus we may speak of the set system of a code or the code of a set system. With slight abuse of notation, we sometimes do not distinguish between the two different notations and this can be readily observed in the text.

### B. Upper Bounds

Given a $t$-subset (i.e., a subset of size $t$) $S$ of $[n]$ and a vector $\boldsymbol{x} \in \{0,1\}^n$, we shall say that $\boldsymbol{x}$ *avoids* $S$ if $\mathrm{supp}(\boldsymbol{x}) \cap S = \varnothing$. The following bounds on LPC codes and CPC codes are easily derived.

*Lemma 1:* Let $\mathbb{C}$ be an $(n, t, w)$-LPC code of size $M$, then

$$M \le \sum_{i=0}^{w} \binom{n-t}{i}.$$

Furthermore, if $\mathbb{C}$ is an $(n, t, w)$-CPC code, then

$$M \le \binom{n-t}{w}.$$

*Proof:* For any given $t$-subset $S$ of $[n]$, each codeset should have at least one codeword which avoids $S$. The number of words with weight $i$ which avoid $S$ is $\binom{n-t}{i}$ and hence there are no more than $\binom{n-t}{w}$ codesets in an $(n, t, w)$-CPC code and no more than $\sum_{i=0}^{w} \binom{n-t}{i}$ codesets in an $(n, t, w)$-LPC code. □

Lemma 1 implies that both CPC codes and LPC codes share the same asymptotic upper bound $O(n^w)$ on the number of codewords, whenever $w$ is fixed. The upper bound of Lemma 1 can be improved for some parameters. For this purpose, we need to define and to introduce some results on Turán systems.

Let $n \ge k \ge r$, and let $X$ be a finite set with $n$ distinct elements. The set $\binom{X}{r}$ is the collection of all $r$-subsets of $X$. A *Turán* $(n, k, r)$-*system* is a set system $(X, \mathcal{B})$, where $|X| = n$ and $\mathcal{B} \subseteq \binom{X}{r}$ is the set of blocks such that each $k$-subset of $X$ contains at least one of the blocks. The *Turán number* $T(n, k, r)$ is the minimum number of blocks in such a system. This number is determined only for $r = 2$ and some sporadic examples (see [12], [15] and references therein). De Caen [6] proved the following general lower bound:

$$T(n, k, r) \ge \frac{n-k+1}{n-r+1} \cdot \frac{\binom{n}{r}}{\binom{k-1}{r-1}}. \tag{1}$$

Note that a codeword $\boldsymbol{x}$ avoids a $t$-subset $S$ if and only if the complement of $S$, which is an $(n-t)$-subset, contains $\mathrm{supp}(\boldsymbol{x})$. Thus from a CPC code we can obtain a collection of disjoint Turán systems by simply taking the supports of the codewords, and vice versa.

*Proposition 2:* A family of codesets $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_M\}$ is an $(n, t, w)$-CPC code if and only if the set system of each $\mathcal{C}_i$ is a Turán $(n, n-t, w)$-system and these $M$ set systems are pairwise disjoint.

Combining the bound in (1) and Proposition 2, we have the following upper bounds on the size of CPC codes and LPC codes.

*Corollary 3:* If $\mathbb{C}$ is an $(n, t, w)$-CPC code of size $M$, then

$$M \le \frac{n-w+1}{t+1} \binom{n-t-1}{w-1}.$$

*Proof:* Proposition 2 implies that

$$M \leq \frac{\binom{n}{w}}{T(n, n-t, w)}. \tag{2}$$

The corollary then follows from (1). □

*Corollary 4:* If $\mathbb{C}$ is an $(n, t, w)$-LPC code of size $M$, then

$$M \leq \sum_{i=0}^{w-1} \binom{n}{i} + \frac{n-w+1}{t+1} \binom{n-t-1}{w-1}.$$

*Proof:* If we consider an $(n, t, w)$-CPC code $\mathbb{C}$, then to form an $(n, t, w)$-LPC code we can add to $\mathbb{C}$ at most $\sum_{i=0}^{w-1} \binom{n}{i}$ codesets, each one containing exactly one codeword of weight less than $w$. □

When $t = \Theta(n)$, we have that $(n-w+1)/(t+1) = O(1)$, and so the upper bound on the size of $(n, t, w)$-CPC codes is improved from $O(n^w)$ implied by Lemma 1 to $O(n^{w-1})$ implied by Corollary 3.

For an $(n, t, w)$-LPC code, we have by Corollary 4 that the size of such a code is at most

$$\sum_{i=0}^{w-1} \binom{n}{i} + \frac{n-w+1}{t+1} \binom{n-t-1}{w-1},$$

which is also $O(n^{w-1})$ when $t$ and $n$ are of the same order of magnitude.

We end this subsection with an example of Proposition 2.

*Example 2:* Consider the $(6, 2, 2)$-CPC code in Example 1. By taking the supports of the codewords we can obtain a set system from each codeset with the set of points being $X \triangleq \{1, 2, 3, 4, 5, 6\}$ and the set of blocks $\mathcal{B}_i$ as follows:

$$\mathcal{B}_1 : \{\{1,2\},\ \{3,4\},\ \{5,6\}\}$$
$$\mathcal{B}_2 : \{\{1,6\},\ \{2,3\},\ \{4,5\}\}$$
$$\mathcal{B}_3 : \{\{1,5\},\ \{2,4\},\ \{3,6\}\}$$
$$\mathcal{B}_4 : \{\{1,4\},\ \{2,6\},\ \{3,5\}\}$$
$$\mathcal{B}_5 : \{\{1,3\},\ \{2,5\},\ \{4,6\}\}$$

Now, for the 4-subset $\{1, 2, 3, 4\}$, we consider the last two wires. As shown in Example 1, there is a codeword from each codeset that avoids $\{5, 6\}$. Thus the blocks, $\{1, 2\}, \{2, 3\}, \{2, 4\}, \{1, 4\}$, and $\{1, 3\}$, are contained in the 4-subset $\{1, 2, 3, 4\}$. In general, for a 4-subset $\{a, b, c, d\}$ of $X$, we consider its complement, which is 2-subset of $X$. According to the definition of CPC codes, from each codeset we can always find a codeword which avoids the 2-subset $X \backslash \{a, b, c, d\}$ and then the support of this codeword is contained in the 4-subset $\{a, b, c, d\}$. Thus, each $(X, \mathcal{B}_i)$ is a $(6, 4, 2)$-Turán system. The disjointness of these Turán systems comes from that of the corresponding codesets.

### C. Some Known Constructions

Chee et. al [3] provided the following construction of LPC/CPC codes.

*Proposition 5 (Decomposition of Complete Hypergraphs):* If $n = (t+1)w$, then there exists an $(n, t, w)$-CPC code of size $\binom{n-1}{w-1}$.

When $w$ is fixed, we have that $t$ and $n$ are of the same order of magnitude and the above construction attains the asymptotic upper bound $O(n^{w-1})$. Unfortunately, usually no efficient encoding and decoding methods are known for this construction and generally the only known encoding method involves listing all the $\binom{n-1}{w-1}$ codesets. The exceptions are for small $n$ or when $w$ is very small, e.g., when $w = 2$ or $w = 3$ [1], [7].

Chee et. al [3] also proposed the following constructions of LPC codes that have both efficient encoding and decoding schemes. We remark that to apply the sunflower construction in Proposition 7, we require known upper bounds on the dimensions of linear codes [11].

*Proposition 6 (Concatenation):* Suppose that $q \leq \sum_{i=0}^{w'} \binom{s}{i}$ and $q$ is a prime power and $t \leq s$.

(i) If $t + 1 \leq m/2$, then there exists an $(ms, t, mw')$-LPC code of size $q^{m-t-1}$.

(ii) If $t+1 \leq m \leq q+1$, then there exists an $(ms, t, mw')$-LPC code of size $q^{m-t}$.

*Proposition 7 (Sunflower Construction):* Let $r + t \leq (n + s)/2$. If a linear $[n, s, w + 1]_2$ code exists and a linear $[n - t, r, w + 1]_2$ code does not exist, then there exists an $(n, t, w)$-LPC code of size $2^{n-t-r}$.

Finally, Proposition 2 also motivates a new method to construct $(n, t, w)$-CPC codes. We just have to find a set with large number of pairwise disjoint Turán $(n, n-t, w)$-systems. One work in this direction was done in [10], where pairwise disjoint Turán $(n, w+1, w)$-systems were considered. Another possible construction based on Proposition 2 is to consider complements of pairwise disjoint Steiner systems. Such pairwise disjoint systems were considered in [2], [18]; and in [8] for Steiner quadruple systems which will be used in the sequel.

## III. Construction of Constant-Power Cooling Codes With Efficient Encoding and Decoding

In this section, we present a new construction of CPC codes that have efficient encoding and decoding algorithms. When $n$ approaches infinity and $w$ is fixed, the codes obtained by this construction asymptotically attain the bound of Corollary 3. As was mentioned before, there are three types of constructions for LPC codes in [3]. The first one is based on decomposition of the complete hypergraph, the second one is a concatenation method based on $q$-ary cooling codes, and the third one is a Sunflower Construction. The construction in this section, is an explicit construction for CPC codes which combines the advantages of the first two types of constructions. We first rephrase the construction based on decomposition of the complete hypergraph in terms of set systems. The construction is based on the following generalization of Proposition 5.

*Proposition 8:* Let $(X, \mathcal{B})$ be a set system of order $n$, where $\mathcal{B}$ is partitioned into $M$ partial parallel classes $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_M$. If $\mathcal{B} \subseteq \binom{X}{w}$ and each $\mathcal{P}_i$ has at least $t + 1$ blocks, then the codesets $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_M$ form an $(n, t, w)$-CPC code.

*Proof:* By definition, each codeword of a codeset has weight $w$. Hence, to show that $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_M$ form an $(n, t, w)$-CPC code, we only have to prove that given a $t$-subset $S$ of $X$ with the list of hottest wires and a codeset $\mathcal{P}_i$,

$1 \leq i \leq M$, there exists a block $B \in \mathcal{P}_i$ such that $B \cap S = \varnothing$. Since $\mathcal{P}_i$ is a partial parallel class with at least $t+1$ codewords, it follows that $S$ intersects at most $t$ blocks of $\mathcal{P}_i$. Hence, there exists a block $B \in \mathcal{P}_i$ such that $B \cap S = \varnothing$. □

A complete $k$-uniform hypergraph $G = (V, E)$ has a vertex set $V$ with $n \geq k$ vertices, and each subset of $\binom{V}{k}$ is connected by a hyperedge. The decomposition of $G$ into perfect matchings is a partition of the set of edges $E$ in $G$ into sets of vertex-disjoint edges, where each vertex of $V$ appears exactly once in each set of the partition. In other words, such a decomposition is a partition of $\binom{V}{k}$ into parallel classes. The celebrated Baranyai's theorem [19, p. 536] asserts that such a decomposition always exists if $k$ divides the number of vertices in $V$. Therefore, we recover Proposition 5.

### A. CPC Codes Based on Linear Codes

A key ingredient of our construction is a $q$-ary linear code. A $q$-ary code $\mathcal{C} \subseteq \mathbb{F}_q^N$ is a *linear code* if $\mathcal{C}$ is an $\mathbb{F}_q$-linear subspace of $\mathbb{F}_q^N$. A linear code $\mathcal{C}$ is an $[N, K, D]_q$ code if $\mathcal{C}$ has dimension $K$ and minimum Hamming distance $D$. Using the codewords of $\mathcal{C}$, we will show how to construct a set system with $q^{K-1}$ partial parallel classes, where each parallel class has blocks of the same size. As a consequence, Proposition 8 yields a CPC code $\mathbb{D}$. To equip $\mathbb{D}$ with efficient encoding and decoding schemes, we utilize the erasure-correcting algorithms of the linear code $\mathcal{C}$. These schemes are discussed in detail in Section III-B.

For a set of coordinates $T$ and a vector $\boldsymbol{\sigma} \in \mathbb{F}_q^{|T|}$, we say that $\boldsymbol{\sigma}$ *appears* $\lambda$ *times in* $\mathcal{C}$ *at* $T$ if there are $\lambda$ codewords in $\mathcal{C}$ whose restriction on $T$ is $\boldsymbol{\sigma}$. Since any two codewords of $\mathcal{C}$ differ in at least $D$ coordinates, it follows that they agree in at most $N - D$ positions. Hence, we have the following observations.

*Lemma 9:* Let $\mathcal{C}$ be an $[N, K, D]_q$ linear code.

(i) For any $(N - D + 1)$-subset of coordinates $T$ and any $\boldsymbol{\sigma} \in \mathbb{F}_q^{N-D+1}$, $\boldsymbol{\sigma}$ appears in at most one codeword of $\mathcal{C}$ at $T$.

(ii) For any $(N - D)$-subset of coordinates $T$ and any $\boldsymbol{\tau} \in \mathbb{F}_q^{N-D}$, $\boldsymbol{\tau}$ appears in at most $q$ codewords of $\mathcal{C}$ at $T$.

*Proof:*

(i) If $\boldsymbol{\sigma} \in \mathbb{F}_q^{N-D+1}$ appears twice in codewords of $\mathcal{C}$ at an $(N-D+1)$-subset of coordinates $T$, then the two related codewords have distance at most $D-1$, a contradiction.

(ii) If $\boldsymbol{\tau} \in \mathbb{F}_q^{N-D}$ appears in $q+1$ codewords of $\mathcal{C}$ at an $(N-D)$-subset of coordinates $T$, then let $t$ be a coordinate not in $T$. In at least two of the related codewords coordinate $t$ has the same symbol. We add this symbol to $\boldsymbol{\tau}$ to obtain $\boldsymbol{\sigma} \in \mathbb{F}_q^{N-D+1}$ which appears in two codewords of $\mathcal{C}$ at the $(N - D + 1)$-subset $T \cup \{t\}$, contradicting claim (i) of this lemma. □

For a code $\mathcal{C}$ and a subset of coordinates $T$, let $\mathcal{C}|_T$ denote the set of codewords restricted to the coordinates of $T$, i.e., the projection of $\mathcal{C}$ into the set of coordinates indexed by $T$. For a word $\boldsymbol{u}$, let $\boldsymbol{u}|_T$ denote the restriction of $\boldsymbol{u}$ to the coordinates of $T$. Finally, for a matrix $\boldsymbol{G}$, let $\boldsymbol{G}|_T$ denote the submatrix of $\boldsymbol{G}$ obtained from the columns indexed by $T$.

*Lemma 10:* Let $\mathcal{C}$ be a linear $[N, K, D]_q$ code. If $\boldsymbol{G}$ is a generator matrix of $\mathcal{C}$, then every $K \times (N - D)$ submatrix of $\boldsymbol{G}$ has rank either $K$ or $K - 1$. Furthermore, there exists a $K \times (N - D)$ submatrix of $\boldsymbol{G}$ whose rank is $K - 1$.

*Proof:* Let $T$ be a subset of $N - D$ coordinate positions and assume the corresponding $K \times (N - D)$ submatrix $\boldsymbol{G}|_T$ has rank $r$, where $r \leq K$. Let $\phi_T$ be the linear map from $\mathbb{F}_q^K$ to $\mathbb{F}_q^{N-D}$ defined by $\phi_T(\boldsymbol{u}) = \boldsymbol{u}\boldsymbol{G}|_T$. Clearly, the dimension of the kernel of $\phi_T$ is $K - r$. Hence, the all-zero vector of length $N - D$ appears in $q^{K-r}$ codewords of $\mathcal{C}|_T$. By Lemma 9 the all-zero vector appears in at most $q$ codewords of $\mathcal{C}|_T$ at $T$, which implies that K-r $\leq 1$ and therefore $r \geq K - 1$.

Let $\boldsymbol{x}$ be a codeword of $\mathcal{C}$ with minimum weight $D$, $T$ be the complement of $\mathrm{supp}(\boldsymbol{x})$ in $[N]$. Let $\boldsymbol{u}$ be the information vector of length $K$ such that $\boldsymbol{x} = \boldsymbol{u}\boldsymbol{G}$. Since $\boldsymbol{x}$ has weight $D$, it follows that $|\mathrm{supp}(\boldsymbol{x})| = D$ and hence the size of $T$ is $N - D$. Since $\boldsymbol{x}$ has *zeroes* in the coordinates of $T$, it follows that for the $K \times (N - D)$ submatrix $\boldsymbol{G}|_T$ of $\boldsymbol{G}$ we have that $\boldsymbol{u}\boldsymbol{G}|_T = \boldsymbol{0}$. Therefore, the rank of $\boldsymbol{G}_T$ is at most $K - 1$.

Thus the rank of $\boldsymbol{G}|_T$ is at least $K - 1$ by the first part of the proof and at most $K - 1$ by the second part of the proof, which implies that the rank of $\boldsymbol{G}|_T$ is $K - 1$. □

We are now ready to present the first new construction for CPC codes. The corresponding encoding and decoding schemes are presented in the next subsection.

*Construction 1:* Let $\mathcal{C}$ be a linear $[N, K, D]_q$ code and $\boldsymbol{G}$ be a generator matrix of $\mathcal{C}$, where the last $N - D$ columns of $\boldsymbol{G}$ form a $K \times (N - D)$ submatrix of $\boldsymbol{G}$ whose rank is $K - 1$.

- Partition the codewords of $\mathcal{C}$ into disjoint codesets such that two codewords $\boldsymbol{x}$ and $\boldsymbol{y}$ are in the same codeset if and only if they agree on their last $N - D$ symbols. Note that the submatrix consisting of the last $N - D$ columns has rank $K - 1$. There are $q^{K-1}$ such codesets and we label them as $\mathcal{C}_{\boldsymbol{\sigma}}$, where $\boldsymbol{\sigma} \in \mathbb{F}_q^{K-1}$.
- For each $\boldsymbol{\sigma} \in \mathbb{F}_q^{K-1}$, truncate the codewords in $\mathcal{C}_{\boldsymbol{\sigma}}$ to length $w$ by removing their last $N - w$ symbols. In other words, set $\mathcal{C}'_{\boldsymbol{\sigma}} \triangleq \{\boldsymbol{x}|_{[w]} : \boldsymbol{x} \in \mathcal{C}_{\boldsymbol{\sigma}}\}$ for each $\boldsymbol{\sigma} \in \mathbb{F}_q^{K-1}$.
- For each $\boldsymbol{\sigma} \in \mathbb{F}_q^{K-1}$ construct the set system $(X, \mathcal{D}_{\boldsymbol{\sigma}})$, where $X = \mathbb{F}_q \times [w]$ and

$$\mathcal{D}_{\boldsymbol{\sigma}} = \{(x_j, j) \ : \ \boldsymbol{x} = x_1 x_2 \cdots x_w \in \mathcal{C}'_{\boldsymbol{\sigma}}, \ j \in [w]\}.$$

In the construction above, each $\mathcal{D}_{\boldsymbol{\sigma}}$ is a collection of subsets of $X$. Using the notational convention introduced in Section II, we may treat it as a codeset of binary words of length $|X|$.

*Theorem 11:* If $N - D + 1 \leq w \leq D$, then the collection of codesets $\mathbb{D} = \{\mathcal{D}_{\boldsymbol{\sigma}} : \boldsymbol{\sigma} \in \mathbb{F}_q^{K-1}\}$ is an $(n, t, w)$-CPC code of size $M = q^{K-1}$, where $n = qw$ and $t = q - 1$.

*Proof:* Clearly, by the definition of the construction we have that $n = qw$ and each codeword has weight $w$. Hence, to complete the proof it is sufficient to show that the $M$ codesets are pairwise disjoint, and for any $t$-subset $S$ of coordinates from $X$ and each $\boldsymbol{\sigma} \in \mathbb{F}_q^{K-1}$, there exists a codeword $\boldsymbol{x}$ in the codeset $\mathcal{D}_{\boldsymbol{\sigma}}$ such that $\mathrm{supp}(\boldsymbol{x}) \cap S = \varnothing$.

(i) Let $\boldsymbol{G}'$ be the $K \times (N-D)$ submatrix of $\boldsymbol{G}$ formed from the last $N - D$ columns of $\boldsymbol{G}$. Consider the linear map $\phi$ from $\mathbb{F}_q^K$ to $\mathbb{F}_q^{N-D}$ defined by $\phi(\boldsymbol{u}) = \boldsymbol{u}\boldsymbol{G}'$. Since the rank of $\boldsymbol{G}$ is $K - 1$, it follows that the kernel of $\phi$ has dimension one. Thus $|\mathcal{C}_{\boldsymbol{\sigma}}| = q$ for each $\boldsymbol{\sigma} \in \mathbb{F}_q^{K-1}$.

(ii) The minimum distance of $\mathcal{C}$ is $D$ and hence each two codewords of $\mathcal{C}$ can agree in at most $N-D$ coordinates, i.e., they differ in any subset of $N-D+1$ coordinates. Since $w \geq N-D+1$ and the codewords of $\mathcal{C}$ were truncated by removing their last $N-w$ coordinates, it follows that all the truncated codewords of $\mathcal{C}$ are distinct. Thus the sets $\mathcal{C}'_{\boldsymbol{\sigma}}$, where $\boldsymbol{\sigma} \in \mathbb{F}_q^{K-1}$, are pairwise disjoint and $|\mathcal{C}'_{\boldsymbol{\sigma}}| = |\mathcal{C}_{\boldsymbol{\sigma}}| = q$. Now, it can be easily verified by the definition that the codesets $\mathcal{D}_{\boldsymbol{\sigma}}$, where $\boldsymbol{\sigma} \in \mathbb{F}_q^{K-1}$, are pairwise disjoint and $|\mathcal{D}_{\boldsymbol{\sigma}}| = q$ for each $\boldsymbol{\sigma} \in \mathbb{F}_q^{K-1}$.

(iii) Each two codewords of $\mathcal{C}_{\boldsymbol{\sigma}}$ agree on their last $N-D$ coordinates and since their distance is at least $D$, it follows that they differ in the first $D$ coordinates. Since $w \leq D$, this implies that any two codewords of $\mathcal{C}'_{\boldsymbol{\sigma}}$ differ in all their $w$ coordinates. Hence, by the definition of $\mathcal{D}_{\boldsymbol{\sigma}}$ this implies that each two codewords of $\mathcal{D}_{\boldsymbol{\sigma}}$ differ in their nonzero coordinates. Therefore, the codewords in $\mathcal{D}_{\boldsymbol{\sigma}}$ are pairwise disjoint, i.e., $\mathcal{D}_{\boldsymbol{\sigma}}$ is a partial parallel class. We also have that $|\mathcal{D}_{\boldsymbol{\sigma}}| = q$ for each $\boldsymbol{\sigma} \in \mathbb{F}_q^{K-1}$. Hence, $\mathcal{D}_{\boldsymbol{\sigma}}$ is a parallel class and as in the proof of Proposition 8 we have that for any $t$-subset $S$, $\mathcal{D}_{\boldsymbol{\sigma}}$ has a codeword $\boldsymbol{x}$ such that $\mathrm{supp}(\boldsymbol{x}) \cap S = \varnothing$.

Thus, the required claims were proved and hence the collection of codesets $\mathbb{D} = \{\mathcal{D}_{\boldsymbol{\sigma}} : \boldsymbol{\sigma} \in \mathbb{F}_q^{K-1}\}$ is a $(qw, q-1, w)$-CPC code of size $q^{K-1}$. □

*Example 3:* Consider the linear $[5,3,3]_5$ code with generator matrix

$$\boldsymbol{G} = \begin{pmatrix} 4 & 3 & 2 & 1 & 0 \\ 2 & 3 & 4 & 0 & 1 \\ 2 & 1 & 2 & 0 & 0 \end{pmatrix}.$$

Since the last two columns form a submatrix of rank 2, we partition the code into the following 25 codesets according to the last two symbols.

$$\begin{aligned} \mathcal{C}_{00} &= \{00000,\ 21200,\ 42400,\ 13100,\ 34300\} \\ \mathcal{C}_{01} &= \{23401,\ 44101,\ 10301,\ 31001,\ 02201\} \\ &\vdots \\ \mathcal{C}_{44} &= \{44444,\ 10144,\ 31344,\ 02044,\ 23244\} \end{aligned}$$

By removing the last two symbols from the codewords in each $\mathcal{C}_{\boldsymbol{\sigma}}$ and replacing each $x_j$ with $(x_j, j)$, we obtain the following codesets $\mathcal{D}_{\boldsymbol{\sigma}}$.

$$\begin{aligned} \mathcal{D}_{00} = \{&\{(0,1),(0,2),(0,3)\},\{(2,1),(1,2),(2,3)\}, \\ &\{(4,1),(2,2),(4,3)\},\{(1,1),(3,2)(1,3)\}, \\ &\{(3,1),(4,2),(3,3)\}\} \\ \mathcal{D}_{01} = \{&\{(2,1),(3,2),(4,3)\},\{(4,1),(4,2),(1,3)\}, \\ &\{(1,1),(0,2),(3,3)\},\{(3,1),(1,2),(0,3)\}, \\ &\{(0,1),(2,2),(2,3)\}\} \\ &\vdots \\ \mathcal{D}_{44} = \{&\{(4,1),(4,2),(4,3)\},\{(1,1),(0,2),(1,3)\}, \\ &\{(3,1),(1,2),(3,3)\},\{(0,1),(2,2),(0,3)\}, \\ &\{(2,1),(3,2),(2,3)\}\} \end{aligned}$$

For a given $[N,K,D]_q$ code $\mathcal{C}$ and its generator matrix $\boldsymbol{G}$ in Construction 1, we need to find a minimum weight codeword in $\mathcal{C}$ in order to determine a $K \times (N-D)$-submatrix of $\boldsymbol{G}$ with rank $K-1$, i.e., to find a permutation of the columns of $\boldsymbol{G}$ such that the last $N-D$ coordinates of $\boldsymbol{G}$ will have rank $K-1$. Finding the minimum distance of a code is an NP-hard problem and the decision problem is NP-complete [20]. Therefore, we focus on certain families of codes where it is computationally easy to find minimum weight codewords. One such family is the family of *maximum distance separable (MDS)* codes. Recall that a linear $[N,K,D]_q$ code is an MDS code if $D = N-K+1$ [13, Ch.11]. If the code $\mathcal{C}$ in Construction 1 is an MDS code, then every $K$ columns of $\boldsymbol{G}$ are linearly independent and hence each $K \times (N-D)$ submatrix of $\boldsymbol{G}$ has rank $K-1$ since $N-D = K-1$. Therefore, we may use any $N-D$ coordinate as the last $N-D$ coordinates of $\mathcal{C}$. It is well known that MDS codes exist for the following parameters.

*Theorem 12 (see [13, Ch.11]):* Let $q$ be a prime power and $D \geq 3$. If $N \leq q+1$ and $2 \leq K \leq q-1$, there exists an $[N,K,D]_q$ MDS code. Furthermore, when $q$ is even and $K \in \{3, q-1\}$, a $[q+2,K,D]_q$ MDS code exists.

Set $N = q+1$, $K = w$ and $D = q-w+2$ and use an $[N,K,D]_q$ MDS code as the code $\mathcal{C}$ in Construction 1. Whenever $q \geq 2w-2$, the condition $N-D+1 \leq w \leq D$ of Theorem 11 is satisfied and hence, we obtain the following corollary.

*Corollary 13:* Let $n, t$ and $w$ be positive integers. If $q = n/w$ is a prime power and $q \geq 2w-2$, then there exists an $(n, q-1, w)$-CPC code of size $(n/w)^{w-1}$.

In Corollary 13, when $w$ is fixed, $t = q-1$ has the same order of magnitude as $n$. Hence, the codes constructed in this case asymptotically attain the upper bound $O(n^{w-1})$. We also note that for some parameters, these CPC codes are much larger than the LPC codes provided by Propositions 6 and 7. This is discussed in the following examples.

*Example 4:* By choosing $n = 96$, $t = 15$ and $w = 6$, Corollary 13 yields a $(96, 15, 6)$-CPC code of size $16^5 = 2^{20}$.

In contrast, suppose we use Proposition 6 to construct a $(96, t, 6)$-LPC code with $t \leq 15$. The largest size $16^5 = 2^{20}$ is obtained by choosing $m = 6$, $t = 1$, $s = 16$, $w' = 1$, and $q = 16$. The resulting $(96, 1, 6)$-LPC code has the same size as the CPC obtained by Corollary 13, but the cooling capability of the former is clearly much weaker. Proposition 7, on the other hand, yields a $(96, 15, 6)$-LPC code of size $2^{16}$ by choosing $s = 81$ and $r = 65$. This code has similar parameters, but its size is much smaller.

The following example shows that a non-MDS code can also be used to obtain a CPC code of large size.

*Example 5:* There exists a $[17, 8, 9]_9$ code (see [11]). Setting $w = 9$ and $t = 8$ in Construction 1 yields a $(81, 8, 9)$-CPC code of size $9^7 \approx 2^{22.189}$.

In contrast, the largest $(81, 8, 9)$-LPC code obtained from Proposition 6 has size $9 \approx 2^{3.17}$ by choosing $m = s = q = 9$, $w' = 1$. Proposition 7, on the other hand, yields an $(81, 8, 9)$-LPC of size $2^{21}$ by choosing $s = 54$ and $r = 52$.

We also note that the corresponding $(81, 8, 9)$-CPC code cannot be constructed using MDS codes, or equivalently, cannot result from Corollary 13. For Corollary 13 to apply,

we require $q \geq 2w-2$, which is not true when $q=8$ and $w=9$.

### B. Encoding and Decoding Schemes

We continue in this subsection and discuss the encoding and decoding schemes for the code $\mathbb{D}$ obtained in Construction 1. Let $\boldsymbol{G}$ be a generator matrix of the $[N,K,D]_q$ code $\mathcal{C}$, where the last $N-D$ columns of $\boldsymbol{G}$ form a $K\times(N-D)$ submatrix $\boldsymbol{G}'$ whose rank is $K-1$. Furthermore, w.l.o.g. we assume that $\boldsymbol{G}$ has the form

$$\boldsymbol{G}=\begin{pmatrix} \boldsymbol{A} & \boldsymbol{I}_{K-1} \\ \boldsymbol{\beta}_K & 0\cdots 0 \end{pmatrix},$$

where $\boldsymbol{I}_{K-1}$ is the identity matrix of order $K-1$.

Each codeset in $\mathbb{D}$ will be identified by the unique vector from $\mathbb{F}_q^{K-1}$. This is possible since the number of codesets is $q^{K-1}$. For $\boldsymbol{\sigma}\in\mathbb{F}_q^{K-1}$, let $\mathcal{C}_{\boldsymbol{\sigma}}$ be the set of $q$ codewords from $\mathcal{C}$ whose suffix of length $K-1$ is $\boldsymbol{\sigma}$. Furthermore, let $\mathcal{C}'_{\boldsymbol{\sigma}}$ and $\mathcal{D}_{\boldsymbol{\sigma}}$ be the derived codesets as defined in Construction 1.

Given a $t$-subset $S$ of $\mathbb{F}_q\times[w]$ and a word $\boldsymbol{\sigma}=(\sigma_1,\sigma_2,\ldots,\sigma_{K-1})\in\mathbb{F}_q^{K-1}$, our objective for encoding of Construction 1 is to find a codeword $\boldsymbol{x}\in\mathcal{D}_{\boldsymbol{\sigma}}$ such that $\mathrm{supp}\,(\boldsymbol{x})\cap S=\varnothing$. For $1\leq i\leq K-1$, let $\boldsymbol{\beta}_i$ be the $i$-th row of $\boldsymbol{A}$. Let

$$\boldsymbol{r}=\boldsymbol{\sigma}\boldsymbol{A}|_{[w]}=\sum_{i=1}^{K-1}\sigma_i\boldsymbol{\beta}_i|_{[w]},$$

and hence the codeset $\mathcal{C}'_{\boldsymbol{\sigma}}$ is

$$\mathcal{C}'_{\boldsymbol{\sigma}}=\{\boldsymbol{r}+\lambda\boldsymbol{\beta}_K|_{[w]}:\lambda\in\mathbb{F}_q\}.$$

The codeset $\mathcal{D}_{\boldsymbol{\sigma}}$ is derived from $\mathcal{C}'_{\boldsymbol{\sigma}}$ as indicated in Construction 1, and hence we can consider the intersection of each one of the $q$ blocks in $\mathcal{D}_{\boldsymbol{\sigma}}$ with $S$ to find the block $B$ such that $B\cap S=\varnothing$.

Hence, for the encoding, $O(n)$ multiplications and $O(n)$ additions over $\mathbb{F}_q$ are required to find $\mathcal{D}_{\boldsymbol{\sigma}}$. During this computation we can also check whether each codeword of $\mathcal{D}_{\boldsymbol{\sigma}}$ has nontrivial intersection with $B$ or not. Therefore, there is no need for further computations to find $B$.

For the decoding, suppose that we have a codeword $\{(x_1,1),(x_2,2),\ldots,(x_w,w)\}$. By our choice we have that $w\geq N-D+1$, which implies that $D-1\geq N-w$ and hence we can correct any $N-w$ erasures in any codeword of $\mathcal{C}$. Hence, the $N-w$ erasures in $(x_1,x_2,\ldots,x_w,?,?,\ldots,?)$ can be recovered and the last $K-1$ symbols, $x_{N-K+2},x_{N-K+3},\ldots,x_N$ are the information symbols. In particular, if the code $\mathcal{C}$ is a Reed-Solomon code, then by using Lagrange interpolation, $O(w^3)$ multiplications are enough to perform the decoding, e.g., [14].

## IV. Error-Correcting CPC Codes

In this section we consider CPC codes that can correct transmission errors ('0' received as '1', or '1' received as '0'). An $(n,t,w)$-CPC which can correct up to $e$ errors will be called an $(n,t,w,e)$-CPECC (constant power error-correcting cooling) code. First, Construction 1 will be used to produce CPECC codes by examining the minimum distance of the constructed codes.

*Theorem 14:* If the code $\mathcal{C}$ used for Construction 1 is an $[N,K,D]_q$ code, then the code $\mathbb{D}$ obtained by Construction 1 is an $(n,t,w,e)$-CPECC code of size $M=q^{K-1}$, where $n=qw$, $t=q-1$, and $e\geq w+D-N-1$.

*Proof:* All the parameters of the code except for $e=w+D-N-1$ were proved in Theorem 11. Since the minimum distance of $\mathcal{C}$ is $D$ and the code $\mathcal{C}$ was punctured in the last $N-w$ coordinates to obtain the code $\mathcal{C}'$ (the union of the codesets $\mathcal{C}'_i$, $1\leq i\leq M$), it follows that the minimum distance of $\mathcal{C}'$ is at least $D-(N-w)$. By the definition of $\mathcal{D}'$ (the union of the codesets $\mathcal{D}'_i$, $1\leq i\leq M$) we have that if $\boldsymbol{x},\boldsymbol{x}'\in\mathcal{C}'$ differ in $\ell$ coordinates, then the related codewords in $\mathcal{D}$ differ in $2\ell$ positions. Hence, the minimum distance of $\mathcal{D}$ is at least $2(D+w-N)$ and thus the number of errors that it can correct is $e\geq w+D-N-1$. □

Next, an algorithm which demonstrates the error-correction for an $(n,t,w,e)$-CPECC code will be given. For simplicity, we will focus on a special example, where our starting point is a Reed-Solomon code $\mathcal{C}$ (which is of course an MDS code), where $K=N-D+1=w-e$.

*Construction 2:* Let $w$ and $e$ be positive integers and $q$ be a prime power such that $q\geq 2w-e-1$. Let $a_1,a_2,\ldots,a_w,b_1,b_2,\ldots,b_{w-e-1}$ be $2w-e-1$ distinct elements of $\mathbb{F}_q$.

- For each polynomial $f(X)\in\mathbb{F}_q[X]$ with $\deg(f)\leq w-e-1$, define the following block on the point set $\mathbb{F}_q\times[w]$,
$$C_f=\{(f(a_j),j):j\in[w]\}.$$
- For each $\boldsymbol{\sigma}=(\sigma_1,\sigma_2,\ldots,\sigma_{w-e-1})\in\mathbb{F}_q^{w-e-1}$, let
$$\mathcal{E}_{\boldsymbol{\sigma}}=\{C_f: f\in\mathbb{F}_q[X],\ \deg(f)\leq w-e-1,\ f(b_i)=\sigma_i \text{ for each } i\in[w-e-1]\}.$$

*Theorem 15:* The code $\mathbb{E}=\{\mathcal{E}_{\boldsymbol{\sigma}}:\boldsymbol{\sigma}\in\mathbb{F}_q^{w-e-1}\}$ is an $(n,t,w,e)$-CPECC code of size $q^{w-e-1}$, where $n=qw$ and $t=q-1$.

*Proof:* It is an immediate observation from the definition of the point set $\mathbb{F}_q\times[w]$ and the codeword $C_f$ that each codeword has length $qw$ and weight $w$. The rest of the proof has four steps. In the first one we will prove that for each $\boldsymbol{\sigma},\boldsymbol{\sigma}'\in F_q^{w-e-1}$, $\mathcal{E}_{\boldsymbol{\sigma}}$ and $\mathcal{E}_{\boldsymbol{\sigma}'}$ are disjoint whenever $\boldsymbol{\sigma}\neq\boldsymbol{\sigma}'$. In the second step we will prove that for each $\boldsymbol{\sigma}\in\mathbb{F}_q^{w-e-1}$ the blocks in $\mathcal{E}_{\boldsymbol{\sigma}}$ are pairwise disjoint. As a result, by a simple counting argument in the third step it will be proved that $\mathbb{E}$ has $q^{w-e-1}$ codesets, each one has a parallel class of size $q$, and as a consequence $\mathbb{E}$ is a $(qw,q-1,w)$-CPC code. In the last step we will find the minimum Hamming distance of $\mathbb{E}$ and as a result the number of errors $e$ that it can correct.

1) Assume that there exist two codewords $C_f\in\mathcal{E}_{\boldsymbol{\sigma}}$ and $C_g\in\mathcal{E}_{\boldsymbol{\sigma}'}$ such that $\boldsymbol{\sigma}\neq\boldsymbol{\sigma}'$ and $C_f=C_g$. Then $f$ and $g$ agree on at least $w$ points and since the degrees of the polynomials are less than $w$, it follows that $f=g$. This implies that $\sigma_i=f(b_i)=g(b_i)=\sigma'_i$ for all $i\in[w-e-1]$ and hence $\boldsymbol{\sigma}=\boldsymbol{\sigma}'$, a contradiction. Thus, $\mathcal{E}_{\boldsymbol{\sigma}}$ and $\mathcal{E}_{\boldsymbol{\sigma}'}$ are disjoint whenever $\boldsymbol{\sigma}\neq\boldsymbol{\sigma}'$.
2) Assume that the blocks $C_f$ and $C_g$ in $\mathcal{E}_{\boldsymbol{\sigma}}$, where $f\neq g$, intersect at the point $(x,i_0)$ for some $x\in\mathbb{F}_q$ and $i_0\in[w]$.

This implies that $f(a_{i_0}) = g(a_{i_0})$ and since $C_f, C_g \in \mathcal{E}_{\boldsymbol{\sigma}}$, it follows that $f(b_i) = g(b_i)$ for each $i \in [w-e-1]$. Therefore, $f$ and $g$ agree on at least $w-e$ points. Since the degrees of $f$ and $g$ are at most $w-e-1$, it follows that $f = g$, a contradiction. Therefore, the blocks in $\mathcal{E}_{\boldsymbol{\sigma}}$ are pairwise disjoint. Recall that each block has size $w$ and the size of the point set of these blocks $\mathbb{F}_q \times [w]$ is $qw$. Hence, each codeset $\mathcal{E}_{\boldsymbol{\sigma}}$ contains at most $q$ blocks.

3) The number of distinct polynomials in $\mathbb{F}_q[X]$ whose degrees are at most $w-e-1$ is $q^{w-e}$. Each polynomial induces exactly one codeword in $\mathbb{E}$. Hence, $\mathbb{E}$ contains exactly $q^{w-e}$ distinct codewords. Since there are $q^{w-e-1}$ codesets and each one contains at most $q$ codewords, it follows that each one contains exactly $q$ codewords. The length of a codeword is $qw$ and the weight of a codeword is $w$, which implies that each codeset is a parallel class. Thus, by Proposition 8, $\mathbb{E}$ is a $(qw, q-1, w)$-CPC code.
4) Finally, for any two distinct codewords $C_f$ and $C_g$, where $f$ and $g$ have degree at most $w-e-1$, we have that $|C_f \cap C_g| \leq w-e-1$ since a larger intersection implies that $f = g$. Therefore, the Hamming distance between $C_f$ and $C_g$ is at least $2e+2$. Thus, the code $\mathbb{E}$ has minimum Hamming distance at least $2e+2$ and it can correct $e$ errors.

Thus, $\mathbb{E}$ is an $(n, t, w, e)$-CPECC code of size $q^{w-e-1}$, where $n = qw$ and $t = q-1$. □

The encoding scheme in Section III-B can be easily adapted for the encoding of the CPECC code $\mathbb{E}$. Algorithm 1 illustrates the decoding scheme for the $(n, t, w, e)$-CPECC code $\mathbb{E}$ obtained in Construction 2.

**Algorithm 1** Error-Correction for the CPECC Codes in Construction 2

**Input:** a binary word $\boldsymbol{u} \subset \mathbb{F}_q \times [w]$ {the word received after the transmission of a codeword}
**Output:** a message $\boldsymbol{\sigma} \in \mathbb{F}_q^{w-e-1}$ {the information word that was sent}

**for** each $i \in [w]$ **do**
  $Y_i \leftarrow \{(y,i) : (y,i) \in \boldsymbol{u}\}$
  **if** $|Y_i| = 1$ **then**
    $y_i \leftarrow y$, where $(y,i)$ is the unique pair in $Y_i$
  **else**
    $y_i \leftarrow$ '?'
$\hat{\boldsymbol{y}} \leftarrow (y_1, y_2, \ldots, y_w)$
apply some error-erasure decoding algorithm for Reed-Solomon codes for $\hat{\boldsymbol{y}}$
The output of the algorithm is a polynomial $L(x)$ of degree $w-e-1$
$\boldsymbol{\sigma} \leftarrow (L(b_1), L(b_2), \ldots, L(b_{w-e-1}))$
**return** $\boldsymbol{\sigma}$

*Theorem 16:* Suppose that the codeword $\boldsymbol{c} \in \mathbb{E}$ obtained in Construction 2 was transmitted and the word $\boldsymbol{u}$ was received from $\boldsymbol{c}$ with at most $e$ errors. Then, Algorithm 1 returns the word $\boldsymbol{\sigma} \in \mathbb{F}_q^{w-e-1}$ such that $\boldsymbol{c} \in \mathcal{E}_{\boldsymbol{\sigma}}$.

*Proof:* Using the notation of Algorithm 1, let $i \in [w]$, $Y_i \triangleq \{(y,i) : (y,i) \in \boldsymbol{u}\}$, and $e' = |\{i : |Y_i| \neq 1\}|$. If $|Y_i| = 0$ then an error occurred and this is reflected as an erasure in $y_i$. If $|Y_i| > 1$ then we also know that an error has occurred for at least one coordinate $(y,i)$. This will be also reflected as an erasure in $y_i$. Hence, at least $e'$ erasure errors are reflected in $\hat{\boldsymbol{y}}$ as a result of at least $e'$ errors in these $Y_i$'s. For the remaining $w-e'$ $Y_i$'s, while there may be errors, we know that each of these $Y_i$'s contains either no errors or two errors. Thus, the number of other erroneous $Y_i$'s is at most $\lfloor (e-e')/2 \rfloor$.

The vector $\hat{\boldsymbol{y}}$ is obtained by mapping the subsets $Y_1, Y_2, \ldots, Y_w$ to the elements of $\mathbb{F}_q \cup \{?\}$. The word $\hat{\boldsymbol{y}}$ was obtained from a codeword $\boldsymbol{x}_f$ of a Reed-Solomon code of length $N = w$, dimension $K = w-e$, and minimum Hamming distance $D = N-K+1 = e+1$. An error-correction algorithm for such a code is capable of correcting $e'$ erasures and at most $\lfloor (e-e')/2 \rfloor$ errors as required by Algorithm 1. □

Using the Berlekamp-Welch algorithm [22] we can correct the errors with $O(q^3)$ operations [22], and hence, Algorithm 1 has complexity $O(n^3)$.

## V. Recursive Construction

Since both Proposition 5 and Construction 1 use disjoint (partial) parallel classes to construct $(n, t, w)$-CPC codes and each (partial) parallel class contains at most $n/w$ blocks, all the codes obtained from these two methods have $t \leq n/w - 1$. In this section, we present a recursive construction that yields $(n, t, w)$-CPC codes which are useful especially for larger values of $t$, i.e., $t \geq n/w$.

The basic idea of our recursive construction is to break the blocks in the (partial) parallel classes using a CPC code of small length. We use the following example to illustrate our idea.

*Example 6:* Let $X = [12]$ and $\mathcal{P} = \{\{1,2,3,4,5,6\}, \{7,8,9,10,11,12\}\}$. Since $\mathcal{P}$ is a parallel class of $X$, for any subset of $X$ consisting of a single point, we can always find a block in $\mathcal{P}$ to avoid this subset. However, for subsets of $X$ with size larger than 1, it may intersect both of these blocks in $\mathcal{P}$.

Now, consider a $(6, 2, 3)$-CPC code consisting of the following two codesets:

$$\mathcal{C}_1 = \{\{a,b,c\}, \{a,b,d\}, \{c,d,e\}, \{c,d,f\}, \{e,f,a\}, \{e,f,b\}\},$$
$$\mathcal{C}_2 = \{\{a,b,e\}, \{a,b,f\}, \{c,d,a\}, \{c,d,b\}, \{e,f,c\}, \{e,f,d\}\}.$$

Using these codesets, we break the blocks in $\mathcal{P}$ of size six into blocks of size three. Specifically, using the codeset $\mathcal{C}_1$, the block $\{1,2,3,4,5,6\}$ is broken into the blocks $\{1,2,3\}$, $\{1,2,4\}$, $\{3,4,5\}$, $\{3,4,6\}$, $\{5,6,1\}$, $\{5,6,2\}$, while the block $\{7,8,9,10,11,12\}$ is broken into $\{7,8,9\}$, $\{7,8,10\}$, $\{9,10,11\}$, $\{9,10,12\}$, $\{11,12,7\}$, $\{11,12,8\}$. This set of blocks of size three then forms our new codeset $\mathcal{D}_1$. We do so similarly for the codeset $\mathcal{D}_2$:

$$\mathcal{D}_1 = \Big\{ \{1,2,3\}, \{1,2,4\}, \{3,4,5\}, \{3,4,6\}, \{5,6,1\}, \{5,6,2\}, \{7,8,9\}, \{7,8,10\}, \{9,10,11\}, \{9,10,12\}, \{11,12,7\}, \{11,12,8\} \Big\},$$

$$\mathcal{D}_2 = \Big\{\{1,2,5\},\{1,2,6\},\{3,4,1\},\{3,4,2\},\{5,6,3\},\{5,6,4\},\{7,8,11\},\{7,8,12\},\{9,10,7\},\{9,10,8\},\{11,12,9\},\{11,12,10\}\Big\}.$$

We show the codesets $\mathcal{D}_1$ and $\mathcal{D}_2$ form a $(12,5,3)$-CPC. For any 5-subset $S$ of $X$, by the pigeonhole principle, it intersects at least one of $\{1,2,3,4,5,6\}$ and $\{7,8,9,10,11,12\}$ in at most 2 points. Since we use a $(6,2,3)$-CPC code to break the blocks of length 6. We can find a block of size 3 from each of $\mathcal{D}_1$ and $\mathcal{D}_2$ to avoid $S$. For example, the subset $\{1,2,7,8,9\}$ intersects the block $\{1,2,3,4,5,6\}$ in two points and we can find $\{3,4,5\} \in \mathcal{D}_1$ and $\{5,6,3\} \in \mathcal{D}_2$ to avoid it.

In general, suppose that we have a set system $(X, \mathcal{B})$ where $\mathcal{B} \subseteq \binom{X}{w}$ and $\mathcal{B}$ can be partitioned into $M$ partial parallel classes $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_M$ with each $\mathcal{P}_i$ containing exactly $q$ blocks. Let $S$ be a $t$-subset of $X$ with $t \geq q$. For each $\mathcal{P}_i$, by the pigeonhole principle, we can find such a block which intersects $S$ in at most $\lfloor t/q \rfloor$ points. If there is a $(w, \lfloor t/q \rfloor, w')$-LPC code $\mathbb{C}$, it is possible to substitute each block of $\mathcal{B}$ with $\mathbb{C}$ by breaking up the block of size $w$ into blocks of size $w'$. This enables us to find a block of size $w'$ which avoids $S$. The following construction is based on this idea, where the code $\mathbb{E}$ is constructed similarly to the code in Construction 2.

*Construction 3:* Let $q \geq n+w-1$ be a prime power and let $a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_{w-1}$ be $n+w-1$ distinct elements of $\mathbb{F}_q$.

- Consider the point set $\mathbb{F}_q \times [n]$ and let
$$\mathcal{B} = \{C_f \triangleq \{(f(a_j), j) : j \in [n]\} : f \in \mathbb{F}_q[x], \deg(f) \leq w-1\}.$$
Note that the size of each block $C_f$ is $n$.
- For each $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \ldots, \sigma_{w-1}) \in \mathbb{F}_q^{w-1}$, let
$$\mathcal{E}_{\boldsymbol{\sigma}} = \{C_f : f \in \mathbb{F}_q[X],\ \deg(f) \leq w-1,\ f(b_i) = \sigma_i \text{ for each } i \in [w-1]\}.$$
Similarly to the proof of Theorem 15, one can show that $\mathcal{B}$ is partitioned by $\mathcal{E}_{\boldsymbol{\sigma}}$, $\boldsymbol{\sigma} \in \mathbb{F}_q^{w-1}$ into $q^{w-1}$ parallel classes, each one of size $q$. Label the parallel classes and their blocks by $\mathcal{P}_i = \{B_{ij} : j \in [q]\}$ for $i \in [q^{w-1}]$.
- Let $\mathbb{D}$ be an $(n,t,w)$-CPC code of size $m$ with point set $[n]$, where $t \geq n/w$.
- For each block $B = \{(x_1,1),(x_2,2),\ldots,(x_n,n)\} \in \mathcal{B}$ and each codeword/block $\{i_i, i_2, \ldots, i_w\} \in \mathbb{D}$, we construct a $w$-subset of $B$, i.e., $\{(x_{i_1}, i_1), (x_{i_2}, i_2), \ldots, (x_{i_w}, i_w)\}$, which is the codeword of the output CPC code.
Since $\mathbb{D}$ has $m$ codesets, from each block $B_{ij} \in \mathcal{B}$ we can get correspondingly $m$ new codesets, denoted as $\mathcal{E}_{ij\ell}$ for $\ell \in [m]$.
- For $(i,\ell) \in [q^{w-1}] \times [m]$, the codeset $\mathcal{E}_{i\ell}$ is defined by $\mathcal{E}_{i\ell} \triangleq \bigcup_{j=1}^{q} \mathcal{E}_{ij\ell}$.

Along the same lines of the proof in Theorem 15 one can prove that

*Theorem 17:* The code $\{\mathcal{P}_i : 1 \leq i \leq q^{w-1}\}$ from Construction 3 is an $(nq, q-1, n)$-CPC code.

*Theorem 18:* The code $\mathbb{E} = \{\mathcal{E}_{i\ell} : i \in [q^{w-1}],\ \ell \in [m]\}$ from Construction 3 is an $(nq, tq, w)$-CPC code of size $mq^{w-1}$.

*Proof:* The size of $\mathbb{E}$, the length of its codewords and their weight follow immediately from the definition of the codewords in $\mathbb{E}$.

Given a $(tq)$-subset $S \subset \mathbb{F}_q \times [n]$ and a codeset $\mathcal{E}_{i\ell}$, $(i,\ell) \in [q^{w-l}] \times [m]$, we should find a codeword $\boldsymbol{x} \in \mathcal{E}_{i\ell}$ such that $\mathrm{supp}(\boldsymbol{x}) \cap S = \varnothing$. Since $\mathcal{E}_{i\ell}$ was constructed from the $q$ blocks of $\mathcal{P}_i$ in which the codewords of the $\ell$-th codeset of $\mathbb{D}$ were substituted, we have to find first a block $B_{ij} \in \mathcal{P}_i$ which contains a subset $S'$ of $S$ whose size is at most $t$. Such a block exists since the number of blocks in $\mathcal{P}_i$ is $q$ and $S$ has size $tq$. Since $\mathcal{E}_{ij\ell}$ is a codeset in an $(n,t,w)$-CPC code, we can find a block $\boldsymbol{x}$ in $\mathcal{E}_{ij\ell}$ which avoids $S'$. As a consequence $\mathrm{supp}(\boldsymbol{x}) \cap S = \varnothing$ as required.

To complete the proof we have to show that all the codesets of $\mathbb{E}$ are pairwise disjoint, i.e., $\mathcal{E}_{i\ell}$ and $\mathcal{E}_{i'\ell'}$ are disjoint whenever $(i,\ell) \neq (i',\ell')$. To this end, it suffices to show $\mathcal{E}_{ij\ell}$ and $\mathcal{E}_{i'j'\ell'}$ are disjoint for any $j, j' \in [q]$. If $(i,j) \neq (i',j')$, it can be verified that $|B_{ij} \cap B_{i'j'}| \leq w-1$ since intersection of size $w$ will imply that the related polynomials are equal. Hence, since each $\mathcal{E}_{ij\ell}$ is a collection of $w$-subsets of $B_{ij}$, we have that $\mathcal{E}_{ij\ell}$ and $\mathcal{E}_{i'j'\ell'}$ are disjoint. If $(i,j) = (i',j')$ then $\mathcal{E}_{ij\ell}$ and $\mathcal{E}_{ij\ell'}$ are from the same $(n,t,w)$-CPC code and therefore they are disjoint. □

Construction 3 only takes an $(n,t,w)$-CPC code with $t \geq n/w$ as input and can yield an $(nq, tq, w)$-CPC code for any prime power $q \geq n+w+1$. Since $tq \geq nq/w$, we can use this $(nq, tq, w)$-CPC code as input and apply Construction 3 again. Hence, from an $(n,t,w)$-CPC code with $t \geq n/w$, we can apply Construction 3 recursively to obtain an infinite class of CPC codes.

We use the following example to illustrate the encoding and decoding for the codes in Construction 3.

*Example 7:* Set $q = 11$, $n = 6$, $t = 2$, $w = 3$ and $m = 2$. Let $a_i = i$ for $1 \leq i \leq 6$, $b_1 = 8$ and $b_2 = 9$. Given a pair $(\boldsymbol{\sigma}, i) \in \mathbb{F}_q^{w-1} \times [m]$, where $\boldsymbol{\sigma} = (0,0)$ and $i = 2$. There are 11 polynomials $f(x)$ such that $\deg(f) \leq 2$ and $f(8) = f(9) = 0$, i.e., $f(x) = c(6 + 5x + x^2)$ where $c \in \mathbb{F}_{11}$. Thus,

$$\mathcal{E}_{\boldsymbol{\sigma}} = \left\{ \begin{array}{c} \{(0,1),(0,2),(0,3),(0,4),(0,5),(0,6)\} \\ \{(1,1),(9,2),(8,3),(9,4),(1,5),(6,6)\} \\ \vdots \\ \{(10,1),(2,2),(3,3),(2,4),(10,5),(5,6)\} \end{array} \right\}.$$

Using the second codeset of the $(6,2,3)$-CPC code in Example 6, we can obtain the following codeset.

$$\mathcal{E}_{\boldsymbol{\sigma},2} = \left\{ \begin{array}{c} \{(0,1),(0,2),(0,5)\},\{(0,1),(0,2),(0,6)\}, \\ \{(0,3),(0,4),(0,1)\},\{(0,3),(0,4),(0,2)\}, \\ \{(0,5),(0,6),(0,3)\},\{(0,5),(0,6),(0,4)\}, \\ \{(1,1),(9,2),(1,5)\},\{(1,1),(9,2),(6,6)\}, \\ \{(8,3),(9,4),(1,1)\},\{(8,3),(9,4),(9,2)\}, \\ \{(1,5),(6,6),(8,3)\},\{(1,5),(6,6),(9,4)\}, \\ \vdots \\ \{(10,1),(2,2),(10,5)\},\{(10,1),(2,2),(5,6)\}, \\ \{(3,3),(2,4),(10,1)\},\{(3,3),(2,4),(2,2)\}, \\ \{(10,5),(5,6),(3,3)\},\{(10,5),(5,6),(2,4)\} \end{array} \right\}$$

Now, we consider the decoding. Assume we receive a block $\{(8,3),(9,4),(9,2)\}$. By interpolating the three points in this block with a polynomial of degree at most 2, we can obtain $f(x)=6+5x+x^2$ and thus decode $\boldsymbol{\sigma}$ as $\boldsymbol{\sigma}=(f(8),f(9))=(0,0)$. Since the set $\{3,4,2\}$ comes from the second codeset of the $(6,2,3)$-CPC code, we have that $\{(8,3),(9,4),(9,2)\}\in\mathcal{E}_{(0,0),2}$.

Generally, for the encoding of the codes in Construction 3, given a message $(\boldsymbol{\sigma},i)\in\mathbb{F}_q^{w-1}\times[m]$ we first encode $\boldsymbol{\sigma}$ into $\mathcal{E}_{\boldsymbol{\sigma}}$. This can be done in $O(q)$, as shown in Section III-B. Then we run the encoding of $\mathbb{D}$ on $i$ and each block in $\mathcal{E}_{\boldsymbol{\sigma}}$ to obtain the codeset $\mathcal{E}_{\boldsymbol{\sigma},i}$. For the decoding, given a block of size $w$, we first find a polynomial $f$ of degree at most $w-1$ which can interpolate the points in the block. Then we evaluate $f$ on $b_1,b_2,\ldots,b_{w-1}$ to decode $\boldsymbol{\sigma}$. Finally, we run the decoder of $\mathbb{D}$ on the set of second coordinates of the given block to decode $i$.

Construction 3 can be applied also on $(n,t,w)$-LPC code (instead of $(n,t,w)$-CPC code). The only condition is that there is no codeset in which there are codewords of different weight. Also, when there are codewords of weight $w'<w$ in the codeset, the whole construction should work with $w'$ instead of $w$, e.g., the degree of the polynomial must be at most $w'-1$.

*Corollary 19:* Let $q$ be a prime power. If $t+w\leq n$ and $q\geq n+w-1$, then

(i) there exists an $(nq,tq,w)$-CPC code of size $q^{w-1}$;
(ii) there exists an $(nq,tq,w)$-LPC code of size $\sum_{i=0}^{w-1}q^i$.

*Proof:*

(i) Since $t+w\leq n$, all the $w$-subset of the set $[n]$ form an $(n,t,w)$-CPC code of size 1. We use this CPC code as the code $\mathbb{D}$ in Construction 3 and apply Theorem 18 with $m=1$ to obtain an $(nq,tq,w)$-CPC code of size $q^{w-1}$.
(ii) Apply Construction 3 and Corollary 19(i) on any $w'\leq w$ to obtain a family of disjoint $(nq,tq,w')$-CPC code of size $q^{w'-1}$. Then combine these CPC codes together to obtain an $(nq,tq,w)$-LPC code of size $\sum_{i=0}^{w-1}q^i$. □

Note that the conditions $t+w\leq n$ and $q\geq n+w-1$ do not exclude the region $t\geq n/w$. Thus in Corollary 19, we may choose proper parameters such that the cooling capability satisfies $tq\geq nq/w$.

The following example shows that although Proposition 7 also works for $t\geq n/w$, Construction 3 can get better code rates in some cases.

*Example 8:* We compare certain CPC codes obtained from Construction 3 and Corollary 19 with the LPC codes obtained from Proposition 7.

(i) Consider the set of five disjoint 3-$(10,4,1)$ designs constructed by Etzion and Hartman [8]. By taking the complements of the blocks we obtain a $(10,3,6)$-CPC code of size five. Applying Construction 3 with $q=16$, we obtain a $(160,48,6)$-CPC code of size $5\cdot16^5\approx2^{22.322}$.
In contrast, Proposition 7 yields a $(160,48,6)$-LPC code of size $2^{17}$ by setting $s=137$ and $r=95$.
(ii) Setting $n=9$, $t=2$, $w=7$, and $q=16$ in Corollary 19 yields a $(144,32,7)$-LPC code of size $\sum_{i=0}^{6}16^i\approx2^{24.093}$.
In contrast, Proposition 7 yields a $(144,32,7)$-LPC code of size $2^{18}$ by setting $s=121$ and $r=94$.

In the regime where $w$ is fixed and $t$ has order of magnitude as $n$, we can show as follows that the codes obtained in this section are asymptotically larger than those obtained from Proposition 7. Namely, the CPC codes obtained from Construction 3 and Corollary 19 attain the asymptotic upper bound $O((nq)^{w-1})$ when $w$ is fixed. In contrast, if we apply Proposition 7 with $s=nq-\lceil\log_2(\sum_{i=0}^{w-1}\binom{nq-1}{i})\rceil$ (the Gilbert-Varshamov lower bound) and $r=nq-tq-\lfloor\log_2(\sum_{i=0}^{\frac{w}{2}}\binom{nq-tq}{i})\rfloor$ (the Hamming upper bound), we obtain an $(nq,tq,w)$-LPC code of smaller size $O((nq)^{w/2})$, or $o((nq)^{w-1})$.

## VI. LPC Codes From Cooling Codes

### A. A Method Based on Domination Mappings

In this section we use a novel method to transform cooling codes into low-power cooling codes, while preserving the efficiency of the cooling codes. The construction is based on an injective mapping called domination mapping which was defined as follows in [5].

The *Hamming ball of radius* $w$ *in* $\{0,1\}^n$ is the set $\mathcal{B}(n,w)$ of all words of weight at most $w$. Explicitly, $\mathcal{B}(n,w)\triangleq\{\boldsymbol{y}\in\{0,1\}^n:\mathrm{wt}(\boldsymbol{y})\leq w\}$. Given $m\leq n$, we are interested in injective mappings $\varphi$ from $\{0,1\}^m$ into $\mathcal{B}(n,w)$ that establish a certain domination relationship between components of $\boldsymbol{x}\in\{0,1\}^m$ and components of its image $\boldsymbol{y}=\varphi(\boldsymbol{x})$. Specifically, one should be able to "switch off" every position $j\in[n]$ in $\boldsymbol{y}$ (that is, ensure that $y_j=0$) by switching off a corresponding position $i\in[m]$ in $\boldsymbol{x}$ (that is, setting $x_i=0$). More precisely, let $G=\big([m]\cup[n],E\big)$ be a bipartite graph with $m$ left vertices and $n$ right vertices. If $G$ has no isolated right vertices, i.e., none of the right vertices has degree zero, we refer to $G$ as a *domination graph*.

*Definition 3:* Given an injective map $\varphi:\{0,1\}^m\to\mathcal{B}(n,w)$ and a graph $G=\big([m]\cup[n],E\big)$, we say that $\varphi$ is a *$G$-domination mapping*, or *$G$-dominating* in brief, if for all $(x_1,x_2,\ldots,x_m)\in\{0,1\}^m$ and $\varphi(x_1,x_2,\ldots,x_m)=(y_1,y_2,\ldots,y_n)$, $x_i=0$ implies $y_j=0$ for all $(i,j)\in E$.

We say that $\varphi$ is an *$(m,n,w)$-domination mapping* if there exists a domination graph $G=\big([m]\cup[n],E\big)$, such that $\varphi$ is $G$-dominating.

Properties of domination mappings, bounds on their parameters, constructions, and existence theorems were given in [5]. In this paper, we develop a method of constructing LPC codes that uses domination mappings. In particular, we demonstrate Theorem 22, which allows one to construct LPC codes from cooling codes and domination mappings. For our purpose, we need some results from [5] and the first lemma, taken from [5], restricts the structure of the domination graph. Given this restricted structure, we provide a technical lemma (Lemma 21) and prove Theorem 22.

*Lemma 20:* The domination graph $G=\big([m]\cup[n],E\big)$ of an $(m,n,w)$-domination mapping has a subgraph in which

every vertex has degree at least one and the degree of the right vertices is exactly one.

In view of Lemma 20 we henceforth assume that our domination graphs have no isolated vertices and all the right vertices have degree exactly one. We will define the *neighborhood* of a vertex $\boldsymbol{v}$ in $G$ as the set of vertices adjacent to $\boldsymbol{v}$ and denote it by $N(\boldsymbol{v})$. The following lemma is an immediate consequence of these observations and definition.

*Lemma 21:* Let $G = \big([m] \cup [n], E\big)$ be the domination graph of an $(m, n, w)$-domination mapping. If $U \subset [n]$ is a set of right vertices of $G$ then $N(U) \triangleq \{N(\boldsymbol{u}) \ : \ \boldsymbol{u} \in U\}$ is a set of vertices in $[m]$ and $|N(U)| \leq |U|$.

Next, the obvious connection between domination mappings, cooling codes, and low-power cooling codes is given in the following theorem.

*Theorem 22:* If there exists an $(m, t)$-cooling code $\mathbb{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_M\}$ and an injective $(m, n, w)$-domination mapping $\varphi$, then the code $\mathbb{C}' = \{\mathcal{C}'_1, \mathcal{C}'_2, \ldots, \mathcal{C}'_M\}$, where

$$\mathcal{C}'_i \triangleq \{\varphi(\boldsymbol{x}) \ : \ \boldsymbol{x} \in \mathcal{C}_i\}, \text{ for each } 1 \leq i \leq M,$$

is an $(n, t, w)$-LPC code.

*Proof:* The length $n$ and the weight which is smaller from or equal to $w$ for the codewords of $\mathbb{C}$ are immediate consequences from the definition of the $(m, n, w)$-domination mapping. Now, suppose we are given a $t$-subset $S' \subset [n]$ and a codeset $\mathcal{C}'_i$ for some $1 \leq i \leq M$. To complete the proof we have to show that there exists a codeword $\boldsymbol{u'} \in \mathcal{C}'_i$ such that $\mathrm{supp}(\boldsymbol{u'}) \cap S' = \varnothing$. The $t$-subset $S'$ can be viewed as a set of right vertices in the domination graph $G = \big([m] \cup [n], E\big)$. By Lemma 21, for the set of neighbors of $S' \subset [n]$, $S \triangleq N(S') \subset [m]$, we have that $|S| \leq |S'|$ and hence $|S| \leq t$. Since $\mathbb{C}$ is an $(m, t)$-cooling code, it follows that there exists a codeword $\boldsymbol{u}$ in $\mathcal{C}_i$ such that $\mathrm{supp}(\boldsymbol{u}) \cap S = \varnothing$, which implies by the domination property that $\mathrm{supp}(\varphi(\boldsymbol{u})) \cap S' = \varnothing$. □

A product construction for domination mappings was presented in [5].

Let $\varphi_1 : \{0,1\}^{m_1} \rightarrow \mathcal{B}(n_1, w_1)$ and $\varphi_2 : \{0,1\}^{m_2} \rightarrow \mathcal{B}(n_2, w_2)$ be arbitrary domination mappings. Then their *product* $\varphi = \varphi_1 \times \varphi_2$ is a mapping from $\{0,1\}^{m_1+m_2}$ into $\mathcal{B}(n_1 + n_2, w_1 + w_2)$ defined as follows:

$$\varphi(\boldsymbol{x}_1, \boldsymbol{x}_2) \ = \ \big(\varphi_1(\boldsymbol{x}_1), \varphi_2(\boldsymbol{x}_2)\big)$$

where $\boldsymbol{x}_1 \in \{0,1\}^{m_1}$, $\boldsymbol{x}_2 \in \{0,1\}^{m_2}$, and $(\cdot, \cdot)$ stands for string concatenation. That is, in order to find the image of a word $\boldsymbol{x} \in \{0,1\}^{m_1+m_2}$ under $\varphi$, we first parse $\boldsymbol{x}$ as $(\boldsymbol{x}_1, \boldsymbol{x}_2)$, then apply $\varphi_1$ and $\varphi_2$ to the two parts.

*Theorem 23:* If $\varphi_1$ is an $(m_1, n_1, w_1)$-domination mapping and $\varphi_2$ is an $(m_2, n_2, w_2)$-domination mapping, then their product $\varphi = \varphi_1 \times \varphi_2$ is an $(m_1 + m_2, n_1 + n_2, w_1 + w_2)$-domination mapping.

The idea in Theorem 23 can be generalized as follows to a large number of domination mappings.

*Theorem 24:* For each $1 \leq i \leq \ell$, let $\varphi_i$ be an $(m_i, n_i, w_i)$-domination mapping. Let $(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_\ell)$ be a binary word, where the length of $\boldsymbol{x}_i$ is $m_i$ for $1 \leq i \leq \ell$. The mapping $\varphi$, defined by

$$\varphi(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_\ell) = (\varphi_1(\boldsymbol{x}_1), \varphi_2(\boldsymbol{x}_2), \ldots, \varphi_\ell(\boldsymbol{x}_\ell)),$$

is also an $(m, n, w)$-domination mapping for $m = \sum_{i=1}^{\ell} m_i$, $n = \sum_{i=1}^{\ell} n_i$, and $w = \sum_{i=1}^{\ell} w_i$.

In [5], the problem of constructing an $(m, n, w)$-domination mapping was reduced to finding a perfect mapping in an associated bipartite graph of size $\Theta(2^m)$. Even though the size of the graph was exponential in $m$, we used symmetry to reduce the problem size and demonstrated that the existence problem can be determined in time polynomial in $m$ and $w$. For small cases, a $(2, 3, 1)$-domination mapping, $(9, 15, 3)$-domination mapping, and $(12, 20, 4)$-domination mapping were explicitly constructed in the same paper. Efficient encoding and decoding procedures for these mappings were also presented.

Using domination mapping with small parameters, we then apply Theorem 24 with Theorem 22 to obtain an infinite family of LPC codes. Specifically, we have the following corollary.

*Corollary 25:* Let $\lambda$, $\mu$, $w_1$, and $w_2$ be integers such that $(\lambda w_i, \mu w_i, w_i)$-domination mappings exist for $i \in [2]$. Suppose that $w$ can be written as $\alpha w_1 + \beta w_2$, where $\alpha$ and $\beta$ are nonnegative integers. If $m = \lambda w$, $n = \mu w$, and there exists an $(m, t)$-cooling code of size $M$, then there exists an $(n, t, w)$-LPC code of size $M$.

Given a $(2, 3, 1)$-domination mapping, we set $\lambda = 2$, $\mu = 3$, $w_1 = w_2 = 1$ in above corollary to obtain the following.

*Corollary 26:* For $w \geq 1$, if there exists a $(2w, t)$-cooling code of size $M$, then there exists a $(3w, t, w)$-LPC code of size $M$.

Given a $(9, 15, 3)$-domination mapping and a $(12, 20, 4)$-domination mapping, we set $\lambda = 3$, $\mu = 5$, $w_1 = 3$, $w_2 = 4$ in Corollary 25 to obtain the following.

*Corollary 27:* For $w \geq 6$, if there exists a $(3w, t)$-cooling code of size $M$, then there exists a $(5w, t, w)$-LPC code of size $M$.

### *B. Encoding and Decoding Schemes*

Therefore, one can use an $(m, n, w)$-domination mapping $\varphi$ to construct an $(n, t, w)$-LPC code $\mathbb{C}$ from an $(m, t)$-cooling code $\mathbb{D}$. The only question is whether there are efficient encoding and decoding schemes for the constructed LPC code $\mathbb{C}$. Such encoding and decoding schemes should be based on efficient encoding and decoding schemes for both the cooling code $\mathbb{D}$ and the domination mapping $\varphi$. While efficient algorithms are known for the cooling code $\mathbb{D}$, less is known for the domination mapping $\varphi$. Hence, we focus our discussion on domination mappings that are obtained from applying the product construction (Theorem 24) on domination mappings with small parameters. Specifically, we describe the encoding procedure for the family of LPC codes obtained from Corollary 25.

Recall that for integers $\lambda$, $\mu$, $w_1$, $w_2$, we write $w = \alpha w_1 + \beta w_2$, where $\alpha$ and $\beta$ are nonnegative integers, and set $m = \lambda w$ and $n = \mu w$. Here, $\lambda$, $\mu$, $w_1$, $w_2$ are constants and so, $\alpha + \beta = O(m)$. In addition, we assume the following ingredients:

(i) For $i \in [2]$, there exist $(\lambda w_i, \mu w_i, w_i)$-domination mappings $\varphi_i$ that compute $\varphi_i$ in constant time.

(ii) There exists an $(m, t)$-cooling code $\mathbb{D} = \{D_1, D_2, \ldots, D_M\}$ of size $M$ with a corresponding encoding

procedure $\psi$ that can be computed in $T_{\mathbb{D}}(m)$ time. Here, given a message $\sigma \in [M]$ and a $t$-subset $S$ of $[m]$, then $\boldsymbol{x} \triangleq \psi(\sigma)$ belongs to $D_\sigma$ and has the property $\mathrm{supp}(\boldsymbol{x}) \cap S = \varnothing$.

Corollary 25 then yields an $(n, t, w)$-LPC code $\mathbb{C} = \{C_1, C_2, \ldots, C_M\}$ of size $M$. In what follows, we provide an encoding scheme that maps messages in $[M]$ to $\mathbb{C}$ using the mappings $\varphi_1$, $\varphi_2$ and encoding $\psi$.

Given a message $\boldsymbol{u} \in [M]$ and a $t$-subset $S'$ of $[n]$, our objective is to find $\boldsymbol{y} \in D_{\boldsymbol{u}}$ such that $\mathrm{supp}(\boldsymbol{y}) \cap S' = \varnothing$.

- To do so, we partition the set of $m$ coordinates into $\alpha+\beta$ blocks: $\alpha$ blocks of size $\lambda w_1$ and $\beta$ blocks of size $\lambda w_2$. Similarly, we partition the set of $n$ coordinates into $\alpha$ blocks of size $\mu w_1$ and $\beta$ blocks of size $\mu w_2$.
- Let $\varphi$ be the $(m, n, w)$-domination mapping obtained by the product construction of Theorem 24 on $\alpha$ copies of $\varphi_1$ and $\beta$ copies of $\varphi_2$.
- Let $S'$ be a $t$-subset of $[n]$ and we compute $S = N(S')$ be a $t'$-subset of $[m]$, where $t' \leq t$ by Lemma 21. Here, $S$ can be computed in $O(\alpha + \beta) = O(m)$ time.
- Applying the encoder $\psi$ to $\boldsymbol{u}$ and $S$, we obtain the word $\boldsymbol{v}$ such that $\boldsymbol{v} \in C_{\boldsymbol{u}}$ and $\mathrm{supp}(\boldsymbol{v}) \cap S = \varnothing$; $\boldsymbol{v}$ can be computed in $T_{\mathbb{D}}(m)$ time.
- Parse $\boldsymbol{v}$ into $\boldsymbol{v}_1\boldsymbol{v}_2 \ldots \boldsymbol{v}_\alpha \boldsymbol{v}'_1\boldsymbol{v}'_2 \ldots \boldsymbol{v}'_\beta$, where $\boldsymbol{v}_i$ and $\boldsymbol{v}'_j$ are of lengths $\lambda w_1$ and $\lambda w_2$, respectively, for $i \in [\alpha]$ and $j \in [\beta]$.
- By using the mappings $\varphi_1$ and $\varphi_2$, we compute $\boldsymbol{y} = \varphi(\boldsymbol{v})$ by setting $\boldsymbol{y}_i = \varphi_1(\boldsymbol{v}_i)$ for $i \in [\alpha]$, $\boldsymbol{y}'_j = \varphi_1(\boldsymbol{v}'_j)$ for $i \in [\beta]$, and $\boldsymbol{y} = \boldsymbol{y}_1\boldsymbol{y}_2 \ldots \boldsymbol{y}_\alpha \boldsymbol{y}'_1\boldsymbol{y}'_2 \ldots \boldsymbol{y}'_\beta$. Since the mappings $\varphi_1$ and $\varphi_2$ can be computed in constant time, this step can be completed in $O(\alpha + \beta) = O(m)$ time.

Therefore, the LPC code constructed in Corollary 25 admits an encoding scheme that computes a codeword in $T_{\mathbb{D}}(m) + O(m)$ time. In [3], whenever $t + 1 \leq m/2$, an $(m, t)$-cooling code with encoding and decoding complexity $O(m^3)$ is constructed (see also Theorem 28). Hence, the two families of LPC codes obtained from Corollaries 26 and 27 have encoding and decoding complexity $O(m^3) = O(n^3)$.

### C. Comparison With Proposition 6

How good are the LPC codes constructed by using the domination mappings? They are incomparable with the code constructions in Sections III and V as the latter admit a different set of parameters. However, the former can be easily compared with the codes obtained in Proposition 6.

In fact, the construction given by Corollary 25 can be viewed as a modification of the concatenation construction (see Proposition 6). In the following examples, we compare the sizes of the LPC codes obtained from Corollaries 26 and 27 with LPC codes from Proposition 6 with similar parameters. We remark that the LPC codes obtained from Corollaries 26 and 27 use three of the most simple (and less powerful) domination mappings: a $(2, 3, 1)$-domination mapping, a $(9, 15, 3)$-domination mapping and a $(12, 20, 4)$-domination mapping.

To this end, we describe the simple and effective construction of cooling codes given in [3]. This construction is based on spreads (or partial spreads). Loosely speaking, a partial $\tau$-spread of the vector space $\mathbb{F}_q^n$ is a collection of disjoint $\tau$-dimensional subspaces of $\mathbb{F}_q^n$. Formally, a collection $V_1, V_2, \ldots, V_M$ of $\tau$-dimensional subspaces of $\mathbb{F}_q^n$ is said to be a *partial $\tau$-spread of* $\mathbb{F}_q^n$ if

$$V_i \cap V_j = \{\mathbf{0}\} \text{ for all } i \neq j.$$

If the $\tau$-dimensional subspaces form a partition of $\mathbb{F}_q^n$ then the partial $\tau$-spread is called a $\tau$-spread. It is well known that such $\tau$-spreads exist if and only if $\tau$ divides $n$, in which case $M = (q^n-1)/(q^\tau-1) > q^{n-\tau}$. For the case where $\tau$ does not divide $n$, *partial $\tau$-spreads* with $M \geq q^{n-\tau}$ have been constructed in [9, Theorem 11].

*Theorem 28 ( [3]):* Let $V_1, V_2, \ldots, V_M$ be a partial $(t+1)$-spread of $\mathbb{F}_2^n$, and define the code $\mathbb{C} = \{V_1^*, V_2^*, \ldots, V_M^*\}$, where $V_i^* = V_i \setminus \{\mathbf{0}\}$ for all $i$. Then $\mathbb{C}$ is an $(n, t)$-cooling code of size $M \geq 2^{n-t-1}$ and has an encoding and decoding scheme that runs in $O(n^3)$ time.

First, we consider the family of LPC codes obtained from Corollary 26.

*Example 9:* For $w \geq 1$ and $t + 1 \leq w$, Theorem 28 provides a $(2w, t)$-cooling code $\mathbb{D}$ of size $2^{2w-t-1}$. Applying Corollary 26 to $\mathbb{D}$, we obtain a $(3w, t, w)$-LPC code $\mathbb{C}$ of size $2^{2w-t-1}$.

Next, we form a comparable code using Proposition 6 and we have the following choice of parameters.

(i) Choose $w' = 1$, $s = 3$, and $m = w$ in Proposition 6. As a consequence, we take $q = 4$, and hence obtain a $(3w, t, w)$-LPC code $\mathbb{C}_1$ of size $2^{2w-2t-1}$ for $t \leq \min\{3, (w/2) - 1\}$. Clearly, the size of $\mathbb{C}_1$ is much smaller than that of $\mathbb{C}$. Furthermore, the range of $t$ is much more restricted as compared to $t \leq w - 1$ for the LPC code $\mathbb{C}$ obtained from the domination mapping.

(ii) A different choice for Proposition 6 is $w' = 3$, $s = 9$, and $m = w/3$. As a consequence, we take $q = 128$, and hence obtain a $(3w, t, w)$-LPC code $\mathbb{C}_2$ of size $2^{7w/3-7t-7}$ for any $t \leq \min\{8, (w/6) - 1\}$. As before, the range of $t$ is much more restricted as compared to $\mathbb{C}$. Also, the size of $\mathbb{C}$ is larger than $\mathbb{C}_2$ for $t \geq (w/18) - 1$.

We continue our discussion with the family of LPC codes obtained from Corollary 27.

*Example 10:* For $w \geq 6$ and $t + 1 \leq 3w/2$, Theorem 28 provides a $(3w, t)$-cooling code $\mathbb{D}$ of size $2^{3w-t-1}$. Applying Corollary 27 to $\mathbb{D}$, we obtain a $(5w, t, w)$-LPC code $\mathbb{C}$ of size $2^{3w-t-1}$.

Next, we form a comparable code using Proposition 6 and we have the following choice of parameters.

(i) Choose $w' = 3$, $s = 15$, and $m = w/3$ in Proposition 6. As a consequence, we take $q = 2^9$, and hence obtain a $(5w, t, w)$-LPC code $\mathbb{C}_1$ of size $2^{3w-9t-9}$ for $t \leq \min\{14, (w/6) - 1\}$. Clearly, the size of $\mathbb{C}_1$ is much smaller than that of $\mathbb{C}$. As before, the range of $t$ is much more restricted.

(ii) Choose $w' = 4$, $s = 20$, and $m = w/4$ in Proposition 6. As a consequence, we take $q = 2^{12}$, and hence obtain a $(5w, t, w)$-LPC code $\mathbb{C}_2$ of size $2^{3w-12t-12}$ for any $t \leq \min\{19, (w/8)-1\}$. Clearly, the size of $\mathbb{C}_2$ is much smaller than that of $\mathbb{C}$. As before, the range of $t$ is much more restricted.

It should be noted that $q$ can be sometimes slightly larger than the one given in the examples. This will not make much difference in the comparison, but the computation in a large field size of odd characteristic is more cumbersome.

We conclude that the codes obtained via domination mapping in most cases have larger size and better capabilities than the best codes obtained by previous known constructions.

## VII. Conclusion

In this work, we studied constructions and efficient encoding and decoding of LPC cods. Such codes can be used to control simultaneously both the peak temperature and the average power consumption of on-chip buses. We first proposed a construction for LPC codes which takes a linear erasure code as input. Using this construction, we obtained a class of LPC codes whose sizes can asymptotically attain the upper bound $O(n^{w-1})$ when $w$ is fixed, as well as a class of LPC codes which is able to correct transmission errors. Efficient encoding and decoding schemes for these two classes of LPC codes are also presented. Then we provided a recursive construction for a special type of LPC codes, i.e., CPC codes. This recursive construction can produce CPC codes of high cooling capability, $t \geq n/w$. Finally, we proposed a method which uses a domination mapping to transform cooling codes into LPC codes, while preserving the efficiency of cooling codes. Compared with the best codes obtained by previous known constructions, the codes obtained by this new method have larger size and better capabilities in most cases.

## References


[1] T. Beth, "Algebraische Auflösungsalgorithmen für einige unendliche familien von 3–designs," *Le Matematiche*, vol. 29, pp. 105–135, Jan. 1974.

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

[3] Y. M. Chee, T. Etzion, H. M. Kiah, and A. Vardy, "Cooling codes: Thermal-management coding for high-performance interconnects," *IEEE Trans. Inf. Theory*, vol. 64, no. 4, pp. 3062–3085, Apr. 2018.

[4] Y. M. Chee, H. M. Kiah, A. Vard, and E. Yaakobi, "Explicit constructions of finite-length WOM codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2017, pp. 2860–2864.

[5] Y. Meng Chee, T. Etzion, H. Mao Kiah, and A. Vardy, "Domination mappings into the Hamming ball: Existence, constructions, and algorithms," 2018, *arXiv:1807.10954*. [Online]. Available: http://arxiv.org/abs/1807.10954

[6] D. de Caen, "Extension of a theorem of Moon and Moser on complete subgraphs," *ARS Combinatoria*, vol. 16, pp. 5–10, Feb. 1983.

[7] N. Deo and P. Micikevicius, "On one-factorization of complete 3-uniform hypergraphs," *Congressus Numerantium*, vol. 158, pp. 153–161, May 2002.

[8] T. Etzion and A. Hartman, "Towards a large set of steiner quadruple systems," *SIAM J. Discrete Math.*, vol. 4, no. 2, pp. 182–195, May 1991.

[9] T. Etzion and A. Vardy, "Error-correcting codes in projective space," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 1165–1173, Feb. 2011.

[10] T. Etzion, V. Wei, and Z. Zhang, "Bounds on the sizes of constant weight covering codes," *Des., Codes Cryptogr.*, vol. 5, no. 3, pp. 217–239, May 1995.

[11] M. Grassl. *Bounds on the Minimum Distance of Linear Codes and Quantum Codes*. Accessed: Mar. 8, 2020. [Online]. Available: http://www.codetables.de

[12] P. Keevash, "Hypergraph Turán problems," in *Surveys in Combinatorics 2011* (London Mathematical Society Lecture Note Series), vol. 392, R. Chapman, Ed. Cambridge, U.K.: Cambridge Univ. Press, 2011, pp. 83–139.

[13] J. MacWilliams, and N. J. A. Sloane, *The Theory of Error Correcting Codes*. Amsterdam, The Netherlands: North Holland, 1978.

[14] R. M. Roth and G. Ruckenstein, "Efficient decoding of reed-solomon codes beyond half the minimum distance," *IEEE Trans. Inf. Theory*, vol. 46, no. 1, pp. 246–257, Jan. 2000.

[15] A. Sidorenko, "What we know and what we do not know about Turán numbers," *Graphs Combinatorics*, vol. 11, pp. 179–199, Jun. 1995.

[16] P. P. Sotiriadis and A. P. Chandrakasan, "A bus energy model for deep submicron technology," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 10, no. 3, pp. 341–350, Jun. 2002.

[17] P. P. Sotiriadis and A. P. Chandrakasan, "Bus energy reduction by transition pattern coding using a detailed deep submicrometer bus model," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 50, no. 10, pp. 1280–1294, Oct. 2003.

[18] C. L. M. van Pul and T. Etzion, "New lower bounds for constant weight codes," *IEEE Trans. Inf. Theory*, vol. 35, no. 6, pp. 1324–1329, Nov. 1989.

[19] J. H. van Lint and N. R. M. Wilson, *A Course in Combinatorics*. Cambridge, U.K.: Cambridge Univ. Press, 1992.

[20] A. Vardy, "The intractability of computing the minimum distance of a code," *IEEE Trans. Inf. Theory*, vol. 43, no. 6, pp. 1757–1766, Nov. 1997.

[21] F. Wang, Y. Xie, N. Vijaykrishnan, and M. J. Irwin, "On-chip bus thermal analysis and optimization," in *Proc. Des. Autom. Test Eur. Conf.*, May 2006, pp. 1–6.

[22] L. R. Welch and E. R. Berlekamp, "Error correction for algebraic block codes," U.S. Patent 4 633 470, Dec. 30, 1986



**Yeow Meng Chee** received the B.Math., M.Math., and Ph.D. degrees in computer science from the University of Waterloo in 1988, 1989, and 1996, respectively. He has held senior positions in public service, including the Head of Security (information infrastructure) and the Assistant Director of Internationalization at the National Computer Board, the Deputy Director of Strategic Programs with the Infocomm Development Authority (IDA), and the Program Director of Interactive Digital Media Research and Development with the Media Development Authority. He deployed South East Asia's First Certification Authority Netrust in 1997, and also founded the Singapore Computer Emergency Response Team (SingCERT). He was the Head of the Division of Mathematical Sciences, Nanyang Technological University, from 2008 to 2010, the Chair of the School of Physical and Mathematical Sciences, Nanyang Technological University, from 2011 to 2017, and the Interim Dean of the College of Science from 2018 to 2019, Nanyang Technological University. He is currently a Professor of industrial systems engineering and management and an Associate Vice President of innovation and enterprise with the National University of Singapore (NUS). His research interest lies in the interplay between combinatorics and computer science, especially coding theory, extremal set systems, and their applications.

Dr. Chee is a Fellow and a Council Member of the Institute of Combinatorics and its Applications. He has represented Singapore in various international forums, including member of the APEC Electronic Commerce Task Force in 1998, a member of the ASEAN Coordinating Committee on Electronic Commerce in 1998, a member of the Working Group on ASEAN Information Infrastructure in 1999, a Secretariat of the Canada–Singapore IT Joint Council in 1998, a Co-Chair of the APEC TEL Public Key Interoperability Expert Group in 1999, a Secretariat of the Australia—Singapore ICT Joint Council in 1999, and a member of APEC Project DARE (Data Analytics Raising Employment) Advisory Board in 2017.

**Tuvi Etzion** (Fellow, IEEE) 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. Since 1984, he has been holding a position in the Department of Computer Science, Technion—Israel Institute of Technology, where he currently holds the Bernard Elkin Chair in computer science. From 1985 to 1987, he was a Visiting Research Professor with the Department of Electrical Engineering—Systems, University of Southern California, Los Angeles. From summers 1990 to 1991, he was visiting Bellcore in Morristown, NJ, USA. From 1994 to 1996, he was a Visiting Research Fellow with the Computer Science Department, Royal Holloway University of London, Egham, U.K. He also had several visits to the Coordinated Science Laboratory, University of Illinois in Urbana–Champaign, from 1995 to 1998, two visits to HP Bristol in summers of 1996 and 2000, a few visits to the Department of Electrical Engineering, University of California at San Diego, from 2000 to 2017, several visits to the Mathematics Department, Royal Holloway University, London, from 2007 to 2017, a few visits to the School of Physical and Mathematical Science (SPMS), Nanyang Technological University, and also to the Department of Industrial Systems Engineering and Management, National University of Singapore, Singapore, from 2016 to 2019, and a few visits to Jiaotong University, Beijing, from 2017 to 2019. His research interests include applications of discrete mathematics to problems in computer science and information theory, coding theory, network coding, and combinatorial designs.

Dr. Etzion was an Associate Editor for Coding Theory for the IEEE TRANSACTIONS ON INFORMATION THEORY from 2006 to 2009. From 2004 to 2009, he was an Editor of the *Journal of Combinatorial Designs*. Since 2011, he has been an Editor of *Designs, Codes, and Cryptography*, and since 2013, he has also been an Editor of *Advances of Mathematics in Communications*.

**Han Mao Kiah** (Member, IEEE) received the Ph.D. degree in mathematics from Nanyang Technological University (NTU), Singapore, in 2014. From 2014 to 2015, he was a Post-Doctoral Research Associate with the Coordinated Science Laboratory, University of Illinois at Urbana–Champaign. From 2015 to 2018, he was a Lecturer with the School of Physical and Mathematical Sciences (SPMS), NTU, Singapore. He is currently an Assistant Professor with SPMS, NTU, Singapore. His research interests include DNA-based data storage, coding theory, enumerative combinatorics, and combinatorial design theory.

**Alexander Vardy** (Fellow, IEEE) was born in Moscow, Russia, in 1963. He received the B.Sc. degree *(summa cum laude)* from the Technion—Israel Institute of Technology, Israel, in 1985, and the Ph.D. degree from Tel-Aviv University, Israel, in 1991. From 1985 to 1990, he was with the Israeli Air Force, where he worked on electronic counter measures systems and algorithms. From 1992 to 1993, he was a Visiting Scientist with the IBM Almaden Research Center, San Jose, CA, USA. From 1993 to 1998, he was with the University of Illinois at Urbana–Champaign, first as an Assistant Professor then as an Associate Professor. Since 1998, he has been with the University of California at San Diego (UCSD), where he is currently the Jack Keil Wolf Endowed Chair Professor with the Department of Electrical and Computer Engineering and the Department of Computer Science. While on sabbatical from UCSD, he has held long-term visiting appointments with CNRS, France, the EPFL, Switzerland, the Technion—Israel Institute of Technology, and Nanyang Technological University, Singapore. His research interests include error-correcting codes, algebraic and iterative decoding algorithms, lattices and sphere packings, coding for storage systems, cryptography and computational complexity theory, and fun math problems.

He has been a member of the Board of Governors of the IEEE Information Theory Society from 1998 to 2006 and from 2011 to 2017. He received an IBM Invention Achievement Award in 1993 and NSF Research Initiation and CAREER Awards in 1994 and 1995. In 1996, he was appointed as a Fellow in the Center for Advanced Study, University of Illinois, and received the Xerox Award for Faculty Research. In 1996, he became a Fellow of the David and Lucile Packard Foundation. He received the IEEE Information Theory Society Paper Award (jointly with Ralf Koetter) in 2004. In 2005, he received the Fulbright Senior Scholar Fellowship, and the Best Paper Award at the IEEE Symposium on Foundations of Computer Science (FOCS). In 2017, his work on polar codes was recognized by the IEEE Communications and Information Theory Societies Joint Paper Award. From 1995 to 1998, he was an Associate Editor for *Coding Theory* and from 1998 to 2001, he was the Editor-in-Chief of the IEEE TRANSACTIONS ON INFORMATION THEORY. From 2003 to 2009, he was an Editor for the *SIAM Journal on Discrete Mathematics*. He is currently serving on the Executive Editorial Board for the IEEE TRANSACTIONS ON INFORMATION THEORY.

**Hengjia Wei** received the Ph.D. degree in applied mathematics from Zhejiang University, Hangzhou, Zhejiang, China, in 2014. He was a Post-Doctoral Fellow with Capital Normal University, Beijing, China, from 2014 to 2016, and a Research Fellow with the School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore, from 2016 to 2019. He is currently a Post-Doctoral Fellow with the Department of Electrical and Computer Engineering, Ben-Gurion University of the Negev, Israel. His research interests include combinatorial design theory, and coding theory and their intersections.

Dr. Wei received the 2017 Kirkman Medal from the Institute of Combinatorics and its Applications.