UDC 621.3.037 **doi:** https://doi.org/10.20998/2522-9052.2024.4.06

Vladyslav Yareshchenko[1], Viktor Kosenko[1, 2]

[1] National University «Yuri Kodratyuk» Poltava Polytechnic, Poltava, Ukraine
[2] Kharkiv National University of Radio Electronics, Kharkiv, Ukraine

UNIT-DISTANCE CODES ENUMERATION

Abstract. The article discusses the current problem of reducing power dissipation in global communication lines while maintaining high performance. With the increasing complexity of system on chip, power consumption has become a major issue in system design. The main source of dynamic power dissipation in digital circuits is buses. Bus switching activity accounts for a significant portion of the total power dissipation. One effective method for reducing switching activity during device-to-device or on-chip communication is the use of low-power encoding techniques. Encoding and decoding methods have been studied to reduce the number of switching on buses. **The purpose of the article** is to develop a method for constructing a set of unit distance codes, determining the types of code transformations, and criteria for assessing the effectiveness of codes. **Research results.** A method for constructively enumerating unit distance codes has been developed, based on an invariant approach and the construction of a system of various representatives. Estimates of their number were obtained, characteristics were determined, and catalogs of typical representatives were generated. **Conclusion.** Application of the developed method will allow us to analyze and select codes with the best properties and, as a result, obtain better results in terms of network delays, energy costs and other design limitations for computer systems.

Keywords: networks on a chip; interconnects; power dissipation; Gray codes; unit distance codes; coding; switching activity.

Introduction

Problem statement. Due to its scalability and high performance, Network-on-Chip (NoC) technology has become a popular choice for the on-chip communication architecture of modern System-on-Chip (SoC) devices. Due to the increasing complexity of SoCs, power consumption has become a major concern in NoC design [1]. Bus switching activity is responsible for a significant portion of the total power dissipation. Power dissipation in digital systems can be static and dynamic. Leakage current in transistors causes static power dissipation, while switching power and short circuit power cause dynamic power dissipation. The main source of dynamic power dissipation in digital circuits is the buses, which are typically loaded with large input and output node capacitances as well as interconnect capacitances. These capacitances grow as technology advances. As SoCs become more complex, the number of buses also increases [2].

Therefore, the problem of reducing power dissipation in global interconnect lines while maintaining high performance is relevant.

Analysis of recent studies and publications. There are a number of challenges in developing data coding methods. Given the different properties of instruction, data, and multiplexed buses, they require different methods. Since additional pins are costly, coding methods must avoid or minimize the number of redundant bits [3].

There are many coding and decoding methods to reduce the number of switching on a data bus [4]. The most popular methods are Bus invert transition signaling and Bus Inversion coding [5]. A second bus line is required to use these methods. The Bus Inversion coding method for low-power I/O was created to minimize switching operations on the data bus. This approach is versatile, but it works best with buses. It is advantageous because buses have large capacity and consequently dissipate a lot of energy [6].

To optimize the bus energy consumption at the system level, a coding method with partial bus inversion has been developed [7]. The partial bus inversion scheme selects a subset of bus lines for bus coding to reduce the total number of bus transitions. This reduces the total number of bus transitions compared to the unencoded patterns.

Flip-N-Write method [8] reduces the number of bit flips by flipping the data. The data is divided into 8 to 32-bit blocks. Each block has a tag bit indicating whether the bits should be inverted or not. Other encoding techniques map the data bits into a set of vectors and select the vector that has the minimum number of bit turns.

Schemes based on the concept of code concatenation [9] are developed for communication applications. Code concatenation is a method that combines two codes in which the codewords of the inner code are the characters in the alphabet by which the outer code is determined [10].

In [11], two data coding methods are proposed to reduce dynamic capacity and improve reliability: an OEFNSC method that considers the overall self-switching and coupled switching activity and an OEFNSC-SEG method with segmentation to reduce switching activity through its own capacity and link capacity. The results confirmed the effectiveness of the proposed data coding schemes in terms of energy efficiency, delay efficiency and energy-delay product.

In the study [12], a dynamic sector coding method is proposed to reduce the bus switching activity. In this method, the source word space is divided into a number of sectors with a unique identifier. The dynamic sector (DS) encoder reduces the number of transitions by 20% more than the binary encoder.

In [5], an adaptive coding scheme called adaptive word reordering is proposed to reduce the inter-chip interconnect power, which effectively reduces the number of signal transitions, resulting in significant power reduction. A scheme is implemented that utilizes the time domain to represent complex bit transition computations as delays and thus limits the excess power due to coding.

Lee E. [13] and Barasch L. [14] investigated a promising class of codes, called generalized (alternative) Gray codes, which have both the reflection property and the unit distance property.

By using codes with the unit distance property, the SoC designer has more choices than when using Gray codes alone. Unfortunately, this type of codes is poorly understood and there are no methods for their construction in the literature [15–17].

The aim of the article is to develop a method of enumeration of unit distance codes, build a system of type representatives and analyze their characteristics.

Presentation of the main material of the study

It is difficult to consider the whole set of objects when solving various combinatorial problems, so a promising direction is to divide them into equivalence classes with respect to a given group of transformations and to study the properties of class representatives [18, 19]. In general, equivalence is a binary relation on a set possessing the properties of reflexivity, symmetry and transitivity. Any two classes of the same equivalence either do not overlap or coincide, i.e., any equivalence defines a partition of a set. Conversely, any partitioning of a set into non-overlapping classes generates an equivalence.

The main goal of classification is to find such a partitioning into classes, in which, on the one hand, equivalent transformations would be easy to implement, and on the other hand, the number of variants would not be very large [20, 21]. As a result of classification, a set of objects is divided into pairwise non-intersecting classes.

The concept of class was introduced by J. Neumann. According to his definition, X is a class if X is a set of all objects possessing some property (invariant) that remains unchanged after applying operations or transformations of a certain type to the objects. More generally, an invariant with respect to an equivalence relation is a property that is constant in every equivalence class [22].

Invariants reflect the invariant most fundamental properties of the studied objects and phenomena. When solving many problems, it is necessary to define a set of invariants, to establish and describe the relations between them, and to construct a complete set of invariants. The study of invariants is directly related to the problems of object classification, since the goal of any mathematical classification is to construct some complete system of invariants separating any two non-equivalent objects from the considered set [23]. Currently, there are different approaches to the definition of an invariant depending on whether the group of transformations is defined explicitly or not [22].

1. The transformation group G is not explicitly introduced, but the equivalence relation η satisfying the requirements of reflexivity, symmetry and transitivity is introduced. As a result, the set M is partitioned into subsets equivalent by η.

2. Transformation group $G = \{g_1, g_2, \ldots, g_d\}$ is defined on the set X. As a result of any transformation of $g_i \in G$ by an element $x \in X$, an element $x^* \in X$ will be obtained.

3. A generalized definition of an invariant is given in [23]: "An invariant is a mapping φ of a considered set M of mathematical objects, equipped with a fixed equivalence relation η, into another set N of mathematical objects, constant on the *-equivalence classes of M by η. If X is an object of M, then $\varphi(M)$ is said to be an invariant of the object X". The mapping φ of some set X into itself is called a transformation.

In essence, the goal of any classification is to construct some complete system of invariants, i.e., such a system that separates any two non-equivalent objects from the considered set.

Since all objects belonging to the same equivalence class pass into each other as a result of a given group of transformations, it is sufficient to define a type representative to describe the equivalence class, as which any element belonging to the equivalence class under consideration can be chosen. For unambiguous description of type representatives it is expedient to define the simplest form to which the investigated mathematical object can be brought by means of the considered group of transformations, i.e. to construct canonical forms.

The canonical form of a mathematical object is a standard way of representing this object in the form of a mathematical expression that allows to identify it in a unique way [24, 25]. Canonical forms are commonly used to make working with equivalence classes more efficient.

More generally, for a class of objects for which an equivalence relation is defined, the canonical form consists of selecting a particular object in each class (a type representative) [26, 27]. The representative is chosen unambiguously among the objects belonging to the class. In some applications, the representative is unambiguously chosen using some deterministic algorithm.

The set of typical representatives of equivalence classes forms a system of different representatives.

Thus, the approach under consideration is reduced to solving combinatorial problems related to determining the existence of a system of different representatives for a family of sets, and determining the number of systems of different representatives satisfying different criteria. The criterion for the existence of a system of different representatives is given by Hall's theorem.

The general approach to determining a type representative consists of the following steps.

1. Selection of any element representative of the equivalence class.

2. Execution of the set of given transformations of the object, i.e. generation of all elements of the equivalence class under consideration.

3. Determination of the canonical form for the elements obtained as a result of each transformation. 4.

4. Determination of a typical representative - an element having a canonical form of a given kind.

If the canonical form is a set of elements, we will choose the minimal canonical form. In this case, we assume that the set $\Lambda = \{\lambda_1, \dots, \lambda_i, \dots, \lambda_\phi\}$ is smaller than the set $\Psi = \{\psi_1, \dots, \psi_i, \dots, \psi_\phi\}$ if there exists a value i such that $\lambda_t = \psi_t$ for $t = 1, \dots, i-1$ and $\lambda_i \le \psi_i$.

Let us consider the application of the method described above to the classification of binary codes.

A code is a bijective (mutually unambiguous) mapping of a finite ordered set of symbols belonging to some finite alphabet Y to another, not necessarily ordered, as a rule more extensive set of symbols X for coding the transmission, storage or transformation of information [28, 29].

For codes the bijective function has the form

$$f : Y \to \mathrm{X},$$

where Y is a finite ordered set of symbols, $Y = \{y_1, y_2, \dots, y_k\}$, X is the set of codewords obtained as a result of the mapping, $X = \{X^1, X^2, \dots, X^k\}$, k is the number of code words, n is the number of digits of the binary code.

Bijective function

$$f : X \leftrightarrow Y$$

has the following features:

– translates different elements of set X into different elements of set Y (injectivity):

$$\forall X^1 \in X,\ \forall X^2 \in X,\ X^1 \ne X^2 \Rightarrow f(X^1) \ne \mathrm{f}(X^2), \quad (1)$$

– any element from Y has its prototype (surjectivity):

$$\forall y \in Y,\ \exists X^* \in X,\ f(X^*) = y. \quad (2)$$

The code word X^i consists of n characters (number of bits):

$$X^i = \{x_{i,1}, \dots, x_{i,n}\};\ x_{i,j} \in \{0,1\},\ i = 1, \dots, k;\ j = 1, \dots, n.$$

The number of changes in the values of the i-th variable in the column is denoted by $\mathrm{h_i}$ and is defined as follows:

$$h_i = \sum_{j=2}^{k} (x_{i,j-1} \oplus x_{i,j}),\ i = 1, \dots, n, \quad (3)$$

Table 1 shows a tabular representation of the code.

Table 1 – **Set of code words and their characteristics**

Y	***X***				
	x_1	…	x_j	…	x_n
y_1	$x_{1,1}$	…	$x_{1,j}$	…	$x_{1,n}$
y_2	$x_{2,1}$	…	$x_{2,j}$	…	$x_{2,n}$
…					
y_i	$x_{i,1}$	…	$x_{i,j}$	…	$x_{i,n}$
…					
y_k	$x_{k,1}$	…	$x_{k,j}$	…	$x_{k,n}$
H	h_1	…	h_j	…	h_n

The subject of this study is codes with unit distance, in which codewords have the following properties:

– two neighboring words differ only in one digit, i.e.

$$\rho(X^i, X^{i+1}) = 1, \quad i = 1, \dots, k-1, \quad (4)$$

where ρ - Hemming distance between codewords X^i and X^{i+1},

– the total number of changes of bit values in codewords is equal to

$$\sum_{j=1}^{k-1} \rho(X^j, X^{j+1}) = k-1. \quad (5)$$

The code in which $\rho(X^1, X^k) = 1$ is called cyclic.

The balance of the code C is defined as follows:

$$C = \sum_{i=1}^{n} \left| h_i - \sum_{j=1}^{n} h_j \Big/ n \right|. \quad (6)$$

The canonical form of a code $W(X)$ is called its representation in the form:

$$W(X) = x_{1,1}, \dots, x_{1,n}, \dots, x_{i,1}, \dots, x_{i,n}, \dots, x_{k,1}, \dots, x_{k,n}.$$

For compactness, the binary form of representation $W(X)$ can be transformed into hexadecimal form.

Among the set of possible transformations of binary matrices, the transformations that preserve the property of unit distance between neighboring binary words in the code are highlighted. These are column permutation (P transformations) and column inversion (N transformations). The code X obtained as a result of a group of transformations τ is denoted as follows $X(\tau)$.

Definition 1. Codes X^1 and X^2 are called P – equivalent if $X^1 = X^2(P)$.

Definition 2. Codes X^1 and X^2 are called N – equivalent if $X^1 = X^2(N)$.

Definition 3. Codes X^1 and X^2 are called PN – equivalent if $X^1 = X^2(PN)$.

The number of transformations is determined by the following expressions:

$$L_P = n!,\ L_N = 2^n. \quad (7)$$

P transformations will be described as a set of permutations of columns of the source code

$$P = \{\pi_1, \dots, \pi_n\},$$

and N transformations in the form of a set

$$N = \{\nu_1, \dots, \nu_n\},$$

where $\nu_I = 1$, if the inversion of the i-th column is performed and $\nu_i = 0$ otherwise.

Table 2 shows the decimal equivalents of the binary codes (D_B), the binary position code (X^B), the Gray code X^c and examples of PN transformations for the Gray code: the code $X^\Gamma(P)$ for $P = \{2,3,4,1\}$, the code $X^\Gamma(N)$ for $N = \{1,0,1,0\}$.

For each code, the number of changes in the values of each digit $H = \{h_1, h_2, h_3, h_4\}$ and the code balance value C are given.

The canonical form for codes X^Γ, $X^\Gamma(P)$, $X^\Gamma(N)$ for the specified transformations has the form:

$W(X^c)$ = 01326754cdfeab98,

$W(X^\Gamma(P))$ = 0264cea89bfd5731,

$W(X^\Gamma(N))$ = ab98cdfe67540132.

Table 2 - **Code Conversion**

D_B	$X^{в}$				$X^{г}$,				$X^{г}(P)$				$X^{г}(N)$			
	x_1	x_2	x_3	x_4	x_1	x_2	x_3	x_4	x_2	x_3	x_4	x_1	$\overline{x_1}$	x_2	$\overline{x_3}$	x_4
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
1	0	0	0	1	0	0	0	1	0	0	1	0	1	0	1	1
2	0	0	1	0	0	0	1	1	0	1	1	0	1	0	0	1
3	0	0	1	1	0	0	1	0	0	1	0	0	1	0	0	0
4	0	1	0	0	0	1	1	0	1	1	0	0	1	1	0	0
5	0	1	0	1	0	1	1	1	1	1	1	0	1	1	0	1
6	0	1	1	0	0	1	0	1	1	0	1	0	1	1	1	1
7	0	1	1	1	0	1	0	0	1	0	0	0	1	1	1	0
8	1	0	0	0	1	1	0	0	1	0	0	1	0	1	1	0
9	1	0	0	1	1	1	0	1	1	0	1	1	0	1	1	1
10	1	0	1	0	1	1	1	1	1	1	1	1	0	1	0	1
11	1	0	1	1	1	1	1	0	1	1	0	1	0	1	0	0
12	1	1	0	0	1	0	1	0	0	1	0	1	0	0	0	0
13	1	1	0	1	1	0	1	1	0	1	1	1	0	0	0	1
14	1	1	1	0	1	0	0	1	0	0	1	1	0	0	1	1
15	1	1	1	1	1	0	0	0	0	0	0	1	0	0	1	0
H	1	3	7	15	1	2	4	8	2	4	8	1	1	2	4	8
C	4,5				2,25				2,25				2,25			

To determine the typical representative T(X) it is necessary to perform all possible code transformations, to determine the canonical form for each type of transformation and to choose the minimal one. Since in the minimum canonical form the first character will be "0", in order to reduce the number of performed transformations it is necessary to determine the code digits in which the first codeword has "1" and invert the corresponding columns.

An example of defining a typical representative for the code given in Table 3 is given below.

Table 3 – **Initial code**

x_1	x_2	x_3	x_4		x_1	x_2	x_3	x_4
1	1	0	1		0	1	0	0
0	1	0	1		1	1	0	0
0	0	0	1		1	0	0	0
0	0	0	0		1	0	0	1
0	0	1	0		1	0	1	1
0	0	1	1		1	0	1	0
0	1	1	1		1	1	1	0
0	1	1	0		1	1	1	1

Since the first code word has the form "1 1 0 1", we perform inversion for the first, second and fourth columns. The resulting code is shown in Table 4.

Table 4 – **Transformed code**

$\overline{x_1}$	$\overline{x_2}$	x_3	$\overline{x_4}$		$\overline{x_1}$	$\overline{x_2}$	x_3	$\overline{x_4}$
0	0	0	0		1	0	0	1
1	0	0	0		0	0	0	1
1	1	0	0		0	1	0	1
1	1	0	1		0	1	0	0
1	1	1	1		0	1	1	0
1	1	1	0		0	1	1	1
1	0	1	0		0	0	1	1
1	0	1	1		0	0	1	0

Canonical form for the transformed code

W_{1234} = 08cdfeab91546732.

Table 5 shows an example of defining a typical representative for a transformed code.

Table 5 – **Definitions of a typical representative for a transformed code**

№	*P*	*W*		W_{min}
1	1 2 3 4	08cdfeab91546732	→	08cdfeab91546732
2	1 2 4 3	08cefd9ba2645731		↓
3	1 3 2 4	08abfecd91326754	→	08abfecd91326754
4	1 3 4 2	089bfdcea2315764	→	089bfdcea2315764
5	1 4 2 3	08aefb9dc4623751		↓
6	1 4 3 2	089dfbaec4513762		↓
7	2 1 3 4	04cdfe675198ab32	→	04cdfe675198ab32
8	2 1 4 3	04cefd5762a89b31		↓
9	2 3 1 4	02abfe673198cd54	→	02abfe673198cd54
10	2 3 4 1	019bfd5732a8ce64	→	019bfd5732a8ce64
11	2 4 1 3	02aefb3764c89d51		↓
12	2 4 3 1	019dfb3754c8ae62		↓
13	3 1 2 4	0467fecd5132ab98		↓
14	3 1 4 2	0457fdce62319ba8		↓
15	3 2 1 4	0267feab3154cd98		↓
16	3 2 4 1	0157fd9b3264cea8	→	0157fd9b3264cea8
17	3 4 1 2	0237fbae64519dc8		↓
18	3 4 2 1	0137fb9d5462aec8	→	0137fb9d5462aec8
19	4 1 2 3	046ef75dc8a23b91		↓
20	4 1 3 2	045df76ec8913ba2		↓
21	4 2 1 3	026ef73ba8c45d91		↓
22	4 2 3 1	015df73b98c46ea2		↓
23	4 3 1 2	023bf76ea8915dc4		↓
24	4 3 2 1	013bf75d98a26ec4		↓
			W_{min} = 0137fb9d5462aec8	

The type representative for the code under consideration is denoted by $T(X)$ and is defined as follows:

$$T(X) = W_{min} = \text{0137fb9d5462aec8}.$$

All unit distance codes for $n = 3$ and $k = 8$ were investigated.

The analysis of the obtained results showed that the set of these codes can be divided into three equivalence classes. Tables 6 – 8 show the codes included in the equivalence classes E_1, E_2 and E_3.

It should be noted that codes belonging to the same class have one type representative (T) and one code structure $S(X) = \langle H(X),> \rangle$.

Table 6 - Codes belonging to the equivalence class E_1

i	$A_{1,i}$	***i***	$A_{1,i}$	***i***	$A_{1,i}$	***i***	$A_{1,i}$	***i***	$A_{1,i}$	***i***	$A_{1,i}$
1	15467320	9	64510237	17	46732015	25	76204513	33	01375462	41	57623104
2	54026731	10	10264573	18	04675132	26	76402315	34	73201546	42	15764023
3	20154673	11	26457310	19	26754013	27	73102645	35	75401326	43	04576231
4	54620137	12	64015732	20	67513204	28	37645102	36	75104623	44	45731026
5	51320467	13	32046751	21	67315402	29	13762045	37	23157640	45	45137620
6	31540267	14	40132675	22	02673154	30	02376451	38	51023764	46	20451376
7	32640157	15	10462375	23	62013754	31	23751046	39	57326401	47	40231576
8	62310457	16	46237510	24	13267540	32	37546201	40	01573264	48	31045762

Table 7 - **Codes belonging to the equivalence class E_2**

i	$A_{2,i}$	***i***	$A_{2,i}$	***i***	$A_{2,i}$	***i***	$A_{2,i}$	***i***	$A_{2,i}$	***i***	$A_{2,i}$
1	04513267	9	62045731	17	64023751	25	37620154	33	73154620	41	57310462
2	45102673	10	64573201	18	26401375	26	76231540	34	75462310	42	15732046
3	51046732	11	46201573	19	26731045	27	76451320	35	75132640	43	15402376
4	54673102	12	04623157	20	67320451	28	73264510	36	37510264	44	54013762
5	01546237	13	23104675	21	67540231	29	23764015	37	32015764	45	45762013
6	02315467	14	20467315	22	62375401	30	20137645	38	31576204	46	40157623
7	01326457	15	40267513	23	31026754	31	10237546	39	51376402	47	10457326
8	02645137	16	46751023	24	32675104	32	13754026	40	57640132	48	13204576

Table 8 – **Codes belonging to the equivalence class E_3**

i	$A_{3,i}$	***i***	$A_{3,i}$	***i***	$A_{3,i}$	***i***	$A_{3,i}$	***i***	$A_{3,i}$	***i***	$A_{3,i}$
1	01546732	9	62045137	17	20467513	25	76451023	33	13754620	41	15732640
2	54673201	10	26401573	18	46751320	26	76231045	34	73154026	42	57640231
3	15402673	11	02645731	19	67540132	27	73264015	35	75132046	43	45762310
4	51046237	12	64573102	20	67320154	28	37620451	36	75462013	44	10457623
5	54013267	13	13204675	21	26731540	29	23764510	37	02315764	45	40157326
6	32015467	14	46201375	22	62375104	30	10237645	38	31576402	46	04513762
7	31026457	15	04623751	23	01326754	31	20137546	39	51376204	47	45102376
8	64023157	16	40267315	24	32675401	32	37510462	40	57310264	48	23104576

Code number 23 in equivalence class T_3 is a Gray code.

The characteristics of equivalence classes for n = 3 and k = 8 are given in Table 9.

Table 9 – **Characteristics of equivalence classes for $n = 3$ и $k = 8$**

Equivalence class	$S(E)$	$C(E)$	$T(E)$
E_1	2, 2, 3	1, 3	01375462
E_2	1, 3, 3	2, 7	01326457
E_3	1, 2, 4	3, 3	01326754

An example of application of the obtained results is given below.

Based on the analysis of Table 8 and Table 9, the code number 33 in the equivalence class E_1 was chosen to realize the code converter with the best balance.

Fig. 1 shows the functional diagram of the developed code converter, which contains:

– three inputs 1, 2, 3,
– three outputs 4, 5, 6,
– AND element 7,
– OR element 8,
– two unequalization elements 9 and 10.

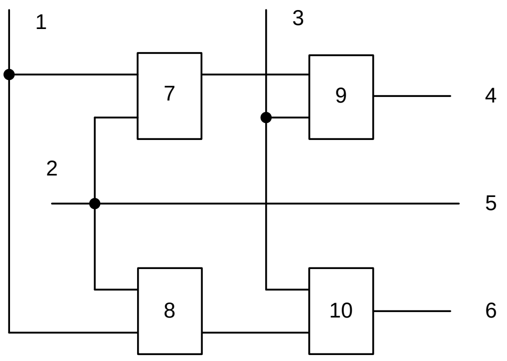

Fig. 1. Code Converter

The device works as follows. At the inputs of the device 1, 2, 3 input binary signals x_1, x_2, x_3 are given, and at the outputs 4, 5, 6 the output binary code y_1, y_2, y_3 is formed.

Table 10 shows the decimal equivalents of the binary codes (D_B), the position code (U_B), the binary equivalent of the Gray code $D_Г$, the binary Gray code $U_Г$, the binary equivalent of the code at the output of the code converter D_C, and the binary code at the output of the code converter U_C. For each code the number of changes of values of each digit $H = \{h_1, h_2, h_3\}$ and the code balance value C are specified.

The given code converter has a code balance $C = 1.3$, which is 2.5 times better than the Gray code converter.

Table 10 – **Binary codes and their characteristics**

D_B	U_B			$D_Г$	$U_Г$			D_C	U_C		
	x_1	x_2	x_3		y_1	y_2	y_3		y_1	y_2	y_3
0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	1	0	0	1	1	0	0	1
2	0	1	0	3	0	1	1	3	0	1	1
3	0	1	1	2	0	1	0	7	1	1	1
4	1	0	0	6	1	1	0	5	1	0	1
5	1	0	1	7	1	1	1	4	1	0	0
6	1	1	0	5	1	0	1	6	1	1	0
7	1	1	1	4	1	0	0	2	0	1	0
H	1	3	7		1	2	4		2	3	2
C	6.7				3.3				1.3		

The number of NP type variants (LSi) for $n = 4$ having the same structure is given in Fig. 2, and their balance score (Ci) is given in Fig. 3 (i is the number of the code structure, $i = 1, \ldots, 18$).

Characteristics of the investigated codes for $n = 3$ and $n = 4$ are given in Table 11.

Designations: N_{code} - total number of binary codes, N_{udc} - number of codes with unit distance, N_{tip} - number of NP types, N_{st} - number of code structures, C_{Gray} value of Gray code balance, C_{min} - minimum value of code balance, L_{Cmin} - number of NP types with minimum balance.

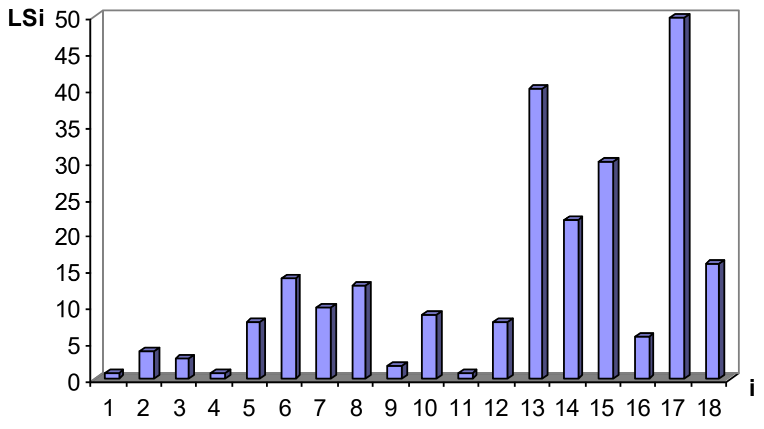

Fig. 2. Number of variants of NP types (LSi) for n = 4 having the same structure

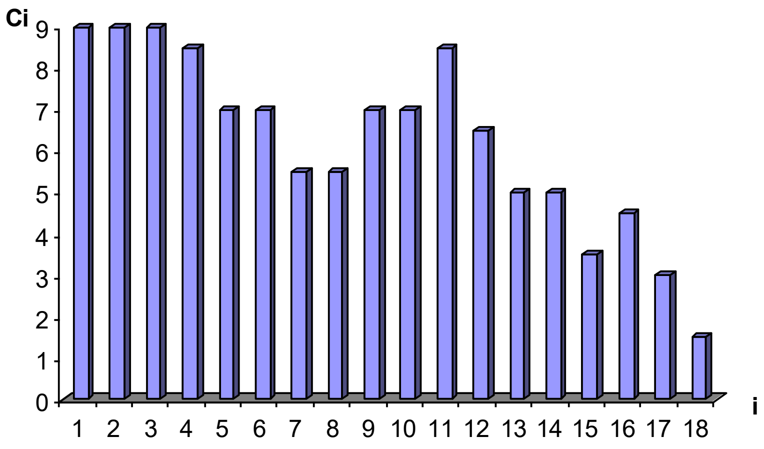

Fig. 3. Balanced structure options for n = 4

Table 11 – **Characteristics of the studied codes for n=3, k=8 and n=4, k=16**

n	N_{code}	N_{udc}	N_{tip}	N_{st}	C_{Gray}	C_{min}	C_{Gray} / C_{min}	L_{Cmin}
3	40320	144	3	3	3,3	1,3	2,5	1
4	2×10^{13}	91392	238	18	9	1,5	6	16

The given results show that the number of typical variants of codes with unit distance is much less than the total number of codes, which allows to choose the optimal coding without enumeration of variants. The efficiency of the method grows with the increase of the number of code digits. So, for $n = 3$ the improvement of balance of optimal codes makes 2,5 times and 6 times for $n = 4$.

Conclusions from this study and prospects for further research in this area

The urgent problem of reducing power dissipation in global interconnect lines while maintaining high performance is considered. It is shown that the main source of dynamic power dissipation in digital circuits are buses. Encoding and decoding methods for reducing the number of switching on the buses are investigated. Codes with the property of unit distance are investigated.

The method of constructive enumeration of unit distance codes is proposed, which allows to choose optimal coding without enumeration of variants. Estimates of the number of typical structures are obtained and catalogs of typical representatives are formed.

Application of the developed method will allow analyzing and selecting codes with better properties and obtaining better results in terms of network delays, power costs and other design constraints for computer systems.

Further researches in this direction: development of the method of construction of codes with specified code structures and construction of code converters.

References

1. Taha, T. B., Barzinjy, A. A., Hussain, F. H. S. and Nurtayeva, T. (2022), “Nanotechnology and computer science: Trends and advances”, *Memories-Materials, Devices, Circuits and Systems*, vol. 2, doi: https://doi.org/10.1016/j.memori.2022.100011
2. Yareshchenko, V. and Kosenko, V. (2023), “Coding to reduce the energy of data movement”, *Control, Navigation and Communication Systems*, vol. 1 (71), pp. 159–162, doi: https://doi.org/10.26906/SUNZ.2023.1.159.
3. Mehta, K. (20150, “A review on strategies and methodologies of dynamic power reduction on low power system design”, *Int. Journal of Computer Science & Communication*, vol. 7, no. 1, pp. 25–33, doi: https://doi.org/10.090592/USC.2016.004
4. Samanth, R., Nayak, S. G. and Nempu, P. B. (2023), “A Novel Multiply-Accumulator Unit Bus Encoding Architecture for Image Processing Applications”, *Iranian Journal of Electrical and Electronic Engineering*, vol. 19, no. 1, pp. 1–11, doi: https://doi.org/10.22068/IJEEE.19.1.2391
5. Maragkoudaki, E. and Pavlidis, V. (2020), “Energyeffic ient time-based adaptive encoding for off-chip communication”, *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 12, pp. 2551–2562, doi: https://doi:10.1109/TVLSI.2020.3018062
6. Mittal, S. and Nag, S. (2019), “A survey of encoding techniques for reducing data-movement energy”, *Journal of Systems Architecture*, vol. 97, pp. 373–396, doi: https://doi.org/10.1016/j.sysarc.2018.11.001
7. Annamalai, N. and Durairajan, C. (2021), “Linear codes from incidence matrices of unit graphs”, *Journal of Information and Optimization Sciences*, vol. 42, no. 8, pp. 1943–1950, doi: https://doi.org/10.1080/02522667.2021.1972617
8. Chang, D., Yan, S., Tan, W. and He, D. (2023), “An optimization scheme to improve the write performance of PCM”, *2023 IEEE 6th Information Technology, Networking, Electronic and Automation Control Conference*, ITNEC, IEEE, vol. 6, pp. 1081–1086, doi: https://doi.org/10.1109/ITNEC56291.2023.10082138
9. Dolecek, L. and Cassuto, Y. (2017), “Channel coding for nonvolatile memory technologies: Theoretical advances and practical considerations”, *Proc. of the IEEE*, vol. 105, no. 9, pp. 1705–1724, doi https://doi.org/10.1109/JPROC.2017.2694613
10. Chamberland, C., Jochym-O'Connor, T. and Laflamme, R. (2017), “Overhead analysis of universal concatenated quantum codes”, *Physical Review A*, vol. 95, no. 2, doi: https://doi.org/10.1103/PhysRevA.95.022313
11. Chennakesavulu, M., Prasad, T. J. and Sumalatha, V. (2022), “Data encoding techniques to improve the performance of system on chip”, *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 2, pp. 492–503, doi: https://doi.org/10.1016/j.jksuci.2018.12.003
12. Chintaiah, N. and Reddy, G. U. (2021), “Low-power sector-based transition reduction bus encoding technique in SOC interconnects”, *International Journal of Computer Aided Engineering and Technology*, vol. 15, no. 2-3, pp. 281–293, doi: https://doi.org/10.1504/IJCAET.2021.117138
13. Lee, D., O’Connor, M. and Chatterjee, N. (2018), “Reducing Data Transfer Energy by Exploiting Similarity within a Data Transaction”, *IEEE International Symposium on High Performance Computer Architecture*, HPCA, pp. 40–51, doi: https://doi.org/10.1109/HPCA.2018.00014
14. Barasch, L. S., Lakshmivarahan, S. and Dhall, S. K. (2020), “Generalized Gray codes and their properties”, *Mathematics for Large Scale Computing*, CRC Press, pp. 203–216, doi: https://doi.org/10.1201/9780429332760-9
15. Serkov, A., Trubchaninova, K. and Lazurenko, B. (2020), “Noise stability of mobile telecommunication systems”, *Control, Navigation and Communication Systems*, vol. 2 (60), pp. 169–172, doi: https://doi.org/10.26906/SUNZ.2020.2.169
16. Herter, F. and Rote, G. (2018), “Loopless Gray code enumeration and the Tower of Bucharest”, *Theoretical Computer Science*, vol. 748, pp. 40–54, doi: https://doi.org/10.1016/j.tcs.2017.11.017
17. Volk, M. and Lunichkin, O. (2022), “Self-healing computer systems”, *Control, Navigation and Communication Systems*, vol. 1 (67), pp. 48–51, doi: https://doi.org/10.26906/SUNZ.2022.1.048

18. Romanenkov, Y., Mukhin, V., Kosenko, V., Revenko, D., Lobach, O., Kosenko, N., Yakovleva, A. (2024), "Criterion for Ranking Interval Alternatives in a Decision-Making Task", International Journal of Modern Education and Computer Science, vol. 16(2), pp. 72–82, doi: https://doi.org/10.5815/ijmecs.2024.02.06
19. Ragulin, V., Owaid, S.R., Kuchuk, H., Gaman, O., Hurskyi, T. (2024), "Development of a method for increasing the efficiency of processing heterogeneous data using a metaheuristic algorithm", Eastern-European Journal of Enterprise Technologies, vol. 4(3(130)), pp. 21–28, doi: https://doi.org/10.15587/1729-4061.2024.309126
20. Filatov, V., Filatova, A., Povoroznyuk, A. and Omarov, S. (2024), "Image classifier for fast search in large databases", *Advanced Information Systems*, vol. 8, no. 2, pp. 12–19, doi: https://doi.org/10.20998/2522-9052.2024.2.02
21. Kuchuk, H., Mozhaiev, O., Kuchuk, N., Tiulieniev, S., Mozhaiev, M., Gnusov, Y., Tsuranov, M., Bykova, T., Klivets, S., and Kuleshov, A. (2024), "Devising a method for the virtual clustering of the Internet of Things edge environment", *Eastern-European Journal of Enterprise Technologies*, vol. 1, no. 9 (127), pp. 60–71, doi: https://doi.org/10.15587/1729-4061.2024.298431
22. Webster, B. (2017), "Knot invariants and higher representation theory", *American Mathematical Society*, vol. 250, no. 1191, doi: https://doi.org/10.48550/arXiv.1309.3796
23. Chang, S., Zhang, Y., Yu, M. and Jaakkola, T.S. (2020), "Invariant rationalization", *International Conference on Machine Learning*, PMLR, pp. 1448–1458, doi: https://doi.org/10.48550/arXiv.2003.09772
24. Petrovska, I., Kuchuk, H., Mozhaiev, M. (2022), "Features of the distribution of computing resources in cloud systems", *2022 IEEE 4th KhPI Week on Advanced Technology, KhPI Week 2022 - Conference Proceedings*, 03-07 October 2022, Code 183771, doi: https://doi.org/10.1109/KhPIWeek57572.2022.9916459
25. Bezkorovainyi, V., Kolesnyk, L., Gopejenko, V. and Kosenko, V. (2024), "The method of ranking effective project solutions in conditions of incomplete certainty", *Advanced Information Systems*, vol. 8, no. 2, pp. 27–38, doi: https://doi.org/10.20998/2522-9052.2024.2.04
26. Petrovska, I. and Kuchuk, H. (2023), "Adaptive resource allocation method for data processing and security in cloud environment", *Advanced Information Systems*, vol. *7*(3), pp. 67–73, doi: https://doi.org/10.20998/2522-9052.2023.3.10
27. Panchenko, S., Prykhodko, S., Kozelkov, S., Shtompel, M., Kosenko, V., Shefer, O. and Dunaievska, O. (2019), "Analysis of efficiency of the Bioinspired method for decoding algebraic convolutional codes", *Eastern-European Journal of Enterprise Technologies*, vol. 2(4-98), pp. 22–30, doi: https://doi.org/10.15587/1729-4061.2019.160753
28. Ding, C. (2019), *Designs from linear codes*, CRC Press, 392 p., doi: https://doi.org/10.1142/11101
29. Huffman, W.C., Kim J.L. and Sole, P. (2021), *Concise encyclopedia of coding theory*, CRC Press, 998 p., ISBN: 9781315147901, doi: https://doi.org/10.1201/9781315147901

Received (Надійшла) 19.06.2024
Accepted for publication (Прийнята до друку) 18.09.2024

ВІДОМОСТІ ПРО АВТОРІВ/ ABOUT THE AUTHORS

Ярещенко Владислав Валерійович – аспірант, Національний університет «Полтавська політехніка імені Юрія Кондратюка», Полтава, Україна;
Vladyslav Yareshchenko – Ph. D. student of National University «Yuri Kondratyuk» Poltava Polytechnic; Poltava, Ukraine; e-mail: vlad.yareschenko@gmail.com, ORCID Author ID: https://orcid.org/0000-0001-7682-0572.

Косенко Віктор Васильович – доктор технічних наук, професор, професор кафедри автоматики, електроніки та телекомунікацій, Національний університет «Полтавська політехніка імені Юрія Кондратюка», Полтава, професор кафедри комп'ютерно-інтегрованих технологій, автоматизації та робототехніки, Харківський національний університет радіоелектроніки, Харків, Україна;
Viktor Kosenko – Doctor of Sciences (Engineering), Professor, Department of Automation, Electronic and Telecommunication Department of National University «Yuri Kondratyuk» Poltava Polytechnic, Poltava, Professor of the Department of Computer-Integrated Technologies, Automation and Robotics, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine;
e-mail: kosvict@gmail.com; ORCID Author ID: https://orcid.org/0000-0002-4905-8508;
Scopus ID: https://www.scopus.com/authid/detail.uri?authorId=57190443921.

Перерахування кодів з одиничною відстанню

В. В. Ярещенко, В. В. Косенко

Анотація. У статті розглядається актуальна проблема зниження потужності, що розсіюється, в глобальних лініях зв'язку при збереженні високої продуктивності. Зі зростанням складності систем на кристалі споживання енергії стало серйозною проблемою під час їх розробки. Основним джерелом розсіювання динамічної потужності у цифрових схемах є шини. Комутаційна активність шин є причиною значної частки загальної потужності, що розсіюється. Одним із ефективних методів зниження комутаційної активності під час зв'язку між пристроями або зв'язку на кристалі є застосування методів кодування з низьким енергоспоживанням. Досліджено методи кодування та декодування, що дозволяють зменшити кількість перемикань на шинах. **Мета статті** полягає у розробці методу побудови безлічі кодів одиничної відстані, визначення видів кодових перетворень, критеріїв оцінки ефективності кодів. **Результати дослідження.** Розроблено метод конструктивного перерахування кодів одиничної відстані, заснований на інваріантному підході та побудові системи різних представників. Отримано оцінки їх кількості, визначено характеристики, сформовано каталоги типових представників. **Висновок.** Застосування розробленого методу дозволить аналізувати та вибирати коди з найкращими властивостями та в результаті отримувати кращі результати з погляду мережевих затримок, витрат на електроенергію та інших конструктивних обмежень для комп'ютерних систем.

Ключові слова: мережі на кристалі; масштабування технології; міжз'єднання; розсіювання потужності; коди Грея; коди одиничної відстані; кодування, комутаційна активність.