

V. Yareshchenko, V. Kosenko

National University "Yuri Kondratyuk Poltava Polytechnic", Poltava, Ukraine

## CODING TO REDUCE THE ENERGY OF DATA MOVEMENT

**Abstract.** The problem of reducing power dissipation in global interconnection lines while maintaining high performance is considered in the paper. High switching activity leads to significant communication losses due to communication capacitances between long lines. New byte-addressed non-volatile memory technologies, such as phase-change memory, enable systems with large persistent memory, improving reliability and potentially reducing power consumption. However, these technologies only support a limited number of write operations over the lifetime per cell, and consume most of their power when the state of a bit changes during a write. Low power coding techniques are required to reduce switching activity during device-to-device or on-chip communications. **The purpose of the article** is to develop a method for constructing a set of unit distance codes, analyzing their characteristics and choosing codes that satisfy the given properties. The address bus coding methods with the least switching activity are considered. To reduce dynamic energy losses in the address bus and minimize communication losses between closely spaced lines, Gray code is used, which has a number of disadvantages. **Conclusions.** The type of codes that have the same properties as Gray codes, i.e. unit distance codes, is determined. A method for constructing a set of unit distance codes, analyzing their characteristics, and choosing codes that satisfy the given properties has been developed. By using the entire set of codes, developers have more choices than using only Gray codes, and this leads to better results.

**Keywords:** Gray code, switching activity, hypergraph, non-volatile memory.

### Introduction

**Statement of the problem.** Computer systems are widely used in various fields of science and technology in the construction of control systems, regulation, transmission and processing of discrete information.

To ensure the required level of quality of their functioning and reliability different approaches are used: improvement of existing and organization of fundamentally new systems, their hardware and software, creation of algorithmic, hardware and software, control and diagnostic support.

Modern processors are becoming increasingly bottlenecked by the energy of moving data across different levels of the memory hierarchy and between different input-output devices, such as sensors and specialized gas pedals.

Reducing power dissipation in global interconnect lines while maintaining high performance is an ongoing challenge for scalable CMOS technology.

**Analysis of recent research and publications.** There are many approaches to reducing input-output power consumption. These approaches fall into two categories. The first consists of methods that optimize the memory hierarchy and data organization to eliminate input-output requirements in the first place. The second category consists of methods that reduce switching activity on buses [1, 2].

External input-output and associated buses are a major contributor to overall system power consumption [3, 4]. The input-output power consumption is in direct relation to the product of the switching activity present on the input-output by the average capacitive load of the switching elements. In [5, 6], it was shown that the capacitive load of the external input/output is several orders of magnitude greater than that of the internal switching nodes.

High switching activity leads to significant communication losses due to communication capacitances between long lines. The long lines of the

address bus are accessed very frequently. As a result of these factors, the instruction memory address bus is usually one of the most power-consuming components of an embedded system [7, 8].

New non-volatile byte addressable memory technologies, such as phase transition memory, enable systems with large persistent memory, improving reliability and potentially reducing power consumption. However, these technologies support only a limited number of write operations over the lifetime per cell and consume most of their power when the bit state changes during writes [9].

Consequently, low-power coding techniques are required to reduce switching activity during device-to-device or on-chip communication.

Various methods are used to encode the address bus with the lowest switching activity. Gray's codes are used to reduce dynamic energy [10, 11]. The use of Gray's coding ensures that only one bit is switched between consecutive addresses. Since access to instruction memory is often continuous, the use of Gray's codes not only reduces dynamic energy loss on the address bus, but also minimizes communication loss between closely spaced lines. However, Gray's code has low balance and a large number of bit switches.

**The aim of the article** is to develop a method for constructing a set of unit distance codes, analyzing their characteristics, and selecting codes that satisfy the given properties.

### Presentation of the main material of the study

Gray codes are known as unit distance codes (single-step or monostrophic). A unit distance code is an unweighted code that changes only one digit position when moving from one number to another in a number sequence [12, 13].

The graph model of Gray's codes is a hypercube graph. In graph theory, a hypercube graph  $Q_n$  is a regular graph with  $2^n$  vertices,  $2^{n-1}n$  edges and  $n$  edges converging to one vertex [14].

A regular (homogeneous) graph is a graph with degrees of all its vertices equal, i.e. each vertex has the same number of neighbors.

Below we consider the hypercube graph  $Q_n$  with labeled vertices.

A labeled vertex is additional information associated with a vertex, which allows to distinguish it from other labeled vertices.

For the considered problem numbering of vertices will start from number "0" and each vertex will correspond to n-digit binary code  $a_1a_2...a_n$ , where  $a_i$  is 0 or 1. Two vertices (or points) of cube  $Q_n$  are adjacent if their binary representations differ only in one position (one digit). Fig. 1 shows the graph of hypercube  $Q_3$ , binary codes of vertices and their notation.

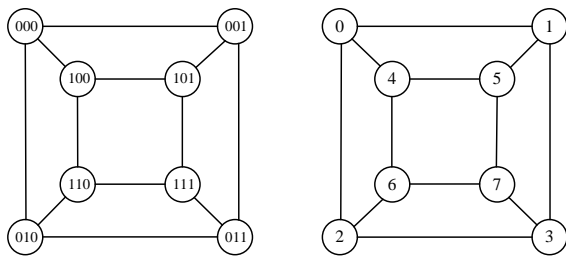


Fig. 1. Hypercube  $Q_3$  graph

A path in graph G is an alternating sequence of vertices and edges that begins and ends with a vertex, each edge of the sequence is incident to two vertices, one of which immediately precedes it, and the other immediately follows it. The specified path connects vertices  $v_0$  and  $v_k$ , and it can be denoted by  $v_0v_1v_2...v_k$ . A path in which all vertices are different is called simple. A path in which,  $v_0=v_k$ , is called closed (or cycle) and open otherwise. A closed path is called a simple cycle if all its n vertices are distinct and  $n \geq 3$  [15].

A simple path passing through each vertex of the graph exactly once is called a Hamiltonian path. A Hamiltonian path differs from a Hamiltonian cycle in that the start and ends of the path may not coincide, unlike a cycle. A Hamiltonian cycle is a Hamiltonian path. Any hypercube  $Q_n$  with  $n > 1$  has a Hamiltonian cycle passing through each vertex exactly once.

The Hamiltonianness of a hypercube is closely related to Gray's code theory. More precisely, there exists a bijective correspondence between the set of n-bit cyclic Gray codes and the set of Hamiltonian cycles in the hypercube  $Q_n$ . A similar property holds for acyclic n-bit Gray's codes and Hamiltonian paths [16].

In terms of graph theory, the problem of constructing a set of unit distance codes is reduced to constructing a set of Hamiltonian paths (cycles) in a hypercube.

To construct a set of Hamiltonian paths in graph  $Q_n$  an algorithm was developed, which was described using the following notations:

$U$  - path to be formed,

$L(U)$  - length of path to be formed,

$U(i)$  - number of vertex at the i-th place in the path  $U$ ,

$v \rightarrow U$  means writing the vertex  $v$  into the path  $U$ ,

$O(v)$  - neighborhood of vertex  $v$  - set of vertices adjacent to vertex  $v$ ,

$A(v) = \{v^{a_1}, \dots, v^{a_e}\}$  is the set of active vertices in the neighborhood of vertex  $v$ . A vertex is called active if it is not included in the formed path  $U$ ,

$e$  is the number of active vertices,

$B = \{B_1, \dots, B_s\}$  - set of promising initial paths (for further consideration),

$s$  - number of promising initial paths,

$d$  - current number of initial paths,

$B_i \rightarrow B$  means writing path  $B_i$  to the set of promising paths  $B$ ,

$R = \{U_1, \dots, U_q\}$  is the set of formed Hamiltonian paths,

$q$  is the number of formed Hamiltonian paths,

$U \rightarrow R$  means the entry of a new formed path,

$t$  is the current number of the vertex in question.

The algorithm for constructing Hamiltonian paths in graph  $Q_n$  consists of the following steps:

1.  $q = 0$ .

2.  $d = 0, s = 0$ .

3.  $d = d + 1$ .

4.  $U = B_d$ .

5.  $t = L(U)$ .

6. Determine the neighborhood of vertex  $O(v), v = U(t)$ .

7. Define the set of active vertices

$A(v) = \{v^{a_1}, \dots, v^{a_e}\}$ .

8. If  $e = 0$ , go to step 13.

9. Write the vertex  $v^a$  into the generated path:  $v^{a_1} \rightarrow U$ .

10. If  $e = 1$ , go to step 6.

11. Form new prospective initial paths:

$s = s + j, v^{a_j} \rightarrow B_s, B_s \rightarrow B, j = 1, \dots, e - 1$ .

12. We proceed to step 6.

13. If  $L(U) = n$ , then a new Hamiltonian path is formed. Write it down:  $q = q + 1; U \rightarrow R$ .

14. If  $d = s$ , then the end of the algorithm.

15. We choose the next promising initial path: go to step 3.

Table 1 shows an example of the algorithm's work, Table 2 shows the process of selecting vertices during the algorithm's work.

Table 1 – An example of algorithm work

d	U	t	O(v)	A(v)	$v^{a_1}$	s	$B_s$
0	0	0	124	124	1	1	02
						2	04
	01	1	35	35	3	3	015
	013	3	127	27	2	4	0137
	0132	2	036	6	6		
	01326	6	247	47	4	5	013267
	013264	4	056	5	5		
	0132645	5	147	7	7		
	01326457	7	-	-	-		
1	02	2	135	35	3	6	026
	023	3	127	17	1	7	0237
	0231	1	035	5	5		
	02315	5	147	47	4	8	023157
	023154	4	056	6	6		
	0231546	6	247	7	7		
	02315467	7	-	-	-		
2	04	4	056	56	5	9	046
	045	5	147	17	1	10	0457
	0451	1	035	3	3		
	04513	3	127	27	2	11	045137

3	045132	2	035	6	6		
	0451326	6	247	7	7		
	04513267	7	-	-	-		
	015	5	147	47	4	12	0157
	0154	4	056	6	6		
	01546	6	247	27	2	13	015467
	015462	2	036	3	3		
4	0137	7	356	56	5	14	01376
	01375	5	147	4	4		
	013754	4	056	6	6		
	0137546	6	247	2	2		
5	01375462	2	-	-	-		
	013267	7	356	5	5	-	
	0132675	5	147	4	4	-	
	01326754	4	-	-	-		
	...						

Table 2 – Selection of vertices during the work of the algorithm

Nv	Hamiltonian path number																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18		
1	0																			
2	1																			
3	3			5			3			6			5			6				
4	2		7		4		7		1		7		4		7		1		7	
5	6		5		6		3		5		6		5		3		3		6	
6	4	7	4	2	7	2	4	7	4	1	7	1	2	7	2	1	7	1	7	
7	5	5	6	3	3	6	6	6	5	3	3	5	6	6	3	5	5	5	3	
8	7	4	2	7	2	4	7	4	1	7	1	4	7	2	1	7	1	1	2	

For all code variants the corresponding binary matrix is formed, the number of changes in the values of each variable (bit) in column  $h_j$  is determined as follows:

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

where  $n$  is the number of digits of the binary code (the number of matrix columns),  $k$  - the number of binary sets (the number of rows of the binary matrix),  $X_i$  - the  $i$ -th binary code,  $X_i = \{x_{i,1} \dots x_{i,n}\}$ , where  $x_{i,1} \dots x_{i,n}$  - bits of the  $i$ -th binary code. The balance of code  $C$  is defined as follows:

$$C = \sum_{i=1}^n |h_i \cdot (\sum_{j=1}^n h_j) / n|. \quad (2)$$

Table 3 shows the Hamiltonian paths ( $U$ ) for  $v_0=0$  and the balance ( $C$ ) of the corresponding binary code obtained with the above algorithm.

The analysis of the obtained unit distance codes given in the table shows that the set of codes consists of acyclic  $n$ -bit codes and corresponding Hamiltonian paths as well as cyclic  $n$ -bit codes and corresponding Hamiltonian cycles in the hypercube  $Q_n$ . Fig. 2 shows examples of cyclic and acyclic codes. Depending on the balance value of the  $C$  code and the number of changes of values of each variable  $H = \{h_1, h_2, \dots, h_n\}$ , the set of unit distance codes can be divided into three classes.

Table 3 – Hamiltonian paths and their characteristics

№	U	C	№	U	C
1	01326457	2.7	10	02645137	2.7
2	01326754	3.3	11	02645731	3.3
3	01375462	1.3	12	02673154	1.3
4	01546237	2.7	13	04513267	2.7
5	01546732	3.3	14	04513762	3.3
6	01573264	1.3	15	04576231	1.3
7	02315467	2.7	16	04623157	2.7
8	02315764	3.3	17	04623751	3.3
9	02376451	1.3	18	04675132	1.3

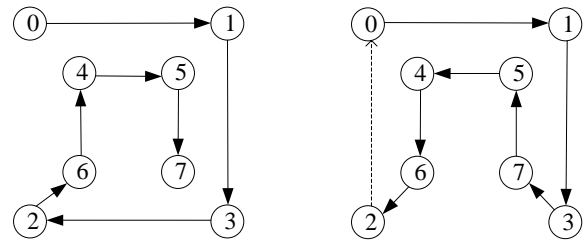


Fig. 2. Example of an acyclic and cyclic path

Table 4 shows decimal equivalents of binary codes ( $D$ ), position code (Bin) and representatives of codes of each class ( $U_1, U_2, U_3$ ) and their characteristics. It should be noted that Gray code belongs to the second class.

Table 4 - Binary codes and their characteristics

$D_0$	Bin			$D_1$	$U_1$			$D_2$	$U_2$			$D_3$	$U_3$		
	$x_1$	$x_2$	$x_3$		$x_1$	$x_2$	$x_3$		$x_1$	$x_2$	$x_3$		$x_1$	$x_2$	$x_3$
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	
2	0	1	0	3	0	1	1	3	0	1	1	3	0	1	
3	0	1	1	2	0	1	0	2	0	1	0	7	1	1	
4	1	0	0	6	1	1	0	6	1	1	0	5	1	0	
5	1	0	1	4	1	0	0	7	1	1	1	4	1	0	
6	1	1	0	5	1	0	1	5	1	0	1	6	1	1	
7	1	1	1	7	1	1	1	4	1	0	0	2	0	1	
H	1	3	7		1	3	3		1	2	4		2	3	
C		6.7				2.7				3.3				1.3	

## Conclusions and prospects for further research in this area

Technological trends and especially portable applications are driving the search for integrated circuits with low power consumption. Solutions are sought that include algorithmic, structural, or physical transformations.

Address bus coding methods with the lowest switching activity are considered. Gray's code, which has low balance and a large number of bit switches, is used to reduce dynamic energy losses in the address bus and to minimize communication losses between closely spaced lines. A type of codes with the same properties

as Gray codes, the unit distance codes, has been defined. We have developed a method for constructing a set of unit distance codes, analyzing their characteristics, and selecting codes that satisfy the given properties. By using the whole set of codes, developers have more choices than when using only Gray codes, and this allows to get better results in terms of branching, propagation delays, power consumption or other related limitations when designing digital systems.

Further research in this direction: developing a method for classifying codes with respect to a given group of transformations, compiling catalogs of typical representatives, and constructing code converters.

### REFERENCES

1. Singh B., Khosla A., Narang S. B. Low power bus encoding techniques for memory testing // *Microelectron Solid State Electron.* – 2013. – V. 2. – № 3. – P. 45-51. DOI: [10.5923/j.msse.20130203.02](https://doi.org/10.5923/j.msse.20130203.02)
2. Serkov A., Trubchaninova K., Lazurenko B. Noise stability of mobile telecommunication systems. *Control, Navigation and Communication Systems.* Poltava: PNTU, 2020. VOL. 2 (60). P. 169-172. doi: <https://doi.org/10.26906/SUNZ.2020.2.169>
3. Bittman D. et al. Optimizing Systems for Byte-Addressable NVM by Reducing Bit Flipping // *FAST.* – 2019. – P. 17-30.
4. Kulandai A. D. R., Rose J., Schwarz T. Gray counters for non-volatile memories // *Memories-Materials, Devices, Circuits and Systems.* – 2022. – V. 2. – P. 100-114. DOI: <https://doi.org/10.1016/j.memori.2022.100014>
5. Mittal S., Nag S. A survey of encoding techniques for reducing data-movement energy // *Journal of Systems Architecture.* – 2019. – V. 97. – P. 373-396.
6. Zhao Z., Wang Z., Min G., Cao Y. Highly-efficient bulk data transfer for structured dissemination in wireless embedded network systems. *Journal of Systems Architecture.* 2017. V. 72. P. 19-28. DOI: <http://dx.doi.org/10.1016/j.sysarc.2016.09.001>
7. Wang S., Ipek E. Reducing data movement energy via online data clustering and encoding // *International Symposium on Microarchitecture (MICRO).* – 2016. – P. 1-13.
8. Korablyov, M., Lutsky, S. System-information models for intelligent information processing // *Innovative Technologies and Scientific Solutions for Industries.* – 2022. – № 3 (21). – P. 26–38. DOI: <https://doi.org/10.30837/ITSSI.2022.21.026>
9. Seol H., Shin W., Jang J., Choi J. Energy efficient data encoding in DRAM channels exploiting data value similarity // *International Symposium on Computer Architecture (ISCA).* – 2016. – P. 719-730.
10. Lee D., O'Connor M., Chatterjee N. Reducing Data Transfer Energy by Exploiting Similarity within a Data Transaction // *IEEE International Symposium on High Performance Computer Architecture (HPCA).* – 2018. – P. 40-51.
11. Volk M., Lunichkin O., Self-healing computer systems // *Control, Navigation and Communication Systems.* *Academic Journal.* – Poltava: PNTU, 2022. – VOL. 1 (67). – P. 48-51. doi: <https://doi.org/10.26906/SUNZ.2022.1.048>
12. Mittal S., Inukonda M. A Survey of Techniques for Improving Error-Resilience of DRAM // *Journal of Systems Architecture.* – 2018. – V. 91. – P. 11-40.
13. Lada N., Rudnytska Y. Implementation of a method for synthesizing groups of symmetric double-operand operations of cryptographic information coding for block encryption systems // *Innovative Technologies and Scientific Solutions for Industries.* – 2022. – № 2 (20). – P. 35– 43. DOI: <https://doi.org/10.30837/ITSSI.2022.20.035>
14. Mütze T., Nummenpalo J. Efficient computation of middle levels Gray codes // *ACM Transactions on Algorithms (TALG).* – 2018. – V. 14. – № 2. – P. 1-29.
15. Kandel A., Bunke H., Last M. *Applied Graph Theory in Computer Vision and Pattern Recognition* // Springer, 2007. – 261 p.
16. Mills, W. H. Some complete cycles on the n-cube // *Proceedings of the American Mathematical Society, American Mathematical Society.* – 1963. – V. 14. – № 4. – P. 640-643.

Received (Надійшла) 20.02.2023

Accepted for publication (Прийнята до друку) 15.03.2023

### Кодування для зменшення енергії руху даних

В. В. Яреценка, В. В. Косенко

**Анотація.** Розглянуто проблему зменшення розсіюваної потужності у глобальних лініях між'єднань за збереження високої продуктивності. Висока комутаційна активність призводить до значних втрат зв'язку через ємності зв'язку між довгими лініями. Нові технології енергонезалежної пам'яті з байтовою адресацією, такі як пам'ять із фазовим переходом, дозволяють створювати системи з великою постійною пам'яттю, підвищуючи надійність та потенційно знижуючи енергоспоживання. Однак ці технології підтримують лише обмежену кількість операцій запису протягом усього терміну служби на комірку та споживають більшу частину своєї потужності при зміні стану біта під час запису. Для зниження комутаційної активності під час зв'язку між пристроями або на кристалі потрібні методи кодування з низьким енергоспоживанням. Мета статті: розробка методу побудови безлічі кодів одичної відстані, аналізу їх характеристик та вибору кодів, що задовольняють задані властивості. Розглянуто методи кодування адресної шини із найменшою комутаційною активністю. Для зменшення динамічних втрат енергії в адресній шині та мінімізації втрат зв'язку між близькими лініями застосовують код Грея, який має ряд недоліків. Визначено вид кодів, що мають ті ж властивості, що і коди Грея - коди одичної відстані. Розроблено метод побудови безлічі кодів одичної відстані, аналізу їх характеристик та вибору кодів, що задовольняють заданим властивостям. Завдяки використанню всієї множини кодів у розробників є більше варіантів вибору, ніж при використанні тільки кодів Грея, і це дозволяє отримати кращі результати.

**Ключові слова:** код Грея, комутаційна активність, гіперграф, енергонезалежна пам'ять.