Micro- and nanoelectronics. Condensed matter physics

Микро- и наноэлектроника. Физика конденсированного состояния

UDC 004.312.44
https://doi.org/10.32362/2500-316X-2023-11-3-46-55

RESEARCH ARTICLE

Method for synthesizing a logic element that implements several functions simultaneously

Stanislav I. Sovetov [1, @], Sergey F. Tyurin [1, 2]

[1] *Perm National Research Polytechnic University, Perm, 614990 Russia*
[2] *Perm State University, Perm, 614068 Russia*
[@] *Corresponding author, e-mail: fizikoz@gmail.com*

Abstract

Objectives. The basic element of a field-programmable gate array is a lookup table (LUT). While in canonical normal form LUTs generally implement only one logical function for a given configuration, in this case, there is always an inactive pass transistor element. Moreover, using a single LUT for a single function reduces system-on-a-chip (SoC) scalability. Therefore, the purpose of the present work is to develop a LUT structure for implementing several logic functions simultaneously on inactive transmitting transistors.

Methods. The evolution of LUT structure is presented for three variables, in which the number of simultaneously implemented functions increases. To implement additional functions, the logical device was decomposed with a different number of variables. The structures were modeled in the *Multisim* electrical simulation system.

Results. The presented simulation of more than two logic functions on inactive parts of the LUT shows the simultaneous operation of two and four logic functions. The complexity for a different number of variables and number of implemented functions is compared.

Conclusions. The simulation results demonstrate the operability of LUT structures in which several logical functions are performed. Thus, when implementing additional functions in the new structure, a smaller number of transmitting transistors is required as compared to a conventional LUT, thus increasing device functionality. The presented solution can be used to increase the number of simultaneously implemented functions of the same variables, which can be important e.g., when implementing code transformations.

Keywords: field-programmable gate array, LUT, transmitting transistors, truth table, logic function

• **Submitted:** 06.12.2022 • **Revised:** 13.01.2023 • **Accepted:** 22.02.2023

For citation: Sovetov S.I., Tyurin S.F. Method for synthesizing a logic element that implements several functions simultaneously. *Russ. Technol. J*. 2023;11(3):46–55. https://doi.org/10.32362/2500-316X-2023-11-3-46-55

Financial disclosure: The authors have no a financial or property interest in any material or method mentioned.

The authors declare no conflicts of interest.

НАУЧНАЯ СТАТЬЯ

Метод синтеза логического элемента, реализующего несколько функций одновременно

С.И. Советов [1, @],
С.Ф. Тюрин [1, 2]

[1] *Пермский национальный исследовательский политехнический университет, Пермь, 614990 Россия*
[2] *Пермский государственный национальный исследовательский университет, Пермь, 614068 Россия*
[@] *Автор для переписки, e-mail: fizikoz@gmail.com*

Резюме

Цель. Базовый элемент программируемой логической интегральной схемы (ПЛИС) реализует логические функции с помощью таблиц истинности (LUT). Строение обычных LUT позволяет реализовывать только одну логическую функцию нескольких переменных в совершенной дизъюнктивной нормальной форме (СДНФ). При этом всегда остается часть неактивных передающих транзисторов. Использование одной LUT для одной функции усложняет масштабирование архитектуры на кристалле (SoC). Целью данной работы является разработка структуры LUT для реализации нескольких логических функций одновременно на неактивных передающих транзисторах.

Методы. Приведена эволюция структуры LUT для трех переменных, в которой увеличивается количество одновременно реализуемых функций. Для реализации дополнительных функций выполнена декомпозиция логического устройства с различным количеством переменных. Проведено моделирование структур в системе электротехнического моделирования *Multisim*.

Результаты. Продемонстрировано моделирование более двух логических функций на неактивных частях LUT, при котором отображена одновременная работа двух и четырех логических функций. Приведено сравнение сложности для разного количества переменных и количества реализованных функций.

Выводы. Результаты моделирования демонстрируют работоспособность структур LUT, в которых выполняется несколько логических функций. Таким образом, при реализации дополнительных функций в новой структуре требуется меньшее количество передающих транзисторов по сравнению с обычным LUT, что увеличивает функциональность устройства. Новое решение позволяет увеличить число одновременно реализуемых функций одних и тех же переменных, что важно при реализации, например, кодовых преобразований.

Ключевые слова: ПЛИС, LUT, передающие транзисторы, таблица истинности, логическая функция

• **Поступила:** 06.12.2022 • **Доработана:** 13.01.2023 • **Принята к опубликованию:** 22.02.2023

Для цитирования: Советов С.И., Тюрин С.Ф. Метод синтеза логического элемента, реализующего несколько функций одновременно. *Russ. Technol. J.* 2023;11(3):46–55. https://doi.org/10.32362/2500-316X-2023-11-3-46-55

Прозрачность финансовой деятельности: Авторы не имеют финансовой заинтересованности в представленных материалах или методах.

Авторы заявляют об отсутствии конфликта интересов.

INTRODUCTION

Field-programmable gate arrays (FPGAs), where a configurable logic unit is used as a basic element, are currently common in electronic devices. The basic element of this block comprises a lookup table (LUT), which implements some logical function. Modern LUTs are configurable multiplexers for realization of logic functions with 2^n inputs and one output for n variables [1, 2]. The use of existing LUTs for n variables requires memory cells and $2^{n+1} - 2$ transistors; in this case, only one logic function is realized. Although,

the same number of LUTs must be used when implementing n logic functions simultaneously, when using a single function in an LUT, the second half of the tree of transistors in the number of 2^{n-1} remains inactive [3, 4]. In previous works, the simultaneous implementation of two logic functions on one tree of transmitting transistors was proposed [5–7].

The 3-LUT tree shown in Fig. 1 consists of three NMOS (N-type metal-oxide-semiconductor) cascades of pass transistors[1] [8, 9]. For any of the eight values on the static random-access memory (SRAM) inputs, only one chain of passing transistors is enabled, while, on the remaining passing transistors, at least one chain is completely inactive.

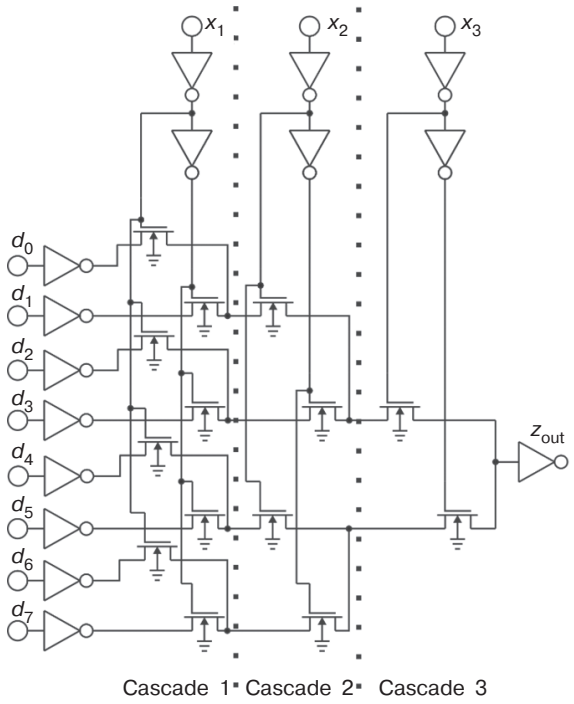

Fig. 1. Tree of LUT of the table of three variables

The 3-LUT logic function can be represented linearly as follows:

$$z(x_3x_2x_1d) = d_0 \cdot \bar{x}_3\bar{x}_2\bar{x}_1 \vee d_1 \cdot \bar{x}_3\bar{x}_2x_1 \vee \vee d_2 \cdot \bar{x}_3x_2\bar{x}_1 \vee d_3 \cdot \bar{x}_3x_2x_1 \vee d_4 \cdot x_3\bar{x}_2\bar{x}_1 \vee \vee d_5 \cdot x_3\bar{x}_2x_1 \vee d_6 \cdot x_3x_2\bar{x}_1 \vee d_7 \cdot x_3x_2x_1, \quad (1)$$

where d_0, d_1, d_2, d_3, d_4, d_5, d_6, d_7 are configuration data of the three variables function $z(x_3x_2x_1)$. By combining d_0, d_1, d_2, d_3, d_4, d_5, d_6, d_7, we can get 2^8 functions.

In minterm canonical form (MCF) or canonical disjunctive normal form (CDNF), all LUTs of n variables perform a single logical function of n arguments [10, 11].

[1] Intel® FPGAs and SoC FPGAs. https://www.intel.in/content/www/in/en/products/details/fpga/cyclone.html. Accessed November 17, 2022.

At the same time, another logical function of the same arguments can be activated on each inactive chain, e.g., summation or carry functions. By combining these chains in OR, we can get a logic element with several outputs.

In the present work, an improved array is proposed that uses the inactive branches of the transmitting transistor tree. By introducing additional LUTs of two variables at stage 2, it is possible to use additional functions on inactive tree circuits.

This novel solution allows the number of simultaneously realized functions of the same variables to be increased; this can important when implementing code conversions.

SYNTHESIS OF A MULTIFUNCTIONAL LOGIC ELEMENT

In order to implement both functions, a single variable LUT was added to the array. This consists of an output inverter and two transmitting transistors, whose outputs are combined and connected to the inverter. LUT inputs of the single variable transistors are connected to the outputs of the second cascade of transmitting transistors. Two transmitting transistors each are added to the input setting inverters; these are controlled by the senior variable, whose outputs are combined and connected to the inverter. This results in a three-variable LUT circuit with two simultaneous functions (Fig. 2).

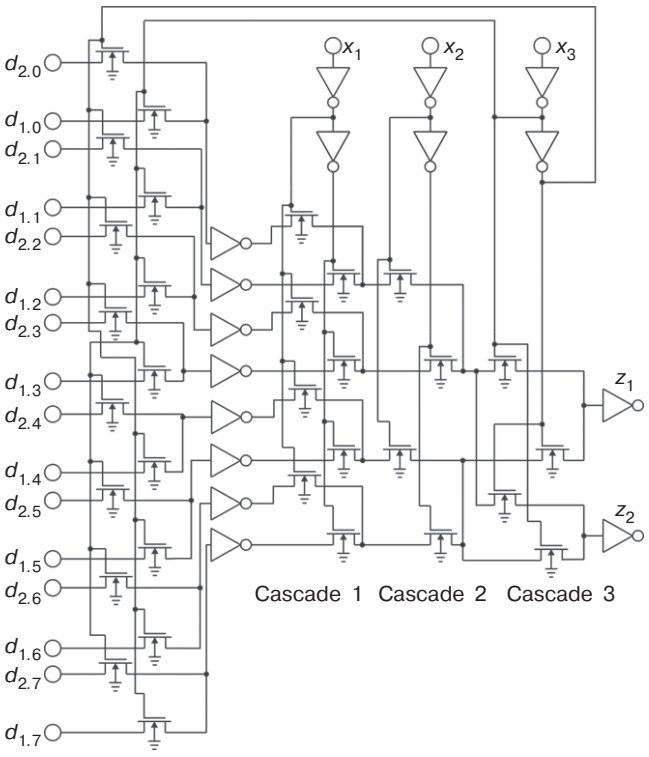

Fig. 2. Tree of three-variable LUT performing two functions simultaneously

To implement two functions, we decompose a logical device by a higher variable (x_3):

First function:

$$z_1(x_3x_2x_1d) = \\ = \bar{x}_3\left[d_{1.0}\cdot\bar{x}_2\bar{x}_1 \vee d_{1.1}\cdot\bar{x}_2x_1 \vee \right. \\ \vee d_{1.2}\cdot x_2\bar{x}_1 \vee d_{1.3}\cdot x_2x_1\left.\right]\vee \\ \vee x_3\left[d_{1.4}\cdot\bar{x}_2\bar{x}_1 \vee d_{1.5}\cdot\bar{x}_2x_1 \vee \right. \\ \vee d_{1.6}\cdot x_2\bar{x}_1 \vee d_{1.7}\cdot x_2x_1\left.\right]. \quad (2)$$

Second function:

$$z_2(x_3x_2x_1d) = \\ = x_3\left[d_{2.4(0)}\cdot\bar{x}_2\bar{x}_1 \vee d_{2.5(1)}\cdot\bar{x}_2x_1 \vee \right. \\ \vee d_{2.6(2)}\cdot x_2\bar{x}_1 \vee d_{2.7(3)}\cdot x_2x_1\left.\right]\vee \\ \vee \bar{x}_3\left[d_{2.0(4)}\cdot\bar{x}_2\bar{x}_1 \vee d_{2.1(5)}\cdot\bar{x}_2x_1 \vee \right. \\ \vee d_{2.2(6)}\cdot x_2\bar{x}_1 \vee d_{2.3(7)}\cdot x_2x_1\left.\right], \quad (3)$$

where $d_{i,j}$; $i = 1, 2(2^v)$; $j = 1, 2, 3, 4, \ldots, 2^3(2^n)$, v is the number of implemented functions.

The setting constant is specified in the format $d_{i,j(k)}$, where k defines the number of the function set and j is the number of the input used to connect it. In our case k is obtained from j by inverting the older variable. Thus, the top half of the SRAM is swapped with the bottom half to use the inactive part of the pass transistors. As a result, two chains from SRAM to output inverters in this LUT configuration of three variables are active, i.e., two functions are implemented simultaneously.

To further increase the number of simultaneously executed functions on the LUT of three variables an additional LUT tree of two variables for each function is introduced, whose inputs are connected to the outputs of the transmitting transistors of the first stage. Additionally, a cascade of transmitting tuning transistors controlled by the second variable (x_2) is introduced. The outputs of the transistors are combined and connected to the inputs of the first transmitting tuning transistor stage. The general schematic of a four-function LUT of three variables is shown in Fig. 3.

To implement four functions, we decompose the logical unit into a higher variable (x_3) and a second variable (x_2):

First function:

$$z_1(x_3x_2x_1d) = \bar{x}_3\bar{x}_2\left[d_{1.0}\cdot\bar{x}_1 \vee d_{1.1}\cdot x_1\right]\vee \\ \vee \bar{x}_3x_2\left[d_{1.2}\cdot\bar{x}_1 \vee d_{1.3}\cdot x_1\right]\vee \\ \vee x_3\bar{x}_2\left[d_{1.4}\cdot\bar{x}_1 \vee d_{1.5}\cdot x_1\right]\vee \\ \vee x_3x_2\left[d_{1.6}\cdot\bar{x}_1 \vee d_{1.7}\cdot x_1\right]. \quad (4)$$

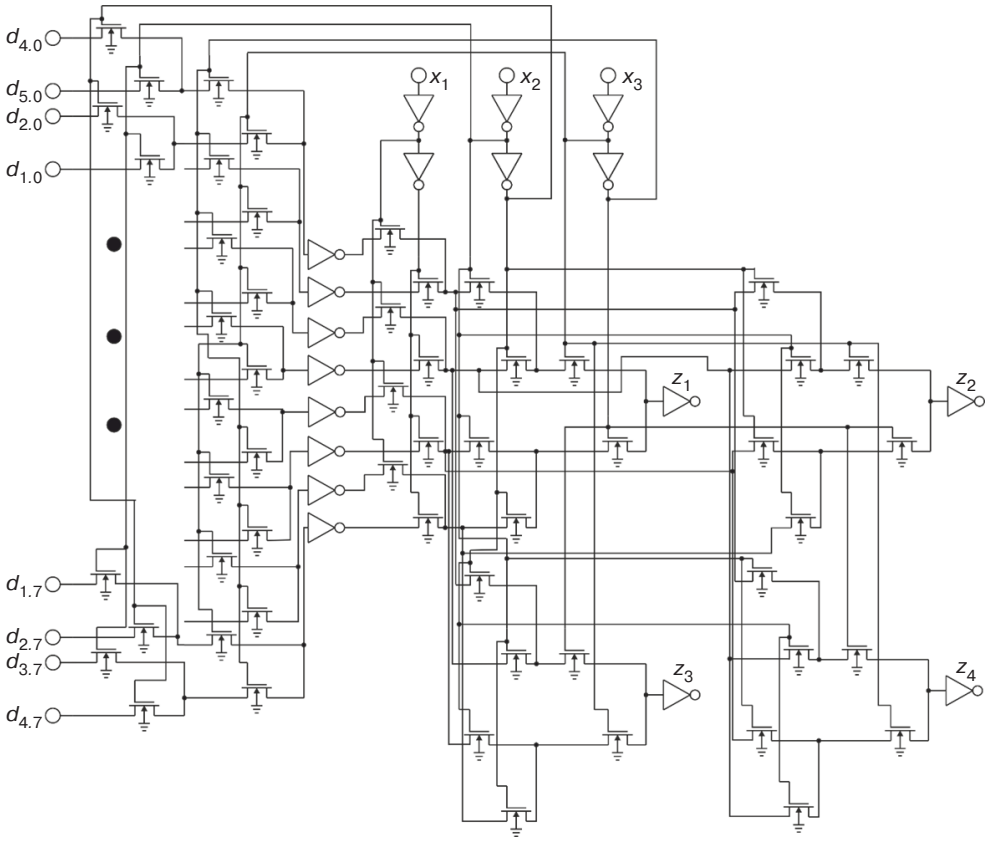

Fig. 3. Tree of three-variable LUT performing four functions simultaneously

Table. Setting the logical levels of SRAM cells for a LUT of three variables performing four functions

No. of input	Configuration for z_1	Configuration for z_2	Configuration for z_3	Configuration for z_4
1	0	0	1	0
2	1	1	1	1
3	1	0	1	0
4	0	0	1	0
5	1	1	0	0
6	0	1	1	0
7	0	0	1	0
8	1	1	1	0

Second function:

$$z_2(x_3x_2x_1d) = \\ = \bar{x}_3x_2\left[d_{2.2(0)}\cdot\bar{x}_1 \vee d_{2.3(1)}\cdot x_1\right]\vee \\ \vee x_3\bar{x}_2\left[d_{2.4(2)}\cdot\bar{x}_1 \vee d_{2.5(3)}\cdot x_1\right]\vee \\ \vee x_3x_2\left[d_{2.6(4)}\cdot\bar{x}_1 \vee d_{2.7(5)}\cdot x_1\right]\vee \\ \vee\bar{x}_3\bar{x}_2\left[d_{2.0(6)}\cdot\bar{x}_1 \vee d_{2.1(7)}\cdot x_1\right]. \quad (5)$$

Third function:

$$z_3(x_3x_2x_1d) = \\ = x_3\bar{x}_2\left[d_{3.4(0)}\cdot\bar{x}_1 \vee d_{3.5(1)}\cdot x_1\right]\vee \\ \vee x_3x_2\left[d_{3.6(2)}\cdot\bar{x}_1 \vee d_{3.7(3)}\cdot x_1\right]\vee \\ \vee\bar{x}_3\bar{x}_2\left[d_{3.0(4)}\cdot\bar{x}_1 \vee d_{3.1(5)}\cdot x_1\right]\vee \\ \vee\bar{x}_3x_2\left[d_{3.2(6)}\cdot\bar{x}_1 \vee d_{3.3(7)}\cdot x_1\right]. \quad (6)$$

Fourth function:

$$z_4(x_3x_2x_1d) = \\ = x_3x_2\left[d_{4.6(0)}\cdot\bar{x}_1 \vee d_{4.7(1)}\cdot x_1\right]\vee \\ \vee\bar{x}_3\bar{x}_2\left[d_{4.0(2)}\cdot\bar{x}_1 \vee d_{4.1(3)}\cdot x_1\right]\vee \\ \vee\bar{x}_3x_2\left[d_{4.2(4)}\cdot\bar{x}_1 \vee d_{4.3(5)}\cdot x_1\right]\vee \\ \vee x_3\bar{x}_2\left[d_{4.4(6)}\cdot\bar{x}_1 \vee d_{4.5(7)}\cdot x_1\right]. \quad (7)$$

The configuration levels derived from the decomposition of the four functions: exclusive OR (z_1), majority (z_2), disjunction (z_3), and conjunction (z_4) are shown in Table.

LUT SIMLATION

Simulation of a three-variable multifunction LUT was performed in the National Instruments *Multisim*[2] dynamic simulation system. For the NMOS transistors, the BSIM 4.8.0[3,4] transistor model was used. The schematic of the inverter implemented on two field-effect transistors is shown in Fig. 4.

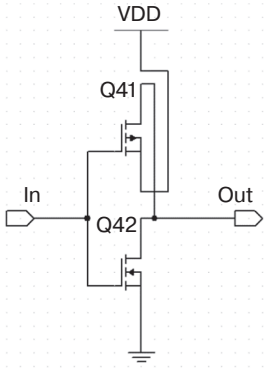

Fig. 4. Field-effect transistor inverter diagram
(In the diagrams that follow, the designations adopted in the GOST 2.710-81[5] standard are used.)

First of all, we consider the general diagram of implementation of LUT for two functions (exclusive OR, majority) as shown in Fig. 5.

[2] https://www.multisim.com/. Accessed November 17, 2022.

[3] Microwind & Dsch Version 3.5. https://www.yumpu.com/en/document/view/40386405/microwind-manual-lite-v35pdf-moodle. Accessed November 17, 2022.

[4] Mead C.A., Conway L. *Introduction to VLSI Systems*. Reading, MA, Addison-Wesley Publishing Co.;1980. 426 p. https://www.researchgate.net/publication/234388249_Introduction_to_VLSI_systems. Accessed November 17, 2022.

[5] GOST 2.710-81. *Unified system for design documentation. Alpha-numerical designations in electrical diagrams*. Moscow: Standartinform; 2008 (in Russ.).

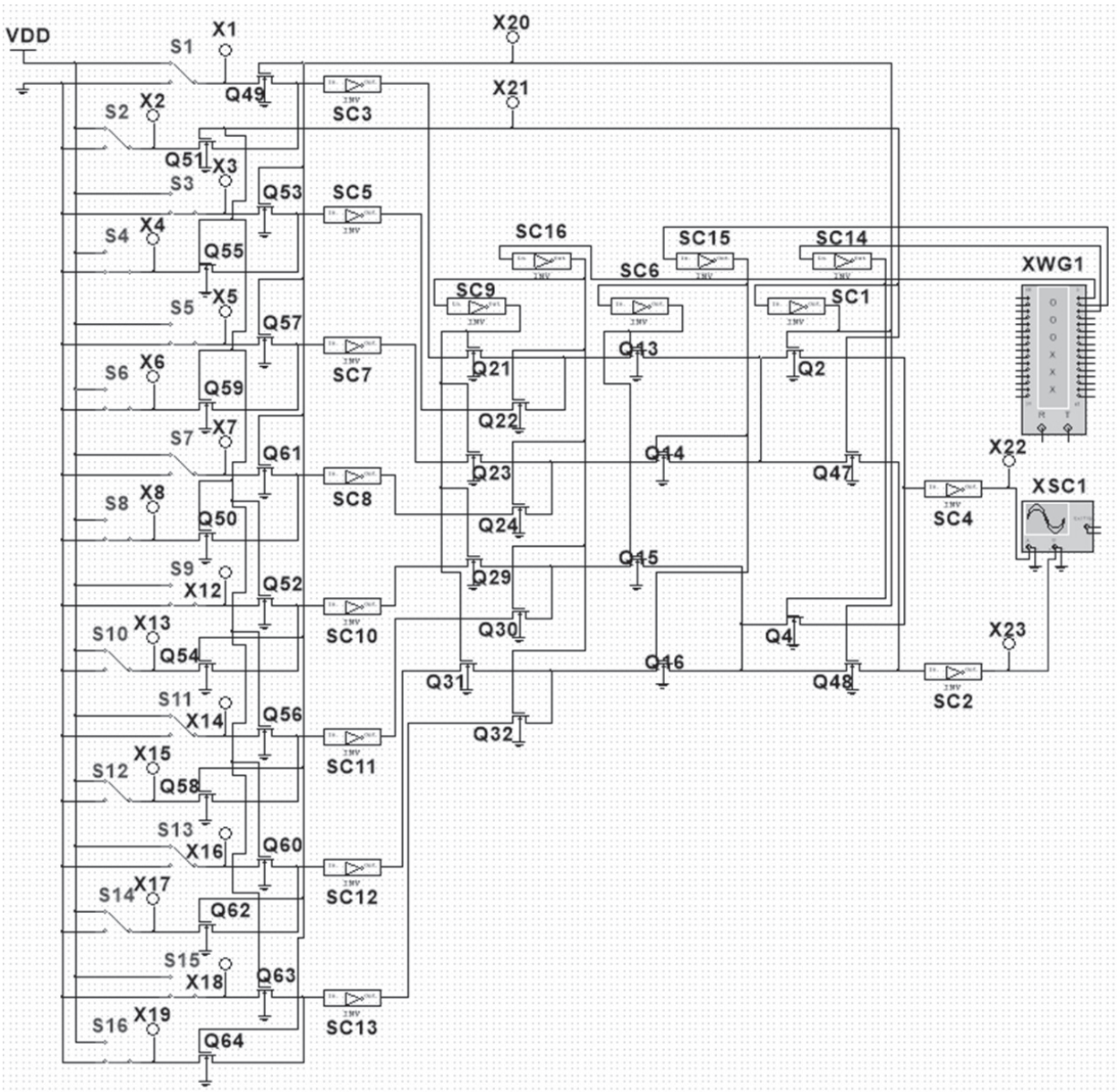

Fig. 5. Diagram of the three-variable LUT performing two functions simultaneously

A modified three-variable LUT scheme for performing four functions simultaneously (exclusive OR, majority, disjunction, and conjunction) is shown in Fig. 6 (see p. 52).

The input setting signal of the SRAM cells was set with the dynamic switches according to Table. The high voltage level (VDD 5V) corresponds to the logical variant, while the low voltage level (GND 0V) corresponds to logical zero.

The variable signals were set using the *Word Generator*, in which a three-bit Gray code is set (Fig. 7). Since the *Word Generator* has a frequency of 1 kHz, it requires 8 ms to completely pass the given Gray code.

Function outputs are connected to the oscilloscope.

The results of the simulation of two and four functions are shown in Figs. 8 and 9 respectively.

The upper oscillogram corresponds to the "exclusive OR" function (z_1). The lower oscillogram corresponds to the majority function (z_2). Each time division of the oscilloscope corresponds to one combination of the Gray code.

In contrast to the basic three-variable LUT diagram, which uses only one half of the transmitting transistors, the proposed new circuit additionally uses the inactive part at 2^{n-1} high order variables.

Complexity in terms of the number of LUT transistors depending on the number of variables n, $n \in \mathbf{N}$ is calculated as follows:

$$L_1(n) =$$
$$= (2+2)\cdot 2^n + (2^{n+1} - 2) + 2n + 2 + 2 =$$
$$= 2^{n+2} + 2^{n+1} + 2n + 2 = 3\cdot 2^{n+1} + 2n + 2.$$

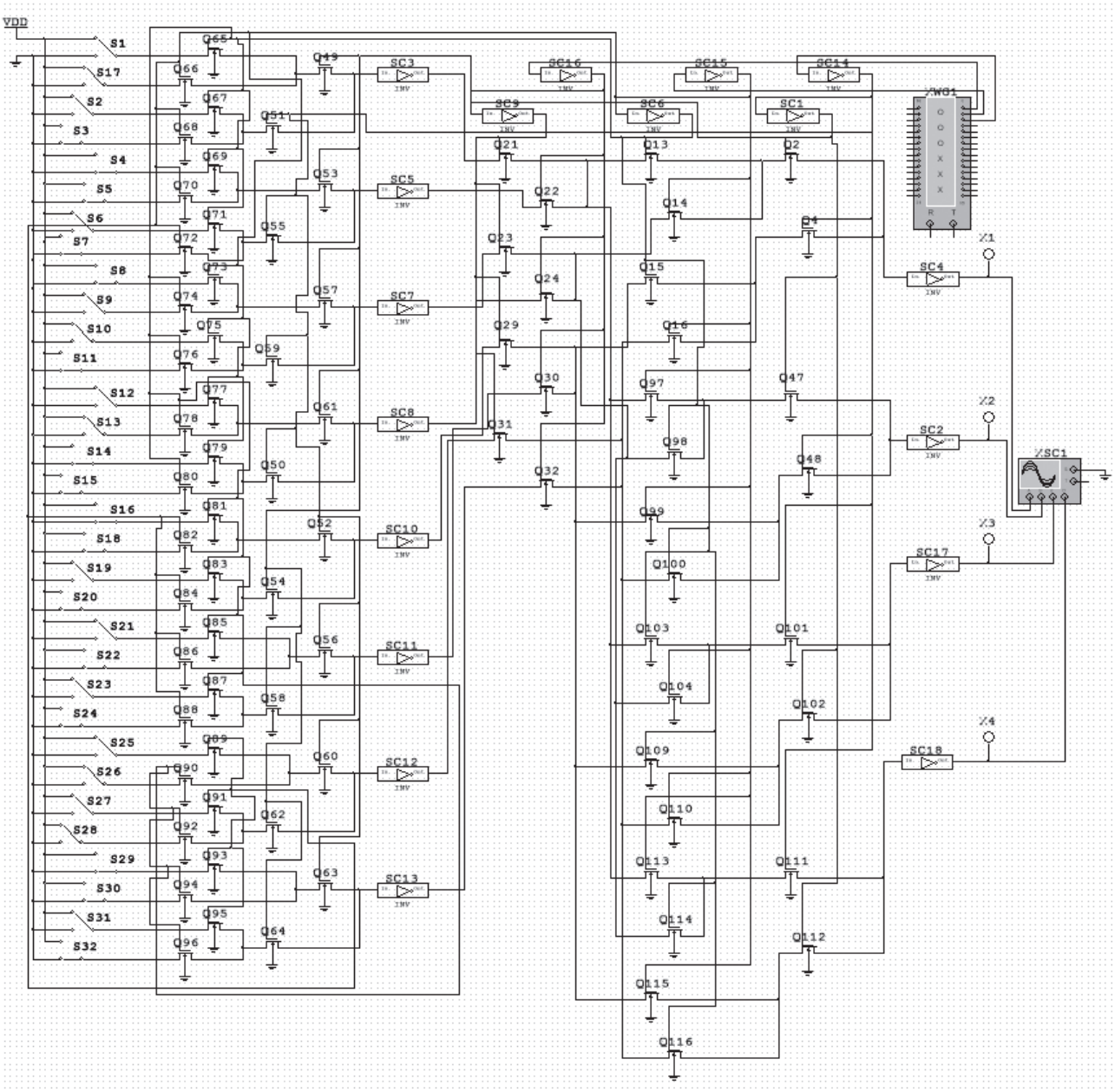

Fig. 6. Diagram of the three-variable LUT performing four functions simultaneously

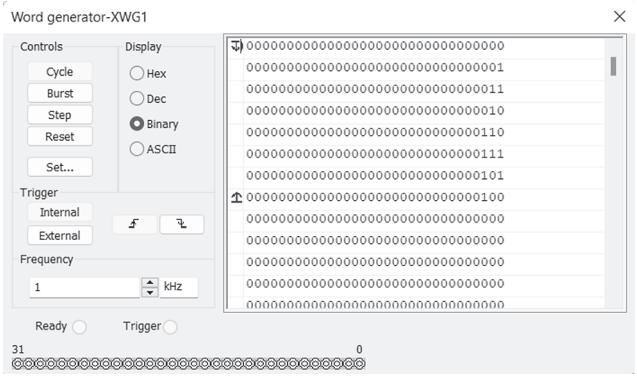

Fig. 7. Gray code in *Word Generator*

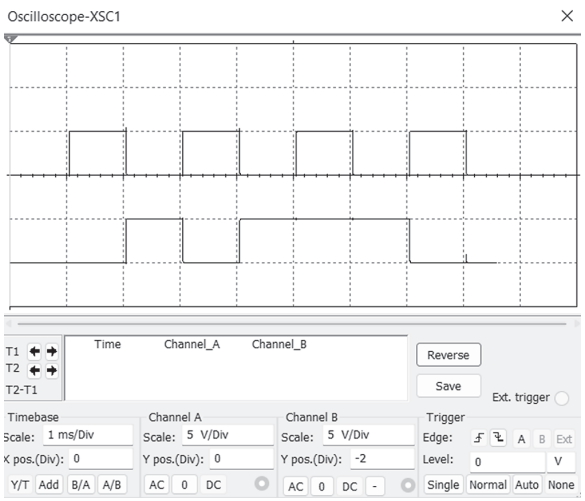

Fig. 8. Oscillogram of two functions of three-variable LUT

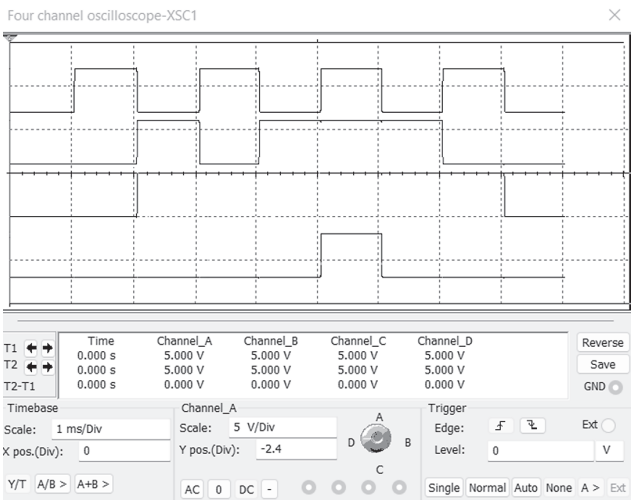

Fig. 9. Oscillogram of four functions of three-variable LUT

To calculate 2^v functions in the proposed scheme we obtain the complexity:

$$L_v(v,n) =$$
$$= (2^{v+1} - 2 + 2) \cdot 2^n + 2^{n+1} + 2n +$$
$$+ (2^{v+1} - 2)(2^v - 1) + 2(2^v - 1) =$$
$$= 2^{v+1} \cdot 2^n + 2^{n+1} + 2n + 2^{v+1} \cdot (2^v - 1).$$

Thus, for example, implementing four logic functions ($v = 2$) of the same variables ($n = 4$) requires two LUTs for a total use of 212 transistors, while, in the proposed LUT design, which implements all four functions, the number of required transistors is 192.

The complexity diagrams representing the usual LUT $L_{v1}(v)$ and the multifunctional LUT $L_v(v)$ implementing v logic functions with the number of variables $n = 8$ are shown in Fig. 10.

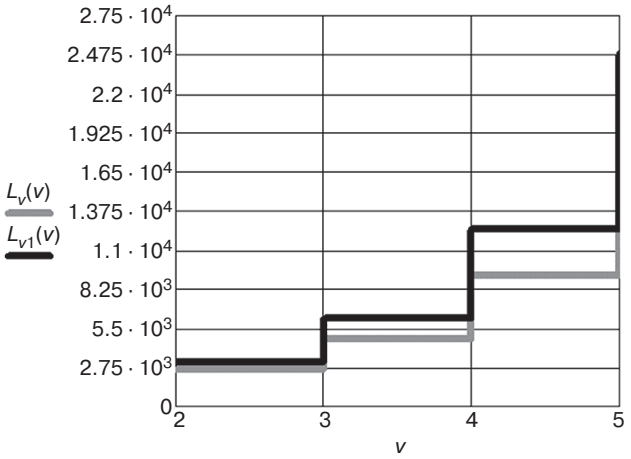

Fig. 10. Comparison of the complexity of a conventional LUT and a multifunctional LUT

As can be seen from the diagram, there is a complexity benefit in terms of the number of transistors for a multifunctional LUT in relation to the corresponding number for conventional LUTs.

CONCLUSIONS

Thus, the possibility of simultaneous implementation of 2^{n-1} logic functions on inactive parts of the circuits of the transmitting transistors in the FPGA, which increases the functionality of the device, is demonstrated. In order to implement the same number of logic functions, 2^{n-2} LUTs are required. Despite the additional cost, there is a benefit in terms of complexity as compared to the number of required transistors.

Authors' contributions

S.I. Sovetov—preparation of research, literature analysis, modeling, writing the text of the article, formulation of conclusions.

S.F. Tyurin—the idea of research, setting goals and objectives, consultations on research issues, editing the article.

REFERENCES

1. Drozd O., Perebeinos I., Martynyuk O., Zashcholkin K., Ivanova O., Drozd M. Hidden fault analysis of FPGA projects for critical applications. In: *2020 IEEE 15th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering* (*TCSET*). Lviv-Slavske, Ukraine, 2020. P. 467–471. https://doi.org/10.1109/TCSET49122.2020.235591
2. Drozd O., Nikul V., Antoniuk V., Drozd M. Hidden faults in FPGA-built digital components of safety-related systems. In: *2018 14th International Conference on Advanced Trends in Radioelecrtronics, Telecommunications and Computer Engineering* (*TCSET*). Lviv-Slavske, Ukraine, 2018. P. 805–809. https://doi.org/10.1109/TCSET.2018.8336320

СПИСОК ЛИТЕРАТУРЫ

1. Drozd O., Perebeinos I., Martynyuk O., Zashcholkin K., Ivanova O., Drozd M. Hidden fault analysis of FPGA projects for critical applications. In: *2020 IEEE 15th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering* (*TCSET*). Lviv-Slavske, Ukraine, 2020. P. 467–471. https://doi.org/10.1109/TCSET49122.2020.235591
2. Drozd O., Nikul V., Antoniuk V., Drozd M. Hidden faults in FPGA-built digital components of safety-related systems. In: *2018 14th International Conference on Advanced Trends in Radioelecrtronics, Telecommunications and Computer Engineering* (*TCSET*). Lviv-Slavske, Ukraine, 2018. P. 805–809. https://doi.org/10.1109/TCSET.2018.8336320

3. Vikhorev R. Universal logic cells to implement systems functions. In: *2016 IEEE NW Russia Young Researchers in Electrical and Electronic Engineering Conference* (*ELConRusNW*). St. Petersburg, Russia, 2016. P. 373–375. https://doi.org/10.1109/EIConRusNW.2016.7448197
4. Vikhorev R. Improved FPGA logic elements and their simulation. In: *2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering* (*EIConRus*). Moscow and St. Petersburg, Russia, 2018. P. 259–264. https://doi.org/10.1109/EIConRus.2018.8317080
5. Skornyakova A.Yu., Vikhorev R.V. Self-timed LUT layout simulation. In: *2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering* (*EIConRus*). St. Petersburg and Moscow, Russia, 2020. P. 176–179. https://doi.org/10.1109/EIConRus49466.2020.9039374
6. Tyurin S.F. Green logic: Green LUT FPGA concepts, models and evaluations. In: Kharchenko V., Kondratenko Y., Kacprzyk J. (Eds.). *Green IT Engineering: Components, Networks and Systems Implementation*. *Studies in Systems, Decision and Control*. 2017;105: 241–261. https://doi.org/10.1007/978-3-319-55595-9_12
7. Tyurin S.F., Prokhorov A.S. *Programmable logic device*: RF Pat. 2637462. Publ. 04.12.2017 (in Russ.).
8. Tyurin S.F. LUT's sliding backup. *IEEE Transactions on Device and Materials Reliability*. 2019;19(1):221–225. https://doi.org/10.1109/TDMR.2019.2898724
9. Strogonov A., Tsybin S. Programmable switching FPGA Insights. *Komponenty i tekhnologii = Components & Technologies*. 2010;11(112):56–62 (in Russ.). Available from URL: https://kit-e.ru/wp-content/uploads/2010_11_56.pdf (accessed November 17, 2022).
10. Strogonov A., Gorodkov P. Current trends in FPGA development: from system integration to artificial intelligence. *Elektronika: Nauka, tekhnologiya, biznes = Electronics: Science, Technology, Business*. 2020;4(195): 46–56 (in Russ.). https://doi.org/10.22184/1992-4178.2020.195.4.46.56
11. Danilova E.Y. FPGAs logic checking method by genetic algorithms. In: *2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering* (*EIConRus*). St. Petersburg and Moscow, Russia, 2020. P. 1787–1790. https://doi.org/10.1109/EIConRus49466.2020.9039052

3. Vikhorev R. Universal logic cells to implement systems functions. In: *2016 IEEE NW Russia Young Researchers in Electrical and Electronic Engineering Conference* (*ELConRusNW*). St. Petersburg, Russia, 2016. P. 373–375. https://doi.org/10.1109/EIConRusNW.2016.7448197
4. Vikhorev R. Improved FPGA logic elements and their simulation. In: *2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering* (*EIConRus*). Moscow and St. Petersburg, Russia, 2018. P. 259–264. https://doi.org/10.1109/EIConRus.2018.8317080
5. Skornyakova A.Yu., Vikhorev R.V. Self-timed LUT layout simulation. In: *2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering* (*EIConRus*). St. Petersburg and Moscow, Russia, 2020. P. 176–179. https://doi.org/10.1109/EIConRus49466.2020.9039374
6. Tyurin S.F. Green logic: Green LUT FPGA concepts, models and evaluations. In: Kharchenko V., Kondratenko Y., Kacprzyk J. (Eds.). *Green IT Engineering: Components, Networks and Systems Implementation*. *Studies in Systems, Decision and Control*. 2017;105: 241–261. https://doi.org/10.1007/978-3-319-55595-9_12
7. Тюрин С.Ф., Прохоров А.С. *Программируемое логическое устройство*: Пат. 2637462 РФ. Заявка № 2016131738; заявл. 01.08.2016; опубл. 04.12.2017.
8. Tyurin S.F. LUT's sliding backup. *IEEE Transactions on Device and Materials Reliability*. 2019;19(1):221–225. https://doi.org/10.1109/TDMR.2019.2898724
9. Строгонов А., Цыбин С. Программируемая коммутация ПЛИС: взгляд изнутри. *Компоненты и технологии*. 2010;11(112):56–62. URL: https://kit-e.ru/wp-content/uploads/2010_11_56.pdf (дата обращения 17.11.2022).
10. Строгонов А., Городков П. Современные тенденции развития ПЛИС: от системной интеграции к искусственному интеллекту. *Электроника: Наука, технология, бизнес*. 2020;4(195):46–56. https://doi.org/10.22184/1992-4178.2020.195.4.46.56
11. Danilova E.Y. FPGAs logic checking method by genetic algorithms. In: *2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering* (*EIConRus*). St. Petersburg and Moscow, Russia, 2020. P. 1787–1790. https://doi.org/10.1109/EIConRus49466.2020.9039052

About the authors

Stanislav I. Sovetov, Postgraduate Student, Department of Automation and Telemechanics, Perm National Research Polytechnic University (29, Komsomolskii pr., Perm, 614990 Russia). E-mail: fizikoz@gmail.com. https://orcid.org/0000-0003-4311-1045

Sergey F. Tyurin, Dr. Sci. (Eng.), Professor, Department of Automation and Telemechanics, Perm National Research Polytechnic University (29, Komsomolskii pr., Perm, 614990 Russia); Professor, Department of Software Computing Systems, Perm State University (15, Bukireva ul., Perm, 614068 Russia). E-mail: tyurinsergfeo@yandex.ru. Scopus Author ID 6603805561, https://orcid.org/0000-0002-5707-5404

Об авторах

Советов Станислав Игоревич, аспирант, кафедра автоматики и телемеханики ФГАОУ ВО «Пермский национальный исследовательский политехнический университет» (614990, Россия, Пермь, Комсомольский пр-т, д. 29). E-mail: fizikoz@gmail.com. https://orcid.org/0000-0003-4311-1045

Тюрин Сергей Феофентович, д.т.н., профессор, кафедра автоматики и телемеханики ФГАОУ ВО «Пермский национальный исследовательский политехнический университет» (614990, Россия, Пермь, Комсомольский пр-т, д. 29); профессор, кафедра «Математическое обеспечение вычислительных систем», ФГАОУ ВО «Пермский государственный национальный исследовательский университет» (614068, Россия, Пермь, ул. Букирева, д. 15). E-mail: tyurinsergfeo@yandex.ru. Scopus Author ID 6603805561, https://orcid.org/0000-0002-5707-5404

Translated from Russian into English by Lyudmila O. Bychkova
Edited for English language and spelling by Thomas A. Beavitt