

Микро- и нанoeлектроника. Физика конденсированного состояния
Micro- and nanoelectronics. Condensed matter physics

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



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.

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.

ВВЕДЕНИЕ

В настоящее время в электронных устройствах распространены программируемые логические интегральные схемы (ПЛИС), где в качестве базового элемента используется конфигурируемый логический блок. Основным элементом данного блока является таблица истинности (англ. lookup table, LUT), которая реализует некоторую логическую функцию. Современные LUT представляют собой конфигурируемые мультиплексоры для реализации логических функций с 2^n входами и одним выходом для n переменных [1, 2]. Использование существующих

LUT для n переменных требует 2^n ячеек памяти и $2^{n+1} - 2$ передающих транзисторов. При этом реализуется только одна логическая функция. Для реализации одновременно n логических функций необходимо использование такого же количества LUT. Однако при использовании одной функции в LUT вторая половина дерева передающих транзисторов в количестве 2^{n-1} остается неактивной [3, 4]. В предыдущих работах предлагалась реализация одновременно двух логических функций на одном дереве передающих транзисторов [5–7].

Дерево 3-LUT (рис. 1) состоит из трех каскадов NMOS (N-type metal-oxide-semiconductor)

проходных транзисторов¹ [8, 9]. Для любого из восьми значений на входах статической памяти с произвольным доступом (static random access memory, SRAM) включается только одна цепочка передающих транзисторов, а на оставшихся передающих транзисторах, как минимум, одна цепочка полностью неактивна.

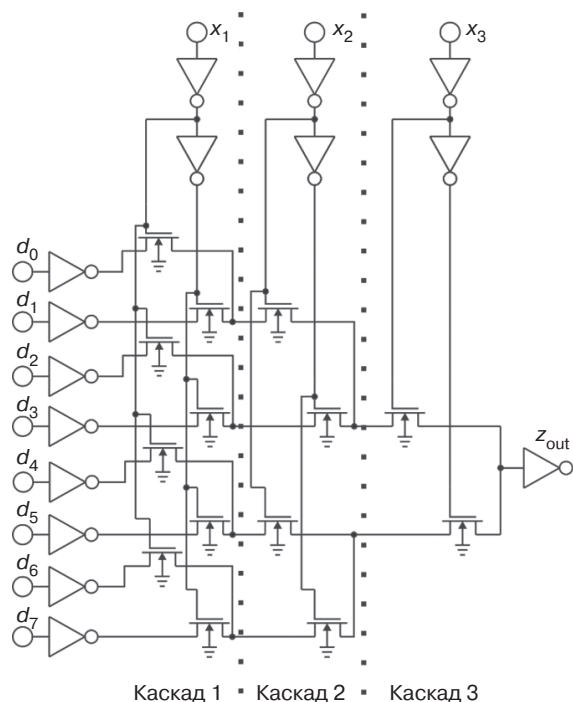


Рис. 1. Дерево LUT таблицы трех переменных

Линейное представление логической функции 3-LUT выглядит как:

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

где $d_0, d_1, d_2, d_3, d_4, d_5, d_6, d_7$ – данные конфигурации функции трех переменных $z(x_3x_2x_1)$. Комбинируя $d_0, d_1, d_2, d_3, d_4, d_5, d_6, d_7$, мы можем получить 2^8 функций.

При этом все LUT n переменных выполняют только одну логическую функцию из n аргументов в дизъюнктивной нормальной форме (ДНФ) или совершенной дизъюнктивной нормальной форме (СДНФ) [10, 11]. В то же время на каждой неактивной цепочке может быть активированная другая логическая функция тех же аргументов, например, функции суммирования или переноса. Комбинируя эти цепочки по ИЛИ, мы можем получить логический элемент с несколькими выходами.

В статье предлагается усовершенствованная схема, использующая неактивные ветви дерева передающих транзисторов. С помощью введения дополнительных LUT двух переменных в каскаде 2 возможно использовать дополнительные функции на неактивных цепях дерева.

Новое решение позволяет увеличить число одновременно реализуемых функций одних и тех же переменных, что важно при реализации, например, кодовых преобразований.

СИНТЕЗ МНОГОФУНКЦИОНАЛЬНОГО ЛОГИЧЕСКОГО ЭЛЕМЕНТА

Для реализации двух функций в схему был добавлена LUT одной переменной, которая состоит из выходного инвертора и двух передающих транзисторов, выходы которых объединены и подключены к инвертору. Входы LUT одной переменной передающих транзисторов подключены к выходам второго каскада передающих транзисторов. Ко входным инверторам настройки добавлены по 2 передающих транзистора, управляемые старшей переменной, выходы которых объединены и подключены к инвертору. Таким образом получается схема LUT трех переменных с одновременным выполнением двух функций (рис. 2).

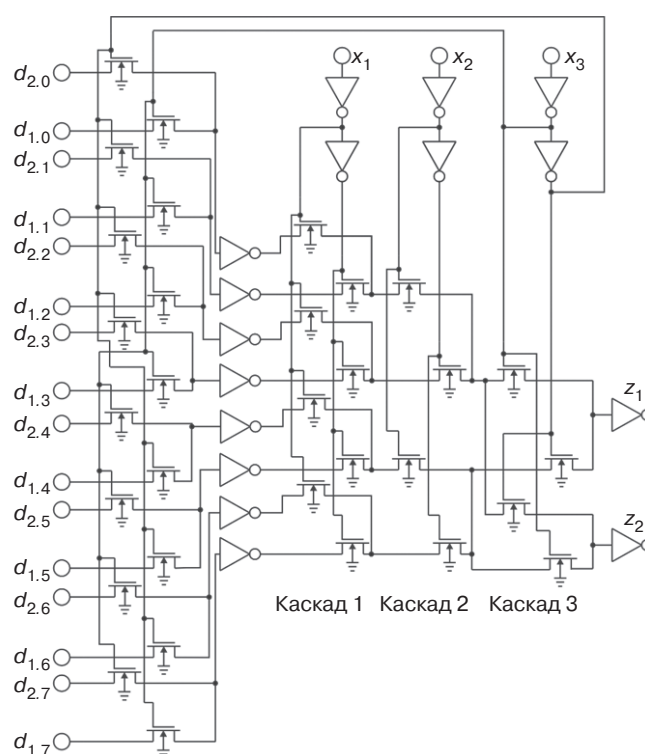


Рис. 2. Дерево LUT трех переменных, выполняющее две функции одновременно

Для реализации двух функций выполним декомпозицию логического устройства по старшей переменной (x_3):

¹ Intel® FPGAs and SoC FPGAs. <https://www.intel.in/content/www/in/en/products/details/fpga/cyclone.html>. Дата обращения 17.11.2022. / Accessed November 17, 2022.

Первая функция:

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

Вторая функция:

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

где $d_{i,j}$; $i = 1, 2(2^v)$; $j = 1, 2, 3, 4, \dots, 2^3(2^n)$; v – количество реализуемых функций. Константа настройки задается в формате $d_{i,j(k)}$, где k определяет номер набора функции, а j – номер входа, который используется для ее подключения. В нашем случае k получается из j путем инвертирования старшей переменной. Таким образом, верхняя половина SRAM меняется местами с нижней половиной, чтобы использовать неактивную часть проходных транзисторов. В результате в данной конфигурации LUT трех

переменных получают активными две цепочки от SRAM до выходных инверторов, т.е. реализуются две функции одновременно.

Для дальнейшего увеличения количества одновременно выполняемых функций на LUT трех переменных дополнительно вводится дерево LUT двух переменных для каждой функции, входы которых подключены к выходам передающих транзисторов первого каскада. Также дополнительно вводится каскад передающих транзисторов настройки, которые управляются второй переменной (x_2). Выходы транзисторов объединены и подключены к входам первого каскада передающих транзисторов настройки. Общая схема четырехфункциональной LUT трех переменных изображена на рис. 3.

Для реализации четырех функций выполним декомпозицию логического устройства по старшей (x_3) и второй переменной (x_2):

Первая функция:

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

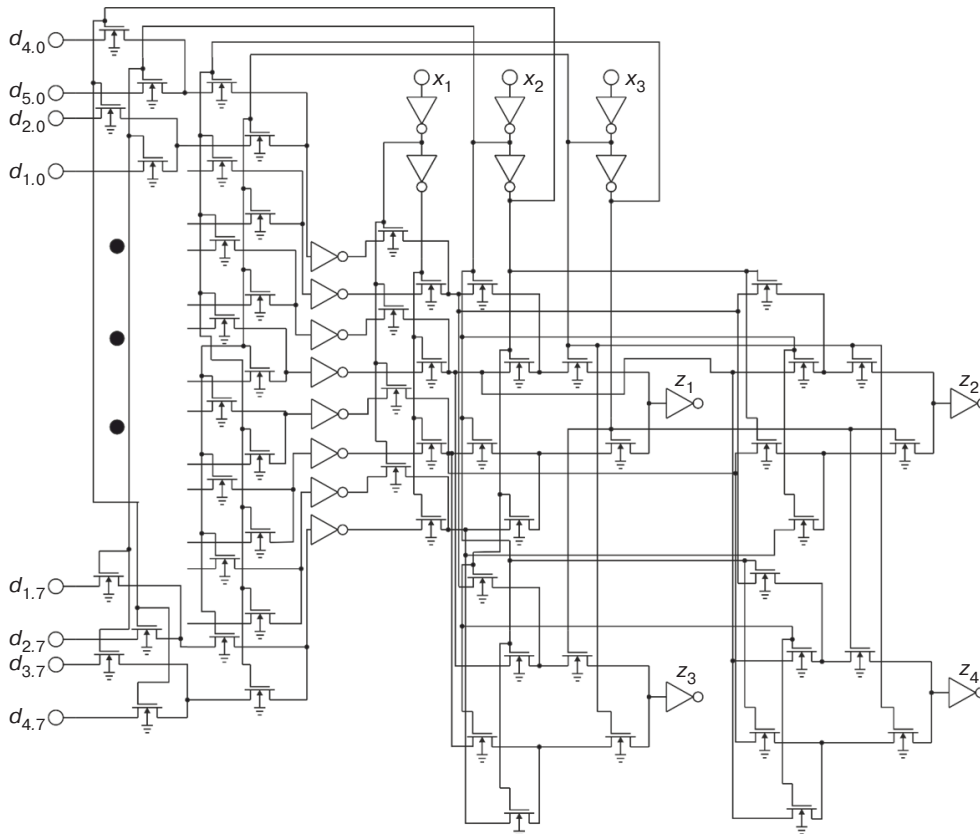


Рис. 3. Дерево LUT трех переменных, которая выполняет четыре функции одновременно

Таблица. Настройка логических уровней ячеек SRAM для LUT трех переменных, выполняющей четыре функции

Номер входа	Конфигурация для z_1	Конфигурация для z_2	Конфигурация для z_3	Конфигурация для 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

Вторая функция:

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

Третья функция:

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

Четвертая функция:

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

Полученные в результате декомпозиции конфигурационные уровни четырех функций: исключающего ИЛИ (z_1), мажоритарной (z_2), дизъюнкции (z_3) и конъюнкции (z_4) – указаны в таблице.

МОДЕЛИРОВАНИЕ LUT

Моделирование многофункциональной LUT трех переменных проводилось в системе динамического моделирования *Multisim*² компании National Instruments. Для NMOS передающих транзисторов была использована модель транзистора BSIM 4.8.0^{3, 4}. Инвертор был реализован на двух полевых транзисторах, его схема приведена на рис. 4.

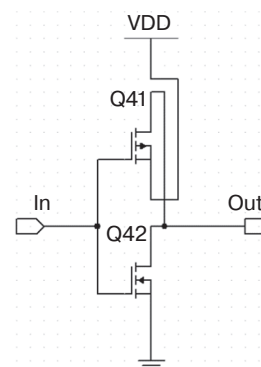


Рис. 4. Схема инвертора на полевых транзисторах.

Здесь и далее на рисунках используются обозначения, принятые в ГОСТ 2.710-81⁵

² <https://www.multisim.com/>. Дата обращения 17.11.2022. / Accessed November 17, 2022.

³ Microwind & Dsch Version 3.5. <https://www.yumpu.com/en/document/view/40386405/microwind-manual-lite-v35pdf-moodle>. Дата обращения 17.11.2022. / 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. Дата обращения 17.11.2022. / Accessed November 17, 2022.

⁵ ГОСТ 2.710-81. *Межгосударственный стандарт. Единая система конструкторской документации. Обозначения буквенно-цифровые в электрических схемах*. М.: Стандартинформ; 2008. [GOST 2.710-81. *Unified system for design documentation. Alpha-numerical designations in electrical diagrams*. Moscow: Standartinform; 2008 (in Russ.).]

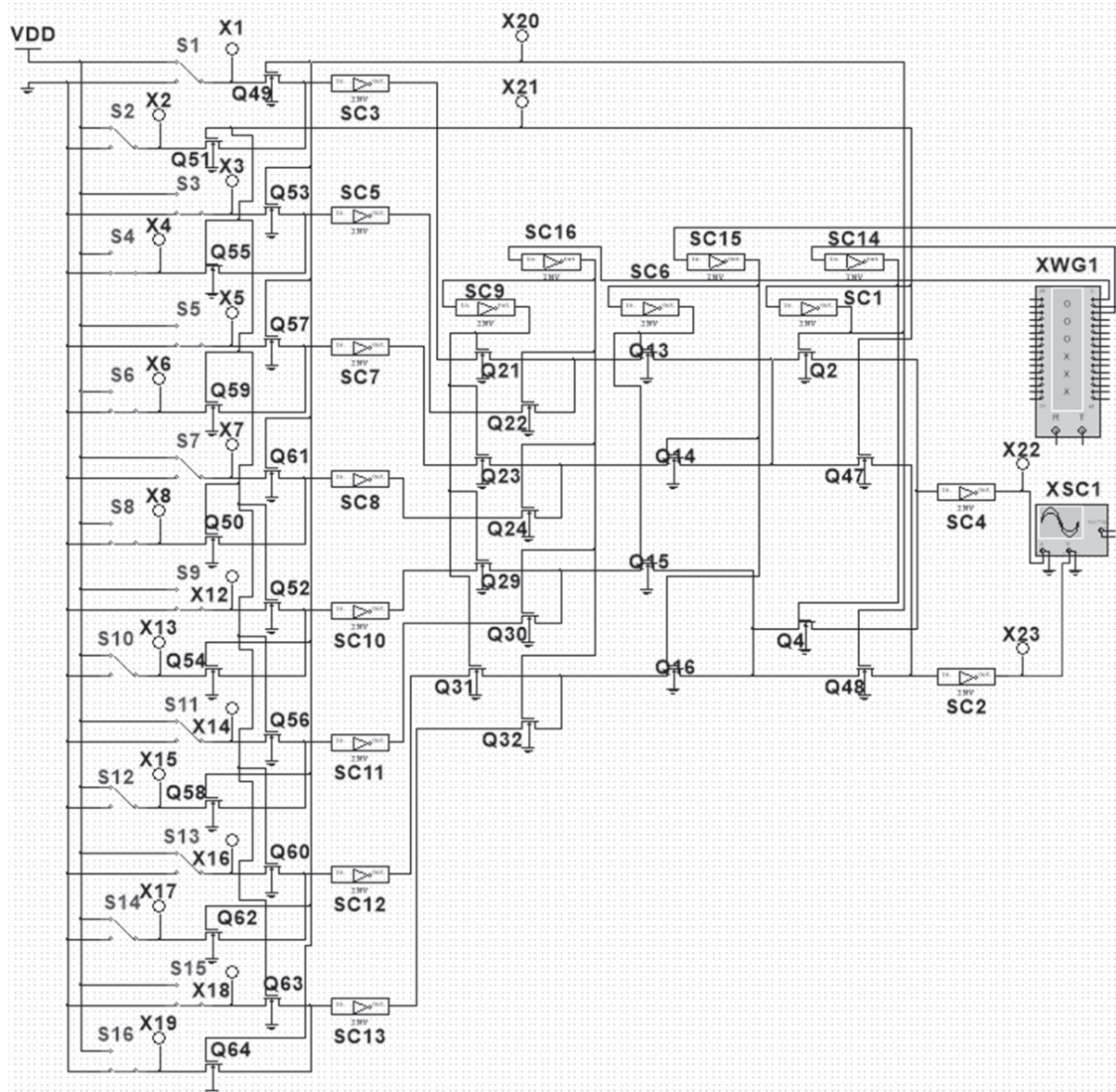


Рис. 5. Схема LUT трех переменных, выполняющая две функции одновременно

В первую очередь рассмотрим общую схему реализации LUT для двух функций (исключающего ИЛИ, мажоритарной), изображенную на рис. 5.

Модифицированная схема LUT трех переменных для выполнения четырех функций одновременно (исключающее ИЛИ, мажоритарная, дизъюнкция, конъюнкция) представлена на рис. 6 (см. с. 52).

Входной сигнал настройки ячеек SRAM задавался с помощью динамических переключателей согласно табл. 1. Высокий уровень напряжения (VDD 5V) соответствует логической единице, а низкий уровень напряжения (GND 0V) соответствует логическому нулю.

Сигналы переменных задавались с помощью *Word Generator*, в котором установлен трехбитный код Грея (рис. 7). Частота работы *Word Generator* составляет 1 кГц, таким образом, требуется 8 мс для полного прохождения заданного кода Грея.

Выходы функций подключены к осциллографу.

Результаты моделирования двух и четырех функций представлены на рис. 8 и 9 соответственно.

Верхняя осциллограмма соответствует функции «исключающее ИЛИ» (z_1). Нижняя осциллограмма соответствует мажоритарной функции (z_2). Каждое временное деление осциллографа соответствует одной комбинации кода Грея.

В отличие от базовой схемы LUT трех переменных, где используется только одна половина передающих транзисторов, в предлагаемой новой схеме дополнительно используется неактивная часть при значении 2^{n-1} переменных высокого порядка.

Сложность по количеству транзисторов LUT в зависимости от количества переменных n , $n \in \mathbb{N}$ рассчитывается следующим образом:

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

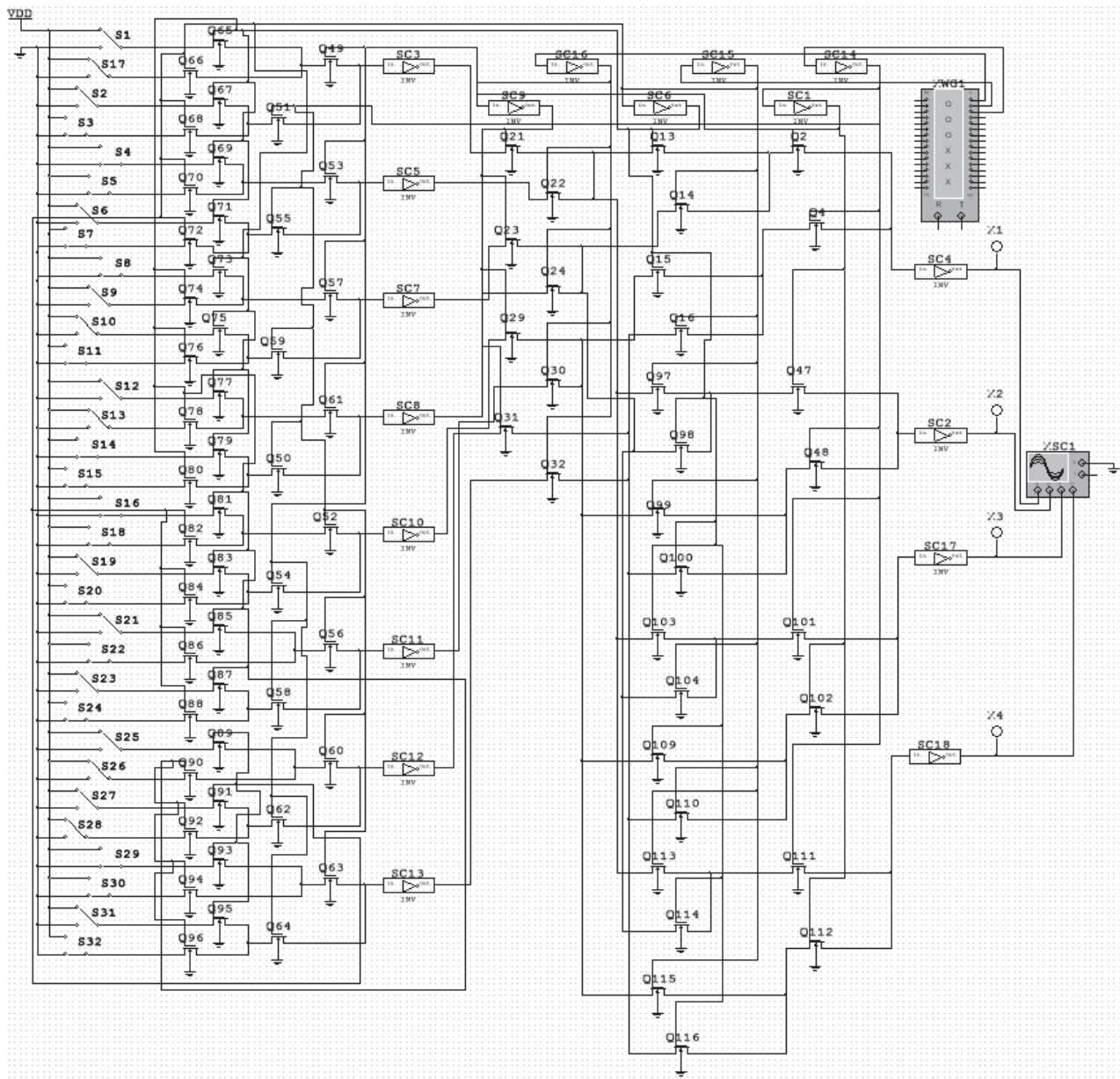


Рис. 6. Схема LUT трех переменных, выполняющая четыре функции одновременно

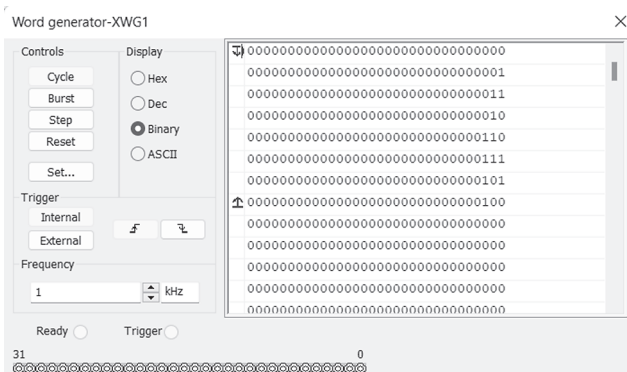


Рис. 7. Код Грея в Word Generator

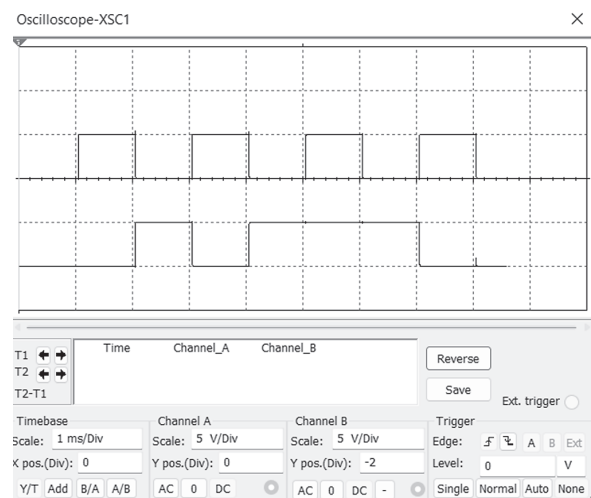


Рис. 8. Осциллограмма двух функций LUT трех переменных

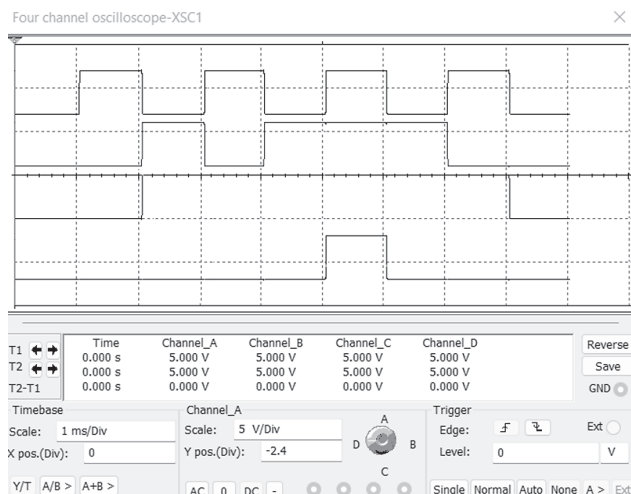


Рис. 9. Осциллограмма четырех функций LUT трех переменных

Для расчета 2^v функций в предложенной схеме получаем сложность:

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

Так, например, для реализации четырех логических функций ($v = 2$) одних и тех же переменных ($n = 4$) требуется две LUT при общем использовании 212 транзисторов, а в предлагаемой новой конструкции LUT, реализующей все четыре функции, количество необходимых транзисторов составляет 192.

Диаграммы сложности, представляющие обычную LUT $L_{v1}(v)$ и многофункциональную LUT $L_v(v)$ реализующую v логических функций с количеством переменных $n = 8$, показаны на рис. 10.

Как видно из диаграммы, имеется выигрыш в сложности по количеству транзисторов для многофункциональной LUT по отношению к соответствующему количеству для обычных LUT.

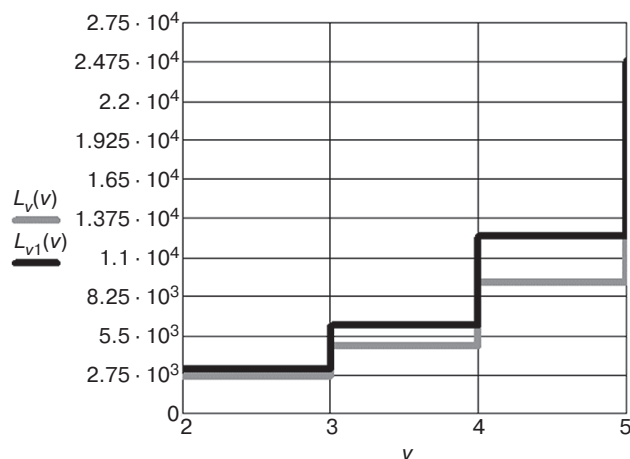


Рис. 10. Сравнение сложности обычной LUT и многофункциональной LUT

ЗАКЛЮЧЕНИЕ

Таким образом, в статье показана возможность одновременной реализации $2^n - 1$ логических функций на неактивных частях цепей передающих транзисторов в ПЛИС, что увеличивает функциональность устройства. Для реализации того же количества логических функций требуется $2^n - 2$ LUT. Несмотря на дополнительные затраты, существует выигрыш в сложности по сравнению с количеством необходимых транзисторов.

Вклад авторов

С.И. Советов – подготовка исследования, анализ литературы, проведение моделирования, написание текста статьи, формулировка выводов.

С.Ф. Тюрин – идея исследования, постановка целей и задач, консультации по вопросам исследования, редактирование статьи.

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.

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>

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 Radioelectronics, 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/ELConRusNW.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 (ELConRus)*. Moscow and St. Petersburg, Russia, 2018. P. 259–264. <https://doi.org/10.1109/ELConRus.2018.8317080>
5. Skorniyakova A.Yu., Vikhorev R.V. Self-timed LUT layout simulation. In: *2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ELConRus)*. St. Petersburg and Moscow, Russia, 2020. P. 176–179. <https://doi.org/10.1109/ELConRus49466.2020.9039374>
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; опублик. 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 (ELConRus)*. St. Petersburg and Moscow, Russia, 2020. P. 1787–1790. <https://doi.org/10.1109/ELConRus49466.2020.9039052>
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 Radioelectronics, 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/ELConRusNW.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 (ELConRus)*. Moscow and St. Petersburg, Russia, 2018. P. 259–264. <https://doi.org/10.1109/ELConRus.2018.8317080>
5. Skorniyakova A.Yu., Vikhorev R.V. Self-timed LUT layout simulation. In: *2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ELConRus)*. St. Petersburg and Moscow, Russia, 2020. P. 176–179. <https://doi.org/10.1109/ELConRus49466.2020.9039374>
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. Stroganov 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. Stroganov 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 (ELConRus)*. St. Petersburg and Moscow, Russia, 2020. P. 1787–1790. <https://doi.org/10.1109/ELConRus49466.2020.9039052>

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: tyurinsergeofeo@yandex.ru. Scopus Author ID 6603805561, <https://orcid.org/0000-0002-5707-5404>

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>