

УДК 681.32

**С.Ф. Тюрин<sup>1,2</sup>, М.А. Чудинов<sup>1,3</sup>**

<sup>1</sup>Пермский национальный исследовательский политехнический университет  
Пермь, Россия

<sup>2</sup>Пермский государственный национальный исследовательский университет  
Пермь, Россия

<sup>3</sup>ПАО «Морион», Пермь, Россия

## **FPGA LUT С ДВУМЯ ВЫХОДАМИ ДЕКОМПОЗИЦИИ ПО ШЕННОНУ**

В настоящее время количество логических элементов, программируемых логических интегральных схем (ПЛИС) типа FPGA (field-programmable gate array) достигает нескольких миллионов, что создает совершенно новые возможности при синтезе цифровой аппаратуры. При этом основой так называемых адаптивных логических модулей (АЛМ) FPGA являются деревья передающих транзисторов LUT (Look Up Table), вычисляющие логические функции в совершенной дизъюнктивной нормальной форме (СДНФ). При работе в арифметическом режиме вычисляются логические функции, отличающиеся значением одной переменной, например, переноса из разряда в разряд, что позволяет ускорить реализацию многоразрядной суммы. Для этого берут два LUT, выходы которых мультиплексируются по значению этой переменной, т.е. как бы вычисляют логические функции «впрок». Похожий принцип использован в архитектуре «Гиперфлекс» (HyperFlex), где разложение логической функции по Шеннону в цепи обратной связи позволяет обеспечить повышение быстродействия автомата с памятью. При этом используются две копии логической функции управления триггером, выбор которых также производится мультиплексором 2-1. Несмотря на отсутствие дефицита логических элементов в некоторых приложениях, например в отказоустойчивой аппаратуре, эти своего рода дублирующие элементы могли бы быть полезны, например, при построении резервированных структур. Поэтому предлагается реализовать декомпозицию по Шеннону на основе одного LUT, для чего дублируются только самые последние два транзистора соответствующего дерева с выходным инвертором, поскольку вычисляется та же самая логическая функция, но на наборе аргументов, отличающемся только одной переменной. В статье описывается предлагаемое техническое решение и оценивается выигрыш в количестве транзисторов по отношению к известному решению.

**Ключевые слова:** программируемая логическая интегральная схема, LUT, разложение Шеннона.

S.F. Tyurin<sup>1,2</sup>, M.A. Chudinov<sup>1,3</sup>

<sup>1</sup>Perm National Research Polytechnic University, Perm, Russian Federation

<sup>2</sup>Perm State National Research University, Perm, Russian Federation  
Morion PJSC, Perm, Russian Federation

## FPGA LUT WITH TWO SHANNON DECOMPOSITION OUTPUTS

At present, the number of logic elements, programmable logic integrated circuits of the FPGA (field-programmable gate array) type reaches several million, which creates completely new possibilities in the design of digital equipment. In this case, the basis of the so-called adaptive logic modules (ALMs) FPGA are the trees of transistors LUT (Look Up Table), which calculate logical functions in full disjunctive normal form (FDNF). When working in arithmetic mode, logical functions are calculated that differ in the value of one variable, for example, carry from discharge to discharge, which allows to speed up the implementation of a multi-digit adder. To do this, take two LUTs, whose outputs are multiplexed by the value of this variable, that is, how to calculate the logical functions "for future use". A similar principle is used in the HyperFlex architecture, where the Shannon decomposition (or Boolean factorization) of the logic function in the feedback loop allows for an increase in the speed of the state machine. Two copies of the flip-flop logic function are used, which are also selected by the 2-1 multiplexer. Despite the lack of shortage of logical elements in some applications, for example, in fault-tolerant equipment, these kind of duplicating elements could be useful, for example, when building redundant structures. Therefore, it is proposed to implement the Shannon decomposition on the basis of one LUT for which only the last two transistors of the corresponding tree with the output inverter are duplicated, since the same logical function is calculated, but on a set of arguments that differ only in one variable. The article describes the proposed technical solution and estimates the gain in the number of transistors in relation to the known solution.

**Keywords:** FPGA, LUT, Shannon decomposition or Boolean factorization.

**Введение.** Программируемые логические интегральные схемы (ПЛИС) являются одним из самых востребованных сегментов рынка элементной базы цифровой аппаратуры, выпускаемой с середины 80-х гг. XX в. [1–3]. К ПЛИС в настоящее время относят также и микросхемы систем на кристалле SoC (System-on-a-Chip) [4] и так называемые системы в пакете-SiP (System-in-Package), представляющие собой объёмные сборки разных микросхем [5, 6]. Имеются многочисленные модификации ПЛИС, но их делят на два основных класса: FPGA (field-programmable gate array), в которых логические функции реализуются в СДНФ (совершенной дизъюнктивной нормальной форме) в генераторах функций, представляющих собой дерево транзисторов (LUT-Look Up Table), CPLD (complex programmable logic devices), в которых вычисляются системы логических функций в дизъюнктивной нормальной форме (ДНФ) [7–11]. Так, на сайте фирмы «Интел» [12] указаны следующие продукты: Stratix10 (14 нм), Stratix V (28 нм), Arria 10 (20 нм), Arria V (28 нм), Cyclone10 (20 нм), Cyclone V (28 нм), MAX10 (55 нм), причём MAX 10 одновременно числится и по разряду

CPLDs (complex programmable logic devices), в который входят также и MAX V (возможно 90 нм), MAX II (скорее всего 0,15 мкм). По классу SoC проходят ПЛИС Stratix10, Arria 10, Arria V, Cyclone 10, Cyclone V. На сайте другой крупнейшей фирмы производителя ПЛИС—Xilinx [13] представлены микросхемы ПЛИС Spartan-6 (45 нм), Virtex-7, Kintex-7, Artix-7, Spartan-7 (28 нм), Virtex UltraScale, Kintex UltraScale (20 нм), Virtex UltraScale+, Kintex UltraScale+ (16 нм). Имеются так называемые адаптивные логические модули АЛМ [14], оптимально конфигурируемые под требуемое число переменных. Всё начиналось с реализации логических функций до 4 переменных, в АЛМ возможна реализация любых функций 7 переменных и некоторых функций 8 переменных. Использование технологии tri-gate [15–17] позволило достичь нового уровня быстродействия или энергоэффективности. Изобилие ресурсов привело к созданию резервированных структур ПЛИС [18]. В связи с этим вызывает интерес вопрос ускорения вычислений логических функций и автоматных отображений за счёт так называемой декомпозиции по Шеннону [19]. Дело в том, что наличие уже десятков миллионов логических элементов дает возможность комбинационной реализации многих алгоритмов, реализуемых ранее только последовательными автоматами. Рассмотрим особенности такой декомпозиции, применяемой в настоящее время, и предложим подход к её дальнейшему развитию.

### 1. Декомпозиция при выполнении арифметических операций.

Дизъюнктивное разложение Шеннона (булева факторизация) по некоторой  $i$ -й переменной [20] позволяет разложить логическую (булеву) функцию  $f$  на две подфункции в виде дизъюнктивной нормальной формы (ДНФ), в одной из которых  $i$ -я переменная равна 0, а в другой – 1. Пусть эта переменная  $x$ , которая может быть в выражении  $f$  как без инверсии, так и с инверсией, так же, как и другие переменные  $y, z, \dots, w$ , тогда разложение Шеннона по  $x$  имеет вид:

$$f[(x, \bar{x}), \tilde{y}, \tilde{z}, \dots, \tilde{w}] = x \cdot f[(1, 0), \tilde{y}, \tilde{z}, \dots, \tilde{w}] \vee \bar{x} \cdot f[(0, 1), \tilde{y}, \tilde{z}, \dots, \tilde{w}]. \quad (1)$$

Конъюнктивное разложение Шеннона (используется конъюнктивная нормальная форма (КНФ)) по  $x$  имеет вид:

$$f[(x, \bar{x}), \tilde{y}, \tilde{z}, \dots, \tilde{w}] = \{x \vee f[(0, 1), \tilde{y}, \tilde{z}, \dots, \tilde{w}]\} \{ \bar{x} \vee f[(1, 0), \tilde{y}, \tilde{z}, \dots, \tilde{w}] \}. \quad (2)$$

Разложение Шеннона функций небольшого числа переменных удобно выполнять по таблице истинности. Например, проанализируем функции полного однобитного сумматора (рис. 1).

Функция суммы  $\text{Sum}(C_{in}AB)$  при  $C_{in} = 0$  (верхняя часть таблицы) становится суммой по модулю два  $\text{Sum}(C_{in} = 0, AB) = A \oplus B$ , а при  $C_{in} = 1$  (нижняя часть таблицы) – эквивалентией  $\text{Sum}(C_{in} = 1, AB) = A \leftrightarrow B$ . Функция переноса  $C_{out}(C_{in}AB)$  при  $C_{in} = 0$  – это конъюнкция  $C_{out}(C_{in}AB) = AB$ , а при  $C_{in} = 1$  – дизъюнкция  $C_{out}(C_{in}AB) = A \vee B$ . Таким образом, при изменении данных (Date 1 = A, Date 2 = B) подфункции вычисляются без учёта переноса, который используется для выбора одной из двух подфункций.

Логический элемент ПЛИС содержит генераторы функций LUT-2 (Look Up Table) и мультиплексы 2-1 (рис. 2).

Carry-in	Date 1 <b>A</b>	Date 2 <b>B</b>	BC	Carry-Out (CinAB)	Sum (CinAB)
0	0	0	0	0	0
0	0	1	1	0	1
0	1	0	2	0	1
0	1	1	3	1	0
1	0	0	4	0	1
1	0	1	5	1	0
1	1	0	6	1	0
1	1	1	7	1	1

Рис. 1. Дизъюнктивное разложение Шеннона функций полного сумматора

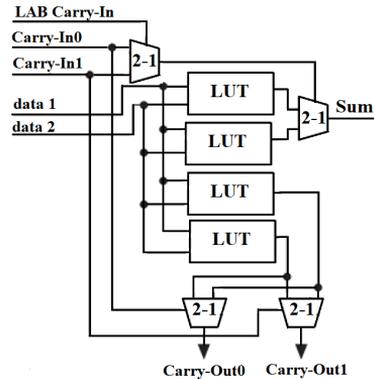


Рис. 2. Использование декомпозиции в логическом элементе при динамическом арифметическом режиме (LEin Dynamic Arithmetic Mode)

Мультиплексы на основе передающих транзисторов (Pass Transistors) 2-1, по существу, представляют собой LUT-1 на одну переменную без настройки входов данных. Следовательно, фактически происходит одновременная реализация двух подфункций суммы и двух подфункций переноса, зависящих от Date 1(A), Date 2 (B), которые затем выбираются старшей переменной Carry-In 0, Carry-In 1.

**2. Декомпозиция при реализации последовательного автомата.** При реализации последовательного автомата (автомата с памятью) для увеличения максимальной частоты реализации

автоматных отображений за счет разложения Шеннона «укорачивается» петля обратной связи [19]. На рис. 3 показана «длинная» обратная связь, реализующая некоторую функцию переходов:

$$y(t+1) = d(t) = ABCy_1. \quad (3)$$

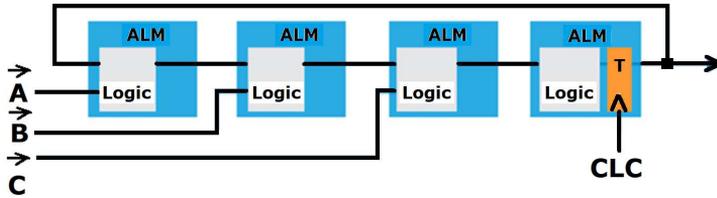


Рис. 3. «Длинная» обратная связь

$A, B, C$  – это векторы входных переменных соответствующего автомата. Здесь указан синхронный триггер (T, flip-flop) на выходе реконфигурируемого LUT, входящего в состав так называемого адаптивного логического модуля ALM. Частота синхронизации CLC рассчитывается с учетом задержки в петле обратной связи на трёх последовательных ALM и логике четвертого ALM, поэтому суммарная задержка относительно велика, и она включает задержку в элементах маршрутизации. Эта обратная связь «длинная», потому что проходит через матрицу локальных связей, где возможны значительные задержки сигналов. Одна переменная  $A, B$  или  $C$  для ALM, реализующего функции 4 и даже в ряде случаев 7 и 8 переменных, – не показательный случай, поэтому пусть это будут некие векторы:

$$y(t+1) = d(t) = \overline{\overline{\overline{ABC}}}y_1, \quad (4)$$

например,

$$A = a_1a_2a_3; B = b_1b_2b_3; C = c_1c_2c_3. \quad (5)$$

Допустим, имеем такую функцию переходов:

$$y_1(t+1) = d(t) = [a_1a_2a_3\overline{y_1}(t) \vee \overline{a_1}\overline{a_2}\overline{a_3}y_1(t)] \vee b_1b_2b_3 \vee c_1c_2c_3. \quad (6)$$

Выполняем разложение Шеннона (Shannon decomposition or Boolean factorization), получаем:

$$y_1 = 0: a_1a_2a_3 \vee b_1b_2b_3 \vee c_1c_2c_3. \quad (7)$$

$$y_1 = 1: \overline{a_1}\overline{a_2}\overline{a_3} \vee b_1b_2b_3 \vee c_1c_2c_3. \quad (8)$$

На рис. 4 показана реализация двух этих функций в разных LUT ALM. При этом число переменных уменьшено на одну за счет исключения переменной состояния триггера  $y$ .

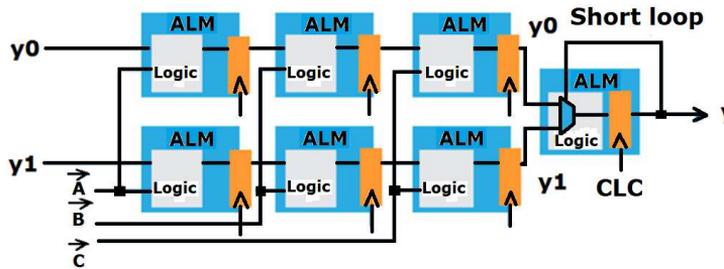


Рис. 4. «Короткая» обратная связь при введении разложения Шеннона

Поэтому петля обратной связи короткая, и задержка определяется только задержкой логики последнего ALM. Такой подход увеличивает аппаратные затраты, но они несущественны для ПЛИС, имеющей уже десятки миллионов таких логических элементов, здесь главное – скорость. Все это называется гипероптимизацией. Поиск возможностей такой оптимизации осуществляется гиперретаймингом. Дальнейшее продвижение этого направления может привести к тому, что такое разложение будет выполнено по всем переменным.

**3. Логический элемент, реализующий декомпозицию по старшей переменной.** Развитие направления декомпозиции по Шеннону может привести к тому, что задержка на вычисление логических функций будет сведена к минимуму, который определяется выбором одной из  $2^n = i$  конституент всего одним транзистором однорангового дерева. В этом случае целесообразно использовать вектор входного набора в унитарном коде:

$$0000\dots 0x^i 0\dots 000000. \quad (9)$$

В выражении (9) позиция  $x^i = 1$  в одном из  $2^n$  разрядов определяет входной вектор функции (рис. 5).

Для уменьшения затрат на реализацию таких «декомпозированных» функций большого числа переменных предлагается использовать модифицированный унитарный код, включающий несколько бит, например три:

$$0000\dots x^i_3 x^i_2 x^i_1 000000. \quad (10)$$

Тогда, например,  $x_3 = 1$  определяет группу из двух переменных  $x_2 x_1$ , которые уже подлежат дешифрации (рис. 6).

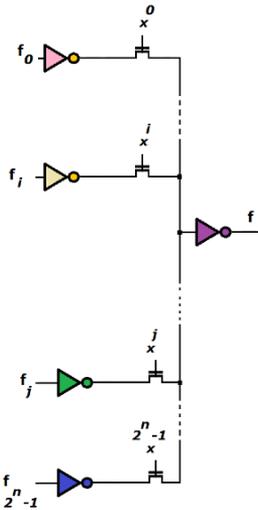


Рис. 5. Выбор одного из  $2^n$  значений функции  $f$  унитарным кодом входного вектора  $0000\dots 0x^i 0\dots 000000$

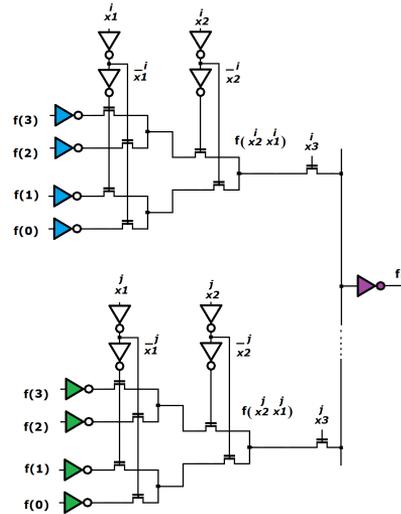


Рис. 6. Выбор одного из  $2^2 \cdot 2^{n-2}$  значений функции  $f$  унитарным кодом входного вектора  $0000\dots x_3^j x_2^j x_1^j 000000$

Возникает задача нахождения оптимальной декомпозиции по Шеннону – по нескольким переменным, так, чтобы и трассировка переменных, и LUT были не слишком сложными, с одной стороны, и задержка на вычисление логической функции не слишком увеличилась, с другой.

Предлагается также LUT с двумя выходами декомпозиции по старшей переменной (рис. 7).

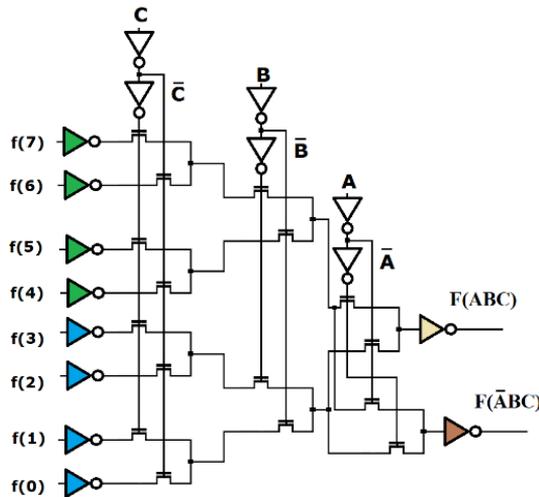


Рис. 7. LUT-3 с двумя предлагаемыми выходами декомпозиции по старшей переменной  $A$ ,  $F(ABC)$  – существующий выход

Функционирование предлагаемого LUT-3 в зависимости от значения старшей переменной показано на рис. 8.

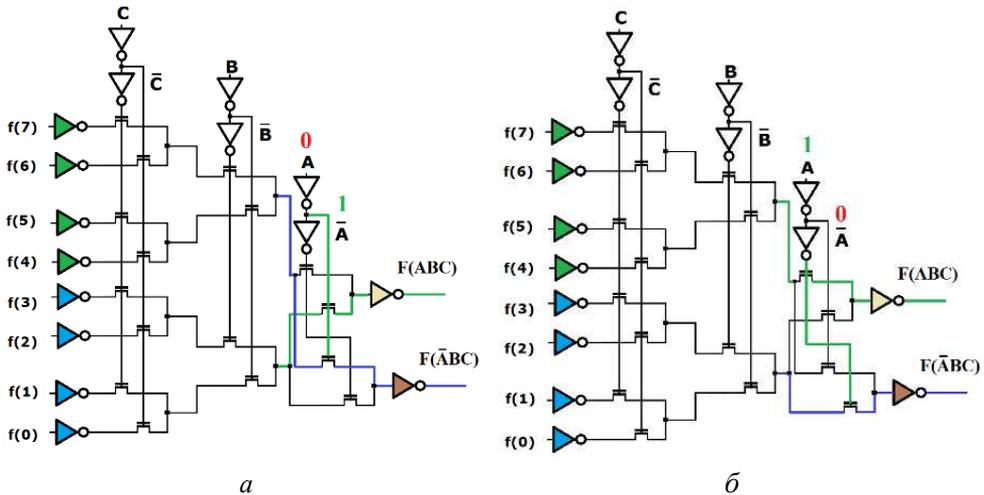


Рис. 8. Функционирование предлагаемого LUT-3:  
а – старшая переменная  $A = 0$ ; б – старшая переменная  $A = 1$

В этом случае создается возможность контроля функционирования LUT путём тестирования с использованием математического аппарата булевых производных, описанного, например, в [20]. Для этого необходимо создать тест, т.е. такие функции, в которых изменение старшей переменной приводит к разным значениям подфункций.

**Выводы.** Таким образом, для обеспечения быстродействия ПЛИС разработчики передовых фирм идут по пути усложнения проектов с использованием разложения Шеннона, что позволяет, кроме прочего, уменьшить длину обратной связи в последовательном автомате.

Следует ожидать увеличение возможностей комбинационных реализаций схем конечных автоматов, традиционно реализуемых последовательно в силу имеющихся ранее ограничений на количество логических элементов и объём памяти.

В связи с этим предлагается унитарный LUT, когда активна только одна входная переменная. Похожий принцип используется при маршрутизации связей в ПЛИС. Например, условия для этого создаст унитарное кодирование памяти автомата. Такой LUT будет обладать максимальным быстродействием. Для снижения аппаратных затрат возможно комбинированное кодирование входного вектора, что, конечно, ухудшит быстродействие вычисления значений логических функций.

С целью упрощения контроля таких вычислений предложен LUT с двумя выходами декомпозиции по старшей переменной. Использование модифицированного LUT позволит уменьшить время диагностирования. Целью дальнейших исследований может быть рассмотрение предложенного LUT с встроенным элементом сложения по модулю два.

### Библиографический список

1. Угрюмов Е.П. Цифровая схемотехника: учеб. пособие. – 2-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2007. – 782 с.
2. Современные реализации ПЛИС [Электронный ресурс]. – URL: <http://fpga.parallel.ru/devices.html> (дата обращения: 31.10.2018).
3. Строгонов А., Цыбин С. Программируемая коммутация ПЛИС: взгляд изнутри [Электронный ресурс]. – URL: [http://www.kit-e.ru/articles/plis/2010\\_11\\_56.php](http://www.kit-e.ru/articles/plis/2010_11_56.php) (дата обращения: 11.10.2018).
4. Intel SoC FPGAs [Электронный ресурс]. – URL: <https://www.intel.com/content/www/us/en/products/programmable/soc.html> (дата обращения: 31.10.2018).
5. Systemin Package [Электронный ресурс]. – URL: <https://amkor.com/technology/system-in-package/> (дата обращения: 31.10.2018).
6. SiP Products [Электронный ресурс]. – URL: <https://www.altera.com/products/sip/overview.html> (дата обращения: 11.10.2018).
7. Виды программируемой логики [Электронный ресурс]. – URL: <http://www.pvsm.ru/programmirovanie/87810> (дата обращения: 10.10.2018).
8. Programmable Logic Devices [Электронный ресурс]. – URL: [http://ee.sharif.edu/~logic\\_circuits\\_t/readings/PLD.pdf](http://ee.sharif.edu/~logic_circuits_t/readings/PLD.pdf) (дата обращения: 04.11.2018).
9. Программируемая логика и её применение в микропроцессорных системах [Электронный ресурс]. – URL: <http://lektsii.org/7-10275.html> (дата обращения: 08.11.2018).
10. CPLD (Complex Programmable Logic Device) [Электронный ресурс]. – URL: <http://www.myshared.ru/slide/981511/> (дата обращения: 09.11.2018).
11. Stephen Brown, Jonathan Rose. Architecture of FPGAs and CPLDs: A Tutorial [Электронный ресурс]. – URL: <http://www.eecg.toronto.edu/~jayar/pubs/brown/survey.pdf> (дата обращения: 10.10.2018).
12. INTEL FPGA [Электронный ресурс]. – URL: <https://www.altera.com/> (дата обращения: 23.10.2018).
13. Xilinx [Электронный ресурс]. – URL: <https://www.xilinx.com/products/silicon-devices/fpga.html> (дата обращения: 31.10.2018).

14. Logic Array Blocks and Adaptive Logic Modules in Stratix III Devices [Электронный ресурс]. – URL: [https://www.altera.com.cn/content/dam/alterawww/global/zh\\_CN/pdfs/literature/hb/stx3/stx3\\_siii51002.pdf](https://www.altera.com.cn/content/dam/alterawww/global/zh_CN/pdfs/literature/hb/stx3/stx3_siii51002.pdf) (дата обращения: 20.10.2018).

15. Ryan Kenny, Jeff Watt. The Breakthrough Advantage for FPGAs with Tri-Gate Technology [Электронный ресурс]. – URL: [https://www.altera.com/en\\_US/pdfs/literature/wp/wp-01201-fpga-tri-gate-technology.pdf](https://www.altera.com/en_US/pdfs/literature/wp/wp-01201-fpga-tri-gate-technology.pdf) (дата обращения: 12.10.2018).

16. Трёхмерные транзисторы 22нм [Электронный ресурс]. – URL: <https://habrahabr.ru/company/intel/blog/118816/> (дата обращения: 15.10.2018).

17. Интегрированные транзисторы CMOS tri-gate [Электронный ресурс]. – URL: <http://compress.ru/article.aspx?id=16789> (дата обращения: 24.10.2018).

18. Carl Carmichael. Triple Module Redundancy Design Techniques for Virtex FPGAs [Электронный ресурс]. – URL: [https://www.xilinx.com/support/documentation/application\\_notes/xapp197.pdf](https://www.xilinx.com/support/documentation/application_notes/xapp197.pdf) (дата обращения: 07.11.2018).

19. Understanding How the New Intel®HyperFlex™ FPGA Architecture Enables Next Generation High-Performance Systems [Электронный ресурс]. – URL: <https://www.altera.com/products/fpga/stratix-series/stratix10/features.html#hyperflexarchitecture> (дата обращения: 27.10.2018).

20. Тюрин С.Ф., Аляев Ю.А. Дискретная математика: практическая дискретная математика и математическая логика. – М.: Финансы и статистика, 2010. – 394 с.

## References

1. Ugriumov E.P. Tsifrovaiia skhemotekhnika [Digitalcircuitry: atutorial]. 2nd ed. Saint Petersburg: BKhV-Peterburg, 2007. 782 p.

2. Sovremennye realizatsii PLIS [Modern FPGA implementation], available at: <http://fpga.parallel.ru/devices.html> (accessed 31 October 2018).

3. Strogonov A., Tsybin S. Programmiruemaia kommutatsiia PLIS: vzgliad iznutri [Programmable switching FPGA: a view from the inside], available at: [http://www.kit-e.ru/articles/plis/2010\\_11\\_56.php](http://www.kit-e.ru/articles/plis/2010_11_56.php) (accessed 11 October 2018).

4. Intel SoC FPGAs, available at: <https://www.intel.com/content/www/us/en/products/programmable/soc.html> (accessed 31 October 2018).

5. Systemin Package, available at: <https://amkor.com/technology/system-in-package/> (accessed 31 October 2018).

6. SiP Products, available at: <https://www.altera.com/products/sip/overview.html> (accessed 11 October 2018).

7. Vidy programmiruemoi logiki [Types of programmable logic], available at: <http://www.pvsm.ru/programmirovanie/87810> (accessed 10 October 2018).

8. Programmable Logic Devices, available at: [http://ee.sharif.edu/~logic\\_circuits\\_t/readings/PLD.pdf](http://ee.sharif.edu/~logic_circuits_t/readings/PLD.pdf) (accessed 04 November 2018).

9. Программируемая логика и её применение в микропроцессорных системах [Programmable logic and its application in microprocessor systems], available at: <http://lektsii.org/7-10275.html> (accessed 08 November 2018).

10. CPLD (Complex Programmable Logic Device), available at: <http://www.myshared.ru/slide/981511/> (accessed 09 November 2018).

11. Stephen Brown, Jonathan Rose. Architecture of FPGAs and CPLDs: A Tutorial, available at: <http://www.eecg.toronto.edu/~jayar/pubs/brown/survey.pdf> (accessed 10 October 2018).

12. INTEL FPGA, available at: <https://www.altera.com/> (accessed 23 October 2018).

13. Xilinx, available at: <https://www.xilinx.com/products/silicon-devices/fpga.html> (accessed 31 October 2018).

14. Logic Array Blocks and Adaptive Logic Modules in Stratix III Devices, available at: [https://www.altera.com.cn/content/dam/alterawww/global/zh\\_CN/pdfs/literature/hb/stx3/stx3\\_siii51002.pdf](https://www.altera.com.cn/content/dam/alterawww/global/zh_CN/pdfs/literature/hb/stx3/stx3_siii51002.pdf) (accessed 20 October 2018).

15. Ryan Kenny, Jeff Watt. The Breakthrough Advantage for FPGAs with Tri-Gate Technology, available at: [https://www.altera.com/en\\_US/pdfs/literature/wp/wp-01201-fpga-tri-gate-technology.pdf](https://www.altera.com/en_US/pdfs/literature/wp/wp-01201-fpga-tri-gate-technology.pdf) (accessed 12 October 2018).

16. Trekhmernye tranzistory 22nm [Three-dimensional transistors 22nm], available at: <https://habrahabr.ru/company/intel/blog/118816/> (accessed 15 October 2018).

17. Integrirovannye tranzistory CMOS tri-gate [Integrated CMOS transistors], available at: <http://compress.ru/article.aspx?id=16789> (accessed 24 October 2018).

18. Carl Carmichael. Triple Module Redundancy Design Techniques for Virtex FPGAs, available at: [https://www.xilinx.com/support/documentation/application\\_notes/xapp197.pdf](https://www.xilinx.com/support/documentation/application_notes/xapp197.pdf) (accessed 07 November 2018).

19. Understanding How the New Intel HyperFlex FPGA Architecture Enables Next Generation High-Performance Systems, available at:

<https://www.altera.com/products/fpga/stratix-series/stratix10/features.html#hyperflexarchitecture> (accessed 27 October 2018).

20. Tiurin S.F. Aliaev Iu.A. Diskretnaia matematika: prakticheskaia diskretnaia matematika i matematicheskaia logika [Discrete mathematics: practical discrete mathematics and mathematical logic]. Moscow: Finansy i statistika, 2010. 394 p.

### **Сведения об авторах**

**Тюрин Сергей Феофентович** (Пермь, Россия) – заслуженный изобретатель Российской Федерации, доктор технических наук, профессор кафедры «Автоматика и телемеханика» Пермского национального исследовательского политехнического университета (614990, Пермь, Комсомольский пр., 29, e-mail: tyurinsergfeo@yandex.ru), профессор кафедры «Математическое обеспечение вычислительных систем» Пермского государственного национального исследовательского университета (614990, Пермь, ул. Букирева, 15).

**Чудинов Максим Андреевич** (Пермь, Россия) – аспирант кафедры «Автоматика и телемеханика» Пермского национального исследовательского политехнического университета (614990, Пермь, Комсомольский пр., 29, e-mail: m-chudinov@list.ru), инженер направления мультисервисного доступа ПАО «Морион» (614066, Пермь, шоссе Космонавтов, 111).

### **About the authors**

**Tyurin Sergey Feofentovich** (Perm, Russian Federation) is a Honored Inventor of the Russian Federation, Doctor of Technical Sciences, Professor at the Department of Automation and Telemechanics Perm National Research Polytechnic University (614990, Perm, 29, Komsomolsky pr., e-mail: tyurinsergfeo@yandex.ru), Professor at the Department of Software Computing Systems Perm State National Research University (614990, Perm 15, Bukireva str.).

**Chudinov Maksim Andreevich** (Perm, Russian Federation) is a Graduate Student the Department of Automation and Telemechanics Perm National Research Polytechnic University (614990, Perm, 29, Komsomolsky pr., e-mail: m-chudinov@list.ru), Engineer at Department of multiservice access, PJSC "Morion" (614066, Perm, 111, highway Kosmonavtov).

Получено 17.01.2019