

<sup>1</sup>Dr.Sc., Professor, Professor of department of automatics and telemechanics, Perm National Research Polytechnic University, Perm, Russia; Professor of department of mathematical support of computer systems, Perm State National Research University, Perm, Russia

<sup>2</sup>PhD, Associate professor of department of software computer technology and automated systems, Perm Military Institute of National Guard Troops of the Russian Federation, Perm, Russia

## STUDY OF THE MULTI-INPUT LUT COMPLEXITY

**Context.** The programmable logic integrated circuits FPGA (field-programmable gate array) used realization of the generator of functions LUT (Look Up Table), which is configured by loading a configuration memory for calculating a logic function in perfect disjunctive normal form (PDNF). The LUT dimension determines the technological limitations of Mead and Conway on the number of series-connected MOS transistors. The standard number of LUT inputs for many years was 3 or 4, and 4-LUT is constructed from two 3-LUTs with an additional 1-LUT. However, in many projects, it is required to calculate functions of a large number of arguments. This requires a multi-input LUT, which is built as a decomposition of 3-LUT, 4-LUT. The speed of computing logic functions determines by the delay in the coupling matrices, so this decomposition leads to a decrease in performance. In recent years, the direction of adaptive logical modules (ALM) has been actively developing, in which the user has access to various versions of logical elements for five, six and even seven, eight variables, which leads to an increase in performance. However, the manufacturer's documentation does not provide a detailed description of the features of such multi-input LUTs, taking into account the Meade-Conway constraints. In addition, there are no estimates of complexity and speed of multi-input LUTs. The analysis of sources allows suggests a further increase in the LUT bit capacity and the convergence of FPGA and CPLD (complex programmable logic devices) capabilities in terms of bit depth. Therefore, studies of the features of constructing multi-input LUTs are relevant and the authors attempted to analyze the implementation of such prospective multi-bit logic

**Objective.** The purpose of this work is to estimate the complexity and speed of the decomposition of a multi-bit LUT.

**Method.** Obtaining expressions for estimating the complexity and speed of decomposition of a multi-bit LUT on a LUT of a lower bit length.

**Results.** A comparison of the complexity and delay in the number of transistors in the decomposition of a multi-bit LUT in the computer mathematics system Mathcad is performed.

**Conclusions.** The conducted researches made it possible to establish the features of constructing multi-bit LUTs and to evaluate various variants of decomposition with further increase in the LUT dimension with the subsequent choice of the optimal ALM variant.

**Keywords:** logic element, FPGA, LUT, transistor, adaptive logic module, decomposition, complexity, speed.

### NOMENCLATURE

ALM – adaptive logic module;

FPGA – field programmable gate array;

LAB – physically grouped set of logical resources Logic Array Block;

LUT – the lookup table;

SRAM – static memory with random access;

LE – logical element;

RAM – random access memory;

ROM – read-only memory;

PDNF – perfect disjunctive normal form;

$k$  – the dimension of the basic LUT;

$n$  – amount of elements;

$x_4, x_3, x_2, x_1$  – input variables.

### INTRODUCTION

LE of programmable logic integrated circuits of FPGA [1–3] type (field-programmable gate array) are ROM permanent memory devices (often called LUT-Look Up Table) implemented on a multiplexer whose data inputs are adjusted by constants. To configure a given logical function in RAM cells (SRAM), the corresponding truth table is loaded. When one of the  $2^n$  paths in the transistor tree is activated by variables, the value of the logic function is read from the corresponding RAM cell and transmitted to the OUT output. Variable inverters ensure the realization of all members of a PDNF.

The optimal speed and complexity of representing typical logic functions is the use of LUT in four variables (4-LUT). Such LUT for the input variables  $x_4, x_3, x_2, x_1$  (setting is 16 bits) is described by the expression:

$$z_{OUT}(x_4x_3x_2x_1) = \overline{ax_4x_3x_2x_1} \vee \overline{bx_4x_3x_2x_1} \vee \overline{cx_4x_3x_2x_1} \vee \overline{dx_4x_3x_2x_1} \vee \overline{ex_4x_3x_2x_1} \vee \overline{fx_4x_3x_2x_1} \vee \overline{gx_4x_3x_2x_1} \vee \overline{hx_4x_3x_2x_1} \vee \overline{ix_4x_3x_2x_1} \vee \overline{jx_4x_3x_2x_1} \vee \overline{kx_4x_3x_2x_1} \vee \overline{lx_4x_3x_2x_1} \vee \overline{mx_4x_3x_2x_1} \vee \overline{nx_4x_3x_2x_1} \vee \overline{ox_4x_3x_2x_1} \vee \overline{px_4x_3x_2x_1}. \quad (1)$$

### 1 PROBLEM STATEMENT

Given: adaptive logic modules FPGA Stratix III in seven variables [4, 5].

In the literature [6–8], the problems of decomposition of multi-bit LUT are not fully covered.

It is required: to assess the complexity and speed of the decomposition of a multi-bit LUT in order to identify features of the construction of adaptive logic modules and the prospects for further increasing the bit capacity.

### 2 REVIEW OF THE LITERATURE

Stratix III FPGAs have adaptive (ALM) logical blocks that are combined into logical blocks (LAB) [4], which implement functions of even seven variables. The peculiarities of the implementation of such LUTs are of interest. The fact is that due to the limitations of Meade and Conway on the number of consecutively connected transistors [5], the tree of transmitting transistors can not contain more than four transistors in the chain. It is necessary to decompose the multi-bit LUT into LUTs of lesser length, that is, to construct a tree from the subtrees.

FPGA Stratix III is described in a sufficient number of sources [6–8]. The structure of such FPGAs includes the so-called logic array blocks containing ALM, which can be configured to implement combinational logic, including arithmetic operations, as well as for the implementation of automata with memory.

The ALM architecture is compatible with the architecture of the 4-input LUTs, and one ALM can also implement any functions up to six variables and certain functions of seven variables. It is noted that such architecture wins on speed and efficiency (probably, it is a question of hardware expenses and the area of a crystal) – Fig. 1 [4].

In Fig. 1 indicates eight inputs of the adaptive LUT, which may give the impression of the possibility of implementing the 8-LUT. Even more confusing is the information contained in the presentation [9], where it is indicated that for the implementation of  $k$ -LUT,  $2k$  bits of SRAM and a multiplexer are also needed  $2k:1$ , but this is impossible. Different modes of using ALM do not clarify the details (Fig. 2) [4].

Let's consider the primary source – the documentation on FPGA Stratix III [4], where the details of ALM are shown (Fig. 3).

Thus, it turns out that ALM is built not only on two 4-LUTs, but there are four LUTs in 3 variables (3-LUT), that so, from two 3-LUTs we can get one 4-LUT. Therefore, there are only four 4-LUTs, then it becomes clear how the 6-LUT is constructed – the two older variables  $e, f$  choose one of the four. In Fig. 5 control signals are not indicated on a number of multiplexers designated by trapezoids (LUT 1-6 are also multiplexers, but they are shown with control signals, the setting is implied).

### 3 MATERIALS AND METHODS

Let  $k$  be the dimension of the basic LUT ( $k \in \{1, 2, 3, 4\}$ ).

For 1-LUT in principle up to  $n=4$  there is no need for an output inverter. More than 4 for the indicated restrictions  $k$  at the moment is not practiced.

Let's estimate the complexity of LUT without

decomposition (“ideal” complexity, since this can only be up to  $n=4$ , no more):

$$L_n = 2^n \cdot 8 + 2^{n+1} + 2n, \quad (2)$$

where  $2^n \cdot 8$  is the number of tuning elements (six SRAM transistors and two transistors are needed for each input of the tuning to implement the inverter at the input of the transistor tree);  $2^n$  – the number of inverters in  $n$  variables;  $2^{n+1}$  – number of elements of the tree of transmitting transistors with the output inverter.

When decomposing an  $n$ -tree with  $k$  LUT,  $k \in \{1, 2, 3, 4\}$ ,  $n \geq k, n \leq 8$ :

$$L_{n,k} = 2^n \cdot 8 + (2^{k+1} + 2k) \cdot 2^{n-k} + (2^{2^{n-k}+1} + 2^{n-k+1}) + 2n, \quad (3)$$

where  $2^{k+1}$  is the complexity of the tree  $k$  LUT;  $2^k$  is the number of transistors in  $k$  inverters,  $2^{n-k}$  need these trees, more LUTs for  $2^{n-k}$  inputs (which can also be decomposed) are needed to connect the trees obtained with decomposition of  $2^{n-k}$  trees, respectively complexity  $2^{n-k+1} + 2 \cdot 2^{n-k} = 2^{n-k+2}$ , where  $2^{n-k+1}$  is the complexity

of the tree with the output inverter,  $2 \cdot 2^{n-k} = 2^{n-k+1}$  – complexity of input inverters. The time delay in the decomposition is estimated by the length of the maximum path in the logical element from the input to the output. At the same time, without decomposition – with the “ideal” version (Figure 2) we get:

$$T_n = n + 2. \quad (4)$$

The path for decomposition in the transmitting transistors is also estimated by the value  $n$ , but due to additional inverters at the input and output in the LUT chain (Fig. 3, 4), it will be larger:

$$T_{n,k} = n + 2 \left\lceil \frac{n}{k} \right\rceil. \quad (5)$$

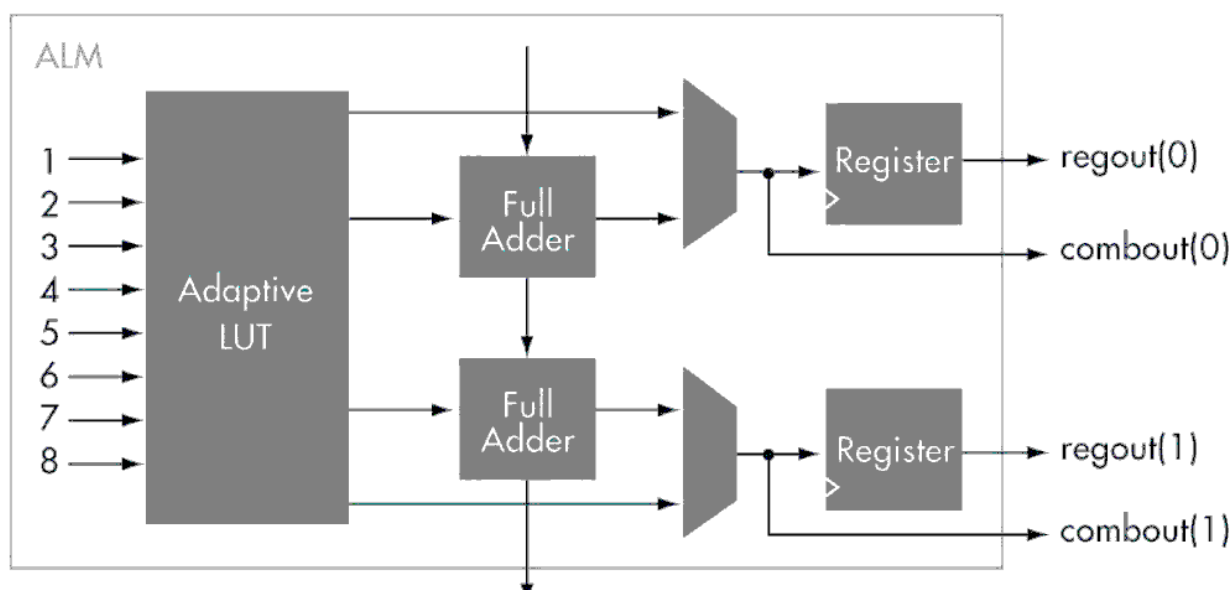


Figure 1 – Stratix III ALM

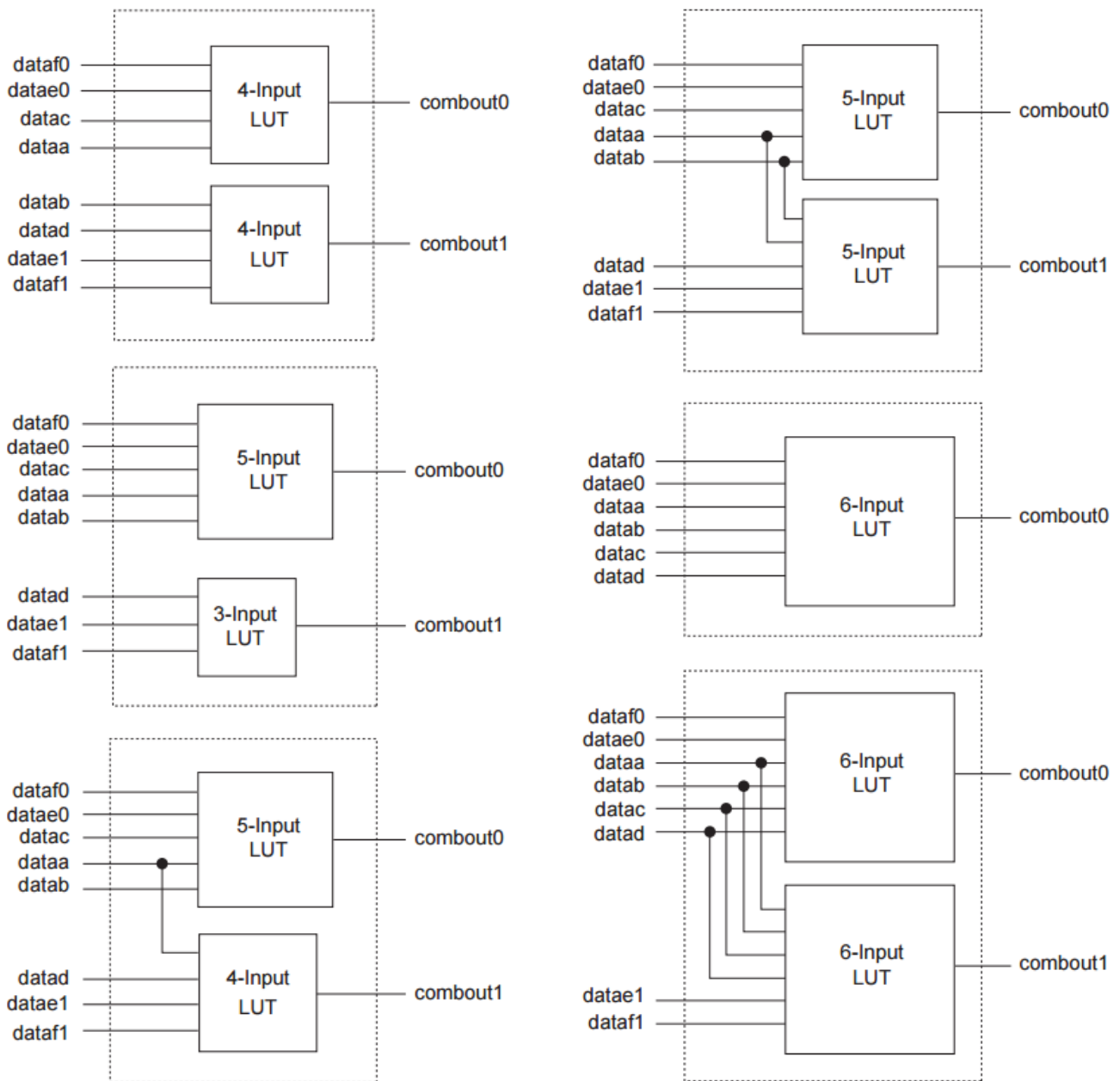


Figure 2 – ALM in Normal Mode

#### 4 EXPERIMENTS

In the process of investigation, schemes of various variants of the multi-bit LUT ( $n > 4$ ) were obtained and modeled. An example of the synthesis of a 6-LUT of four 4-LUTs and one 2-LUT is shown in Fig. 4.

In Fig. 4 2-LUT inputs have inverters, therefore, since the number of inverters on the signal path is even, the settings are recorded as usual.

#### 5 RESULTS

We restrict ourselves to  $n=8$ , so it is assumed that the additional LUT will fit into the required decomposition parameters with  $k$  LUT,  $k \in \{1, 2, 3, 4\}$ . We use the computer mathematics system Mathcad. The graphs for comparing the complexity of the decomposition according to the expression (3)  $n$  LUT over  $k$  are shown in Fig. 5

The result is expected – the larger is the building block, the less is the cost for implementing a complex LUT for 5, 6, 7 and 8 variables. The graphs of the change (5) for  $n=5\dots 8$  are shown in Fig. 6.

The graphs of the change (5) for  $n=7\dots 10$  are shown in Fig. 7.

#### 6 DISCUSSION

Thus, in the adaptive logic modules of the Stratix III FPGA there are two 4-LUTs, as indicated in the translation articles. However, in fact there are two more LUTs in 3 variables (3-LUT), from which two additional 4-LUTs can be built. In total, four 4-LUTs are obtained. It is clear how 5-LUT and 6-LUT are built from them. There is no difficulty in obtaining of two 5-LUTs. Therefore, the setting must contain at least 64 bits to specify any function of the six variables. It

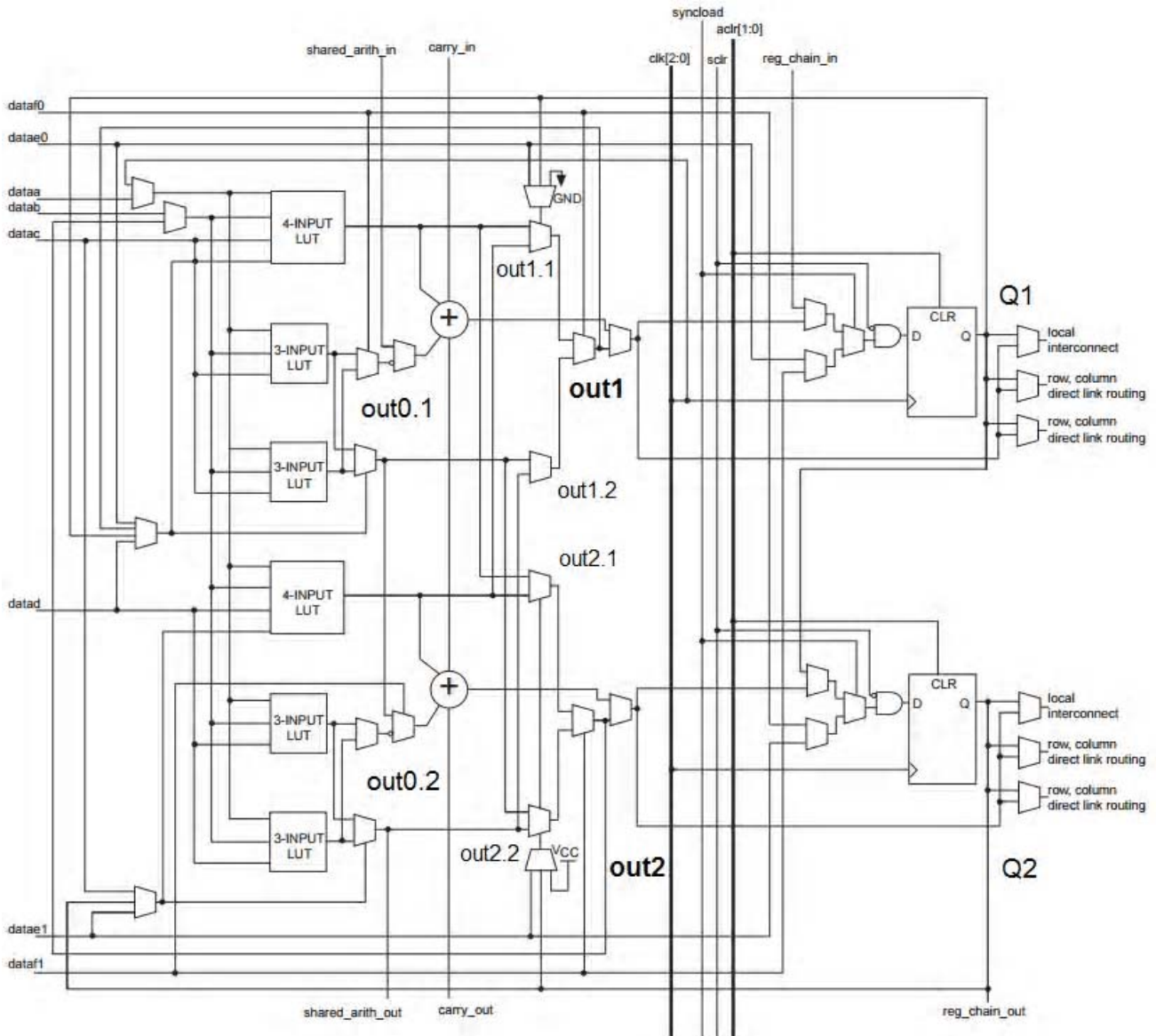


Figure 3 – Stratix III ALM Details

is advisable in the future by analyzing the ALM setup to obtain a logical model and check on it the compliance of the declared capabilities of ALM with the variants depicted in the documentation.

### CONCLUSIONS

Analyzed decomposition of multi-bit LUT shows, that the most effective in terms of complexity and speed is the use of “building blocks” 4-LUT, as it is indicated in the available sources. It is interesting to build LUT on the basis of so-called 3D transistors, which are already actively used by leading firms. There is information about mitigating the limitations of Meade and Conway in such “advanced” technologies. In addition, it is advisable to investigate the problem of decomposition when introducing the fault tolerance facilities proposed in [10] into the LUT.

### ACKNOWLEDGEMENTS

This research was carried out with the support of the Department of Automation and Remote Control of the Perm

National Research Polytechnic University, and of the Department of Mathematical Support of Computer Systems of the Perm State National Research University. Special thanks to the honored inventor of Ukraine, Doctor of Technical Sciences, Professor Kharchenko Vyacheslav Sergeevich (National Aerospace University “Kharkiv Aviation Institute”), and to the Doctor of Technical Sciences, Professor Drozd Alexander Valentinovich (Odessa National Polytechnic University).

### REFERENCES

1. Kharchenko V. Green IT Engineering / V. Kharchenko, Y. Kondratenko, J. Kacprzyk (Eds.) // *Concepts, Models, Complex Systems Architectures, Studies in Systems, Decision and Control.* – Berlin, Springer. – 2017. – Vol. 74. DOI: 10.1007/978-3-319-44162-7.
2. Kharchenko V. Concepts of Green IT Engineering: Taxonomy, Principles and Implementation / [Kharchenko V., Illiashenko O.] // *Green IT Engineering: Concepts, Models, Complex Systems Architectures, Studies in Systems, Decision and Control.* – 2017. – Vol. 74. – P. 3–20. DOI: 10.1007/978-3-319-44162-7\_1.

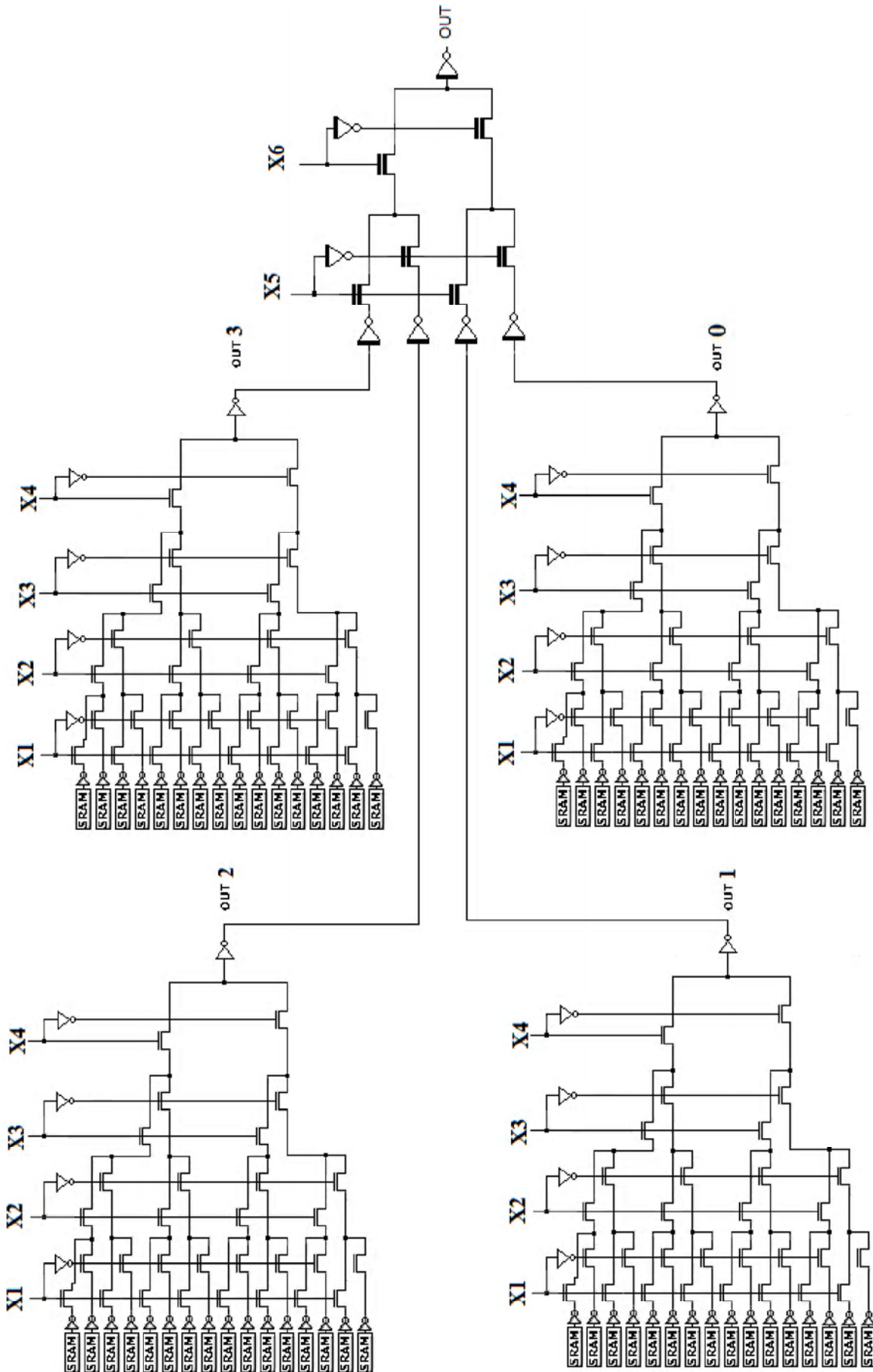


Figure 4 – 6-LUT, consisting of four 4-LUTs and one 2-LUT

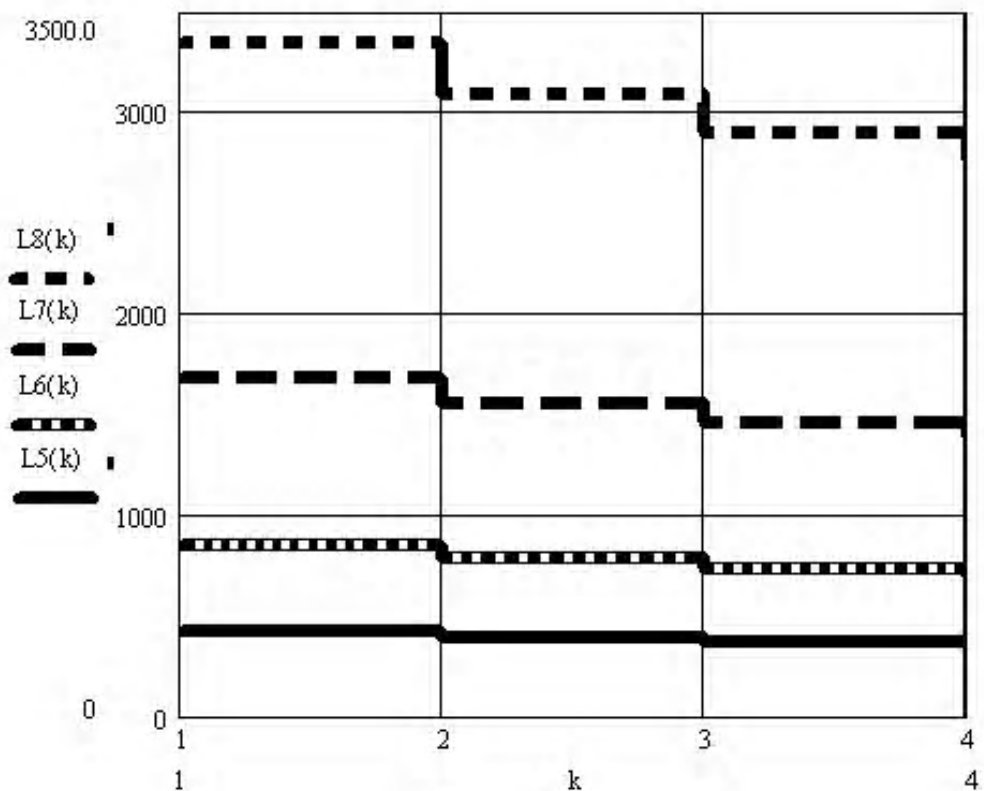


Figure 5 – Comparison of complexity of decomposition  $n$  LUT by  $k$

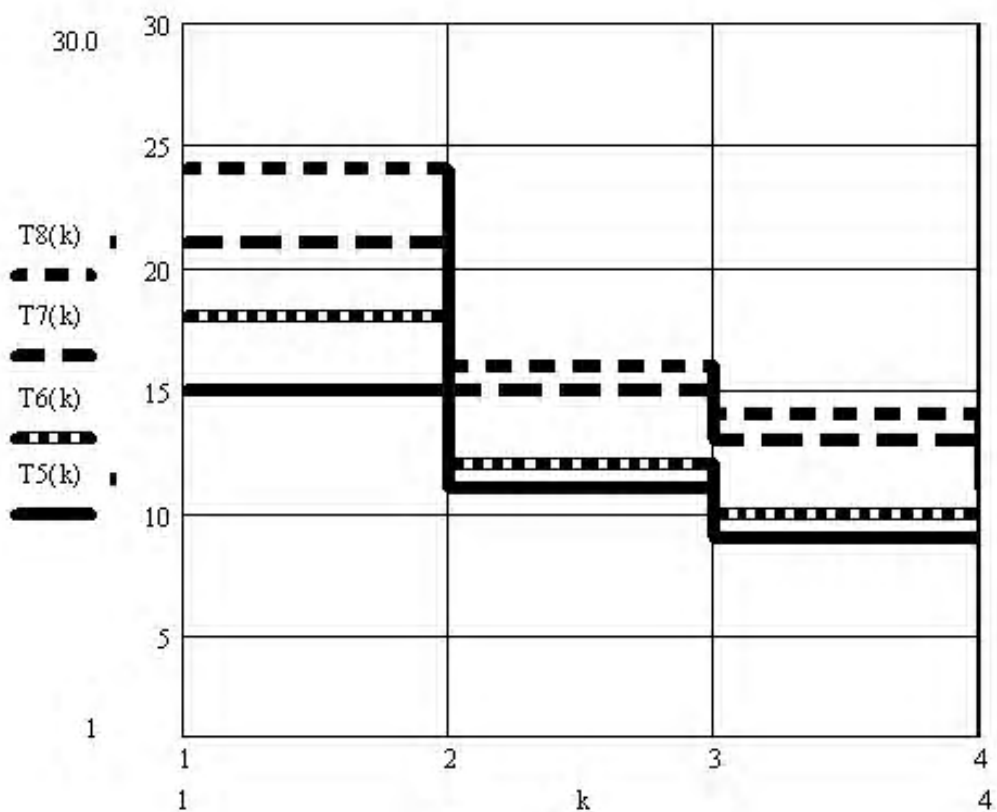


Figure 6 – Comparison of the delay LUT with decomposition for  $n=5...8$  by  $k$

3. Kondratenko Y. P. PLC-Based Systems for Data Acquisition and Supervisory Control of Environment-Friendly Energy-Saving Technologies / Y. P. Kondratenko, O. V. Korobko, O. V. Kozlov // Green IT Engineering: Concepts, Models, Complex Systems Architectures, Studies in Systems, Decision and Control. – 2017. – Vol. 74. – P. 247–267. DOI: 10.1007/978-3-319-44162-7\_13.
4. Stratix III Device Handbook, Volume 1 [electronic resource], access mode: [https://www.altera.com/content/dam/altera-www/global/en\\_US/pdfs/literature/hb/stx3/stratix3\\_handbook.pdf](https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/hb/stx3/stratix3_handbook.pdf)
5. Intel HyperFlex FPGA Architecture Overview [electronic resource], access mode: <https://www.altera.com/products/fpga/stratix-series/stratix-10/features.html#hyperflexarchitecture>
6. Understanding How the New Intel HyperFlex FPGA Architecture Enables Next-Generation High-Performance Systems [electronic resource], access mode: <https://www.altera.com/products/fpga/stratix-series/stratix-10/features.html#hyperflexarchitecture>.
7. Tyurin S. F. The decoding of LUT FPGA configuration of the finite state machine with Quartus II / S. F. Tyurin, A. V. Grekov // International Journal of Applied Engineering Research. – 2016. – Vol. 11, No. 20. – P. 10264–10266.
8. Tyurin S. Green Logic: Models, Methods, Algorithms / S. Tyurin, A. Kamenskih // Green IT Engineering: Concepts, Models, Complex Systems Architectures, Studies in Systems, Decision and Control. – 2017. – Vol. 74. – P. 69–86. DOI: 10.1007/978-3-319-44162-7\_13.
9. Presentation on ALTERA'S FPGA Technology [electronic resource], Access mode: <http://www.authorstream.com/Presentation/hsrathore158-1410279-fpga>
10. Tyurin S. Green Logic: Green LUT FPGA Concepts, Models and Evaluations / S. Tyurin // Green IT Engineering: Concepts, Models, Complex Systems Architectures, Studies in Systems, Decision and Control. – 2017. – Vol. 74. – P. 241–261. ISBN: 978-3-319-55594-2.

Article was submitted 22.09.2017.

After revision 16.11.2017.

Тюрін С. Ф.<sup>1</sup>, Греков А. В.<sup>2</sup>

<sup>1</sup>Заслужений винахідник Російської Федерації, д-р техн. наук, професор, професор кафедри автоматичної та телемеханіки Пермського національного дослідницького політехнічного університету, Перм, Росія; професор кафедри математичного забезпечення обчислювальних систем Пермського державного національного дослідницького університету, Перм, Росія

<sup>2</sup>Канд. техн. наук, доцент кафедри програмного забезпечення обчислювальної техніки і автоматизованих систем Пермського військового інституту військ національної гвардії Російської Федерації, Перм, Росія

#### ДОСЛІДЖЕННЯ СКЛАДНОСТІ БАГАТОРОЗРЯДНИХ LUT FPGA

**Актуальність.** У програмованих логічних інтегральних схемах FPGA (field-programmable gate array) використовується реалізація генератора функцій LUT (Look Up Table), який налаштовується шляхом завантаження конфігураційної пам'яті на обчислення однієї логічної функції в досконалії диз'юнктивній нормальній формі (СДНФ). Розмірність LUT визначають технологічні обмеження Міда – Конвей на число послідовно з'єднаних МОП транзисторів. Стандартним числом входів LUT довгі роки було 3, 4, причому 4-LUT будується з двох 3-LUT з додатковим 1-LUT. Однак у багатьох проєктах потрібно обчислювати функції великого числа аргументів. Для цього необхідний багаторозрядний LUT, який будується як декомпозиція 3-LUT, 4-LUT. Швидкодія обчислення логічних функцій визначається затримкою в матрицях зв'язків, тому така декомпозиція призводить до зниження швидкодії. В останні роки активно розвивається напрямок адаптивних логічних модулів (АЛМ), в яких користувачеві доступні різні варіанти логічних елементів на п'ять, шість і навіть на сім, вісім змінних, що призводить до підвищення швидкодії. Однак, детальний опис особливостей таких багаторозрядних LUT з урахуванням обмежень Міда-Конвей, оцінок складності і швидкодії в документації виробників відсутня. У той же час аналіз джерел дозволяє зробити висновок про подальше збільшення розрядності LUT і зближення можливостей FPGA і CPLD (complex programmable logic devices) в плані розрядності. Тому дослідження особливостей побудови багаторозрядних LUT є актуальними і авторами зроблена спроба аналізу реалізації такої перспективної багаторозрядної логіки.

**Мета роботи** – оцінка складності і швидкодії при декомпозиції багаторозрядного LUT.

**Метод.** Отримання виразів для оцінок складності і швидкодії декомпозиції багаторозрядного LUT на LUT меншої розрядності.

**Результати.** Виконано порівняння складності та затримки в кількості транзисторів при декомпозиції багаторозрядного LUT в системі комп'ютерної математики Mathcad.

**Висновки.** Проведені дослідження дозволили встановити особливості побудови багаторозрядних LUT і оцінювати різні варіанти декомпозиції при подальшому збільшенні розмірності LUT з подальшим вибором оптимального варіанта АЛМ.

**Ключові слова:** логічний елемент, ПЛІС типу FPGA, LUT, транзистор, адаптивний логічний модуль АЛМ, декомпозиція, складність, швидкодія.

Тюрін С. Ф.<sup>1</sup>, Греков А. В.<sup>2</sup>

<sup>1</sup>Заслуженный изобретатель Российской Федерации, д-р техн. наук, профессор, профессор кафедры автоматичної та телемеханіки Пермского национального исследовательского политехнического университета, Пермь, Россия; профессор кафедры математического обеспечения вычислительных систем Пермского государственного национального исследовательского университета, Пермь, Россия

<sup>2</sup>Канд. техн. наук, доцент кафедры программного обеспечения вычислительной техники и автоматизированных систем Пермского военного института войск национальной гвардии Российской Федерации, Пермь, Россия

#### ИССЛЕДОВАНИЕ СЛОЖНОСТИ МНОГОРАЗРЯДНЫХ LUT FPGA

**Актуальность.** В программируемых логических интегральных схемах FPGA (field-programmable gate array) используется реализация генератора функций LUT (Look Up Table), который настраивается путем загрузки конфигурационной памяти на вычисление одной логической функции в совершенной диз'юнктивной нормальной форме (СДНФ). Размерность LUT определяют технологические ограничения Мида и Конвей на число последовательно соединенных МОП транзисторов. Стандартным числом входов LUT долгие годы было 3, 4, причем 4-LUT строится из двух 3-LUT с дополнительным 1-LUT. Однако во многих проєктах требуется вычислять функции большого числа аргументов. Для этого необходим многоразрядный LUT, который строится как декомпозиция 3-LUT, 4-LUT. Быстродействие вычисления логических функций определяется задержкой в матрицах связей, поэтому такая декомпозиция приводит к снижению быстродействия. В последние годы активно развивается направление адаптивных логических модулей (АЛМ), в которых пользователю доступны различные варианты логических элементов на пять, шесть и даже на семь, восемь переменных, что приводит к повышению быстродействия. Однако, детальное описание особенностей таких многоразрядных LUT с учетом ограничений Мида-Конвей, оценок сложности и быстродействия в документации производителей отсутствует. В то же время анализ источников позволяет

сделать вывод о дальнейшем увеличении разрядности LUT и сближении возможностей FPGA и CPLD (complex programmable logic devices) в плане разрядности. Поэтому исследования особенностей построения многоразрядных LUT являются актуальными и авторами предпринята попытка анализа реализации такой перспективной многоразрядной логики.

**Цель работы** – оценка сложности и быстродействия при декомпозиции многоразрядного LUT.

**Метод.** Получение выражений для оценок сложности и быстродействия декомпозиции многоразрядного LUT на LUT меньшей разрядности.

**Результаты.** Выполнено сравнение сложности и задержки в количестве транзисторов при декомпозиции многоразрядного LUT в системе компьютерной математики Mathcad.

**Выводы.** Проведенные исследования позволили установить особенности построения многоразрядных LUT и оценивать различные варианты декомпозиции при дальнейшем увеличении размерности LUT с последующим выбором оптимального варианта ALM.

**Ключевые слова:** логический элемент, ПЛИС типа FPGA, LUT, транзистор, адаптивный логический модуль ALM, декомпозиция, сложность, быстродействие.

## REFERENCES

1. Kharchenko V., Kondratenko Y., Kacprzyk J. (Eds.) *Green IT Engineering: Concepts, Models, Complex Systems Architectures, Studies in Systems, Decision and Control*. Berlin, Springer, 2017, Vol. 74. DOI: 10.1007/978-3-319-44162-7.
2. Kharchenko V., Iliashenko O. *Concepts of Green IT Engineering: Taxonomy, Principles and Implementation, Green IT Engineering: Concepts, Models, Complex Systems Architectures, Studies in Systems, Decision and Control*, Berlin, Springer, 2017, Vol. 74, pp. 3–20. DOI: 10.1007/978-3-319-44162-7\_1.
3. Kondratenko Y. P., Korobko O. V., Kozlov O. V. *PLC-Based Systems for Data Acquisition and Supervisory Control of Environment-Friendly Energy-Saving Technologies, Green IT Engineering: Concepts, Models, Complex Systems Architectures, Studies in Systems, Decision and Control*, Berlin, Springer, 2017, Vol. 74, pp. 247–267. DOI: 10.1007/978-3-319-44162-7\_13.
4. *Stratix III Device Handbook, Volume 1* [electronic resource], access mode: [https://www.altera.com/content/dam/altera-www/global/en\\_US/pdfs/literature/hb/stx3/stratix3\\_handbook.pdf](https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/hb/stx3/stratix3_handbook.pdf)
5. *Intel HyperFlex FPGA Architecture Overview* [electronic resource], access mode: <https://www.altera.com/products/fpga/stratix-series/stratix-10/features.html#hyperflexarchitecture>
6. *Understanding How the New Intel HyperFlex FPGA Architecture Enables Next-Generation High-Performance Systems* [electronic resource], access mode: <https://www.altera.com/products/fpga/stratix-series/stratix10/features.html#hyperflexarchitecture>
7. Tyurin S. F., Grekov A. V. The decoding of LUT FPGA configuration of the finite state machine with Quartus II, *International Journal of Applied Engineering Research*, 2016, Vol. 11, No. 20, pp. 10264–10266.
8. Tyurin S., Kamenskih A. Green Logic: Models, Methods, Algorithms, *Green IT Engineering: Concepts, Models, Complex Systems Architectures, Studies in Systems, Decision and Control*. Berlin, Springer, Vol. 74, 2017, pp. 69–86. DOI: 10.1007/978-3-319-44162-7\_13.
9. *Presentation on ALTERA'S FPGA Technology* [electronic resource], access mode: <http://www.authorstream.com/Presentation/hsrathore158-1410279-fpga>
10. Tyurin S. Green Logic: Green LUT FPGA Concepts, Models and Evaluations, *Green IT Engineering: Concepts, Models, Complex Systems Architectures, Studies in Systems, Decision and Control*, Berlin, Springer, 2017, Vol. 74, pp. 241–261. ISBN: 978-3-319-55594-2.