

## ЛОГИЧЕСКИЙ ЭЛЕМЕНТ FPGA ДЛЯ РЕАЛИЗАЦИИ ДИЗЬЮНКТИВНОЙ НОРМАЛЬНОЙ ФОРМЫ

А. В. ГРЕКОВ<sup>1</sup>, С. Ф. ТЮРИН<sup>2</sup>

*Пермский военный институт войск национальной гвардии Российской Федерации,  
614112, Пермь, Россия  
E-mail: grekartemvl@mail.ru*

*Пермский национальный исследовательский политехнический университет,  
614990, Пермь, Россия  
E-mail: tyurinsergfeoyandex.ru*

Логический элемент программируемых логических интегральных схем типа FPGA называется в англоязычных работах LUT (Look Up Table), что переводится как „просмотровая таблица“, т.е. по существу таблица истинности некоторой логической функции, основанная на совершенных дизьюнктивных нормальных формах представления логических функций. Это требует значительных аппаратных затрат, в отличие от дизьюнктивной нормальной формы (ДНФ), используемой с 1970-х гг. в программируемых логических матрицах (ПЛМ). В то же время большая часть реализуемых логических функций имеет число конъюнкций  $k$  много меньшее, чем общее число наборов  $n$  двоичных переменных. Для программирования ПЛМ необходимы специальные программаторы, тогда как конфигурация постоянного запоминающего устройства (ПЗУ) загружается в оперативные запоминающие устройства. Предложен логический элемент ДНФ-ПЗУ, реализующий систему функций в ДНФ, такой подход существенно снижает аппаратные затраты на реализацию логики программируемых логических интегральных схем при неухудшении быстродействия. Проанализирована сложность предлагаемого технического решения, приведены результаты функционального моделирования.

**Ключевые слова:** логический элемент, ПЛИС типа FPGA, Look Up Table, дизьюнктивная нормальная форма, транзистор

**Введение.** Логические функции в программируемых логических интегральных схемах (ПЛИС) типа FPGA (field-programmable gate array) [1, 2] вычисляются на основе совершенной дизьюнктивной нормальной формы (ДНФ). При этом используются постоянные запоминающие устройства (ПЗУ), называемые также Look Up Table (LUT), реализованные на мультиплексоре, входы данных которого настраиваются константами-значениями из таблиц истинности.

Оптимально по быстродействию и сложности представления типовых логических функций использовать ПЗУ на четыре переменных (4-ПЗУ) [1, 2]. Такое ПЗУ для входных переменных  $x_4, x_3, x_2, x_1$  приведено на рис. 1, оно описывается выражением с шестнадцатью битами настройки:

$$\begin{aligned}
 z_{\text{вых}}(x_4x_3x_2x_1) = & \overline{a}x_4\overline{x_3}\overline{x_2}\overline{x_1} \vee \overline{b}x_4\overline{x_3}\overline{x_2}x_1 \vee \overline{c}x_4x_3\overline{x_2}\overline{x_1} \vee \overline{d}x_4x_3x_2\overline{x_1} \vee \\
 & \overline{e}x_4x_3\overline{x_2}x_1 \vee \overline{f}x_4\overline{x_3}\overline{x_2}x_1 \vee \overline{g}x_4x_3x_2\overline{x_1} \vee \overline{h}x_4x_3x_2x_1 \vee \\
 & \overline{i}x_4\overline{x_3}\overline{x_2}\overline{x_1} \vee \overline{j}x_4\overline{x_3}\overline{x_2}x_1 \vee \overline{k}x_4\overline{x_3}x_2\overline{x_1} \vee \overline{l}x_4\overline{x_3}x_2x_1 \vee \\
 & \overline{m}x_4x_3\overline{x_2}\overline{x_1} \vee \overline{n}x_4x_3\overline{x_2}x_1 \vee \overline{o}x_4x_3x_2\overline{x_1} \vee \overline{p}x_4x_3x_2x_1.
 \end{aligned}
 \tag{1}$$

В ПЛИС Stratix III имеются адаптивные (перестраиваемые под требуемую задачу) логические блоки (ALM), которые объединяются в логические блоки (Logic Array Block, LAB) [3, 4], реализующие функции даже семи переменных. Но каждое ПЗУ реализует только одну логическую функцию для этой конфигурации. При увеличении числа переменных экспоненциально возрастают затраты на настройку. В статье предложен логический элемент ДНФ-ПЗУ, реализующий дизъюнктивную нормальную форму, что позволяет сократить затраты на использование логических функций большого числа переменных.

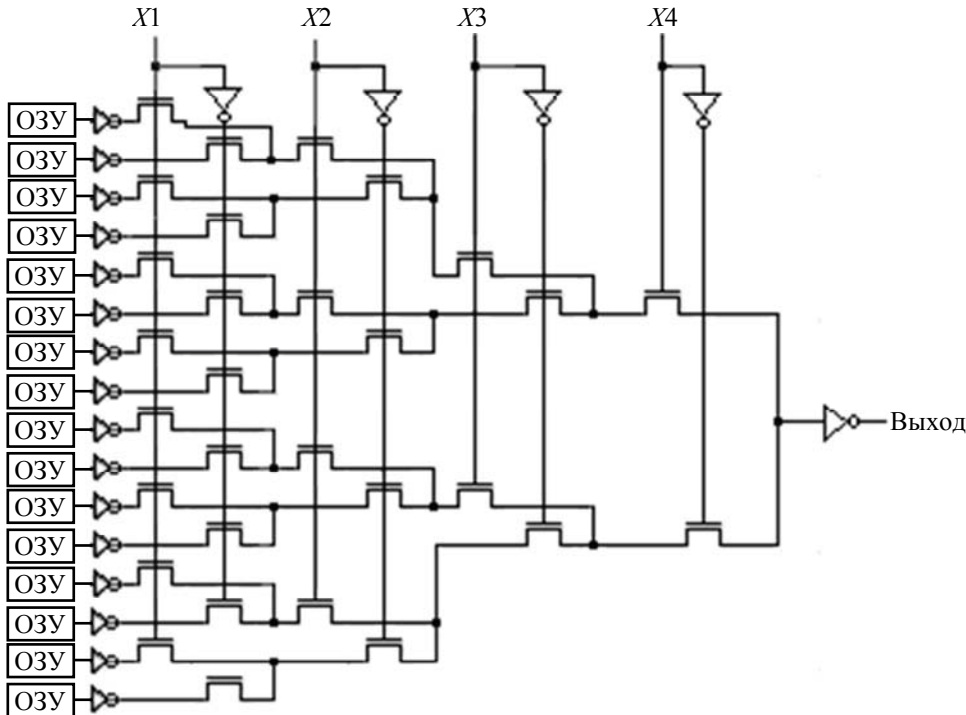


Рис. 1

**Структура ДНФ-ПЗУ** конфигурируется пользователем, как и программируемая логическая матрица [1, 5], для которой необходим специальный программатор. В ДНФ-ПЗУ конфигурирование происходит путем загрузки оперативной памяти ОЗУ. На рис. 2 представлена структурная схема блока ДНФ-ПЗУ (настройка конъюнкций загружается в ОЗУ, не указанное на рис. 2).

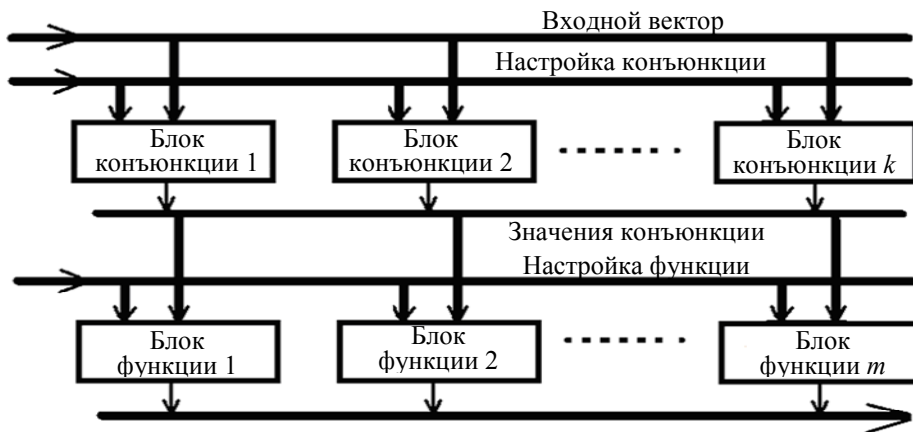


Рис. 2

Таким образом, вместо загрузки значений таблицы истинности задаются лишь значения конъюнкций длиной  $n$ , где  $n$  — число переменных  $m$  логических функций. Вхождение  $k$  конъюнкций в  $m$  функций также программируется настройкой. По заданному входному набору (вектору)  $n$  переменных блоки конъюнкций вычисляют значения  $k$  конъюнкций, которые далее формируют по „ИЛИ“ значения  $m$  логических функций. Рассмотрим внутреннюю структуру блоков конъюнкций и функций.

**Структура блоков конъюнкций и функций ДНФ-ПЗУ.** Структурная схема блока конъюнкции [6] ДНФ-ПЗУ приведена на рис. 3.

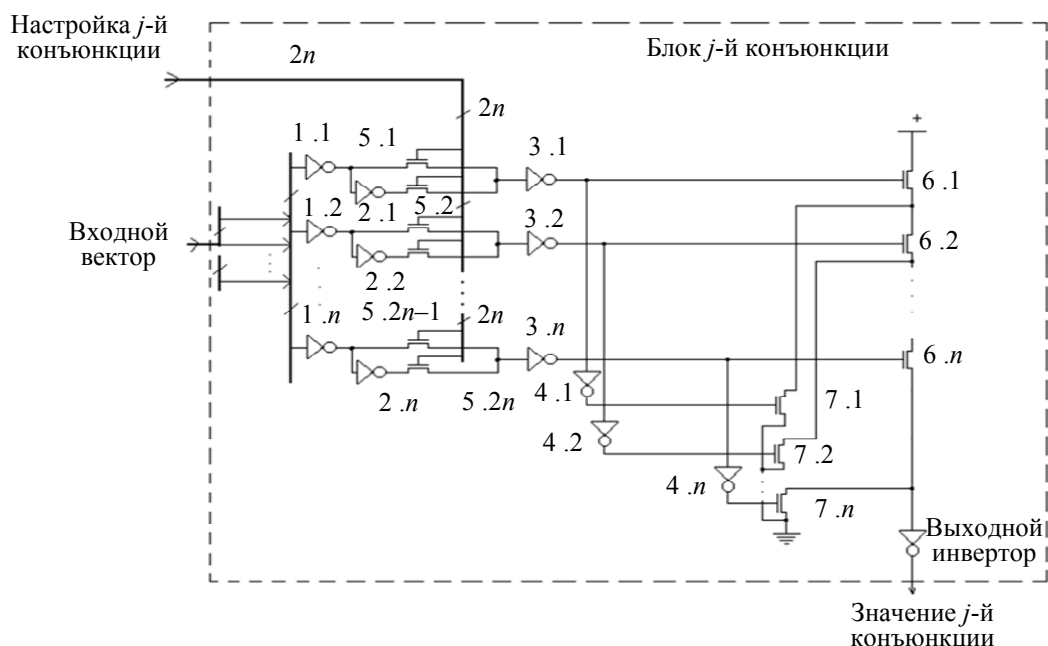


Рис. 3

Для задания значения одной из  $n$  переменных в  $j$ -й конъюнкции используются три группы инверторов 1, 2, 3, 4 так, что если переменная входит в конъюнкцию без инверсии, то активируются нечетные транзисторы группы 5 транзисторов. Соответственно активируются затворы транзисторов группы 6. Если переменная входит в конъюнкцию с инверсией, активируются четные транзисторы группы 5 транзисторов. В случае несущественной переменной активируются и четные, и нечетные транзисторы в каждой из пар группы 5, чтобы соответствующий транзистор в группе 6 был активирован. Если во входном наборе значение переменной не равно заданному, то соответствующий транзистор в группе 6 не будет активирован, но для исключения „третьего состояния“ в цепочке транзисторов 6 с помощью инверторов группы 4 на его выход будет подключен потенциал „Нуль вольт“. Таким образом обеспечивается требуемая ортогональность сигнала на выходах транзисторов группы 6 [7].

Следовательно, если заданная конъюнкция равна единице, на выходе будет нуль, иначе — единица. Для ограничения по числу последовательно соединенных транзисторов [7, 8] могут быть введены дополнительные промежуточные инверторы. Так, для функции  $S0 \oplus S1 \oplus S2$  настройка приведена в таблице.

**Активирование транзисторов группы 5 четырех блоков конъюнкции ДНФ-ПЗУ для  $S0 \oplus S1 \oplus S2$**

№ блока конъюнкции	5.1	5.2	5.3	5.4	5.5	5.6	Конъюнкция
1	0	1	0	1	1	0	$\overline{S0} \cdot \overline{S1} \cdot S2$
2	0	1	1	0	0	1	$\overline{S0} \cdot S1 \cdot \overline{S2}$
3	1	0	0	1	0	1	$S0 \cdot \overline{S1} \cdot \overline{S2}$
4	1	0	1	0	1	0	$S0 \cdot S1 \cdot S2$

Структурная схема блока функций [6] ДНФ-ПЗУ приведена на рис. 4. При задании необходимых конъюнкций для данного блока функции ДНФ-ПЗУ используется внешняя настройка, активирующая затворы транзисторов  $T$ . В случае хотя бы одного требуемого логического нуля по входу значений конъюнкций на выходе блока формируется логическая единица.

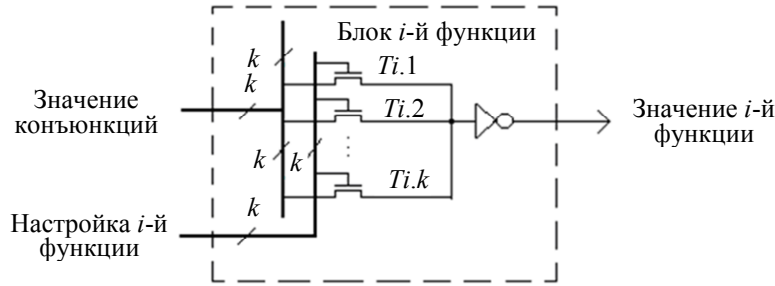


Рис. 4

**Сравнительная оценка аппаратных затрат на реализацию ДНФ-ПЗУ.** Оценим затраты на реализацию традиционного ПЗУ:

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

где  $2^n \cdot 8$  — число элементов настройки (на каждый вход настройки необходимо шесть транзисторов ОЗУ и два транзистора для реализации инвертора на входе дерева передающих транзисторов);  $2^n$  — число инверторов по  $n$  переменным;  $2^{n+1}$  — число элементов дерева передающих транзисторов с выходным инвертором.

В предлагаемом ДНФ-ПЗУ на один разряд требуется двенадцать транзисторов и два транзистора на отдельный инвертор блока конъюнкции. На каждый из  $m$  блоков вычисления функций требуется  $k$  транзисторов и два транзистора на отдельный инвертор блока вычисления функций. Итого получаем:

$$L_{\text{ДНФ}}(n, k, m, \text{ОЗУ}) = k(12n + 2) + m(k + 2) + 6k(2n) + 6mk. \quad (3)$$

Сравнение числа транзисторов ДНФ-ПЗУ (сплошная линия) и традиционного ПЗУ (пунктир) для  $n=2-4$ ,  $m=16$ ,  $k=8$  представлено на рис. 5, а; для  $n=4-8$ ,  $m=16$ ,  $k=8$  — на рис. 5, б.

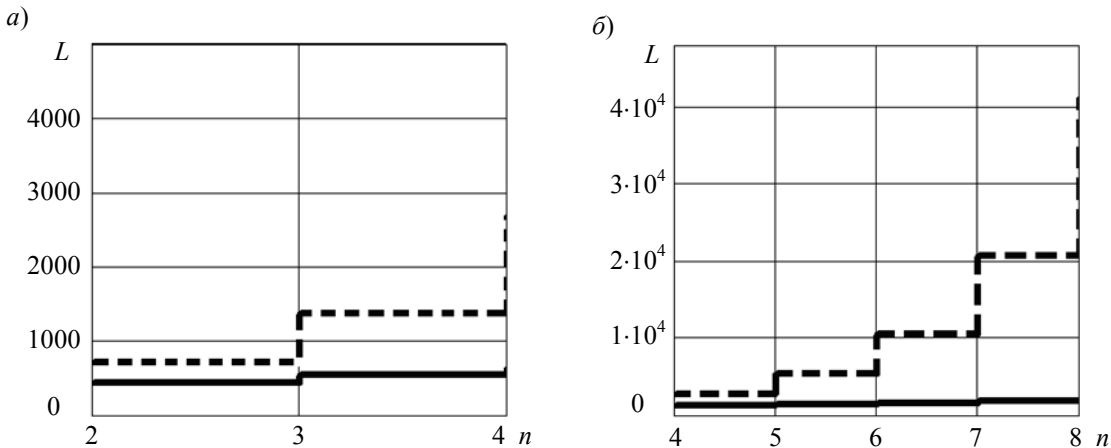


Рис. 5

Моделирование логического элемента ДНФ-ПЗУ подтвердило работоспособность предлагаемого технического решения.

**Вывод.** Таким образом, использование ДНФ-ПЗУ позволяет сократить число элементов для реализации функций множества числа переменных. Выигрыш тем больше, чем меньше используется конъюнкций из  $2^n$ . При этом затраты растут линейно, в отличие от традиционного ПЗУ. Особенно велик выигрыш при  $n \geq 7$ . Функциональное моделирование подтверждает работоспособность предлагаемого технического решения — конфигурируемого логического элемента ДНФ-ПЗУ для ПЛИС FPGA.

## СПИСОК ЛИТЕРАТУРЫ

1. Угрюмов Е. П. Цифровая схемотехника: учебное пособие для вузов. СПб: БХВ-Петербург, 2007. 800 с.
2. Цыбин С. Программируемая коммутация ПЛИС: взгляд изнутри [Электронный ресурс]: <[http://www.kite.ru/articles/plis/2010\\_11\\_56.php](http://www.kite.ru/articles/plis/2010_11_56.php)>.
3. Золотуха Р., Комолов Д. Stratix III — новое семейство FPGA фирмы Altera [Электронный ресурс]: <[http://kite.ru/assets/files/pdf/2006\\_12\\_30.pdf](http://kite.ru/assets/files/pdf/2006_12_30.pdf)>.
4. Использование ресурсов ПЛИС Stratix III фирмы Altera при проектировании микропроцессорных ядер [Электронный ресурс]: <<file:///C:/Users/%D0%A2%D1%8E%D1%80%D0%B8%D0%BD/Desktop/%D0%A6%D1%8B%D0%B1%D0%B8%D0%BD%2010%20%D0%B3%D0%BE%D0%B4.pdf>>.
5. Пат. 2503993 РФ. Программируемое логическое устройство / С. Ф. Тюрин, А. В. Набатов, О. А. Громов, А. В. Греков, Д. А. Карлов. Опубл. 10.01.2014. Бюл. № 1.
6. Пат. 2544750 РФ. Программируемое логическое устройство / С. Ф. Тюрин. Опубл. 20.03.2015. Бюл. № 8.
7. Ульман Дж. Д. Вычислительные аспекты СБИС / Пер. с англ. А. В. Неймана. Под ред. П. П. Пархоменко. М.: Радио и связь, 1990. 480 с.
8. Tyurin S. F., Grekov A. V. Functionally Complete Tolerant Elements // Intern. J. of Applied Engineering Research. 2015. Vol. 14, N 10. P. 34433—34442. ISSN 0973-4562.

**Сведения об авторах****Артем Владимирович Греков**

— канд. техн. наук, доцент; Пермский военный институт войск национальной гвардии Российской Федерации, кафедра программного обеспечения вычислительной техники и автоматизированных систем; E-mail: grekartemvl@mail.ru

**Сергей Феофанович Тюрин**

— д-р техн. наук, профессор; Пермский национальный исследовательский политехнический университет, кафедра автоматизации и телемеханики; E-mail: tyurinsergfeo@yandex.ru

Рекомендована кафедрой  
программного обеспечения  
вычислительной техники  
и автоматизированных систем

Поступила в редакцию  
20.01.17 г.

**Ссылка для цитирования:** Греков А. В., Тюрин С. Ф. Логический элемент FPGA для реализации дизъюнктивной нормальной формы // Изв. вузов. Приборостроение. 2017. Т. 60, № 6. С. 513—518.

**FPGA LOGIC ELEMENT FOR IMPLEMENTATION OF DISJUNCTIVE NORMAL FORM****A. V. Grekov<sup>1</sup>, S. F. Tyurin<sup>2</sup>**

<sup>1</sup>Perm Military Institute of National Guard Troops of the Russian Federation,  
614112, Perm, Russia  
E-mail: grekartemvl@mail.ru

<sup>2</sup>Perm National Research Polytechnic University, 614990, Perm, Russia  
E-mail: tyurinsergfeo@yandex.ru

Logical element of programmable logic integrated circuits of FPGA (field-programmable gate array) type is called in foreign sources LUT (Look Up Table), i.e. it is essentially a truth table of a logic function. This implementation is based on the perfect disjunctive normal forms (PDFNF) representation of the logic functions that require significant hardware costs, as opposed to the implementation in the disjunctive normal form (DNF), used with the 70-ies of XX century in the programmable logic array (PLA). At the same time, much of the logic functions implemented by a number k of conjunctions is much smaller than the total number of sets of n binary variables. However, for programming the PLA requires special programmers, whereas the configuration LUT is loaded into random access memory (RAM or SRAM). A DNF-LUT is proposed to implement a system of logical functions in DNF which significantly reduces the hardware cost of implementing the logic of programmable logic integrated circuits with no deterioration of the performance. Complexity of the proposed technical solutions and the results of functional simulation are analyzed.

**Keywords:** logical element, PLD of FPGA type, Look Up Table, disjunctive normal form, transistor

**Data on authors**

- Artem V. Grekov** — PhD, Associate Professor; Perm Military Institute of National Guard Troops of the Russian Federation, Department of Software Computer Technology and Automated Systems; E-mail: grekartemvl@mail.ru
- Sergey F. Tyurin** — Dr. Sci., Professor; Perm National Research Polytechnic University, Department of Automation and Telemechanics; E-mail: tyurinsergfeo@yandex.ru

**For citation:** Grekov A. V., Tyurin S. F. FPGA logic element for implementation of disjunctive normal form. *Journal of Instrument Engineering*. 2017. Vol. 60, N 6. P. 513—518 (in Russian).

DOI: 10.17586/0021-3454-2017-60-6-513-518