

Разработка методов архитектурно-ориентированного ресинтеза в маршруте автоматизированного проектирования ПЛИС

И.В. Тиунов, И.А. Липатов, Д.А. Железников

Институт проблем проектирования в микроэлектронике РАН, г. Зеленоград,

tiunov_i@ippm.ru, lipatov_i@ippm.ru, zheleznikov_d@ippm.ru

Аннотация — С увеличением сложности схем, проектируемых в базе программируемых логических интегральных схем (ПЛИС), растет и занимаемая ими площадь. Большая занимаемая площадь ведет к увеличению общей длины межсоединений, что является причиной увеличения задержек распространения сигналов и, как следствие, общего ухудшения быстродействия. Кроме того, это приводит к трудностям на этапе трассировки межсоединений ввиду ограниченности трассировочных ресурсов ПЛИС. Таким образом, возрастает потребность в разработке алгоритмов логического ресинтеза и оптимизации проектируемых схем для уменьшения занимаемой ими площади на кристалле.

В данной работе предложены архитектурно-ориентированные методы ресинтеза схем для ПЛИС. Результатом применения таких методов является объединение логических элементов схемы с триггерами для случаев, когда выход логического элемента одновременно связан с несколькими логическими элементами или триггерами. Методы апробированы на наборах тестовых схем сборника IWLS 2005, а также некоторых реально разрабатываемых схемах. Описанные в статье методы реализованы в виде программного модуля и внедрены в существующий маршрут проектирования схем для ПЛИС.

Ключевые слова — автоматизация проектирования, технологическое отображение, логический ресинтез, САПР, ПЛИС.

I. ВВЕДЕНИЕ

Различные схемотехнические решения, такие как объединение программируемой таблицы соответствия (ТС или LUT в англоязычной литературе) и триггера (DFF) в одной логической ячейке (ЛЯ или LE) (рис. 1), встречаются в большинстве распространенных архитектур ПЛИС, а использование этих ресурсов для оптимизации в большинстве случаев приведет к значительному уменьшению площади занимаемой проектируемой схемой. Также, это приводит к улучшению временных характеристик конечного устройства.

Из этого следует, что разработка и внедрение в существующие маршруты проектирования схем для ПЛИС архитектурно-ориентированных методов

ресинтеза, осуществляемых на этапе технологического отображения является актуальной задачей.

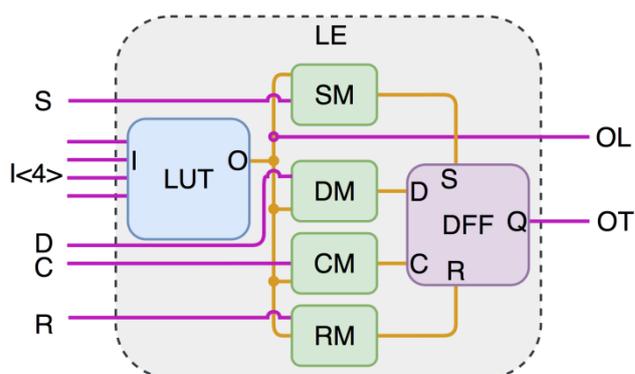


Рис. 1. Архитектура "идеальной" логической ячейки

В предыдущей работе [1] было показано, что наличие таких архитектурных особенностей ПЛИС позволяет проводить оптимизацию по площади результатов технологически независимого синтеза. В ней рассматривается лишь простейший случай, когда к выходу логического элемента подключен только один триггер.

Помимо распространенного подхода к объединению таблицы соответствия и триггера в одной логической ячейке также существуют и другие возможности для архитектурно-зависимой оптимизации схем. Например, в некоторых архитектурах логическая ячейка может иметь два выхода: выход ТС и выход триггера. Возможность использования данных одновременно с двух выходов обуславливает потенциал множественной оптимизации схем.

Далее будет приведено описание "идеальной" логической ячейки. Представленная архитектура позволяет применять все из рассматриваемых далее методов ресинтеза на этапе технологического отображения.

II. ОПИСАНИЕ "ИДЕАЛЬНОЙ" ЛОГИЧЕСКОЙ ЯЧЕЙКИ

Для начала опишем "идеальную" (с точки зрения применимости рассматриваемых в статье методов)

программируемую логическую ячейку ПЛИС, для которой далее проводится исследование.

Такая ячейка должна содержать: набор вентилях, отвечающий за реализацию ТС, триггер с входами синхронизации, сброса и установки, четыре мультиплексора на каждый из входов триггера для перенаправления сигнала с программируемой ТС на один из входов триггера.

Список внешних интерфейсов должен содержать: информационные входы ТС (I), информационный (D) и управляющие (синхронизации (C), сброса (R) и установки (S)) входы триггера, набор конфигурационных входов для программирования ячейки (в том числе мультиплексоров), а также, выход триггера (OT) и выход таблицы соответствия (OL). Схематично такая ячейка представлена на рис. 1.

В реальных ПЛИС часто используется только часть дополнительного функционала управления мультиплексорами данных и управляющих входов, которые упомянуты в структуре рассматриваемой ячейки. Например, чаще всего встречается использование двух выходов (OL и OT) и возможность перенаправления сигнала с ТС на триггер. Такие подходы применяются в ПЛИС компаний Altera (Stratix II [3], Cyclone V [4] и др.) и Xilinx (XC4000E [5] и др.), а также других распространенных архитектурах. Гораздо реже программируемая логическая ячейка содержит логику, отвечающую за перенаправление сигнала с ТС на управляющие входы триггера.

III. ОБЪЕДИНЕНИЕ В СЛУЧАЕ МНОЖЕСТВЕННОЙ НАГРУЗКИ И ОБЪЕДИНЕНИЕ ПО УПРАВЛЯЮЩИМ ВХОДАМ

A. Объединение по управляющим входам

Архитектура ячейки (рис. 1) позволяет перенаправлять данные с выхода ТС на один из входов триггера, будь то информационный или один из управляющих. Из этого следует, что объединение ячеек может быть произведено не только по информационному входу триггера, но и по любому из управляющих триггером входов.

В предыдущей работе [1], как упоминалось ранее, объединение ячеек происходило лишь в том случае, когда ТС и триггер были соединены между собой по входу данных. Очевидно, что для данной ячейки подобного ограничения нет, и оптимизация может быть проведена и в тех случаях, когда требуется перенаправить сигнал не на информационный вход, а на один из управляющих посредством изменения сигнала на адресном входе соответствующего мультиплексора (CM, RM или SM). Также очевидно, что для проведения подобной оптимизации в проектируемой схеме должна присутствовать логика, управляющая сигналами синхронизации (C), сброса (R) или установки (S) триггера.

B. Объединение с одним из триггеров в случае множественной нагрузки

Одним из ограничений работы алгоритма оптимизации технологического отображения из

предыдущей работы также была невозможность его применения для случаев, когда на выходе логического элемента нагружено больше одного триггера, однако, часто возникают ситуации, когда в качестве нагрузки выступают несколько элементов, как триггеров, так и элементов ТС.

Рассмотрим одну из таких ситуаций, когда в качестве нагрузки выступает множество элементов, среди которых есть хотя бы один триггер (рис. 2).

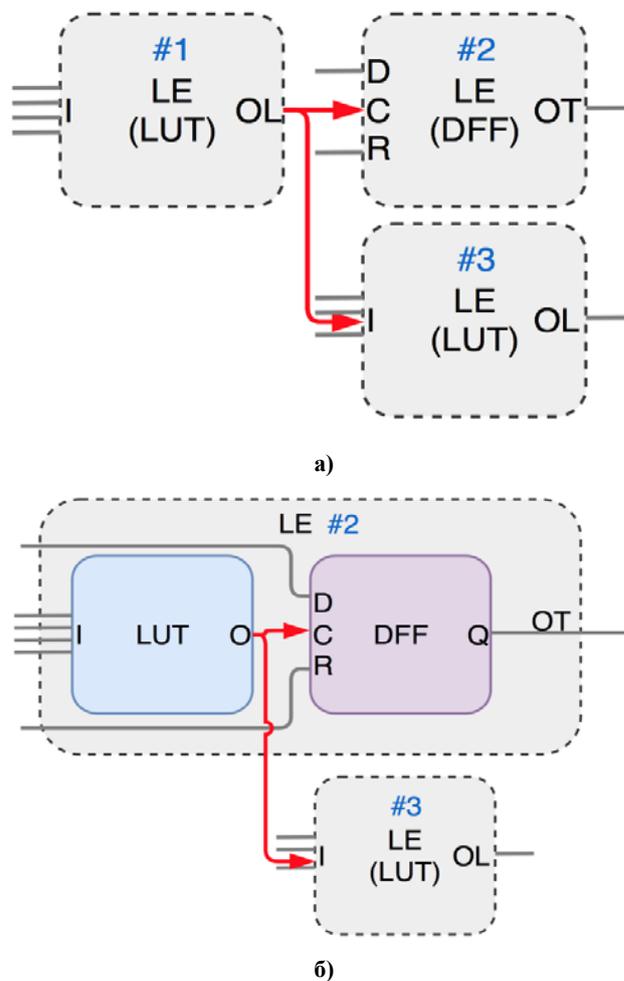


Рис. 2. Объединение ТС и триггера по входу синхронизации: а) до; б) после

Поскольку в рассматриваемой ячейке предусмотрен вывод данных и с ТС и с триггера, то становится возможным объединение ТС с одним из нагрузочных триггеров, а оставшиеся нагрузочные элементы подключить к выходу OL объединенной ячейки. Для этого необходимо, для начала, разорвать цепь между ТС и объединяемым триггером и со всеми нагрузочными элементами. Затем следует произвести объединение ячеек ТС и выбранного триггера. Наконец, необходимо восстановить соединение с оставшимися нагрузочными элементами с выходом OL объединенной ячейки.

Несмотря на то, что на рис. 2 представлен случай, когда ТС и триггер объединяются по входу

синхросигнала, очевидно, что данный метод можно применять как для объединения по информационному входу, так и по другим управляющим входам триггера (сброса и установки).

С. Объединение с несколькими триггерами в случае множественной нагрузки.

Другой вариант множественного объединения можно провести также для случаев, когда в качестве нагрузки выступают несколько триггеров (не исключая вариант ТС в качестве нагрузки). В такой ситуации возможно объединение сразу с несколькими триггерами. Для этого необходимо создать копии ТС для каждого триггера и произвести объединение также как в случае с одним триггером для каждой из пар ТС-триггер. Пример такой ситуации продемонстрирован на рис. 3.

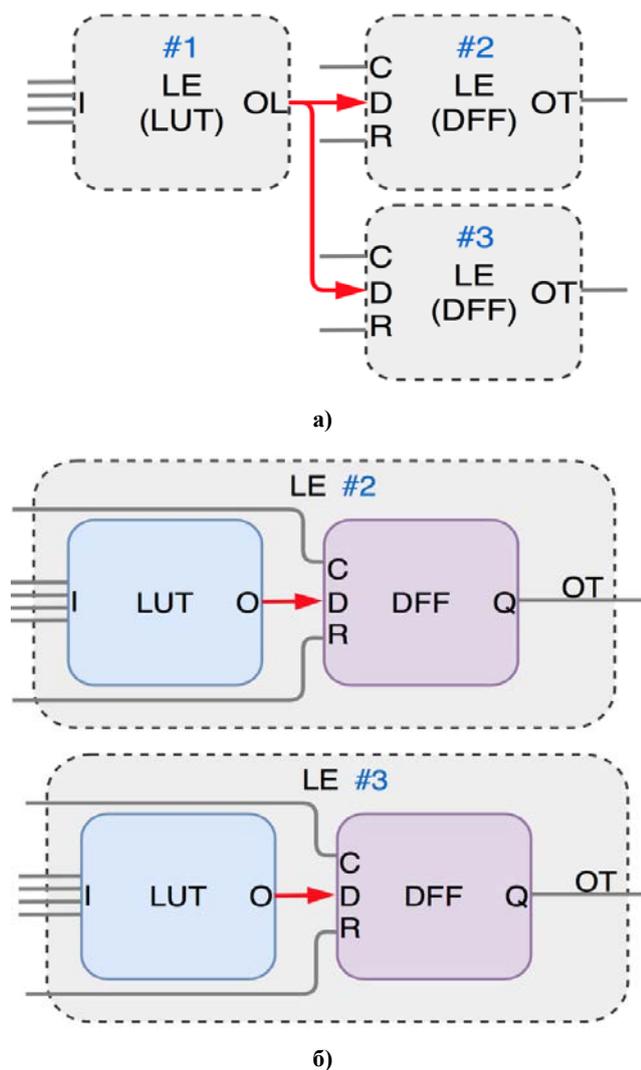


Рис. 3. Множественное объединение ТС и триггера: а) до; б) после

Как и в случае объединения с одним триггером, данный метод можно применять для всех доступных вариантов объединения, как по информационному, так и по управляющим входам.

IV. ВНЕДРЕНИЕ ТЕХНОЛОГИЧЕСКОГО РЕСИНТЕЗА В МАРШРУТ ПРОЕКТИРОВАНИЯ СХЕМ НА ПЛИС

Разработанные методы архитектурно-ориентированного ресинтеза на этапе технологического отображения были реализованы в виде кроссплатформенного программного пакета ХСУ, предназначенного для технологического отображения схем в базис логических элементов ПЛИС и их ресинтеза на данном этапе проектирования. Программное обеспечение было внедрено в существующий маршрут проектирования цифровых схем для отечественных ПЛИС, разработанный ИППМ РАН [2].

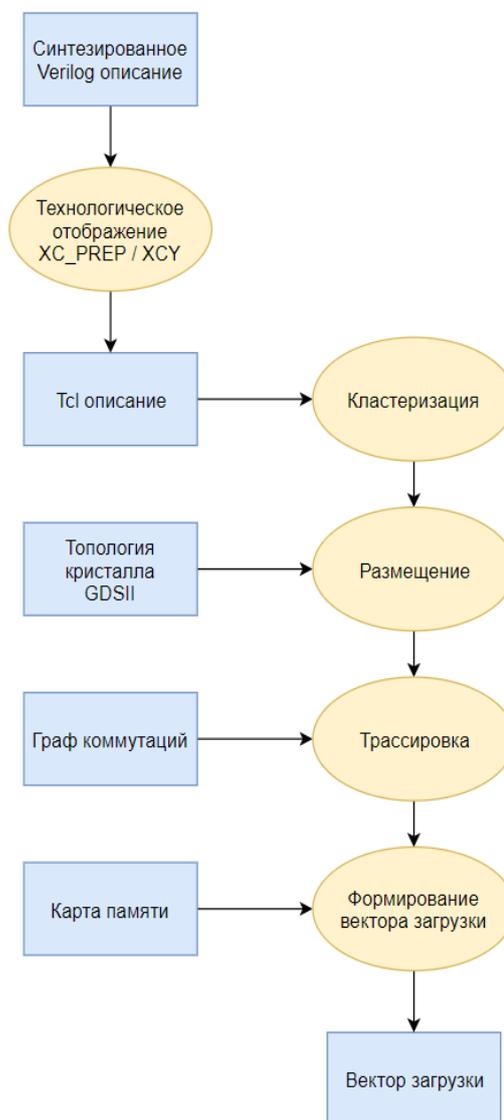


Рис. 4. Основные этапы маршрута проектирования, осуществляемые программными средствами собственной разработки

На рис. 4 показаны этапы маршрута автоматизированного проектирования схем в базисе

ПЛИС, для которых были реализованы собственные программные средства и алгоритмы.

Программный пакет ХСУ выполняет этап технологического отображения в разработанном маршруте проектирования. Программа принимает на вход синтезированное описание схемы на языке Verilog в терминах ТС или библиотечных элементов, и преобразует его в технологически зависимое описание на языке Tc1. Далее список элементов и межсоединений в терминах языка Tc1 передается программе ХС, решающей задачи кластеризации и размещения логических элементов, трассировки межсоединений, конфигурирования и генерации файла прошивки для ПЛИС.

После считывания входного Verilog-описания и создания структуры данных в памяти программы ХСУ начинается процесс ресинтеза схемы в соответствии с архитектурными особенностями целевой ПЛИС по описанным методам.

Процесс ресинтеза выполняется в два этапа путем обхода всех логических элементов проектируемой схемы. При первом обходе, программа ориентируется только на пары ТС-триггер, связанные по информационному входу данных. При нахождении подходящей пары запускается процесс объединения элементов. Предпочтение на первом этапе отдается связям по информационным входам, поскольку от этого напрямую зависит общая длина межсоединений, а также длина критического пути, что в свою очередь оказывает влияние на быстроедействие проектируемой схемы. На втором этапе выполняются проверки на наличие оставшихся возможных связей и оптимизации по следующему приоритету: сигнал сброса (R), установки (S), синхронизации (C).

По завершению процесса ресинтеза программа при необходимости может создать дополнительные периферийные ячейки (недостающие ячейки определяются автоматически на основе входного описания), после чего будут сформированы библиотека логических элементов и структурное описание проектируемой схемы на языке Tc1 в терминах командного интерфейса программы ХС. Также программа, по желанию пользователя, может генерировать Verilog-описание в терминах ТС, прошедших оптимизацию, для логического моделирования схемы после этапа технологического отображения.

В результате работы программы ХС формируется файл прошивки ПЛИС и ряд вспомогательных файлов: файл описания проектируемой схемы на языке SPICE, файл с входными векторами воздействий для моделирования, графическое представление схемы в формате GDS-II и др.

Данный маршрут апробирован в ИПИМ РАН и успешно применяется при проектировании схем для ПЛИС семейства 5510ХС в АО «НИИМЭ» и ПАО «Микрон».

V. РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

Апробация приведенных методов ресинтеза технологического отображения с учетом архитектуры ячейки была произведена как на сборнике наборов тестовых схем IWLS 2005 [6] (куда входят наборы ISCAS'89, ITC'99, Faraday Benchmark и др.), так и на реально проектируемых схемах. Результаты экспериментов отражены в таблице 1.

Таблица 1

Результаты тестирования разработанных методов для "идеальной" ячейки

Набор тестовых схем	Схема	Количество элементов до архитектурной оптимизации	Количество элементов после одиночного объединения	Количество элементов после множественного объединения и объединения по управляющим входам	Улучшение (%)	Суммарное улучшение (%)
ISCAS'89	s382	66	46	45	2,17	31,82
	s400	69	49	48	2,04	30,43
	s444	69	49	48	2,04	30,43
	s641	91	77	74	3,90	18,68
	s713	91	77	74	3,90	18,68
	s5378	576	457	425	7,00	26,22
	s9234	431	347	333	4,04	22,74
	s13207	1204	884	864	2,26	28,24
s38584	4457	3369	3350	0,56	24,84	
ITC'99	b14	2076	1832	1831	0,06	11,80
	b17	12544	11131	11129	0,02	11,28
Faraday	DSP	17470	14076	14000	0,54	19,86
Реальные схемы	test_1	738	561	553	1,43	25,07
	test_2	792	550	539	2,00	31,94
	test_3	657	483	477	1,24	27,40
Среднее значение:					2,21	23,96

Для реализации этапа логического синтеза было использовано свободно распространяемое программное средство логического синтеза – Yosys [7], [8], которое, в свою очередь, для генерации описания схемы в терминах ТС и триггеров использует программный пакет Berkeley ABC [9].

Как видно из таблицы результатом применения методов множественной оптимизации является уменьшение занимаемой проектируемой схемой площади до 7%. Среднее значение по протестированным схемам составляет ~2%. Суммарный (с применением методов этой и прошлой работы) средний выигрыш по площади для рассматриваемых схем составил ~24%. Максимальное значение при этом составило ~32%.

VI. ЗАКЛЮЧЕНИЕ

Разработка методов и алгоритмов архитектурно-ориентированной ресинтеза является очень важной задачей при проектировании схем в базе ПЛИС. Как можно видеть из полученных результатов применение таких методов позволяет в некоторых случаях значительно сократить площадь, занимаемую схемой на ПЛИС.

Очевидно, что результатом применения этих методов также станет и общее сокращение длины межсоединений и, как следствие, сокращение задержек распространения сигналов, что позволяет говорить об общем повышении быстродействия проектируемой схемы.

Рассмотренные методы архитектурно-ориентированного ресинтеза могут быть применены

для любой ПЛИС, имеющей некоторые или все из рассмотренных архитектурных особенностей.

ЛИТЕРАТУРА

- [1] Lipatov, I. A., & Tiunov, I. V. Performance-driven technology mapping for XC5510 family FPGAs. // Young Researchers in Electrical and Electronic Engineering (EIConRus), 2017 IEEE Conference of Russian. IEEE, 2017. P. 477-479.
- [2] Гаврилов С.В., Железников Д.А., Липатов И.А., Тиунов И.В. Маршрут проектирования для отечественных программируемых интегральных схем специального назначения: интеграция с существующими промышленными средствами автоматизированного проектирования и решение проблем импортозамещения // Электронная техника. Серия 3. Микроэлектроника. Москва. 2017. С. 5-11.
- [3] URL: https://www.altera.com/en_US/pdfs/literature/wp/wp-01003.pdf
- [4] URL: https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/hb/cyclone-v/cv_51001.pdf
- [5] URL: <http://www.seas.upenn.edu/~ese170/handouts/FPGA.pdf>
- [6] URL: <http://iwls.org/iwls2005/benchmarks.html> (дата обращения: 22.04.2018)
- [7] Clifford Wolf, Johann Glaser. Yosys - A Free Verilog Synthesis Suite. // Proceedings of the 21st Austrian Workshop on Microelectronics (Austrochip). 2013.
- [8] Glaser, J., & Wolf, C. Methodology and Example-Driven Interconnect Synthesis for Designing Heterogeneous Coarse-Grain Reconfigurable Architectures. // Models, Methods, and Tools for Complex Chip Design. Springer, Cham, 2014. P. 201-221.
- [9] URL: <http://www.eecs.berkeley.edu/~alanmi/abc/> (дата обращения: 22.04.2018)

Development of Methods for Architecturally-Oriented Resynthesis in the Computer-Aided Design Flow for FPGAs

I.V. Tiunov, I.A. Lipatov, D.A. Zheleznikov

Institute for Design Problems in Microelectronics of RAS (IPPM RAS), Zelenograd,
tiunov_i@ippm.ru, lipatov_i@ippm.ru, zheleznikov_d@ippm.ru

Abstract — While a complexity of the circuits designed for FPGAs is increased, the occupied area also grows. This leads to difficulties in the stages of placement and tracing due to the limited traceability of the FPGA resources. Thus, to reduce the used area on the crystal, there is a growing need for the development of logic synthesis algorithms and optimization of the designed circuits. In the previous article, an architecturally-oriented algorithm for optimizing circuits was described, as a result of which the logic elements are merged with triggers on the example of 5510XC FPGAs family. That paper was devoted to the simplest case, when one trigger was connected to one logic element only by data input. In this paper, we propose architecturally-oriented resynthesis algorithm for cases where several elements, both logic and triggers, are connected to the output of a single logical element. The algorithm was tested on IWLS 2005 Benchmarks, and the results were arranged in a comparative table.

In the introduction the quick overview of the problem, the results and the limitations of the previous technology mapping method are given. The second section describes an "ideal" FPGA cell, for which the methods described in this paper can be applied. The real cells of FPGAs may be differ from "ideal", but they have some of its capabilities. For each trigger input, there is a multiplexer in the cell that allows to connect the LUT output to one of the corresponding trigger inputs. This opens the door to further optimizations. In addition, the cell must have two outputs: the output of the trigger and the output directly from the LUT. The third section focuses on improving of the technology mapping method compared to previous work. The essence of the method is that the merge of LUT and the trigger can be performed not only on the data input, but also on the control inputs: clock, reset and set. It is possible due to architectural features of an "ideal" cell. In addition, one more technology mapping method description when several elements are

connected to LUT as load, among which there are one or more triggers, is given. In this case LUT is combined with one or several triggers. It occurs with creating copies of LUT. The remaining elements are connected as a load to the combined element to the output of the LUT. The fourth section demonstrates the real design flow of digital circuits for FPGAs where the described methods have been successfully integrated in. The fifth section presents the results of the work of technology mapping algorithms for the "ideal" FPGA. In the conclusion the results are summarized.

Keywords – technology mapping, logical resynthesis, design flow, digital circuit, CAD, FPGA.

REFERENCES

- [1] Lipatov, I. A., & Tiunov, I. V. Performance-driven technology mapping for XC5510 family FPGAs. // Young Researchers in Electrical and Electronic Engineering (EIconRus), 2017 IEEE Conference of Russian. IEEE, 2017. P. 477-479.
- [2] Gavrilov S.V., Zheleznikov D.A., Lipatov I.A., Tiunov I.V. Marshrut proyektirovaniya dlya otechestvennykh programiruyemykh integral'nykh skhem spetsial'nogo naznacheniya: integratsiya s sushchestvuyushchimi promyshlennymi sredstvami avtomatizirovannogo proyektirovaniya i resheniye problem importozameshcheniya (Design flow for domestic programmable integrated circuits for special purpose: integration with existing computer-aided design systems and solution for problems of import substitution) // Elektronnyaya tekhnika. Seriya 3. Mikroelektronika. 2017. S. 5-11.
- [3] URL: https://www.altera.com/en_US/pdfs/literature/wp/wp-01003.pdf
- [4] URL: https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/hb/cyclone-v/cv_51001.pdf
- [5] URL: <http://www.seas.upenn.edu/~ese170/handouts/FPGA.pdf>
- [6] URL: <http://iwls.org/iwls2005/benchmarks.html> (access date: 22.04.2018)
- [7] Clifford Wolf, Johann Glaser. Yosys - A Free Verilog Synthesis Suite. // Proceedings of the 21st Austrian Workshop on Microelectronics (Austrochip). 2013.
- [8] Glaser, J., & Wolf, C. Methodology and Example-Driven Interconnect Synthesis for Designing Heterogeneous Coarse-Grain Reconfigurable Architectures. // Models, Methods, and Tools for Complex Chip Design. Springer, Cham, 2014. P. 201-221.
- [9] URL: <http://www.eecs.berkeley.edu/~alanmi/abc/> (access date: 22.04.2018)