

Эффективный метод локализации ошибок при проектировании специализированных БИС

Л.Г. Гагарина¹, И.О. Гайдук¹, Е.А. Кремер¹, А.В. Можжухина^{1,2}

¹Национальный исследовательский университет «МИЭТ»,
г. Москва, Россия

²НПК «Технологический центр», г. Москва, Россия

reania@rambler.ru

В настоящее время в отечественных САПР отсутствует подсистема локализации ошибок на этапе функционально-логического проектирования. В работе проанализированы существующие методы САПР и рассмотрен необходимый минимальный функционал для метода локализации ошибок в рамках улучшения САПР «Ковчег». Предложен эффективный метод локализации ошибок на основе требований отечественного технологического процесса и проведенного анализа САПР. Представлена формализация задачи сокращения времени проектирования БИС. Дополнительно приведена схема данных разрабатываемой подсистемы локализации ошибок, учитывающая особенности САПР «Ковчег». К ним, в частности, относятся структура проектов БИС и особенности хранения в них информации. Рассмотрена зависимость времени разработки БИС от степени интеграции микросхемы до и после встраивания в САПР подсистемы локализации ошибок. Сокращение времени разработки БИС при данных упрощениях и усреднениях составляет около 7 % при идеальных условиях работы, что соответственно позволит уменьшить и стоимость микросхемы.

Ключевые слова: функционально-логическое моделирование; БИС; логическое проектирование; САПР

Для цитирования: Гагарина Л.Г., Гайдук И.О., Кремер Е.А., Можжухина А.В. Эффективный метод локализации ошибок при проектировании специализированных БИС // Изв. вузов. Электроника. – 2019. – Т. 24. – № 5. – С. 530–538. DOI: 10.24151/1561-5405-2019-24-5-530-538

Efficient Error Localization Method during Developing Specialized LSI

L.G. Gagarina¹, I.O. Gayduk¹, E.A. Kremer¹, A.V. Mozhukhina^{1,2}

¹National Research University of Electronic Technology, Moscow,
Russia

²SMC «Technological Center», Moscow, Russia

reania@rambler.ru

Abstract: During functional logic modeling in the process of developing specialized large-scale integrated circuits (LSI) various tasks have to be solved. Currently, the domestic CAD systems lack the error localization subsystem at the stage of the functional-logical design. In the work the existing CAD methods have been analyzed, where the necessary minimum functionality for the error localization technique in the context of improving the «Kovcheg» CAD system had been considered. An effective method of localizing errors based on the requirements of the domestic process and the CAD analysis has been proposed. The formalization of the task of reducing the time for designing LSI has been presented. In addition, a diagram of the data of the developed error localization subsystem has been given, taking into account the features of the “Kovcheg” CAD system. These features, in particular, include the structure of the LSI projects and the peculiarities of storing information in them. When considering the dependence of the LSI development time on the degree of integration of the chip before and after embedding the error localization subsystem in the CAD system, the theoretical shortening of the LSI development time with these simplifications and averages is about 7% in ideal working conditions.

Keywords: LSI; visualization; structural scheme; CAD; FLM

For citation: Gagarina L.G., Gayduk I.O., Kremer E.A., Mozhukhina A.V. Efficient error localization method during developing specialized LSI. *Proc. Univ. Electronics*, 2019, vol. 24, no. 5, pp. 530–538. DOI: 10.24151/1561-5405-2019-24-5-530-538

Введение. В настоящее время одно из приоритетных направлений в изготовлении отечественных БИС – совершенствование САПР, в том числе за счет улучшения подсистем тестирования и отладки.

При разработке современных БИС используется широкий спектр разноплановых подсистем проектирования. В НПК «Технологический центр» (г. Москва) в маршрут изготовления микросхем входят следующие этапы: системное проектирование; функциональное проектирование; логическое проектирование; физическое (топологическое) проектирование; производство БИС; тестирование БИС в составе пластины и в составе корпуса; квалификационные испытания БИС; испытания БИС в составе аппаратуры заказчика. Поскольку при поиске ошибок на каждом из этапов могут возникнуть дополнительные временные задержки и возврат на предыдущие этапы, то процесс создания микросхем на сложных проектах значительно замедляется.

Многие отечественные САПР на сегодняшний день в полной мере не удовлетворяют всем потребностям в сфере разработки БИС. Наиболее полной и доступной САПР является «Ковчег» [1]. Однако вследствие отсутствия подсистемы локализации ошибок

на этапе логического проектирования разработчикам приходится обращаться к дорогостоящим зарубежным аналогам. Это может приводить не только к встраиванию в электрическую схему аппаратных закладок, но и к утечке информации и разрабатываемых проектов, поэтому необходимо создание собственной методики и подсистемы локализации ошибок в проектах специализированных БИС.

Анализ методик в современных действующих САПР. На основе анализа данных о различных САПР для разработки БИС выбраны две крупные фирмы – Cadence и Synopsys, а также Aldec, Xilinx, Altera и Icarus. В таблице представлены результаты анализа методик САПР выбранных компаний на основе требований отечественных разработчиков и технологического процесса.

**Преимущества и недостатки существующих методик поиска ошибок
на этапе логического проектирования
Advantages and disadvantages of existing methods for finding errors
at the stage of logical design**

Методика	Преимущества методики	Недостатки методики
Cadence NC-Verilog [2]	<ul style="list-style-type: none"> • Обширный набор средств и настроек для проверки модели микросхемы. • Визуальное отображение схемы. • Возможность раскрытия схемы от одного элемента. • Возможность просмотра значения сигнала на входах и выходах в процессе моделирования 	<ul style="list-style-type: none"> • Работает только под Unix-подобными системами. • Вся информация о среде и сама среда только на английском языке. • Освоение системы управления функциями среды занимает много времени. • Не содержит в себе необходимого спектра функций для сквозного проектирования микросхемы
Synopsys VCS и DVE [3]	<ul style="list-style-type: none"> • Возможность отображения информации о микросхеме в разных представлениях (ASCII таблица моделирования и диаграмма сигналов). • Визуальное отображение схемы. • Возможность отслеживания пути сигнала в схеме БИС в двух режимах – «Path» и «Design» 	<ul style="list-style-type: none"> • Совместимость только с Unix- системами. • Среда и пользовательские инструкции на английском языке. • Просмотр схемы целиком без возможности свертки/развертки от элемента. • Не содержит в себе необходимого спектра функций для сквозного проектирования микросхемы
Aldec Active-HDL [4]	<ul style="list-style-type: none"> • Возможность внесения изменений и правок в проект в процессе моделирования и поиска ошибок. • Визуальное отображение схемы. • Возможность просмотра значения сигнала на входах и выходах элементов схемы в процессе моделирования 	<ul style="list-style-type: none"> • Инструкции по работе со средой и сама среда только на английском языке. • Просмотр схемы целиком без возможности свертки/развертки от элемента
Xilinx ISE [5]	<ul style="list-style-type: none"> • Возможность внесения изменений и правок в проект в процессе моделирования и поиска ошибок. • Возможность отслеживания пути сигнала в схеме микросхемы 	<ul style="list-style-type: none"> • Визуальное отображение схемы без отображения сигнала. • Просмотр схемы целиком без возможности свертки от одного элемента. • Множество различных подсистем, входящих в состав среды, не нужных на этапе поиска ошибок в проекте микросхемы. • Информация по работе со средой и сама среда только на английском языке

Методика	Преимущества методики	Недостатки методики
Altera ModelSim [6]	<ul style="list-style-type: none"> • Возможность внесения изменений и правок в код проекта в процессе моделирования и поиска ошибок. • Совместимость с Windows. • Визуальное отображение схемы микросхемы через Quartus II 	<ul style="list-style-type: none"> • Инструкции по работе со средой и сама среда только на английском языке. • Просмотр схемы целиком без возможности свертки от одного элемента. • Бесплатная версия ограничена в функционале, полная версия платная
Icarus Verilog [7]	<ul style="list-style-type: none"> • Бесплатная версия 	<ul style="list-style-type: none"> • Правки осуществляются внешними программистами, что ставит под угрозу данные проекта. • Инструкции по работе со средой и сама среда только на английском языке. • Совместимость только с Unix-подобными системами. • Полностью отсутствует какое-либо графическое отображение схемы

Цель настоящей работы – разработка методики локализации ошибок на этапе логического проектирования для дальнейшего использования при создании соответствующей подсистемы САПР «Ковчег».

Формализация проблемы локализации ошибок в проекте БИС. Для четкого понимания того, что именно необходимо улучшить в процессе разработки БИС, следует подробно рассмотреть формализацию поставленной задачи.

Основная формула расчета стоимости вывода микросхемы на рынок имеет вид

$$C = F(\tau, \theta) \rightarrow \min,$$

где F – функциональная зависимость; τ – время разработки БИС; θ – время производства БИС.

Поскольку использование САПР происходит на этапах разработки, то следует более подробно рассмотреть составляющие τ . В процесс разработки БИС включаются системное, функциональное, логическое и топологическое проектирование, а также фиксированное время на квалификационные испытания и испытания в аппаратуре заказчика. Необходимо учитывать, что с любого этапа возможен возврат для внесения правок. При этом все операции технологического маршрута между двумя этапами придется повторить:

$$\tau = \sum_{i=1}^n \frac{T_i}{K_i} N_i + \begin{cases} \max(T_{ex}, T_{exc}) \\ T_{ex} + T_{exc} \end{cases}, \quad (1)$$

где n – количество этапов разработки БИС; T_i – трудоемкость i -го этапа разработки; K_i – количество специалистов, выполняющих работу на i -м этапе; N_i – количество повторений i -го этапа разработки; T_{ex} – квалификационные испытания; T_{exc} – испытания в составе аппаратуры заказчика. Квалификационные испытания могут проводиться параллельно с испытаниями в составе аппаратуры заказчика либо отдельно в течение определенного времени, оговариваемого с заказчиком [8]. Принято, что в формуле (1) берется максимум из T_{ex} , T_{exc} или их сумма. В n учитывается базовое количество этапов разработки микросхемы. В N_i хранится число повторов каждого из этапов. Оно зависит от количества ошибок и сложности требований к изготавливаемой микросхеме.

В трудоемкости любого из этапов разработки учитывается назначение микросхемы. Например, если БИС будет использоваться заказчиком в космосе, то по сравнению с микросхемами обычного назначения у нее должны быть повышенная отказоустойчивость, большой срок работы и хранения, пониженное энергопотребление и др. В зависимости от этого варьируются требования к БИС.

Трудоемкость i -го этапа разработки выражается формулой

$$T_i = \frac{S(D, E)}{P_i},$$

где S – сложность разработки БИС, т.е. функциональная зависимость от E – степени интеграции БИС и D – коэффициента усложнения в зависимости от класса (назначения); P_i – производительность труда на i -м этапе разработки.

Степень интеграции БИС вычисляется по формуле

$$E = \lg U,$$

где U – количество элементов в составе БИС.

Наиболее важным параметром для рассмотрения является P_i [9]:

$$P_i = \frac{Q(1 - k_p)}{t_{ki} K_i},$$

где Q – объем продукции; k_p – коэффициент простоев; t_{ki} – трудозатраты работников на i -м этапе. В трудозатратах работников учитывается наличие и совершенствование подсистем САПР, используемых для разработки.

Методика локализации ошибок в проекте БИС. На основе требований отечественного технологического процесса и проведенного анализа САПР разработана методика локализации ошибок, содержащая основные этапы.

Этап 1. Обнаружение ошибок при сравнении результатов функционально-логического моделирования (ФМЛ) и требуемой реакции на выводах БИС (например, на выходе необходимо получить определенный сигнал – единицу, а в результате моделирования получено неопределенное состояние).

Этап 2. Определение необходимого количества контрольных точек, число которых зависит от сложности микросхемы. Контрольная точка – файл, содержащий состояния всех контактов элементов логической схемы БИС в выбранный момент времени. Чем сложнее микросхема, тем сложнее с первого цикла разработки определить причину ошибок.

Этап 3. Расстановка контрольных точек и запуск моделирования. Каждая контрольная точка имеет свое уникальное имя.

Этап 4. Визуализация логической схемы БИС и состояний контактов в контрольных точках. Сначала подключается одна из контрольных точек, выбранных на этапе 3, и выбирается один элемент схемы. От выбранного элемента через соединения разворачивается остальная схема. Около выводов подписываются значения сигналов из выбранной контрольной точки. При необходимости можно сменить контрольные точки (перейти к этапу б).

Этап 5. При успешной локализации проблемного элемента осуществляются правки в схеме.

Этап 6. При неудачной локализации проблемного элемента выбирается другая контрольная точки и происходит возврат к этапу 4.

Этап 7. При переборе всех доступных контрольных точек из этапа 7 необходимо вернуться на этап 3 и выбрать другой набор контрольных точек. В случае высокой сложности БИС требуется большое количество повторов всех этапов при постепенном сужении области поиска проблемного элемента.

Особенности подсистемы локализации ошибок для САПР «Ковчег». Для создания эффективной подсистемы локализации ошибок для САПР «Ковчег» на этапе логического проектирования необходимо наличие определенных функций, в том числе графической визуализации схемы, связи моделирования с визуализированной схемой и развертывания схемы от одного элемента.

Первая функция должна находить в библиотеке элементов файл с графическим отображением элемента и выводить его на экран в определенном месте. В этой же функции выбирается первый элемент для дальнейшего развертывания схемы. Также необходимы несколько переменных для хранения специфических данных:

- список графических элементов, в котором хранится информация об уже выведенных на экран графических элементах схемы;

- матрица, представляющая собой структуру данных для расчетов, содержащую схематичное отображение элементов схемы в графическом окне, в частности информацию о занятости точки поля графическим элементом и его характеристиках. Эта структура нужна для правильной расстановки и трассировки элементов.

Кроме того, необходима функция для состыковки подсистемы моделирования с подсистемой локализации ошибок с помощью контрольных точек. Таким образом, после выбора контрольной точки в меню подсистемы из файла считывается информация о состоянии всех контактов, выводится на экран над контактами визуализированных элементов схемы и вносится в поле. Функция развертывания схемы от одного элемента необходима для минимизации выведения ненужных элементов на экран. В результате разработчик выводит на экран только необходимую для проверок часть микросхемы.

Для корректной работы при выводе новых элементов на экран функция отображения элементов всегда должна сверяться с матрицей (свободна ли область) и предусматривать механизмы выбора другого места, если область уже занята. Сюда же включен алгоритм соединения элементов. В процессе трассировки проверяется возможность проведения соединения и занесение проведенных линий в поле. За основу алгоритма трассировки взят волновой алгоритм (алгоритм Ли) ввиду его высокой результативности [10].

Из дополнительных функций специалисты указывают на необходимость функций отслеживания пути; удаления из графического окна каких-либо ранее выведенных элементов; поиска и выведения на экран ближайших триггеров в цепи по контакту какого-либо элемента. Последняя поможет выяснить, с какого триггера в цепи за счет задержек и ошибок в схеме подается неправильный сигнал (все данные не успели установиться на всех элементах схемы при переключении основного сигнала clock за период).

При разработке схем алгоритмов функций необходимо учитывать специфику хранения информации об элементах схемы, а также проектов БИС, особенности реализации структур данных и специфику хранения данных контрольных точек. При создании подсистемы локализации ошибок в проектах БИС также проанализированы данные и составлена схема [11], представленная на рисунке (где УГО – условные графические обозначения). Она, согласно ГОСТу, отражает взаимосвязи подсистем моделирования и локализации ошибок. Методика локализации ошибок частично выполняется в подсистеме моделирования, где можно проследить взаимосвязи перечисленных функций с требуемой для их работы информацией. Для работы подсистемы в проекте БИС должна содержаться информация обо всех условных графических обозначениях и контрольных точках. Разработчик сам выбирает первый элемент, контрольные точки и схему, на базе которых осуществляется работа подсистемы.

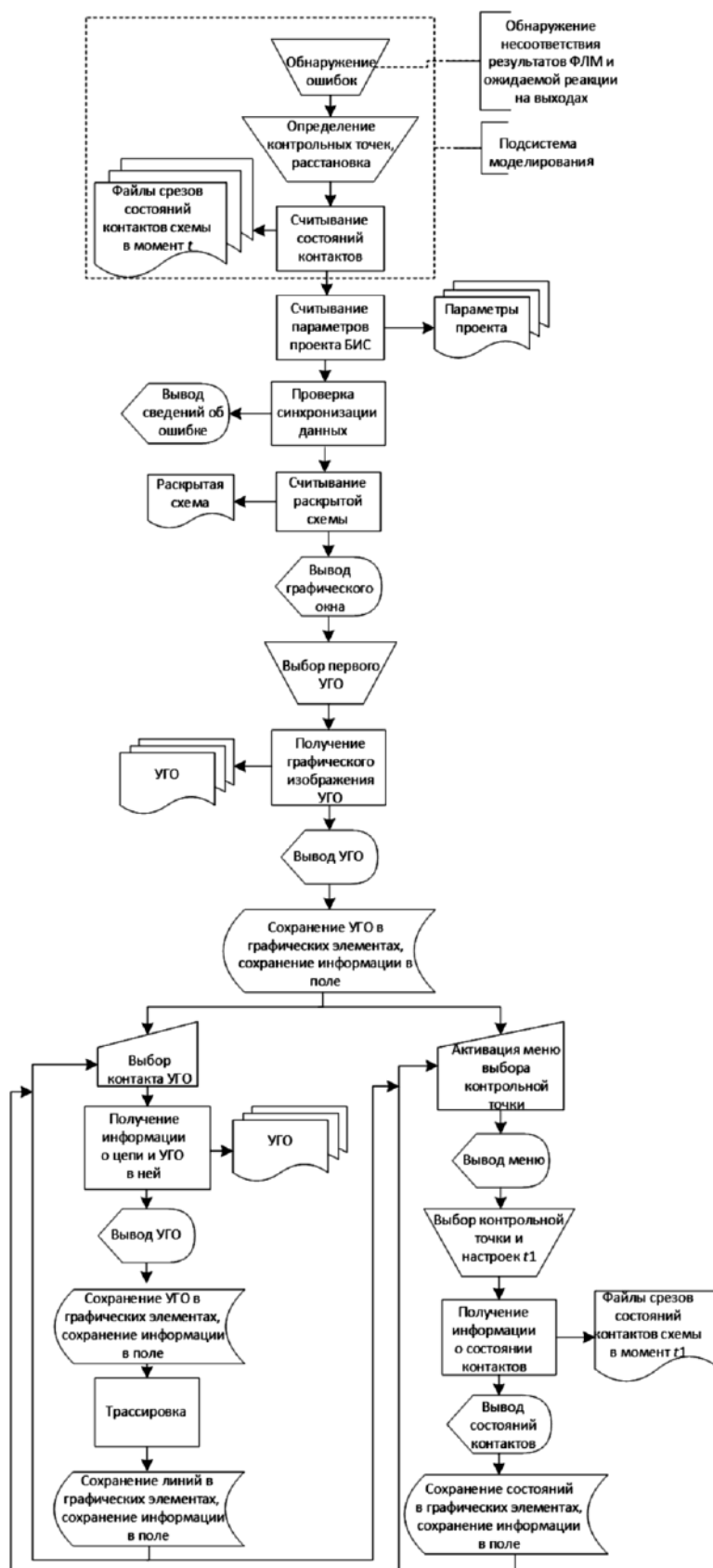


Схема данных подсистемы локализации ошибок
Data scheme of the error localization subsystem

Рассмотрена зависимость времени разработки САПР от степени интеграции микросхемы до и после встраивания подсистемы локализации ошибок. С учетом определенных упрощений и принятия рабочего дня, равного 8 ч, при всех идеальных значениях на разработку микросхемы тратится $\tau = \sum_1^4 8 + 1$ неделя + 1 неделя = $8 \cdot 4 + 8 \cdot 5 \cdot 2 = 112$ ч, или 14 рабочих дней. С учетом времени производства БИС θ и минимальных производственных рисков итоговое время вывода микросхемы на рынок примерно один месяц.

При более детальном рассмотрении зависимости времени разработки БИС от степени интеграции микросхемы до и после встраивания в САПР подсистемы локализации ошибок сокращение времени разработки БИС при данных упрощениях и усреднениях составляет около 7 % при идеальных условиях работы. Процент обращений к подсистеме локализации при этом от 9 до 15 % от общего числа повторения этапов разработки N_i и n .

Заключение. В соответствии с проведенными расчетами смоделирован процесс вывода микросхем в среде MatLab на рынок.

В формализации проблемы локализации ошибок в проекте БИС улучшение происходит за счет сокращения времени и снижения трудозатрат специалистов при локализации ошибок. При снижении трудоемкости t_{ki} сокращается время этапа логического проектирования T_i . Это достигается путем использования совместной визуализации структурной схемы и результатов ФЛМ. Таким образом, сокращается время разработки БИС τ , а также уменьшается число повторов этапов N_i . Разработанный метод позволит уменьшить время разработки микросхемы, соответственно, и ее стоимость C .

Литература

1. *Гаврилов С.В., Денисов А.Н., Коняхин В.В.* Система автоматизированного проектирования «Ковчег 2.1» / под ред. Ю.А. Чаплыгина. – М.: Микрон-Принт, 2001. – 210 с.
2. *Manikas T., Thornton M.* Tutorial for cadence simvision verilog simulator tool. – Southern Methodist University, 2013. – 9 p.
3. *Hima Bindu Kommuru, Hamid Mahmoodi.* ASIC design flow tutorial using synopsys tools. – San Francisco: State University San Francisco, 2009. – 124 p.
4. *Шалагинов А.* Изучаем Active-HDL // Компоненты и технологии. – URL: http://www.kit-e.ru/articles/circuit/2009_03_134.php (дата обращения: 18.09.2018).
5. Synthesis and simulation design guide. – Xilinx, 2007. – 218 p.
6. ModelSim-altera software simulation. User Guide. – Altera, 2013. – 12 p.
7. User Guide // Icarus Verilog. – URL: http://iverilog.wikia.com/wiki/User_Guide (дата обращения: 24.09.2018).
8. ГОСТ 16504-81. Испытания и контроль качества продукции. – М., 1991. – 24 с.
9. Основные показатели и формула для расчета производительности труда // Делать Дело. Кадровое делопроизводство. – URL: <https://delatdelo.com/spravochnik/osnovy-biznesa/proizvoditelnost/truda-formula-gascheta.html> (дата обращения: 24.10.2018).
10. Волновой алгоритм поиска пути. Статьи и студенческие работы. – URL: <http://www.100byte.ru/100btwrks/wv/wv.html> (дата обращения: 24.10.2018).
11. ГОСТ 19.701-90. Схемы алгоритмов, программ, данных и систем. – М., 1990. 26 с.

Поступила в редакцию 24.04.2019 г.; после доработки 29.04.2019 г.; принята к публикации 18.06.2019 г.

Гагарина Лариса Геннадьевна – доктор технических наук, профессор, директор Института системной и программной инженерии и информационных технологий Национального исследовательского университета «МИЭТ» (Россия, 124498, г. Москва, г. Зеленоград, пл. Шокина, 1), incos@miee.ru

Гайдук Игорь Олегович – ассистент Института системной и программной инженерии и информационных технологий Национального исследовательского университета «МИЭТ» (Россия, 124498, г. Москва, г. Зеленоград, пл. Шокина, 1), igolgai@mail.ru

Кремер Евгений Александрович – аспирант Института системной и программной инженерии и информационных технологий Национального исследовательского университета «МИЭТ» (Россия, 124498, г. Москва, г. Зеленоград, пл. Шокина, д.1), kremerea@gmail.com

Можжухина Арина Валерьевна – магистр Института системной и программной инженерии и информационных технологий Национального исследовательского университета «МИЭТ» (Россия, 124498, г. Москва, г. Зеленоград, пл. Шокина, 1), инженер НПК «Технологический центр» (124498, г. Москва, г. Зеленоград, пл. Шокина, 1), reania@rambler.ru

References

1. Gavrillov S.V., Denisov A.N., Konyahin V.V. Ed. By Chaplugin Y.A. *CAD system «Kovcheg 2.1»*. Moscow, Micron-Print Publ., 2001. 210 p. (in Russian).
2. Manikas T., Thornton M. *Tutorial for Cadence simvision verilog simulator tool*. Southern Methodist University, 2013. 9 p.
3. Hima Bindu Kommuru, Hamid Mahmoodi. *ASIC Design flow tutorial using Synopsys tools*. Nano-Electronics & Computing Research Lab School of Engineering San Francisco State University San Francisco, CA, 2009. 124 p.
4. Shalaginov A. Studying Active-HDL. *Components and Technology*. Available at: http://www.kite.ru/articles/circuit/2009_03_134.php (accessed: 18.09.2018). (in Russian).
5. *Synthesis and Simulation Design Guide*. Xilinx, 2007. 218 p.
6. *ModelSim-Altera Software Simulation*. User Guide. Altera, 2013. 12 p.
7. *Icarus Verilog. User Guide: Digest of articles on using Icarus Verilog*. Available at: http://iverilog.wikia.com/wiki/User_Guide (accessed: 24.09.2018).
8. *GOST 16504-81. Testing and quality control of products*. Moscow, 1991. 24 p. (in Russian).
9. Key indicators and formula for calculating productivity. *To do business. Online journal, personnel office work*. Available at: <https://delatdelo.com/spravochnik/osnovy-biznesa/proizvoditelnost/truda-formularascheta.html> (accessed: 24.10.2018). (in Russian).
10. *Wave algorithm for finding the path. Articles and student work*. Available at: <http://www.100byte.ru/100btwrks/wv/wv.html> (accessed 24.10.2018). (in Russian).
11. *GOST 19.701-90. Schemes of algorithms, programs, data and systems*. Moscow, 1990. 26 p. (in Russian).

Received 24.04.2019; Revised 29.04.2019; Accepted 18.06.2019.

Information about the authors:

Larisa G. Gagarina – Dr. Sci. (Eng.), Prof., Director of the Institute of System and Computer Science and Information Technology, National Research University of Electronic Technology (Russia, 124498, Moscow, Zelenograd, Shokin sq., 1), incos@miee.ru

Igor O. Gayduk – Assistant of the Institute of System and Computer Science and Information Technology, National Research University of Electronic Technology (Russia, 124498, Moscow, Zelenograd, Shokin sq., 1), igolgai@mail.ru

Evgeny A. Kremer – PhD student of the Institute of System and Computer Science and Information Technology, National Research University of Electronic Technology (Russia, 124498, Moscow, Zelenograd, Shokin sq., 1), kremerea@gmail.com

Arina V. Mozzhukhina – Master's Degree of the Institute of System and Computer Science and Information Technology, National Research University of Electronic Technology (Russia, 124498, Moscow, Zelenograd, Shokin sq., 1), Engineer, SIC «Technological Center» (Russia, 124498, Moscow, Zelenograd, Shokin sq., 1), reania@rambler.ru