

УДК 621.3.049.77

ВЕРИФИКАЦИЯ ПОВЕДЕНЧЕСКИХ МОДЕЛЕЙ БЛОКОВ ИНТЕГРАЛЬНЫХ СХЕМ С ПОМОЩЬЮ СРЕДСТВ ПРОГРАММИРУЕМОЙ ЛОГИКИ

Н.М. Чернецкая^a, А.А. Михтеева^{a,b}, Н.Н. Невирковец^{a,c}, Д.В. Костыгов^a, Я.В. Беляев^a

^a АО «Концерн «ЦНИИ «Электронприбор», Санкт-Петербург, 197046, Российская Федерация

^b Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация

^c Санкт-Петербургский государственный электротехнический университет (ЛЭТИ) им. В.И. Ульянова (Ленина), Санкт-Петербург, 197376, Российская Федерация

Адрес для переписки: nevirkovets.nn@gmail.com

Информация о статье

Поступила в редакцию 21.01.18, принята к печати 27.03.18

doi: 10.17586/2226-1494-2018-18-3-479-486

Язык статьи – русский

Ссылка для цитирования: Чернецкая Н.М., Михтеева А.А., Невирковец Н.Н., Костыгов Д.В., Беляев Я.В. Верификация поведенческих моделей блоков интегральных схем с помощью средств программируемой логики // Научно-технический вестник информационных технологий, механики и оптики. 2018. Т. 18. № 3. С. 479–486. doi: 10.17586/2226-1494-2018-18-3-479-486

Аннотация

Предмет исследований. Исследованы методы верификации поведенческих моделей с использованием средств программируемых логических интегральных схем. Показано, что применение верификации с использованием средств программируемой логики позволяет выявить функциональные ошибки, не определяемые на этапе верификации с помощью традиционного маршрута проектирования интегральных схем. **Метод.** Предложен подход к двухэтапному прототипированию интерфейсных блоков интегральных схем с помощью средств программируемой логики с использованием стандартных IP-блоков и внешних устройств. Предложено добавление дополнительного этапа верификации, осуществляемого после выполнения верификации согласно традиционному маршруту проектирования интегральных схем в системе автоматического проектирования Cadence. **Основные результаты.** Подход использован для верификации блока последовательного периферийного интерфейса в составе интегральной схемы микромеханического акселерометра. В результате верификации модели интерфейсного блока определено, что первый этап прототипирования с использованием стандартных интерфейсных IP-блоков позволяет выявить имеющиеся функциональные ошибки в устройстве с минимальными временными затратами. При отсутствии стандартных IP-блоков модель верифицирующего устройства необходимо разработать отдельно, что может привести к возникновению рисков неработоспособности конечного устройства. Второй этап прототипирования с использованием внешнего подключаемого верифицирующего устройства позволил избавиться от ошибок, связанных с задержками распространения сигнала вне интегральной схемы, и избежать ограничений, связанных с отсутствием необходимых IP-блоков.

Практическая значимость. Двухэтапное прототипирование может быть использовано при проектировании интерфейсных блоков интегральных схем с целью минимизации вероятности появления ошибок при передаче данных. По результатам двухэтапного прототипирования модели блока последовательного периферийного интерфейса выявлены и исправлены функциональные ошибки, которые не были выявлены с помощью этапа верификации традиционного маршрута проектирования. Спроектированная с помощью предлагаемого подхода модель использована при разработке интегральной схемы для микромеханического акселерометра.

Ключевые слова

интегральные схемы, ПЛИС, верификация, прототипирование, поведенческая модель, интерфейс

Благодарности

Работа выполнена при поддержке гранта РФФИ 16-08-000640.

VERIFICATION OF INTEGRATED CIRCUIT BEHAVIORAL MODELS BY PROGRAMMABLE LOGIC

N.M. Chernetskaya^a, A.A. Mikhteeva^{a,b}, N.N. Nevirkovets^{a,c}, D.V. Kostygov^a, Y.V. Belyaev^a

^a Concern "CSRI Elektropribor", Saint Petersburg, 197046, Russian Federation

^b ITMO University, Saint Petersburg, 197101, Russian Federation

^c Saint Petersburg Electrotechnical University (LETI), Saint Petersburg, 197376, Russian Federation

Corresponding author: nevirkovets.nn@gmail.com

Article info

Received 21.01.18, accepted 27.03.18
 doi: 10.17586/2226-1494-2018-18-3-479-486
 Article in Russian

For citation: Chernetskaya N.M., Mikhiteeva A.A., Nevirkovets N.N., Kostygov D.V., Belyaev Y.V. Verification of integrated circuit behavioral models by programmable logic. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2018, vol. 18, no. 3, pp. 479–486 (in Russian). doi: 10.17586/2226-1494-2018-18-3-479-486

Abstract

Subject of Research. The paper considers the research of verification methods for behavioral models by the field-programmed gate array (FPGA). Applying verification with FPGA gives the possibility to identify functional errors, which are not determined at the verification phase of the traditional integrated circuits (IC) design. **Method.** The approach is recommended to two-stage IC interface blocks prototyping with FPGA by means of standard IP-blocks and external devices. The addition of an extra verification phase was proposed carried out after the verification phase according to the traditional IC design path in the Cadence automated design system. **Main Results.** The approach was used to verify the block of the serial peripheral interface (SPI), which was included in the IC of the micromechanical accelerometer. The result of the interface block model verification showed that the first stage of prototyping with the use of standard interface IP blocks gives the possibility to reveal the existing functional errors in the device with minimal time. Without standard IP-blocks, the model of the verification device ought to be developed separately that can lead to malfunction risks of the final device. The second prototyping stage applying an external plug-in verification device makes it possible to get out of errors connected with signal propagation delays outside the IC and to avoid limitations connected with lack of necessary IP-blocks. **Practical Relevance.** Two-stage prototyping can be used in the design of IC interface blocks with a view to minimize the probability of errors in data transmission. Functional errors not detected during the verification phase of the traditional IC design were identified and corrected based on the results of two-stage prototyping of the SPI block model. The model designed by this approach was used to develop an IC for a micromechanical accelerometer.

Keywords

integrated circuits, FPGA, verification, prototyping, behavioral model, interface

Acknowledgements

This work was supported by the project no. 16-08-00640 of the Russian Foundation for Basic Research, Russian Federation.

Введение

На сегодняшний день степень интеграции интегральных схем (ИС) растет экспоненциально, что позволяет разместить на кристалле проект практически любой сложности [1]. Несмотря на успехи в развитии технологий создания ИС все более высокой степени интеграции, актуальной проблемой является ее верификация на этапе проектирования. Исследования [2] показывают, что выход годных ИС на первом этапе составляет всего 33%, причем наиболее частыми причинами неработоспособности являются функциональные ошибки. Данные ошибки возникают на этапе разработки поведенческих моделей на языках описания аппаратуры и не могут быть полностью определены на этапе верификации традиционного маршрута проектирования в таких программных средах, как Cadence, Synopsys и Mentor Graphics. Следовательно, необходимо дополнительно исследовать модели с целью минимизации возможных функциональных ошибок.

Одним из возможных решений является выполнение прототипирования – реализации блоков ИС с помощью средств программируемой логики [3–5]. Прототипирование разрабатываемого устройства в программируемой логической интегральной схеме (ПЛИС) является одним из способов проверки корректности его функционирования и позволяет определить работоспособность поведенческой модели, синтезированной в логике, на реальных задачах в устройстве. Решения для ПЛИС-прототипирования предлагают компании, специализирующиеся на разработке средств автоматизированного проектирования (САПР) для микроэлектроники, такие как Cadence, Synopsys и другие. Однако эти решения имеют большую стоимость и предназначены для верификации крупных цифровых проектов.

Для создания прототипов ИС или отдельных их блоков разработчики зачастую используют ПЛИС таких компаний, как Altera, Xilinx [6]. Причем тестовое окружение, разработанное на этапе верификации в САПР, адаптируется для возможности его использования в ПЛИС.

Основной задачей прототипирования является поиск функциональных ошибок в поведенческой модели, связанных с взаимодействием с внешними устройствами. Однако общая методика разработки прототипов для отдельных блоков интегральных схем отсутствует.

Для интерфейсных блоков ИС предлагается подход к использованию двухэтапного прототипирования с целью устранения возможных функциональных ошибок. В работе приведен анализ особенностей верификации двух вариантов конфигурации прототипов поведенческих моделей интегральных схем с использованием ПЛИС на примере последовательного интерфейса SPI (Serial Peripheral Interface).

Верификация поведенческих моделей

Верификация поведенческой модели цифрового блока ИС заключается в выполнении тестов и анализе их результатов. Для этого необходимо определить типы возможных ошибок, создать верификационный план и разработать тестовое окружение, включающее в себя набор тестов, генерирующих входные

воздействия и анализирующих выходные сигналы. Верификация разрабатываемой модели выполняется с помощью компьютерного моделирования в САПР интегральных схем. При обнаружении несоответствия полученных и ожидаемых сигналов в процессе верификации следует выявить причину возникновения ошибки, после чего скорректировать поведенческую модель и повторить моделирование. Верификация цифровых поведенческих моделей является итеративным процессом, что подразумевает повторное тестирование после каждого изменения поведенческой модели [7]. После исправления ошибки необходимо повторять тест, который ее обнаружил. Особенно важно обеспечить полноту тестового окружения, чтобы гарантировать исчерпывающую проверку функционирования модели блока.

Полнота тестового окружения определяется количественными показателями следующих метрик [8]: покрытие строк кода, покрытие операторов, покрытие переходов управления, покрытие состояний автомата, покрытие входных последовательностей. Численное значение метрики покрытия – это отношение числа достигнутых (покрытых) ситуаций к общему числу ситуаций в модели [9]. Программные инструменты для сбора и анализа покрытия тестируемого блока, такие как Cadence Coverage Tool [10], входят в большинство современных САПР.

Для поведенческих моделей цифровых блоков ИС различают следующие классы ошибок:

- функциональные;
- семантические;
- ошибки использования конструкций языков описания аппаратуры;
- синтаксические ошибки;
- временные ошибки, возникающие вследствие появления задержек при передаче данных.

Среди функциональных ошибок в интерфейсных блоках можно выделить следующие ошибки:

- нарушение целостности переданных данных;
- ошибки синхронизации данных на разных частотах;
- некорректная интерпретация переданных данных;
- несоответствие требуемому режиму работы устройства.

При верификации поведенческих моделей в программных средах с использованием тестового окружения однозначно можно исключить лишь ошибки синтаксиса и некоторые ошибки использования конструкций языка описания аппаратуры. Не могут быть однозначно определены некоторые семантические, временные и функциональные ошибки, ошибки возникновения триггеров-защелок [11], ошибки неправильного определения типа данных и др.

Блоки с высокой степенью надежности и непосредственно отвечающие за передачу данных, например, интерфейсные блоки, с целью минимизации функциональных ошибок необходимо исследовать дополнительно. Для этого вводится дополнительный этап верификации с использованием средств ПЛИС, осуществляемый после выполнения этапа верификации традиционного маршрута проектирования в САПР Cadence (рис. 1).

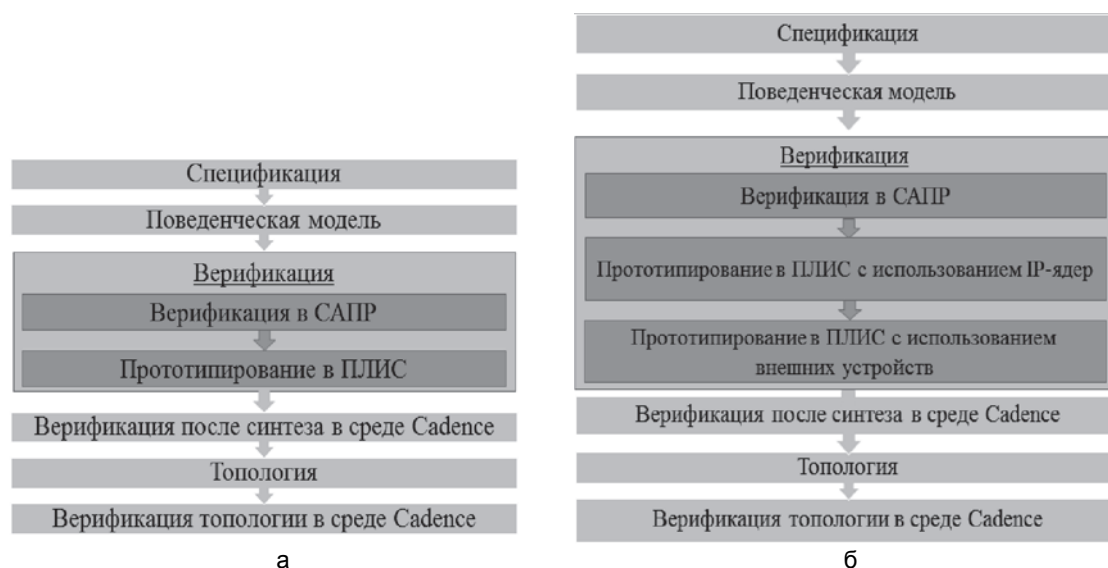


Рис. 1. Место верификации с помощью программируемых логических интегральных схем в традиционном маршруте проектирования интегральных схем (а); предлагаемый подход к двухэтапному прототипированию с помощью программируемых логических интегральных схем для интерфейсных блоков (б)

Для верификации интерфейсных блоков ИС предлагается использовать два этапа прототипирования на ПЛИС:

1. прототипирование с использованием средств ПЛИС и сложнофункциональных блоков (IP-блоков (Intellectual Property), IP-ядер);
2. прототипирование с использованием средств ПЛИС с подключением внешних устройств.

Первый вариант конфигурации прототипа позволяет провести функциональную верификацию разрабатываемого устройства, а также проверить возможность синтеза конечной схемы. Однако в этом случае сигнал не выходит за пределы кристалла ПЛИС. Таким образом, практически не учитываются временные задержки, связанные с временем распространения сигнала и, как следствие, возможное появление ошибок. Применение второго варианта конфигурации прототипа интерфейсного блока позволяет учесть вышеприведенный недостаток. В качестве внешнего устройства может быть использован микроконтроллер (МК) с реализованным в нем интерфейсом.

Для анализа особенностей верификации двух вариантов конфигурации прототипов использована модель последовательного интерфейса SPI [12], предназначенного для обмена данными с блоком коррекции аналого-цифрового преобразователя ИС для микромеханического акселерометра. Разработанная поведенческая модель блока интерфейса SPI выступает в роли ведомого устройства, реализуя функцию последовательной передачи и приема бит. Полученные от внешнего ведущего устройства данные конфигурируют остальные блоки ИС. Пакет передаваемых данных содержит 16 бит.

Верификация модели интерфейсного блока в САПР

Первый этап верификации поведенческой модели блока интерфейса с помощью тестового окружения выполнен в программной среде Cadence Incisive. Для проведения верификации было разработано тестовое окружение для проверки корректности принимаемых и передаваемых данных. В состав тестового окружения входит модель ведущего блока интерфейса, формирующая случайную посылку данных. Временные диаграммы моделирования блока интерфейса (рис. 2) показали успешное выполнение разработанных тестов. Анализ тестового покрытия подтвердил 100% полноту тестового окружения (рис. 3).

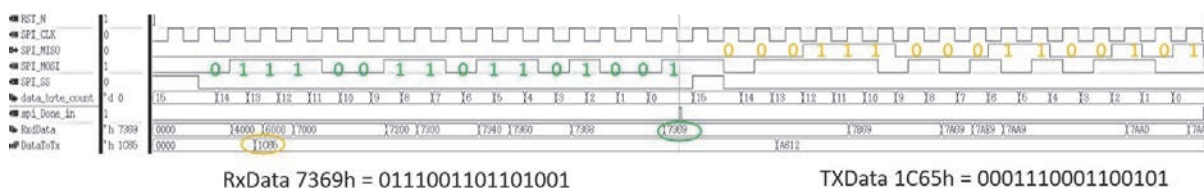


Рис. 2. Временная диаграмма функционирования блока интерфейса

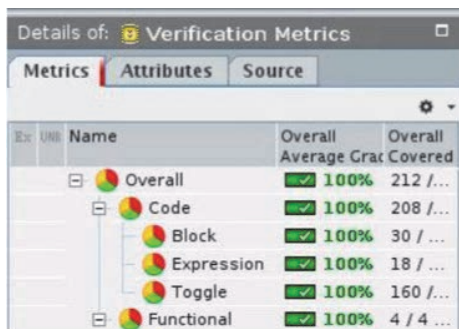


Рис. 3. Анализ полноты тестирования

Прототипирование модели интерфейсного блока в ПЛИС с использованием IP-блоков

Для верификации интерфейсного блока SPI с использованием средств ПЛИС необходимо адаптировать ранее разработанное тестовое окружение, что подразумевает замену несинтезируемых конструкций в тестах. Кроме того, для данной конфигурации прототипа вместо модели ведущего блока интерфейса, формирующего случайную посылку данных для тестируемого блока интерфейса, используется IP-блок интерфейса SPI. IP-блок интерфейса SPI реализован в программном ядре Nios II [13], который возможно реализовать на ПЛИС фирмы Altera семейства Cyclone III и выше. В качестве отладочной платы использовалась DE0-CV фирмы Terasic.

Для унификации и удобства сопоставления результатов с этапом верификации модели в САПР тестируемый блок интерфейса включается в прототип совместно с блоком управляющего автомата обработки команд.

При данной конфигурации прототипа исследуются внутренние сигналы ПЛИС, поэтому для отладки и визуализации временных диаграмм применен внутрисхемный логический анализатор SignalTap [14], входящий в состав САПР Quartus II. Общая схема прототипа представлена на рис. 4.

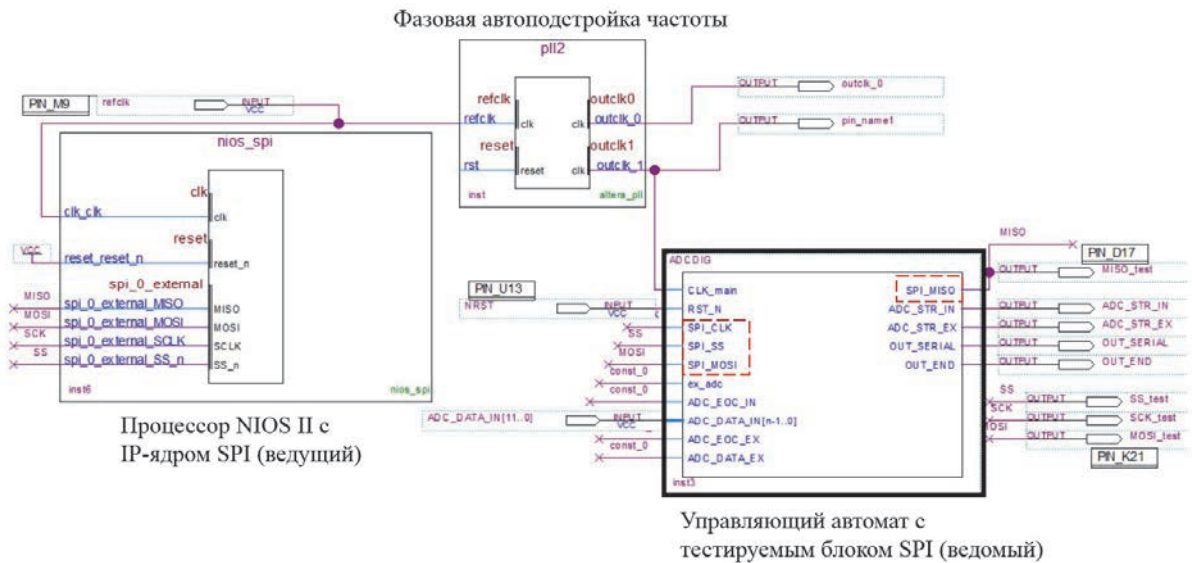


Рис. 4. Схема подключения прототипа интерфейсного блока с использованием IP-блока микропроцессора Nios II

На этапе верификации первого варианта конфигурации прототипа на ПЛИС были проанализированы временные диаграммы функционирования устройства. Однако метрики полноты покрытия не были использованы, поскольку для их анализа используются несинтезируемые конструкции языка. Временная диаграмма, полученная с помощью логического анализатора SignalTap, отражает некорректную инициализацию счетчика, которая ведет к функциональной ошибке в передаче данных (рис. 5, а). Данная ошибка не была выявлена на этапе верификации в САПР, поскольку при инициализации в программном обеспечении не учитываются временные задержки, связанные со временем установления необходимых данных.

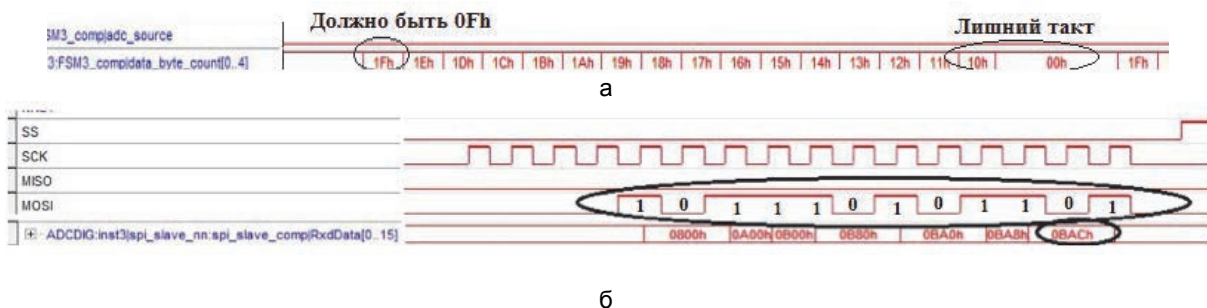


Рис. 5. Временная диаграмма: некорректной инициализации счетчика (а), функционирования прототипа блока интерфейса с использованием IP-блока (б)

Для устранения данной функциональной ошибки было изменено направление счета для корректного выполнения инициализации счетчика в значение 0. Результаты тестирования показали корректное функционирование интерфейса (рис. 5, б).

Прототипирование модели интерфейсного блока в ПЛИС с подключением внешних устройств

При данной конфигурации прототипа блока интерфейса в качестве ведущего блока используется внешнее устройство. Руководствуясь тенденциями импортозамещения, в качестве ведущего устройства выбран микроконтроллер отечественного производства 1986BE1T [15] фирмы Миландр. В данном МК реализован последовательный синхронный интерфейс SPI, который обеспечивает полнодуплексный обмен данными по четырехпроводной линии и программное задание фазы и полярности тактового сигнала. Для проведения верификации тестируемого блока интерфейса необходимо адаптировать тестовое окружение, использованное при верификации в САПР Cadence, в программное обеспечение для МК.

Тестируемый блок интерфейса входит в состав прототипа с использованием средств ПЛИС совместно с блоком преобразования данных для их последующего вывода на семисегментные индикаторы платы DE0-CV и отправки данных ведущему устройству в МК (рис. 6).

Корректность передаваемых данных и успешность выполнения тестирования определялась по значениям с семисегментных индикаторов, которые должны совпадать с данными, реализованными в программном обеспечении для МК. Макет для тестирования прототипа представлен на рис. 7.

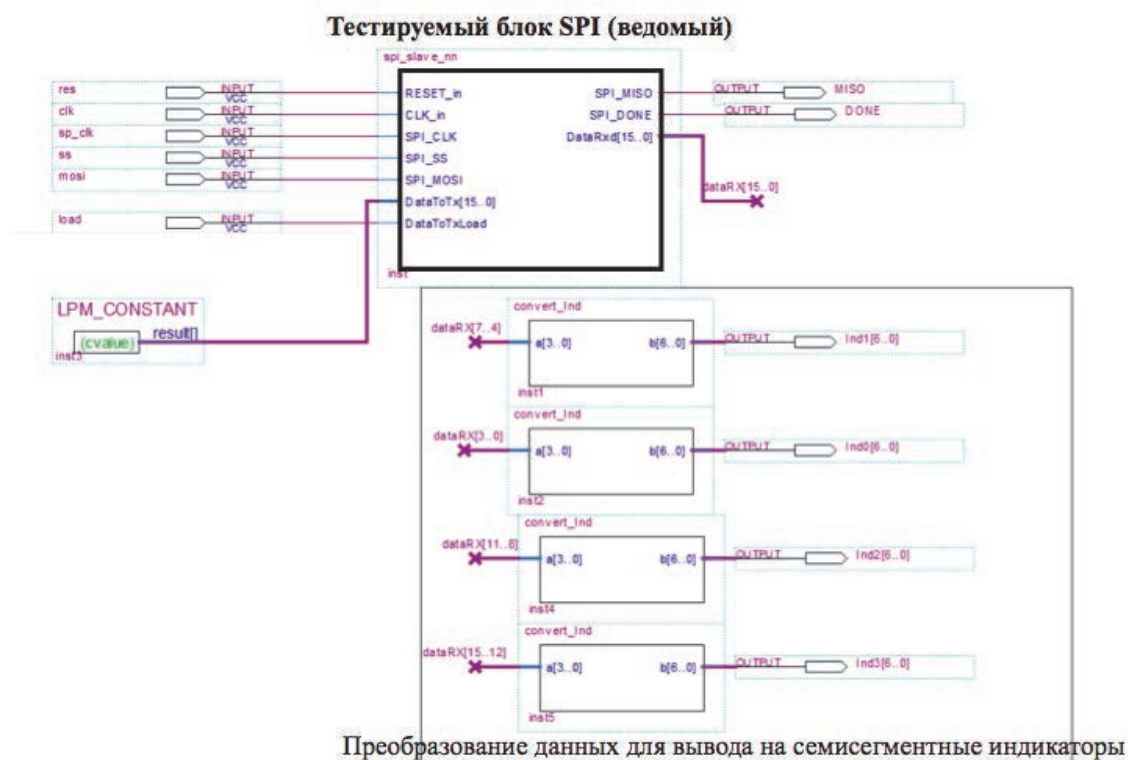


Рис. 6. Прототип интерфейсного блока с использованием средств программируемой логики для подключения с внешним микроконтроллером

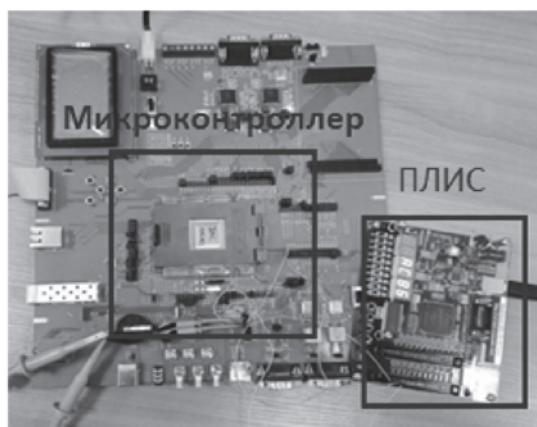


Рис. 7. Макет для тестирования прототипа блока интерфейса

По результатам верификации прототипа модели интерфейса были выявлены функциональные ошибки, которые не возникали на этапе моделирования. При передаче данных от МК в ПЛИС, от ведущего устройства ведомому, принятые данные были сдвинуты вправо относительно переданных данных. При обратной передаче наблюдался сдвиг данных влево. Для устранения возникшей ошибки в блок интерфейса SPI внесены дополнительные триггеры для синхронизации приема и передачи данных (SS_old, SS_latched).

После коррекции ошибок, возникших на двух этапах прототипирования, было проведено повторное тестирование блока интерфейса в САПР Cadence (рис. 8), которое подтвердило правильное функционирование устройства.

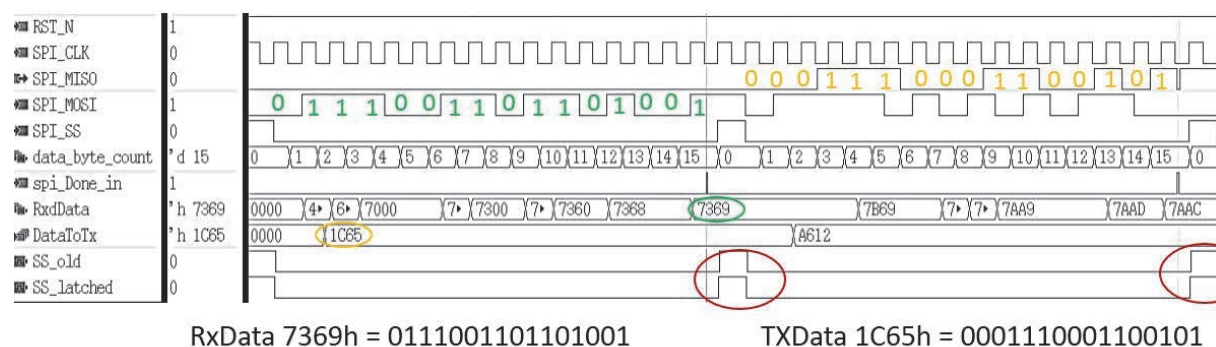


Рис. 8. Временная диаграмма функционирования интерфейсного блока после двухэтапного прототипирования

Заключение

Исследованы особенности верификации интерфейсного блока интегральной схемы с использованием средств программируемых логических интегральных схем. Предложен подход к проведению двухэтапного прототипирования на программируемых логических интегральных схемах с использованием сложнофункциональных блоков (IP-блоков) и внешних устройств. Применение данного подхода позволяет выявить функциональные ошибки, не определяемые на этапе верификации в средствах автоматизированного проектирования.

В результате исследования модели интерфейса Serial Peripheral Interface определено, что первый этап прототипирования с использованием стандартных интерфейсных IP-блоков позволяет выявлять имеющиеся функциональные ошибки в устройстве с минимальными временными затратами. При отсутствии стандартных IP-блоков модель верифицирующего устройства необходимо разработать отдельно. Однако это может привести к возникновению рисков неработоспособности конечного устройства, так как модель IP-блока может содержать функциональные ошибки.

Второй этап прототипирования с использованием внешнего подключаемого верифицирующего устройства позволил избавиться от ошибок, связанных с задержками распространения сигнала вне интегральной схемы, и избежать ограничений, связанных с отсутствием необходимых IP-блоков.

По результатам двухэтапного прототипирования модели интерфейса Serial Peripheral Interface выявлены и исправлены функциональные ошибки. Блок тестируемого интерфейса вошел в состав интегральной схемы для микромеханического акселерометра.

Литература

1. Титовская Т.С., Непомнящий О.В., Леонова А.В., Комаров А.А. Формальная верификация при проектировании сверхбольших интегральных схем // Вестник КрасГАУ. 2014. № 4. С. 87–89.
2. Foster H. Conclusion: The 2016 Wilson Research Group Functional Verification Study. URL: <https://blogs.mentor.com/verificationhorizons/blog/2017/01/03/conclusion-the-2016-wilson-research-group-functional-verification-study/> (accessed 5.02.18).
3. Бухтеев А.В. Методы и средства проектирования систем на кристалле // Chip News. 2003. № 4. С. 4–14.
4. Юрлин С.В. Разработка специализированных прототипов на основе программируемой логики для эффективной функциональной верификации многоядерных микропроцессоров: дис. ... канд. техн. наук. М., 2014.
5. Wojcikowski M., Pankiewicz B. ASIC design example of complex SoC with FPGA prototyping // Przegląd Elektrotechniczny. 2013. P. 156–158.
6. Юрлин С.В., Бычков И.Н. Прототипирование на основе ПЛИС для верификации многоядерных микропроцессоров // Проблемы разработки перспективных микро- и наноэлектронных систем. 2014. № 4. С. 45–50.
7. Майерс Г. Искусство тестирования программ. Вильямс, 2012. 272 с.
8. Piziali A. Functional Verification Coverage Measurement and Analysis. New York: Kluwer Academic Publishers, 2004. 216 p.
9. Хисамбеев И.Ш., Чибисов П.А. Об одном методе построения метрик функционального покрытия в тестировании микропроцессоров // Проблемы разработки перспективных микро- и наноэлектронных систем. 2014.

References

1. Titovskaya T.S., Nepomnyashchii O.V., Leonova A.V., Komarov A.A. Formal verification in very large-scale integration designing. *The Bulletin of KrasGAU*, 2014, no. 4, pp. 87–89. (in Russian)
2. Foster H. *Conclusion: The 2016 Wilson Research Group Functional Verification Study*. Available at: <https://blogs.mentor.com/verificationhorizons/blog/2017/01/03/conclusion-the-2016-wilson-research-group-functional-verification-study/> (accessed 5.02.18).
3. Bukhteev A.V. Methods and tools for designing systems on a chip. *Chip News*, 2003, no. 4, pp. 4–14. (in Russian)
4. Yurlin S.V. *Development of Specialized Prototypes Based on Programmable Logic for Efficient Functional Verification of Multi-Core Microprocessors*. Dis. PhD Eng. Sci. Moscow, 2014. (in Russian)
5. Wojcikowski M., Pankiewicz B. ASIC design example of complex SoC with FPGA prototyping. *Przegląd Elektrotechniczny*, 2013, pp. 156–158.
6. Yurlin S.V., Bychkov I.N. FPGA prototyping for functional verification of multi-core processors. *Problems of Development of Advanced Micro and Nanoelectronic Systems*, 2014, no. 4, pp. 45–50. (in Russian)
7. Myers G.J. *The Art of Software Testing*. Wiley, 2004.
8. Piziali A. *Functional Verification Coverage Measurement and Analysis*. New York, Kluwer Academic Publishers, 2004, 216 p.
9. Khisambeev I.Sh., Chibisov P.A. On one method of defining functional coverage metrics for microprocessor testing. *Problems of Development of Advanced Micro and*

- № 2. С. 63–68.
10. Incisive Coverage User Guide 15.2. Cadence, 2016. 350 p.
 11. Sutherland S., Mills D. *Synthesizing SystemVerilog Busting the Myth that SystemVerilog is only for Verification*. SNUG Silicon Valley, 2013. 45 p.
 12. Лапин А.А. Интерфейсы. Выбор и реализация. М.: Техносфера, 2005. 168 с.
 13. NIOS II Processor - Overview [Электронный ресурс]. Режим доступа: <https://www.altera.com/products/processors/overview.html>, свободный. Яз. англ. (дата обращения 13.02.2018).
 14. Ковач Н. Логический анализатор SignalTap [Электронный ресурс]. Режим доступа: <https://marsohod.org/11-blog/213-signalTap>, свободный. Яз. рус. (дата обращения 13.02.2018).
 15. Спецификация микроконтроллера 1986BE1T Миландр [Электронный ресурс]. Режим доступа: https://ic.milandr.ru/products/mikrokontrollery_i_protessory/1986ve1t/, свободный. Яз. рус. (дата обращения 10.02.2018).
 10. *Nanoelectronic Systems*, 2014, no. 2, pp. 63–68. (in Russian)
 10. *Incisive Coverage User Guide 15.2*. Cadence, 2016, 350 p.
 11. Sutherland S., Mills D. *Synthesizing SystemVerilog Busting the Myth that SystemVerilog is only for Verification*. SNUG Silicon Valley, 2013, 45 p.
 12. Lapin A.A. *Interfaces. Selection and Implementation*. Moscow, Tehnosfera Publ., 2005, 168 p. (in Russian)
 13. *NIOS II Processor - Overview*. Available at: <https://www.altera.com/products/processors/overview.html> (accessed 13.02.2018).
 14. Kovach N. *Logical Analyzer SignalTap*. Available at: <https://marsohod.org/11-blog/213-signalTap> (accessed 13.02.2018).
 15. *Milandr 1986BE1T Microcontroller Specification*. Available at: https://ic.milandr.ru/products/mikrokontrollery_i_protessory/1986ve1t/ (accessed 10.02.2018).

Авторы

Чернецкая Наталья Михайловна – инженер, АО «Концерн «ЦНИИ «Электроприбор», Санкт-Петербург, 197046, Российская Федерация, ORCID ID: 0000-0002-7085-0348, natali030594@gmail.com

Михтеева Анна Александровна – младший научный сотрудник, АО «Концерн «ЦНИИ «Электроприбор», Санкт-Петербург, 197046, Российская Федерация; сотрудник подразделения Международная лаборатория «Интегрированные системы ориентации и навигации», Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, Scopus ID: 57195502932, ORCID ID: 0000-0002-0637-6987, designcenter.spb@mail.ru

Невирковец Николай Николаевич – младший научный сотрудник, АО «Концерн «ЦНИИ «Электроприбор», Санкт-Петербург, 197046, Российская Федерация; аспирант, Санкт-Петербургский государственный электротехнический университет (ЛЭТИ) им. В.И. Ульянова (Ленина), Санкт-Петербург, 197376, Российская Федерация, Scopus ID: 57195506419, ORCID ID: 0000-0003-2062-506X, nevirkovets.nn@gmail.com

Костыгов Дмитрий Вадимович – младший научный сотрудник, АО «Концерн «ЦНИИ «Электроприбор», Санкт-Петербург, 197046, Российская Федерация, Scopus ID: 57189243101, ORCID ID: 0000-0003-4379-5803, dkost92@mail.ru

Беляев Яков Валерьевич – кандидат технических наук, начальник НТЦ «Дизайн-центр», АО «Концерн «ЦНИИ «Электроприбор», Санкт-Петербург, 197046, Российская Федерация, Scopus ID: 56382180100, ORCID ID: 0000-0003-1971-9069, jak0b@mail.ru

Authors

Natalia M. Chernetskaya – engineer, Concern "CSRI Elektropribor", Saint Petersburg, 197046, Russian Federation, ORCID ID: 0000-0002-7085-0348, natali030594@gmail.com

Anna A. Mikhteeva – junior scientific researcher, Concern "CSRI Elektropribor", Saint Petersburg, 197046, Russian Federation; Employee of International Laboratory "Integrated Navigation And Attitude Reference Systems", ITMO University, Saint Petersburg, 197101, Russian Federation, Scopus ID: 57195502932, ORCID ID: 0000-0002-0637-6987, designcenter.spb@mail.ru

Nikolay N. Nevirkovets – junior scientific researcher, Concern "CSRI Elektropribor", Saint Petersburg, 197046, Russian Federation; postgraduate, Saint Petersburg Electrotechnical University (LETI), Saint Petersburg, 197376, Russian Federation, Scopus ID: 57195506419, ORCID ID: 0000-0003-2062-506X, nevirkovets.nn@gmail.com

Dmitry V. Kostygov – junior scientific researcher, Concern "CSRI Elektropribor", Saint Petersburg, 197046, Russian Federation, Scopus ID: 57189243101, ORCID ID: 0000-0003-4379-5803, dkost92@mail.ru

Yakov V. Belyaev – PhD, Head of department, Concern "CSRI Elektropribor", Saint Petersburg, 197046, Russian Federation, Scopus ID: 56382180100, ORCID ID: 0000-0003-1971-9069, jak0b@mail.ru