

---

---

# ПРОГРАММНО-АППАРАТНЫЕ СРЕДСТВА ИНФОРМАЦИОННО-УПРАВЛЯЮЩИХ СИСТЕМ

---

---

УДК 004.27  
DOI: 10.17586/0021-3454-2017-60-10-967-972

## ПОДХОД К РАЗРАБОТКЕ СИСТЕМ ПОТОКОВОЙ ОБРАБОТКИ ДАННЫХ НА ПЛИС С ВОЗМОЖНОСТЬЮ КОМБИНИРОВАННОЙ ОТЛАДКИ

В. Ю. ПИНКЕВИЧ

*ООО „ЛМТ“, 199034, Санкт-Петербург, Россия  
E-mail: vasilij.pinkevich@lmt.spb.ru*

Рассмотрен вопрос отладки аппаратных вычислительных систем, ориентированных на потоковую обработку данных на ПЛИС. Для эффективной отладки моделей системы и ее прототипов требуется возможность переносить тестовые данные между разными представлениями системы. Предлагаемый подход основан на наборе правил разработки систем на ПЛИС и тестового окружения, что позволяет обеспечить совместимость форматов данных, получение необходимых входных и выходных данных, а также выполнение отладки в разных представлениях системы. Предлагается набор компонентов повторного использования для создания отладочной инфраструктуры. Подход ориентирован на реализацию в рамках существующих маршрутов проектирования, САПР и сред разработки аппаратных систем на ПЛИС.

**Ключевые слова:** *отладка, верификация, разработка, ПЛИС, система на ПЛИС, система на кристалле, САПР*

**Введение.** При проведении вычислений, связанных с цифровой обработкой сигналов, изображений, звука и т.п., часто применяются решения на ПЛИС, обладающие особой структурой — системы потоковой обработки данных (СПОД). Использование ПЛИС позволяет реализовывать СПОД в мелкосерийных и экспериментальных изделиях с возможностью перепрограммирования после проведения испытаний. СПОД обычно представляют собой набор функциональных вычислительных блоков, соединенных потоковыми интерфейсами для передачи массивов однотипных данных [1].

Одной из центральных задач является отладка разрабатываемых микросхем, систем на кристалле и систем на ПЛИС. Отладка должна выполняться на всех этапах разработки — от высокоуровневых моделей до реализации в изделии [2—4]. Однако между отладкой моделей аппаратной системы до ее изготовления и отладкой в готовом прототипе существует разрыв, который заключается в несовместимости применяемых методов и средств, форматов тестовых данных и т.п. Это усложняет процесс локализации и исправления ошибок [5, 6].

В настоящей статье предлагается подход к разработке СПОД на ПЛИС, который позволяет унифицировать и оптимизировать процесс их отладки до и после изготовления прототипа с учетом специфики систем этого класса. Основу предлагаемого подхода, названного комбинированной отладкой, составляет обмен тестовыми данными между различными представлениями вычислительной системы.

**Существующие решения.** Существует общая тенденция к усилению связей между разными уровнями представления проекта и задачами (Activity) в процессе разработки сложных вычислительных систем с целью повышения качества и скорости разработки [7, 8]. В частности, ведущими поставщиками САПР для разработки систем на ПЛИС и заказных микросхем прилагаются усилия по интеграции всех процессов разработки и отладки в рамках одного маршрута и одной САПР. Предлагаются решения для конкретных прикладных областей, например, в области цифровой обработки сигналов — Vivado System Generator for DSP (Xilinx) [9]; недостатком этого решения является привязка к платформе и библиотеке компонентов конкретного производителя. В области разработки процессоров существуют решения [6, 10] по унификации процесса отладки с помощью системы генерации тестов и специальных средств, позволяющих запускать одни и те же тесты в симуляторе и в готовом изделии.

В [11] предлагается метод долговременного наблюдения за отклонениями от нормы поведения функционирующей системы. Поведение сравнивается с моделью, созданной на этапе разработки.

Компании Exostiv Labs [12] и Tektronix (технология Certus) [13] предлагают средства для сбора отладочных данных с системы на ПЛИС на целевой скорости работы. Компания DAFSA предлагала набор компонентов для ввода и вывода отладочных данных из системы на ПЛИС [14].

Стандарт IEEE 1500 ориентирован на отладку IP-ядер в составе микросхемы по аналогии с интерфейсом JTAG [15]. Недостатком этого подхода является ограничение максимальной скорости работы отлаживаемого блока скоростью отладочного интерфейса.

Проанализировав имеющиеся решения, можно сделать вывод, что для маршрутов проектирования существующих САПР, с учетом доступных сторонних решений, в настоящее время отсутствуют методика и средства, которые позволяли бы решать поставленную задачу для произвольного проекта СПОД на ПЛИС.

**Маршруты отладки систем потоковой обработки данных.** СПОД обычно является частью более сложной вычислительной системы, которая выполняет задачи настройки режимов работы СПОД, ввода/вывода данных и др. Также СПОД взаимодействует с внешним окружением вычислительной системы, осуществляя потоковый ввод и вывод данных в цифровом виде или через АЦП/ЦАП (рис. 1).

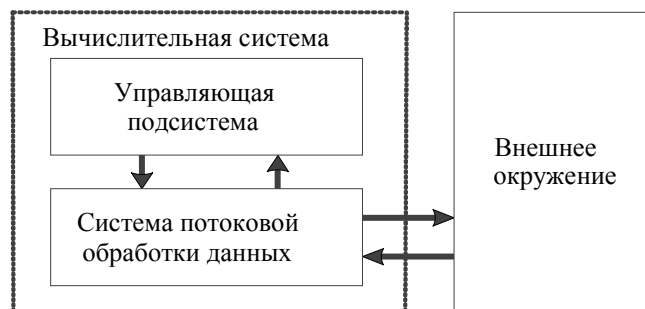


Рис. 1

При разработке таких систем стоит задача отладки:

- целевых алгоритмов работы СПОД;
- реализации целевых алгоритмов работы в прототипе СПОД (верификация).

Выполнять верификацию реализованной СПОД необходимо на эталонных наборах данных. Для контроля адекватности целевого алгоритма необходимо обеспечить возможность наблюдения за работой СПОД на целевой скорости, с данными, получаемыми от целевого внешнего окружения. Зачастую только в таких условиях можно обеспечить корректное взаимодействие с внешним окружением. На рис. 2 показан пример использования различных

типов входных данных и приведены выводы, которые можно сделать на основе соответствующих тестовых запусков.

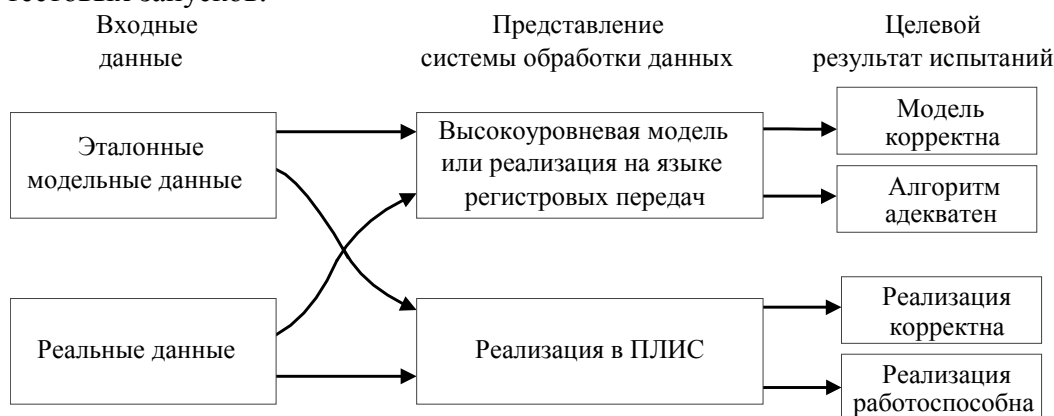


Рис. 2

Возможны два сценария разработки СПОД. Целевое внешнее окружение:

- 1) доступно и может использоваться при тестировании;
- 2) недоступно, СПОД не может быть протестирована на реальных данных до окончания разработки.

В первом случае отладка целевых алгоритмов с реальными данными может проводиться уже на этапе высокоуровневых моделей, а во втором варианте высокоуровневые модели СПОД отлаживаются с применением модельных тестовых данных.

**Подход к разработке систем потоковой обработки данных.** Предлагаемый подход ориентирован на работу с различными представлениями СПОД при использовании стандартных САПР и сред программирования, применяемых при разработке систем на ПЛИС. Рассматриваются следующие основные представления СПОД: программная модель, реализация на языке описания аппаратуры, готовое изделие (система на ПЛИС). Возможны и другие варианты представления, в частности, дополнительные абстрактные модели алгоритмов.

Для достижения поставленной цели предлагаемый подход требует:

- соответствия различных представлений системы общей базовой структурной модели;
- обеспечения возможности тестирования системы и ее отдельных элементов в рамках каждого представления.

Базовая структурная модель представляет собой ориентированный граф (рис. 3). Вершинами являются минимальные функциональные вычислительные блоки, которые требуется отлаживать по отдельности. Дуги представляют собой интерфейсы передачи данных. Необходима совместимость интерфейсов между вычислительными блоками во всех представлениях системы. Если совместимость не является побитовой, то требуются специальные средства преобразования данных для их передачи между представлениями.

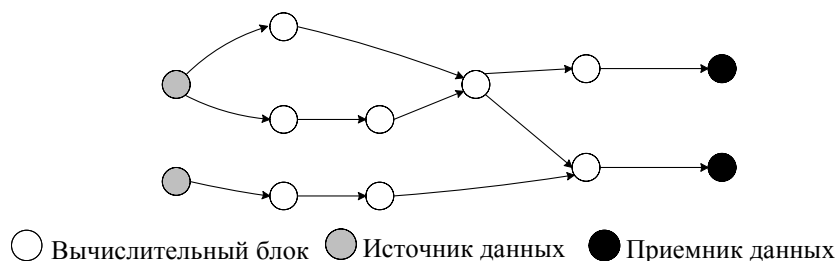


Рис. 3

В каждом из представлений системы должна быть обеспечена возможность запуска отдельных вычислительных блоков и их цепочек с выдачей входных тестовых данных и сохранением результатов в каждой дуге графа. Для высокоуровневой модели и используемого в симуляции описания на уровне регистровых передач это может быть сделано с помощью

стандартных средств декомпозиции программ, характерных для применяемого языка (С, С++, MATLAB, Python и др. для высокоуровневой модели, VHDL или Verilog на уровне регистровых передач). Инфраструктура для обеспечения необходимых требований в уже реализованной системе на ПЛИС должна включать блоки сбора/выдачи данных и контроллер эксперимента. Пример реализации системы потоковой обработки данных с элементами отладочной инфраструктуры приведен на рис. 4.

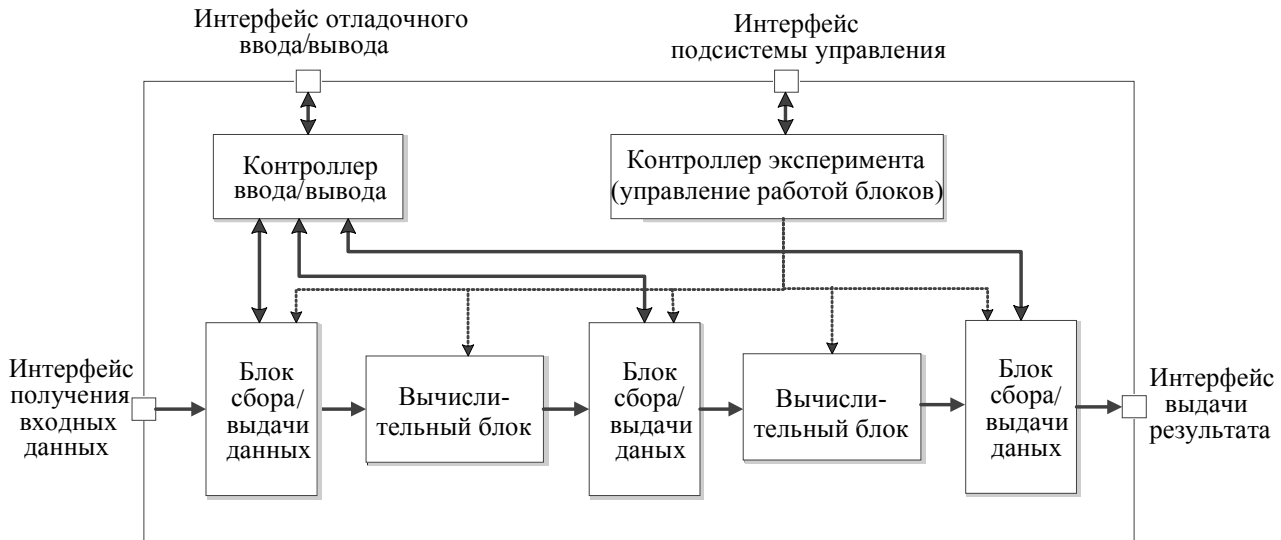


Рис. 4

Шаблоны тестового окружения для представления на конкретном языке и элементы отладочной инфраструктуры в ПЛИС являются компонентами повторного использования.

**Применение предлагаемого подхода.** Рассмотрим реализацию комбинированной отладки в системе на ПЛИС, разработанной с применением предложенного подхода. Пусть имеется следующий набор представлений системы: высокоуровневая модель алгоритма (на языке MATLAB, Python, С, С++, SystemC или др.), реализация на уровне регистровых передач (VHDL или Verilog), целевая реализация в ПЛИС. Возможны четыре основных варианта использования предлагаемой инфраструктуры:

1) контроль корректности реализации (верификация), который выполняется путем запуска всех представлений системы с одинаковыми тестовыми данными и проверкой соответствия полученных результатов ожидаемым;

2) отладка целевых алгоритмов работы системы с реальным внешним окружением (для второго сценария разработки СПОД, когда реальное внешнее окружение недоступно перед началом разработки). После получения целевой реализации системы на ПЛИС в реальных условиях работы собираются входные данные и с ними выполняются запуски высокоуровневой модели. По результатам этих запусков высокоуровневая модель может быть скорректирована, и затем изменения должны быть реализованы в остальных представлениях;

3) тестовые запуски системы на ПЛИС с тестовыми данными позволяют, не используя внешнее окружение, выполнять комплексную отладку СПОД во взаимодействии с управляющей подсистемой. Кроме этого, это может использоваться для более быстрой (за счет аппаратного ускорения) проверки разных вариантов алгоритма, чем при исполнении кода высокоуровневой модели или симуляции представления системы на языке уровня регистровых передач;

4) элементы отладочной инфраструктуры могут быть использованы в целевом режиме работы для обмена выходными и/или входными данными с управляющей подсистемой, а также для управления СПОД.

Предложенный подход был реализован при разработке многоканальных систем цифровой обработки сигналов, получаемых с АЦП, по обоим сценариям. Высокоуровневые модели аппаратных блоков разрабатывались на языке Python, СПОД на ПЛИС были реализованы на языке Verilog для систем на кристалле семейства Xilinx Zynq. Тестовая инфраструктура использовалась во всех описанных выше вариантах. Это обеспечило ускорение разработки за счет более эффективного процесса отладки.

Возможности применения предлагаемого подхода зависят от свободных аппаратных ресурсов ПЛИС для реализации отладочной инфраструктуры, а также объема внутрисистемной памяти либо пропускной способности отладочного интерфейса ввода-вывода.

**Заключение.** Предложенный подход к разработке систем потоковой обработки данных на ПЛИС позволяет связать этапы отладки моделей системы и отладки ее реализации после изготовления прототипа. Подход ориентирован на реализацию в рамках стандартных САПР и сред программирования, применяемых при разработке систем на ПЛИС и заказных микросхем.

Предлагаются компоненты повторного использования: шаблоны тестового окружения для языков уровня регистровых передач и элементы отладочной инфраструктуры для системы на ПЛИС.

Возможно использование предложенного подхода в качестве расширения к существующим САПР.

#### СПИСОК ЛИТЕРАТУРЫ

1. Meyer-Baese U. Digital Signal Processing with Field Programmable Gate Arrays. Springer, 2001.
2. Sinha R., Roop P., Basu S. Correct-by-Construction Approaches for SoC Design. Springer, 2013.
3. Mitra S., Seshia S. A., Nicolici N. Post-Silicon Validation Opportunities, Challenges and Recent Advances // 47th ACM/IEEE Design Automation Conf. (DAC). 13—18 June 2010. DOI: 10.1145/1837274.1837280
4. Goeders J., Wilton S. J. E. Effective FPGA Debug for High-Level Synthesis Generated Circuits // 24th Intern. Conf. on Field Programmable Logic and Applications (FPL). 2—4 September 2014. DOI: 10.1109/FPL.2014.6927498.
5. Stollon N., Leatherman R. Integrating on Chip Debug Instrumentation and EDA Verification Tools // DesignCon East, 2005 [Электронный ресурс]: <www.fs2.com/pdfs/DesignCon%20East%202005\_FS2\_ID1\_202\_final.pdf>.
6. Adir A., Coptly S., Landa S., Nahir A., Shurek G., Ziv A., Meissner C., Schumann J. A Unified Methodology for Pre-Silicon Verification and Post-Silicon Validation // Design, Automation & Test in Europe Conference & Exhibition (DATE), 14—18 March 2011. DOI: 10.1109/DATE.2011.5763252
7. Teich J. Hardware/software codesign: the past, the present, and predicting the future // Proc. IEEE. 2012. Vol. 100. P. 1411—1430.
8. Platonov A., Kluchev A., Penskoi A. Expanding Design Space for Complex Embedded Systems with HLD-methodology // 6th Intern. Congr. on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT). 2014. P. 157—164.
9. Xilinx System Generator for DSP. Xilinx, Inc. System Generator for DSP Getting Started Guide UG639 (v11.4). December 2, 2009 [Электронный ресурс]: <www.xilinx.com>.
10. Wagner I., Bertacco V. Reversi: Post-Silicon Validation System for Modern Microprocessors // IEEE Intern. Conf. on Computer Design. 12—15 Oct. 2008. DOI: 10.1109/ICCD.2008.4751878
11. Антонов А. А., Быковский С. В., Кустарев П. В. Монитор временных ограничений для систем на кристалле // Изв. вузов. Приборостроение. 2014. Т. 57, № 4. С. 53—57.
12. Leens F. FPGA verification tools need an upgrade. Exostiv Labs, 2015.
13. Certus ASIC Prototyping Debug Solution. Tektronix, Inc. 2012.
14. Abramovici M., Bradley P., Dwarakanath K., Levin P., Memmi G., Miller D. A Reconfigurable Design-for-Debug Infrastructure for SoCs // Proc. of the 43rd Design Automation Conf. DAC. San Francisco, CA, USA, July 24—28, 2006. DOI: 10.1145/1146909.1146916.

15. *Da Silva F., McLaurin T., Waayers T.* The Core Test Wrapper Handbook. Rationale and Application of IEEE Std. 1500. Springer Science+Business Media, LLC, 2006.

**Сведения об авторе**

**Василий Юрьевич Пинкевич**

— ООО „ЛМТ“; инженер; E-mail: vasilii.pinkevich@lmt.spb.ru

Рекомендована кафедрой  
вычислительной техники

Поступила в редакцию  
03.07.17 г.

**Ссылка для цитирования:** Пинкевич В. Ю. Подход к разработке систем потоковой обработки данных на ПЛИС с возможностью комбинированной отладки // Изв. вузов. Приборостроение. 2017. Т. 60, № 10. С. 967—972.

**AN APPROACH TO DESIGN OF FPGA-BASED SYSTEMS FOR STREAM DATA PROCESSING  
WITH CAPABILITY OF COMBINED DEBUGGING**

**V. Yu. Pinkevich**

*LMT Ltd., 199034, St. Petersburg, Russia  
E-mail: vasilii.pinkevich@lmt.spb.ru*

Debugging of computational FPGA-based systems oriented on stream data processing is considered. An effective debugging procedure for a systems models and prototypes it is necessary to use unified input and output data. The proposed approach is based on a specific procedure of design of FPGA projects and their test environments. This provides compatibility of data formats, generation the necessary input/output test data, and debugging in various representations of designed system. A set of components to be reused when creating a debug infrastructure is presented. The described approach is oriented for implementation in the framework of existing design flows and CAD tools for FPGA-based projects.

**Keywords:** debugging, verification, design, FPGA, FPGA based system, chip based system, CAD

**Data on author**

**Vasilii Yu. Pinkevich**

— LMT Ltd.; Engineer; E-mail: vasilii.pinkevich@lmt.spb.ru

**For citation:** Pinkevich V. Yu. An approach to design of FPGA-based systems for stream data processing with capability of combined debugging. *Journal of Instrument Engineering*. 2017. Vol. 60, N 10. P. 967—972 (in Russian).

DOI: 10.17586/0021-3454-2017-60-10-967-972