

ПРОВЕДЕНИЕ ФУНКЦИОНАЛЬНОГО КОНТРОЛЯ ПРОГРАММИРУЕМЫХ ЛОГИЧЕСКИХ ИНТЕГРАЛЬНЫХ СХЕМ

© 2017

А. А. Огурцов инженер-исследователь;
АО «Российские космические системы», г. Москва;
sanoyashi@mail.ru

Описываются подходы к проведению функционального контроля программируемых логических интегральных схем (ПЛИС). Рассматривается существующее оборудование, позволяющее проводить функциональный контроль ПЛИС. Приводится таблица сравнения основных параметров представленной на рынке аппаратуры. Даны основные параметры аппаратуры, позволяющие упростить процесс проведения функционального контроля ПЛИС. По итогам анализа приводятся основные достоинства и недостатки использования данной аппаратуры. Описываются основные методы проведения функционального контроля ПЛИС. При разработке алгоритмов вся внутренняя структура ПЛИС разделяется на две отдельные функционально независимые части. Первая часть – это внутренняя программируемая логика, а вторая – встроенные в ПЛИС функциональные блоки. Приводятся примеры алгоритмов, разработанных на основе описанных методов, и конфигурационные прошивки, созданные при помощи этих алгоритмов. Разработанные алгоритмы осуществляют функциональный контроль внутренней логики, а также встроенных в ПЛИС функциональных блоков. При создании алгоритмов функционального контроля внутренней логики использован метод повторяющихся тестовых множеств. Для контроля встроенных в ПЛИС функциональных блоков используется метод создания схем самодиагностики.

ПЛИС; тестирование; автоматизированное тестовое оборудование; конфигурируемые логические блоки; алгоритмы функционального контроля ПЛИС; испытания; аппаратура для функционального контроля.

Цитирование: Огурцов А.А. Проведение функционального контроля программируемых логических интегральных схем // Вестник Самарского университета. Аэрокосмическая техника, технологии и машиностроение. 2017. Т. 16, № 4. С. 137-146. DOI: 10.18287/2541-7533-2017-16-4-137-146

Введение

В настоящее время при разработке радиоэлектронной аппаратуры широкое применение получили программируемые логические интегральные схемы, что связано в первую очередь с большой логической ёмкостью современных ПЛИС, позволяющей реализовывать функционально сложные устройства с высокими частотными характеристиками на одном кристалле интегральной схемы. Высокая степень интеграции и большое количество встроенных функциональных блоков ПЛИС ведёт к увеличению вероятности возникновения функционального дефекта, приводящего в дальнейшем к неправильному функционированию разрабатываемой аппаратуры. Поэтому при испытаниях важно не только проведение параметрического, но и функционального контроля ПЛИС. Возникновение дефектов также может быть обусловлено нарушением правил транспортировки ПЛИС и наличием на рынке большого числа контрафактных изделий, в том числе микросхем, уже снятых с производства, но до сих пор применяемых в некоторых образцах аппаратуры. Поэтому проведение функционального контроля позволяет подтвердить возможность использования ПЛИС в высоконадёжной аппаратуре при заданных условиях внешних воздействующих факторов и при заданных характеристиках.

Аппаратура для проведения функционального контроля

Для проведения испытаний микросхем существует определённый тип автоматической тестерной аппаратуры, в зарубежных источниках также называемой АТЕ (Automatic Test Equipments). Данная аппаратура позволяет проведение параметрического и функционального контроля в автоматическом режиме с заданными характеристиками. На рынке представлено множество как отечественных, так и иностранных производителей данной аппаратуры. В таблице приведено сравнение аппаратуры различных фирм-производителей по параметрам, наиболее сильно влияющим на возможности проведения функционального контроля ПЛИС.

Сравнительный анализ основных параметров аппаратуры для контроля ПЛИС

Наименование параметра	Фирма-производитель							
	Teradyne		Форм		HILEVEL Technology		Chroma	
Модель	Integra J750Ex	FLEX	FORMULA HF3	FORMULA 2K	Titan	Griffin III	3600	3380P
Максимальное количество каналов	1024	2240	512	256	1024	512	256	576
Частота смены векторов, МГц, не более	100**	500**	200 (300**)	20	200**	200**	100**	100**
Память на каждый тестовый канал, М состояний, (каждое состояние кодируется тремя битами)	64	256	64	1	64	64	16	64
Уровни выходного сигнала драйвера, В	-1.0...+6.5	-2.6...+7.6	-2.0...+7.5	-1.0...+6.0	--*	-1.25...+6.75	-8...+8	-8...+8
Диапазон активных нагрузок, мА	0..50	0..100	0..24	0..24	--*	--*	0..35	--*

*Данные характеристики не указаны в описании аппаратуры.

**Частота в мультиплексированном режиме.

Подробное описание и сравнение возможностей современной аппаратуры имеется в [1]. Проведение функционального контроля ПЛИС возможно и при помощи логических анализаторов, но при условии наличия в них специальных модулей генераторов тестовых последовательностей или программаторов и при наличии в ПЛИС поддержки определённых стандартов JTAG-тестирования.

Помимо определённых достоинств описанной выше аппаратуры, таких как автоматизация проведения испытаний, высокие частоты проведения контроля и точностные характеристики аппаратуры, существует и ряд недостатков, существенно затрудняющих или делающих весьма трудоёмким процесс проведения

функционального контроля ПЛИС. Основными недостатками являются: сложность реализации функционального контроля, отсутствие заложенных в аппаратуру специализированных алгоритмов функционального контроля, отсутствие возможности конфигурирования ПЛИС посредством интерфейсов SERIAL, JTAG или SPI либо отсутствие соответствующих библиотек для конфигурирования ПЛИС, высокая стоимость оборудования.

Методы проведения функционального контроля

В ПЛИС, в отличие от ASIC и других типов СБИС с фиксированной структурой, функциональное тестирование не зависит от функционирования конечного, созданного пользователем устройства и требует разработки универсальных высокоэффективных алгоритмов функционального контроля всех встроенных узлов и блоков [2]. Функциональный контроль ПЛИС требует множественных переконфигураций, так как невозможно провести полное функциональное тестирование всей внутренней логики только одной тестовой прошивкой.

Для проведения функционального контроля ПЛИС выделим три основных метода: метод использования схем самодиагностики (в зарубежной литературе называемый BIST (Built-in-self-test) [2–4], метод повторяющихся тестовых множеств (ILA-Iterative Logic Arrays) [5] и метод считывания выходных откликов посредством различных интерфейсов.

Метод использования схем самодиагностики подразумевает реализацию в прошивке специализированных схем, позволяющих проводить тестирование внутри самой ПЛИС. Данные схемы имеют определённую структуру (рис. 1), состоящую из генератора тестовых сигналов (TPG), анализатора отклика (ORA) и логики управления. Главным преимуществом метода является возможность тестирования без использования дорогостоящего тестового оборудования (ATE). Но реализация алгоритма на основе данного метода подразумевает использование определённого процента внутренней логики ПЛИС, а при проведении функционального контроля необходимо проверять весь объём внутренней логики. Поэтому данный метод наиболее удобно применять для тестирования всевозможных внутренних блоков ПЛИС.



Рис. 1. Архитектура стандартной схемы самодиагностики (BIST)

Для тестирования внутренней логики ПЛИС наиболее удобным является метод повторяющихся тестовых множеств. Основная идея данного метода заключается в том, что конфигурационные логические блоки (CLB), из которых состоит ПЛИС, конфигурируются таким образом, что выходные сигналы предыдущего CLB-блока являются входными сигналами следующего блока. Поэтому появляется возможность соединить блоки тестируемой логики в тестовые цепочки (рис. 2).

Использование ИЛА-метода позволяет значительно сократить число тестовых конфигураций и снизить время проведения функционального контроля. Одним из достоинств метода является высокая наблюдаемость, позволяющая определить место воз-

никновения ошибки. Использование метода ИЛА является оптимальным для проведения функционального контроля внутренней логики ПЛИС. Основным недостатком метода является необходимость наличия специализированного оборудования АТЕ для проведения тестирования, а также низкая, по сравнению с BIST, скорость тестирования.

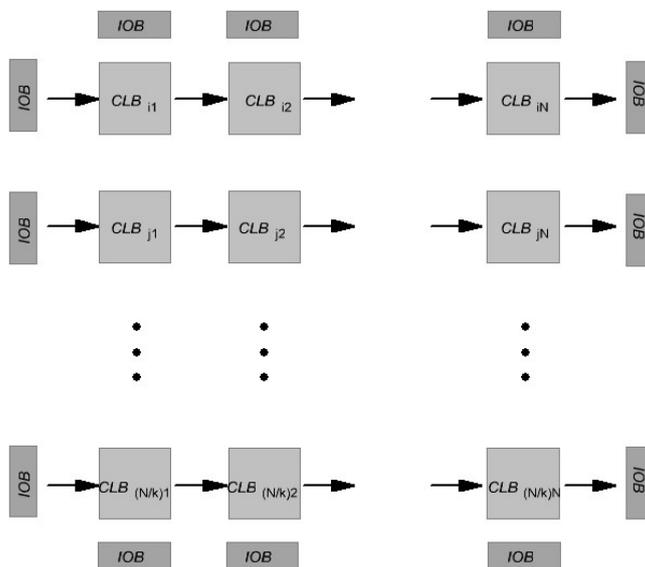


Рис. 2. Схема работы метода ИЛА

Метод считывания выходных откликов посредством различных интерфейсов требует наличия в ПЛИС интерфейса, например JTAG, поддерживающего расширенный функционал [6–8]. Использование данного метода облегчает проведение функционального контроля некоторых встроенных блоков ПЛИС. Достоинствами данного метода являются высокое быстродействие и возможность проведения функционального контроля без использования дорогостоящего оборудования. Основным недостатком является то, что данный метод применим только к новым семействам ПЛИС и может тестировать только некоторые встроенные блоки. Тестирование всей ПЛИС является довольно трудоёмким и заключается в объединении данного метода с методом встроенных схем самодиагностики посредством конфигурационной прошивки.

Алгоритмы функционального контроля

Для проведения функционального контроля ПЛИС на основе вышеописанных методов разрабатываются специализированные алгоритмы тестирования, представленные в виде тестовых конфигурационных прошивок. Для простоты понимания приведём разработанные алгоритмы в виде RTL-схем, т.к. VHDL код более сложен для восприятия. Первым этапом проведения функционального контроля является проверка внутренней логики ПЛИС. Основной объём внутренней логики современных ПЛИС составляют конфигурационные логические блоки CLB, количество которых достигает сотен тысяч. Каждый CLB-блок имеет одинаковую структуру и соединён с соседними блоками посредством коммутируемой матрицы межсоединений. На рис. 3 приведена стандартная архитектура ячейки ПЛИС.

Конфигурационный логический блок состоит из трёх основных частей: программируемого автомата состояний (таблицы перекодировки LUT), программируемого регистра и логики переноса. Проанализировав внутреннюю структуру блока, можно увидеть, что для полного функционального контроля CLB блока необходимо проверять:

- таблицы перекодировки (LUT's – Look-up tables);
- блоки распределённой памяти на основе LUT;
- блоки сдвиговых регистров (SRL – Shift register logic) на основе LUT;
- блоки программируемых регистров;
- логику переноса.

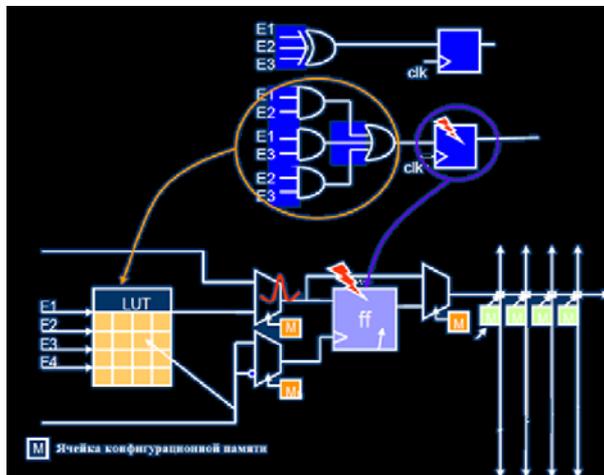


Рис. 3. Архитектура ячейки ПЛИС

Для каждого описанного блока разработаны алгоритмы функционального контроля на основе метода повторяющихся множеств.

Одним из основных компонентов конфигурационного логического блока является LUT-блок, способный также выполнять функции распределённой памяти и сдвигового регистра. Все вышеописанные функции проверяются при проведении функционального контроля. Для проверки LUT-блока используются алгоритмы, описанные ниже.

Как показали предыдущие исследования [9], для полной проверки одного LUT-блока необходимы только две тестовые конфигурации на основе логических функций XOR и XNOR. Основной предлагаемый принцип функционального контроля LUT показан на примере элемента с четырьмя входами. Для формирования одной ячейки сканирующей цепочки (рис. 4) используется восемь LUT-блоков: четыре из них сконфигурированы на выполнение функции исключающего ИЛИ (XOR), а остальные конфигурируются как исключающие ИЛИ с инверсией (XNOR). Соединены восемь LUT-блоков как показано на рис. 4.

Тестовые сигналы на входы A1, A2, A3 и A4 поступают на тестируемую ячейку с выводов микросхемы либо с предыдущей сканирующей ячейки. Т.к. первые 4 ячейки выполняют функции логического XOR, а оставшиеся ячейки – функции логического XNOR, тогда, если все LUT функционируют правильно, то выполняются следующие тождества:

$$B1 = B2 = B3 = B4 = A1 \wedge A2 \wedge A3 \wedge A4, \quad (1)$$

$$C1 = B1 \wedge A2 \wedge A3 \wedge A4 = A1. \quad (2)$$

Нетрудно доказать, что C2, C3 и C4 соответственно равны A2, A3 и A4, т.е. сигналы на входах ячейки эквивалентны сигналам на выходе. Следовательно, если происходит ошибка в любом LUT-блоке, то это обнаружится на выходе сканирующей цепочки.

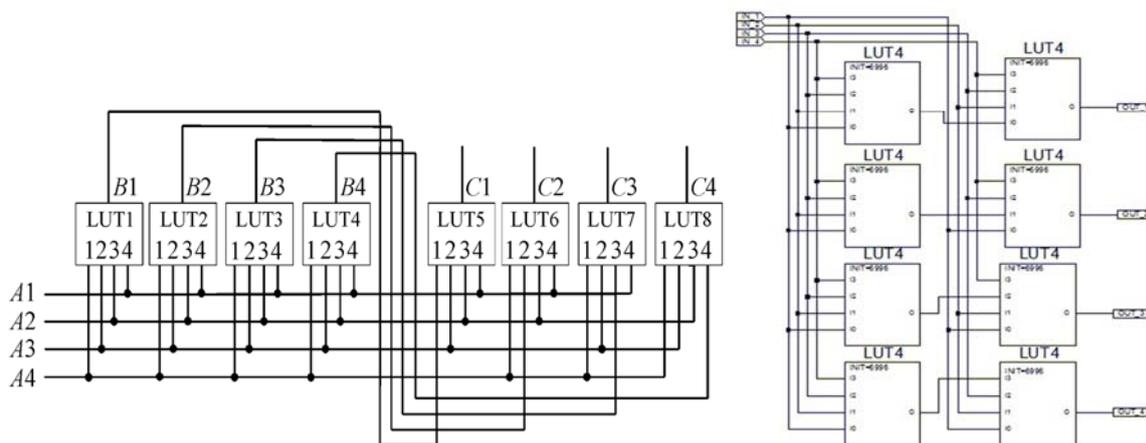


Рис. 4. Ячейка сканирующей цепочки (слева), пример конфигурационной прошивки (справа)

Как было отмечено ранее, у большинства современных ПЛИС ячейки LUT-блоков могут быть сконфигурированы на выполнение функции распределённой RAM или ROM памяти, а также на выполнение функции сдвигового регистра SRL. Для проверки функции распределённой памяти предлагается алгоритм функционального контроля, показанный на рис. 5. Функциональная схема работы алгоритма зависит от типа используемого LUT, на основе которого строится распределённая память. На рис. 5 приведён пример для распределённой памяти на основе LUT с четырьмя входами.

Для этого 4 RAM блока соединяются параллельно, как показано на рис. 5, и формируют одну сканирующую ячейку. Запись информации происходит по положительному перепаду тактового сигнала, а чтение – по отрицательному, для чего на вход OUTCLK подаётся инвертированный сигнал INCLK. Чтобы правильно детектировать возможные возникающие ошибки, в памяти предустановлены значения, противоположные подаваемым тестовым сигналам. Как можно увидеть из схемы, сигналы на входе сканирующей ячейки совпадают с сигналами на выходе, что позволяет объединить ячейки в сканирующую цепь. Для проверки всех возможных ошибок требуется четыре конфигурационные прошивки, где к каждому входу данных ячеек памяти подключаются разные сигналы in.

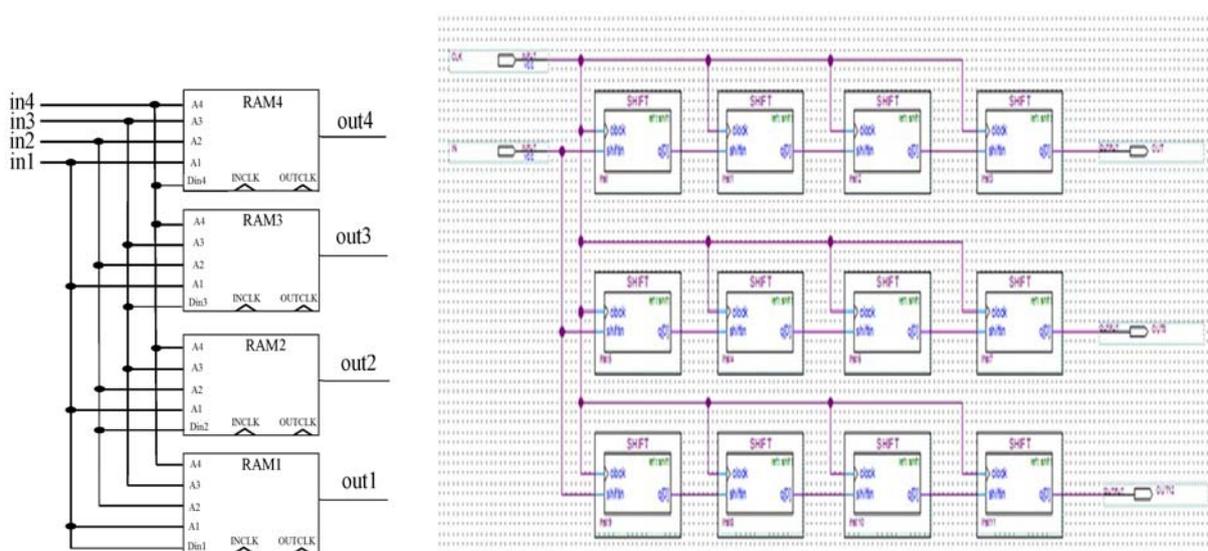


Рис. 5. Ячейка сканирующей цепочки для проверки памяти (слева) и пример реализации конфигурационной прошивки для проверки сдвиговых регистров на основе LUT (справа)

Как отмечено ранее, LUT-блок большинства современных ПЛИС может также конфигурироваться и как блок сдвигового регистра. Для проверки этой функции LUT используются алгоритмы, принцип работы которых схож с вышеописанными. Для этого сдвиговые регистры объединяются в сканирующую цепь, где функции сканирующей ячейки выполняет один сдвиговый регистр. При возникновении ошибки сигнал на выходе сканирующей цепочки будет отличаться от входного сигнала. Пример реализации конфигурационной прошивки приведён на рис. 6.

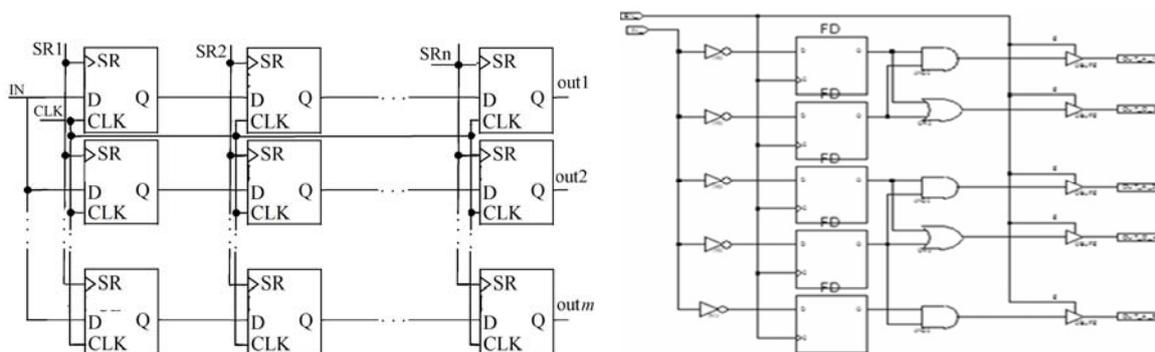


Рис. 6. Алгоритм проверки программируемых регистров в CLB (слева) и пример реализации в виде конфигурационной прошивки (справа)

Помимо LUT-блоков CLB-блок состоит из программируемых регистров, которые могут быть сконфигурированы на выполнение функции триггера или защёлки (latch) и логики быстрого переноса. Для проверки логики быстрого переноса используется тот же принцип, что и для проверки сдвиговых регистров. Алгоритм проверки программируемых регистров следующий. Все регистры конфигурируются как D-триггеры и объединяются в последовательности. Но поскольку триггер может иметь сигналы сброса/установки, функционирование которых также должно быть проверено, то последовательности соединяются так, как показано на рис. 6, и представляют собой тестовую матрицу.

На вход каждой сканирующей ячейки подаётся одинаковый тестовый сигнал. Если в каком-либо из триггеров возникает сбой, то его можно детектировать на выходах. Проверка сигналов сброса/установки происходит аналогично. В зависимости от запрограммированной полярности подаются управляющие сигналы на вход сброса SR. Соответственно, если отсутствует реакция какого-либо триггера на управляющие сигналы, то это может детектироваться на выходах тестовых цепочек.

Современные ПЛИС помимо внутренней программируемой логики содержат в своём составе огромное количество встроенных блоков. В ПЛИС встречаются следующие функциональные блоки: блоки синхронизации DCM и PLL, арифметические блоки (умножители), процессорные ядра (DSP – digital signal processor), блочная память, высокоскоростные приёмопередатчики, блоки аналоговой обработки сигналов (встроенные АЦП) и т.д. Для проверки каждого из блоков разрабатываются алгоритмы функционального контроля на основе метода схем самодиагностики BIST.

На рис. 7 приведён пример реализации алгоритма функционального контроля блока PLL. Принцип работы заключается в том, что на вход PLL подаётся сигнал с эталонного внешнего генератора, а полученные после умножения/деления частоты контролируются схемой измерения частоты. На рис. 7 приведён пример контроля блока PLL методом считывания выходных откликов при помощи интерфейса JTAG и утилиты SignalTap II фирмы Altera.

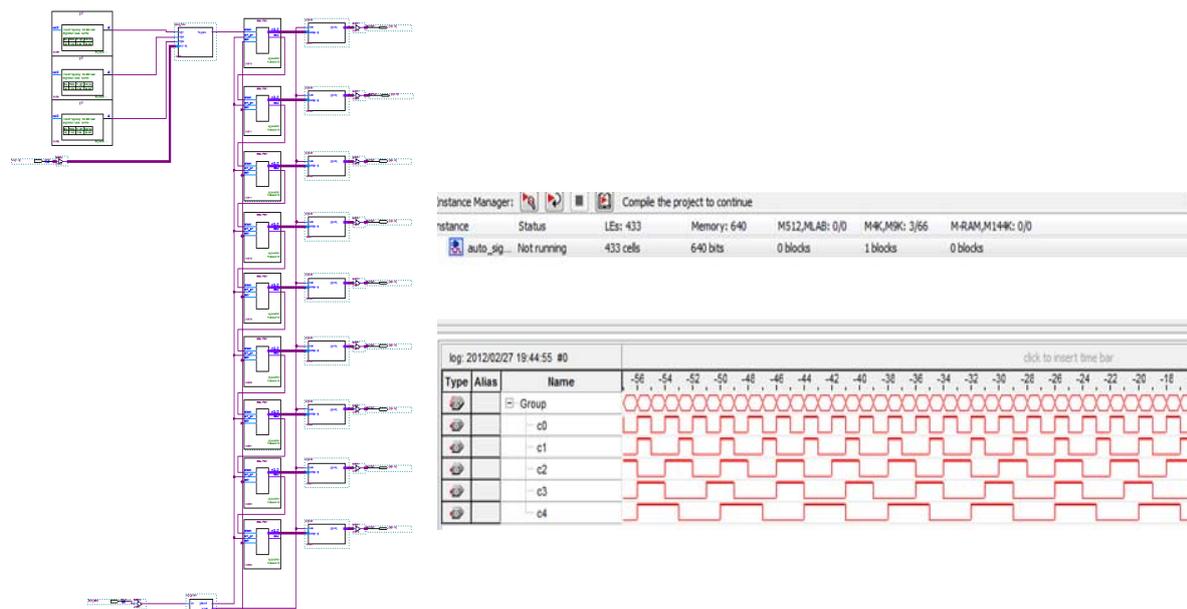


Рис. 7. Алгоритм проверки блоков PLL методом схем самодиагностики BIST (слева) и методом считывания выходных откликов (справа)

Заключение

Проведённый анализ методов показал, что для проведения контроля внутренней логики ПЛИС наиболее удобно использовать метод повторяющихся тестовых множеств (ILA-Iterative Logic Arrays), а для контроля встроенных функциональных блоков ПЛИС наиболее эффективно применять метод с использованием схем самодиагностики BIST. По результатам сравнительного анализа имеющейся аппаратуры и на основе разработанных алгоритмов функционального контроля ведётся работа по разработке тестера функционального контроля ПЛИС и по оценке и улучшению эффективности алгоритмов функционального контроля ПЛИС.

Библиографический список

1. Краснов М.И., Огурцов А.А. Аппаратура для функционального контроля ПЛИС // Контроль. Диагностика. 2013. № 9 (12). С. 49-54.
2. Toutounchi S., Lai A. FPGA test and coverage // IEEE International Test Conference (TC). 2002. V. 67. P. 599-607. DOI: 10.1109/TEST.2002.1041811
3. Dutton B.F., Stroud C.E. Built-in self-test of configurable logic blocks in Virtex-5 FPGAs // 41st Southeastern Symposium on System Theory. 2009. P. 230-234. DOI: 10.1109/ssst.2009.4806778
4. Zhang Z., Zhiping Wen Z., Chen L., Zhou T., Zhang F. BIST approach for testing configurable logic and memory resources in FPGAs // IEEE Asia-Pacific Conference on Circuits and Systems, Proceedings, APCCAS. 2008. P. 1767-1770. DOI: 10.1109/apccas.2008.4746383
5. Huang W.K., Meyer F.J., Lombardi F. Multiple fault detection in logic resources of FPGAs // IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems. 1997. P. 186-194. DOI: 10.1109/dftvs.1997.628324
6. Bareisa E., Jusas V., Motiej K., Seinauskas R. Black box fault models // Information Technology and Control. 2006. V. 35. P. 177-186.

7. Rearick J., Eklow B., Posse K., Crouch A., Bennetts B. IJTAG (Internal JTAG): A Step Toward a DFT Standard // IEEE International Conference on Test. 2005. DOI: 10.1109/test.2005.1584044

8. IEEE Std. 1149.6-2003. IEEE Standard for Boundary-Scan Testing of Advanced Digital Networks. New York: Institute of Electrical and Electronics Engineers, 2003.

9. IEEE Std. 1500-2005. IEEE Standard Testability Method for Embedded Core-Based Integrated Circuits. New York: Institute of Electrical and Electronics Engineers, 2005.

CONDUCTING FUNCTIONAL CONTROL OF FIELD-PROGRAMMABLE GATE ARRAYS

© 2017

A. A. Ogurtsov Research Engineer;
Joint-Stock Company Russian Space Systems, Moscow, Russian Federation
sanoyashi@mail.ru

Approaches to accomplishing functional control of field-programmable gate arrays (FPGA) are described. The first part of the article describes the existing equipment that allows conducting functional tests of FPGAs. A table comparing the main parameters of the equipment available in the market is given. The table contains the main parameters of the equipment that make it possible to simplify the process of conducting functional tests of FPGAs. The main advantages and disadvantages of using this equipment are shown on the basis of the analysis. In the second part of the article the author describes the main methods of conducting functional tests of FPGAs. When developing algorithms, the author divides all the internal structure of a FPGA into two functionally independent parts. The first part is the internal programmable logic, and the second part contains functional blocks built into a FPGA. The article gives examples of algorithms developed on the basis of the methods described and configuration firm-ware produced with the help of these algorithms. The developed algorithms carry out functional control of the internal logic, as well as of function blocks built into a FPGA. To develop algorithms for functional control of internal logic the author uses the method of iterative logic arrays. A method of creating built-in self-test structures is used for the control of function blocks built into a FPGA. In the final part the author draws conclusions and describes the directions of his further research.

Field-programmable gate array (FPGA); FPGA test; automatic test equipment; configurable FPGA; FPGA functional control; equipment for functional control.

Citation: Ogurtsov A.A. Conducting functional control of field-programmable gate arrays. *Vestnik of Samara University. Aerospace and Mechanical Engineering*. 2017. V. 16, no. 4. P. 137-146. DOI: 10.18287/2541-7533-2017-16-4-137-146

References

1. Krasnov M.I., Ogurtsov A.A. Equipments for FPGA functional test. *Testing. Diagnostics*. 2013. No. 9 (12). P. 49-54. (In Russ.)

2. Toutouchi S., Lai A. FPGA test and coverage. *IEEE International Test Conference (TC)*. 2002. V. 67. P. 599-607. DOI: 10.1109/TEST.2002.1041811

3. Dutton B.F., Stroud C.E. Built-in self-test of configurable logic blocks in Virtex-5 FPGAs. *41st Southeastern Symposium on System Theory*. 2009. P. 230-234. DOI: 10.1109/ssst.2009.4806778

4. Zhang Z., Zhiping Wen Z., Chen L., Zhou T., Zhang F. BIST approach for testing configurable logic and memory resources in FPGAs. *IEEE Asia-Pacific Conference on Circuits and Systems, Proceedings, APCCAS*. 2008. P. 1767-1770. DOI: 10.1109/apccas.2008.4746383

5. Huang W.K., Meyer F.J., Lombardi F. Multiple fault detection in logic resources of FPGAs. *IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*. 1997. P. 186-194. DOI: 10.1109/dftvs.1997.628324
6. Bareisa E., Jusas V., Motiej K., Seinauskas R. Black box fault models // *Information Technology and Control*. 2006. V. 35. P. 177-186.
7. Rearick J., Eklow B., Posse K., Crouch A., Bennetts B. IJTAG (Internal JTAG): A Step Toward a DFT Standard. *IEEE International Conference on Test*. 2005. DOI: 10.1109/test.2005.1584044
8. IEEE Std. 1149.6-2003. IEEE Standard for Boundary-Scan Testing of Advanced Digital Networks. New York: Institute of Electrical and Electronics Engineers, 2003.
9. IEEE Std. 1500-2005. IEEE Standard Testability Method for Embedded Core-Based Integrated Circuits. New York: Institute of Electrical and Electronics Engineers, 2005.