

Методы встраивания средств тестирования в устройства с использованием средств автоматизации реинжиниринга

О.В. Ненашев

Санкт-Петербургский государственный политехнический университет,
nenashev@kspt.icc.spbstu.ru

Аннотация — В работе описан подход к встраиванию средств тестирования и самодиагностики, основанный на средствах автоматизации реинжиниринга (САР) цифровых устройств. Предлагаемый подход использует гибридную модель для представления информации из различных уровней описания в алгоритмах анализа и трансформации устройств. В работе описаны архитектуры встраиваемых в устройство тестовых агентов и интерфейсов. Предлагаемые подходы были реализованы на базе прототипа САР, называемого PHRT. Прототип был успешно внедрён в ряде проектов, результаты подтвердили применимость предлагаемых подходов. В работе показан пример создания на базе PHRT интегрированной среды разработки, решающей задачу автоматизированного тестирования на ПЛИС сложной системы на кристалле. В предлагаемом примере удалось снизить общие затраты на тестирование более чем в два раза.

Ключевые слова — автоматизация реинжиниринга, гибридная модель, инструментирование, автоматизация тестирования, САПР, внутрисхемное тестирование.

I. ВВЕДЕНИЕ

Для того чтобы соответствовать изменяющимся требованиям, системы автоматизации и вычислительной техники требуют доработки в течение своего жизненного цикла. Изменение характеристик системы требует внесения в устройства функциональных и архитектурных изменений. Этот процесс называют реинжинирингом электронных систем. В современном мире стремятся повторно использовать существующие наработки для снижения проектных затрат, поэтому реинжиниринг составляет значительную часть цикла разработки систем.

Одной из типовых задач реинжиниринга является встраивание в устройства средств тестирования и диагностики, позволяющих снизить затраты на выявление и устранение неисправностей системы, что улучшает технико-экономические и эксплуатационные характеристики изделий в целом.

Как и иные задачи реинжиниринга, встраивание средств тестирования требует большое количество временных и человеческих ресурсов проекта, поэтому

актуальна задача автоматизации данного процесса. Для ее решения используется специальный класс систем автоматизации проектирования (САПР), называемый средствами автоматизации реинжиниринга (САР). Они могут представлять собой отдельные инструментарии или входить в состав интегрированных сред разработки. В САР для описания алгоритмов анализа и трансформации устройств используется внутренняя модель представления. В общем случае к этой модели и САР предъявляются следующие требования:

- 1) Возможность интеграции в существующие процессы разработки электронных устройств.
- 2) Поддержка популярных языков описания устройств (HDL – Hardware Description Language): VHDL, Verilog, SystemC и SystemVerilog.
- 3) Возможность повторного использования ранее разработанных алгоритмов реинжиниринга.
- 4) Работа в рамках заданных ограничений по системным ресурсам и временным характеристикам.

В данной работе предложен подход к встраиванию средств тестирования в САР на базе многоуровневых моделей устройств, что позволяет модифицировать структуру на уровне низкоуровневых компонентов устройства (логических вентилях, блочной памяти на ПЛИС и т.п.) и при этом анализировать архитектуру устройства на уровне исходных HDL-описаний. Подход основан на гибридной модели PHRT, которая позволяет хранить максимально полную информацию об устройстве и при этом не зависеть от исходного представления устройства [1]. При этом возможно избежать множественных реализаций одного алгоритма для разных HDL и связанных с этим сложностей внесения изменений и поиска ошибок.

Подход в соответствии с постановкой задачи реинжиниринга предназначен для встраивания средств тестирования, диагностики и самодиагностики (BIST – Built-In Self-Test) в готовые архитектуры и реализации, но также применим и в методологиях разработки архитектуры с учётом средств тестирования (DFT – Design For Testability). В последнем случае объектом реинжиниринга могут быть прототипы устройств, а целью – выбор оптимальных для проекта тестовых интерфейсов и архитектуры тестовой оснастки.

II. СВЯЗАННЫЕ РАБОТЫ

В предыдущих работах автором была предложена новая гибридная модель представления устройств и методики работы с ней для типовых задач реинжиниринга. На их основе построен прототип САПР и произведена апробация на примерах частных задач реинжиниринга устройств, результаты которой подтвердили применимость предлагаемой гибридной модели и основанных на ней методик. Ниже описаны гибридная модель и разработанный прототип САПР, использованные в данной работе.

A. Гибридная модель представления устройств

Гибридная модель была предложена как основа для построения алгоритмов реинжиниринга устройств в расширяемых САПР, которые играют роль базовых инструментариев при разработке специфичных для пользователя алгоритмов реинжиниринга. При этом требуется предоставить максимально полную информацию об устройстве для анализа, для чего в предлагаемой гибридной модели совмещается несколько уровней представления.

Гибридная модель является многоуровневым объектным представлением систем и сочетает в себе преимущества низкоуровневых структурных описаний (нетлистов) и высокоуровневых описаний на HDL. Модель обладает следующими основными свойствами:

- совместимость с наиболее распространёнными HDL: VHDL, Verilog, SystemC и SystemVerilog;
- независимость от исходного формата описания устройства, переносимость основанных на модели алгоритмов реинжиниринга;
- возможность одновременной обработки нескольких устройств в единой модели;
- минимальный набор типов элементов;
- возможность специализации модели для частных задач реинжиниринга за счёт точек расширения и программной реализации.

Гибридная модель включает минимальный набор типов элементов, которые имеют сложную объектную структуру, предназначенную для программной обработки в САПР. На рис. 1 приведена структура гибридной модели, серым цветом обозначено структурное подмножество элементов.

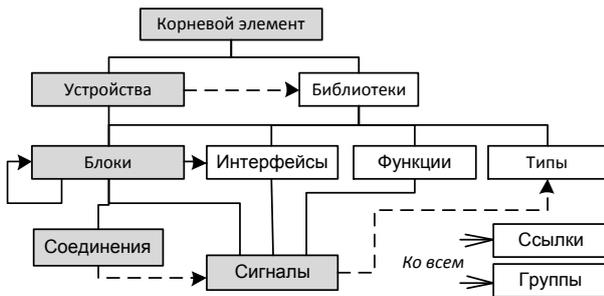


Рис. 1. Типы элементов гибридной модели

Предлагаемая модель поддерживает структурное наследование и обратные вызовы для верификации устройств. Поведенческие описания и условная генерация элементов поддерживаются через механизм функций. Также предлагается механизм ссылок, который позволяет наследовать реализацию элементов из других элементов или библиотек. Более полное описание модели приводится в [1].

B. Прототип САПР

Программный инструментарий PHRT является прототипом расширяемого САПР для автоматизации частных задач анализа и модификации устройств. PHRT основан на гибридной модели устройств, что даёт ряд преимуществ перед работающими с исходными кодами средствами: интегрированными средами разработки, DMS SRT [2] и другими САПР. Основная область применения PHRT - это повышение надёжности устройств, оптимизация и встраивание средств тестирования и самодиагностики. Прототип интегрируется с внешними САПР для использования реализованных в них возможностей анализа и синтеза устройств. Структура прототипа приведена на рис. 2.

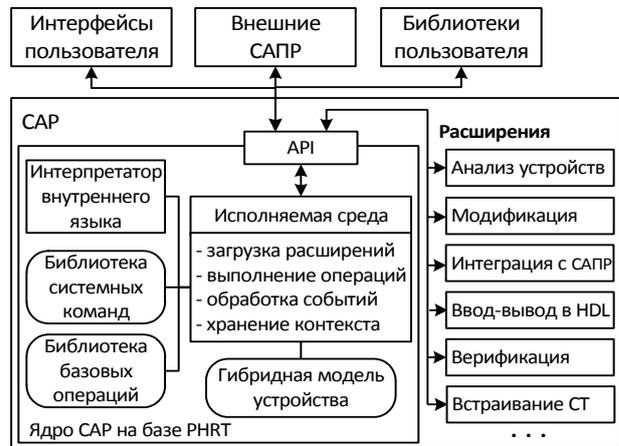


Рис. 2. Структура разработанного прототипа САПР

Ядро средства предоставляет исполняемую среду и набор базовых операций для работы с гибридной моделью. Алгоритмы проведения реинжиниринга и библиотеки операций реализуются в PHRT в виде функциональных расширений на языках Java или Tcl. Совокупность ядра среды и расширений формирует специализированную САПР, способную решать частные пользовательские задачи реинжиниринга.

В рамках предыдущих исследований автором разработан ряд расширений PHRT для типовых задач реинжиниринга. Ниже приведены примеры типовых задач при встраивании средств тестирования, поддерживаемых существующими расширениями:

- импорт/экспорт синтезируемых описаний архитектур устройств на языках EDIF и VHDL;
- базовые операции над гибридной моделью: добавление, перенос и удаление элементов, переименование, доступ к свойствам и пр.;

- соединение сигналов на разных уровнях устройства с модификацией интерфейсов;
- анализ архитектуры устройств: оценка площади кристалла, временной анализ;
- интеграция с САПР Quartus II фирмы Altera для прототипирования на ПЛИС: импорт/экспорт проектов, частичный синтез компонентов.

Более подробно прототип RHRT описан в [3].

III. ВСТРАИВАНИЕ СРЕДСТВ ТЕСТИРОВАНИЯ

В данной главе описан подход к созданию оснастки для внутрисхемного тестирования компонентов устройства. Рассмотрены решаемые при проведении реинжиниринга задачи построения тестовых агентов и интерфейсов, а также их соединения в единую систему тестирования устройства.

A. Задачи, решаемые при встраивании средств тестирования в устройство

Для проведения внутрисхемного тестирования нужно обеспечить полную управляемость и частичную наблюдаемость тестируемого компонента системы. Полная управляемость требуется для задания начального состояния компонента перед запуском теста, а наблюдаемость необходима для принятия решения о корректности работы компонента. Критерии наблюдаемости тестируемого модуля определяются его архитектурой и набором тестов.

Ниже приведены задачи реинжиниринга, которые требуется решить при встраивании в устройство средств внутрисхемного тестирования:

- 1) Изоляция тестируемого компонента от влияния остальной части системы.
- 2) Изоляция системы от влияния компонента при проведении тестирования. В ряде случаев может быть использован останов системы.
- 3) Встраивание средств генерации тестовых векторов и/или установки требуемого состояния модуля для обеспечения управляемости.
- 4) Встраивание диагностических интерфейсов для наблюдения за состоянием системы.
- 5) Соединение тестовых агентов с интерфейсами через уровни иерархии устройства.
- 6) Контроль соответствия инструментированного устройства поставленным ограничениям по системным ресурсам и временным характеристикам.

B. Тестовые агенты

На рис. 3 приведена структура предлагаемого тестового агента, который решает перечисленные выше задачи изоляции устройства. Для изоляции остальной части системы тестовый агент может быть подключен непосредственно к выходам устройства, если у него есть соответствующие интерфейсы управления (например, общий сигнал сброса).

При реинжиниринге в САПР тестовый агент можно реализовать как универсальный параметризуемый компонент. Поскольку гибридная модель независима от исходных языков описания, то требуется только одна реализация, которая может быть подключена к описанным на других языках модулям системы.

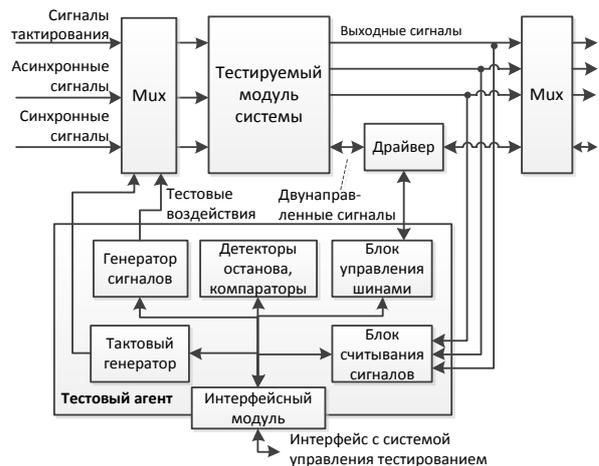


Рис. 3. Организация тестового агента

При реализации алгоритмов на базе САПР тестовые агенты могут быть реализованы как конфигурируемые блоки, которые могут быть параметризованы в соответствии с диагностируемыми сигналами.

C. Построение системы тестирования

В устройстве может присутствовать большое число тестовых агентов, поэтому возникает проблема их подключения к внешним средствам тестирования. При реинжиниринге устройства предлагается также встраивать систему управления тестированием (СУТ), которая будет подключаться ко всем агентам и при этом взаимодействовать с внешним окружением через стандартный интерфейс (например, JTAG). Могут быть встроены следующие классы подобных систем:

- дешифратор тестовых агентов, управляемый внешней системой;
- СУТ с памятью для промежуточного хранения тестовых векторов при проведении тестов на реальных скоростях работы;
- модуль самодиагностики устройства (BIST).

IV. МЕТОДИКА ВСТРАИВАНИЯ ТЕСТОВЫХ АГЕНТОВ

В данной главе рассмотрены методики встраивания тестируемого модуля в устройство. В качестве входных описаний одновременно берутся исходный HDL-код и сформированный при синтезе в САПР список соединений. Требования по тестовым агентам задаются при помощи списка портов и сигналов, к которым требуется подключить внешние выводы. По итогам трансформации в САПР генерируется синтезируемое описание, которое подаётся во внешние средства синтеза. Порядок проведения реинжиниринга в этом случае показан на рис. 4.

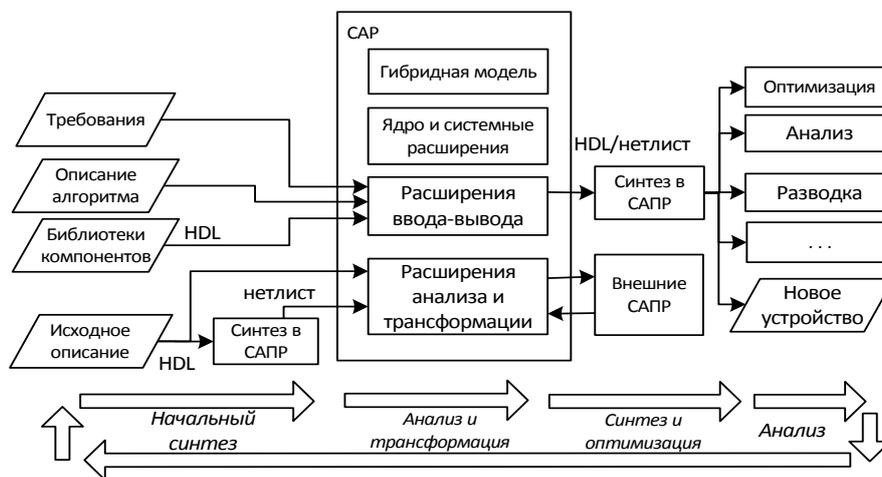


Рис. 4. Процесс проведения реинжиниринга при встраивании средств тестирования

Предлагается использовать итеративный подход к проведению реинжиниринга, в цикле которого производится модификация архитектуры устройства в САПР и промежуточный синтез внешними САПР для контроля соответствия поставленным требованиям и ограничениям. Решение об окончании реинжиниринга принимается разработчиком или САПР при полной автоматизации процесса. За счёт реализации эффективных алгоритмов анализа и трансформации в САПР возможно снизить число итераций вплоть до одной. Исходные нетлисты синтезируются внешними средствами для построения низкоуровневой части гибридной модели.

При встраивании средств тестирования в устройства выделяются следующие стадии алгоритма реинжиниринга:

- 1) Синтез исходного нетлиста в сторонней САПР.
- 2) Импорт гибридной модели из исходных описаний.
- 3) Импорт списка компонентов, которые должны быть инструментированы для проведения тестов.
- 4) Импорт библиотечных компонентов.
- 5) Генерация тестовых агентов из библиотечных компонентов в соответствии со списком.
- 6) Генерация СУТ из библиотечных компонентов на основании сгенерированных агентов.
- 7) Подключение тестовых агентов к СУТ через уровни иерархии устройства.
- 8) Генерация выходного описания.
- 9) Повторный синтез сгенерированного описания для получения тестовой конфигурации устройства.
- 10) Анализ соответствия устройства требованиям и принятие решения о завершении реинжиниринга.

Импорт модели и генерация выходного описания осуществляется средствами САПР. Шаги 5-7 алгоритма оперируют с внутренней моделью устройства, поэтому они независимы от исходных описаний устройства. Таким образом, однажды реализованный алгоритм может быть применён для произвольных устройств, описанных на любых HDL, поддерживаемых САПР.

V. АПРОБАЦИЯ ПРЕДЛОЖЕННЫХ ПОДХОДОВ

В данной главе описаны результаты апробации предложенных подходов с использованием прототипа PHRT на примере встраивания средств тестирования в систему на кристалле. Показан пример построения САПР и его встраивания в маршрут проектирования системы. Также приведены сравнительные результаты синтеза устройств.

A. Подход к проведению апробации

Для апробации предлагаемых подходов поставлена задача автоматизации внутрисхемного тестирования устройств на ПЛИС в процессе разработки. При этом тестовая версия устройства может значительно отличаться из-за необходимости встраивания средств тестирования в случае, когда они не поддерживаются аппаратной платформой. При частом проведении тестирования требуется параллельная поддержка нескольких версий системы, что увеличивает затраты и риск возникновения ошибок.

Решением проблемы является автоматизированная генерация тестовой версии системы на основании исходной версии системы и требований к проведению тестирования. Такая задача является частным случаем реинжиниринга и может решаться при помощи предлагаемых в работе подходов. В рамках апробации решены следующие задачи:

- 1) разработка расширения PHRT для встраивания тестовых агентов и интерфейсов в устройство;
- 2) интеграция САПР в процесс проектирования посредством разработки плагина к среде проектирования на базе Eclipse;
- 3) тестирование разработанного прототипа САПР на примерах реальных устройств.

B. Построение специализированного САПР

Предложенные подходы внедрены в ряде проектов, посвящённых разработке специализированных систем на кристалле. Они успешно опробованы на сложных

системах на кристалле, включающих процессорные ядра LEON3 и OpenRISC с открытыми исходными описаниями на HDL [4]. Реализация подхода в САП позволила автоматизировать встраивание средств тестирования во внутренние компоненты устройства (регистровая память, АЛУ) без модификации исходных описаний. На рис. 5 приведена схема устройства с тестовой обвязкой, генерируемой при помощи прототипа САП.

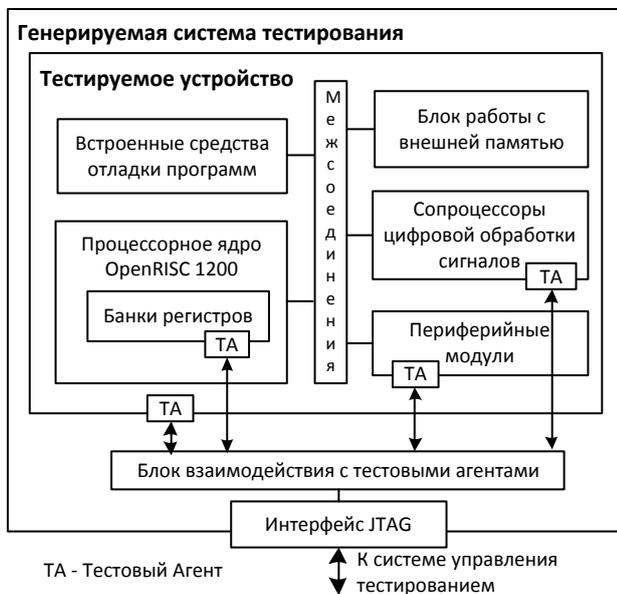


Рис. 5. Пример встраивания тестовых агентов в систему

В предлагаемой реализации интегрированной среды пользователь редактирует исходные описания устройств на HDL и задаёт тесты при помощи редактора тестовых векторов или языка VHDL. После подготовки тестов пользователю достаточно нажать кнопку в графическом интерфейсе, после чего система автоматически произведёт инструментирование устройства и запустит тесты на ПЛИС. По завершении тестов результаты отображаются в графическом интерфейсе системы. На рис. 6 приведен пример специализированной САПР, построенного на базе интегрированной среды разработки Eclipse и ранее созданного прототипа САП.

С. Результаты проведения реинжиниринга

Предложенная автоматизация позволила сократить общие проектные затраты на функциональное тестирование реализаций на ПЛИС более чем в два раза. Таким образом, подтверждена эффективность применения предложенных подходов для встраивания средств тестирования в сложные устройства.

После реализации всех необходимых компонентов генерация и синтез тестовой версии устройства на типовой персональной ЭВМ занимают менее получаса, из которых лишь десятая часть тратится на выполнение реинжиниринга в САП. Решение аналогичной задачи в типовых САПР занимает более восьми часов из-за большого числа уровней иерархии в

тестируемом устройстве, сквозь которые проводятся сигналы для подключения к тестовому интерфейсу.

В таблице 1 приведены сравнительные результаты ручного и автоматизированного инструментирования для системы на кристалле, включающей минимальный набор периферии и процессорное ядро OpenRISC1200 с внешней памятью программ и данных. Восемь тестовых агентов встраиваются в интерфейсы устройства и в несколько компонентов на разных уровнях иерархии (рис. 6), агенты подключают к СУТ 68 управляемых сигналов и 84 - наблюдаемых. В качестве тестового стенда использованы отладочные платы на базе ПЛИС Cyclone IV компании Altera.

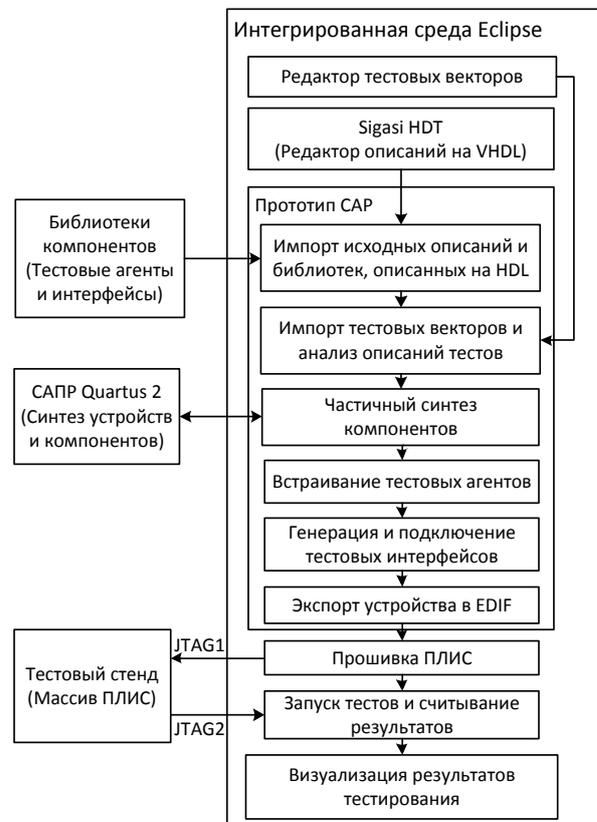


Рис. 6. Процесс проведения тестирования устройства при помощи интегрированной среды на базе САП

Видно, что автоматическое встраивание средств тестирования требует больше системных ресурсов и приводит к снижению максимальной частоты устройства. Результат обусловлен восстановлением низкоуровневой архитектуры из нетлиста из-за “сквозных” соединений между уровнями иерархии, что приводит к отличиям в исходных и выходных нетлистах даже при отсутствии преобразований в PHRT. В свою очередь это приводит к разным результатам оптимизации при синтезе устройства в САПР Quartus2. Различия в результатах синтеза составляют не более 5% от общих системных ресурсов, что приемлемо для тестовых прототипов устройств. Для улучшения показателей требуется доработка используемых в прототипе PHRT методик импорта и объединения нетлистов.

Таблица 1

Результаты встраивания устройства

Эксперимент	F_{max} , МГц	LCELL	Регистры	Блочная память, бит
Исходное устройство	39,31	8055	3105	53888
Тестовые интерфейсы	56,14	516	270	0
Ввод-вывод в САР без преобразований	37,18	8247	3105	53888
Ручная модификация	39,15	8752	3417	53888
Автоматическая модификация	37,02	8916	3417	53888

D. Анализ недостатков предлагаемых подходов

В работе выявлены следующие проблемы использования САР и предложенных подходов при решении задач автоматизации внутрисхемного тестирования:

- 1) Значительные различия исходных HDL-описаний и синтезированных нетлистов, которые возникают при оптимизации устройства или экспорта в форматы без сохранения иерархии. Например, формат нетлистов EDIF допускает “сквозные соединения”, что требует дополнительной обработки при импорте модели.
- 2) Слабая поддержка поведенческих описаний в гибридной модели. Фактически, для их анализа нужно проводить частичный синтез компонентов, что снижает скорость выполнения операций в САР.
- 3) Необходимость дополнения исходной гибридной модели метаданными для задания ограничений по системным ресурсам и временным характеристикам целевой системы.

Несмотря на указанные недостатки подходов, подтверждена их применимость для решения задач автоматизации тестирования. В свою очередь, это подтверждает эффективность ранее предложенных моделей и методик для построения САР.

VI. ЗАКЛЮЧЕНИЕ

В работе предложен подход к решению задачи встраивания средств тестирования и самодиагностики в сложные цифровые устройства и системы посредством расширяемых САР общего назначения. Показана актуальность применения многоуровневых моделей в САР для расширения возможностей анализа и трансформации устройств. Предложенные в работе архитектуры тестовых агентов и методики их встраивания в систему реализованы на базе прототипа САР и успешно внедрены в ряде проектов по разработке сложных систем на кристалле, что подтверждает применимость предложенных подходов.

Созданные прототипы программных средств могут быть использованы для апробации при решении задач повышения надёжности устройств и их диагностики. Предлагаемая схема встраивания тестовых агентов позволяет реализовать предложенные в работах [5] и [6] подходы с минимальными затратами на реализацию САР.

Дальнейшим развитием работы является расширение области применения на задачи, которые не поддерживаются существующими САР. Одной из таких задач является запуск тестов из средств симуляции на ПЛИС, что позволит снизить затраты на проведение тестирования за счёт повторного использования тестов. При этом можно будет поддержать кросс-верификацию модели устройства и его аппаратной реализации. Также планируется расширить анализ архитектуры устройств с целью снижения аппаратных затрат на реализацию тестовых агентов в иерархических устройствах.

БЛАГОДАРНОСТИ

Автор приносит благодарность сотрудникам научно-производственных предприятий, на которых проводилась апробация и внедрение прототипа САР и рассмотренных в работе методик встраивания средств тестирования в устройства.

Автор также выражает благодарность своему научному руководителю к.т.н. Филиппову Алексею Семёновичу (доцент кафедры КСПТ, СПбГПУ) за поддержку при проведении исследований.

ЛИТЕРАТУРА

- [1] Ненашев О.В. Расширяемый инструментарий для автоматизации реинжиниринга цифровых систем на кристалле // Университетский научный журнал. СПб.: Санкт-Петербургский университетский консорциум, 2013. №5. С. 194-203.
- [2] Baxter I.D., Pidgeon C., Mehlich M. DMS: Program Transformations for Practical Scalable Software Evolution // Proceedings of the 26th International Conference on Software Engineering ICSE'04. Washington, DC, USA: IEEE Computer Society, 2004. С. 625-634.
- [3] Nenashev O. PHRT: a model and programmable tool for hardware reengineering automation // Proceeding of 9th European Software Engineering Conference. New York: ACM, 2013. С. 719-722.
- [4] Tong J.G., Anderson I.D., Khalid M.A. Soft-core processors for embedded systems // Microelectronics, 2006. ICM'06. International Conference on. IEEE, 2006. С. 170-173.
- [5] Yang J.-S., Nadeau-Dostie B., Touba N.A. Test point insertion using functional flip-flops to drive control points // Test Conference, 2009. ITC 2009. International. IEEE, 2009. С. 1-10.
- [6] Мамутова О.В. Использование встроенного процессора для управления эмуляцией внесения неисправностей типа «сбой» в блоки памяти // Университетский научный журнал. СПб.: Санкт-Петербургский университетский консорциум, 2013. С. 194-203.