УДК 004.4
DOI: 10.15827/0236-235X.119.401-408
Дата подачи статьи: 17.04.17
2017. Т. 30. № 3. С. 401–408

АНАЛИЗ СОВРЕМЕННЫХ МЕТОДОВ ТЕСТИРОВАНИЯ И ВЕРИФИКАЦИИ ПРОЕКТОВ СВЕРХБОЛЬШИХ ИНТЕГРАЛЬНЫХ СХЕМ

Д.И. Слинкин, зав. группой, slin@nm.ru
(Федеральный научный центр Научно-исследовательский институт системных исследований РАН (ФНЦ НИИСИ РАН), Нахимовский просп., 36, корп. 1, г. Москва, 117218, Россия)

Компании, разрабатывающие *сверхбольшие интегральные схемы* (СБИС), придерживаются определенных маршрутов проектирования и входящих в них маршрутов тестирования. Считается, что 60–80 % усилий команд-разработчиков аппаратных средств затрачивается на верификацию и отладку. Важным этапом является отладка моделей уровня регистровых передач (RTL). При этом не существует универсальной методики для решения этой задачи.

В статье анализируются зарубежные и отечественные публикации, посвященные отладке и верификации промышленных проектов СБИС. Разбираются четыре основные методологии: формальная верификация, имитационное тестирование, использование аппаратных ускорителей, создание прототипов на основе *программируемых логических интегральных схем* (ПЛИС). Для каждой из этих методологий приводится информация о способе выявления ошибок, существующих программных и аппаратных инструментах, которые могут быть применены в процессе отладки. Анализируются их особенности, такие как трудоемкость, требования к квалификации и численности команд верификаторов, стоимость необходимых инструментов, наличие метрик для оценки тестового покрытия. Приводятся названия конкретных промышленных проектов СБИС, в которых эти методологии были использованы: микропроцессоры, высокопроизводительные сетевые коммутаторы, графические процессоры. Называются примененные разработчиками инструменты.

Особое внимание уделено отладке проектов на ПЛИС. Разбираются следующие подходы: использование встроенного логического анализатора, внешнее контрольно-измерительное оборудование и их комбинирование.

На основе практического опыта показано, что четыре рассмотренные методологии отладки и верификации проектов СБИС имеют разграниченные области применения. Называются типы микросхем, для которых может быть использована конкретная методология. Кратко резюмируются их достоинства и недостатки.

Ключевые слова: *формальная верификация, имитационное тестирование, аппаратные ускорители, ПЛИС-прототипы, практическое применение.*

Ошибка в проекте *сверхбольших интегральных схем* (СБИС), обнаруженная на ранней стадии, удешевляет ее исправление. Наиболее дорогостоящим является изготовление новых фотошаблонов. Поэтому число итераций, связанных с изменением проекта на этапе тестирования готовых кристаллов, желательно минимизировать. Модель уровня RTL определяет основные характеристики будущей микросхемы. Статья посвящена анализу существующих методов отладки RTL-моделей с целью определения областей их применения. Рассмотрим основные методы отладки RTL-моделей.

Формальная верификация (ФВ). Применяются следующие методы ФВ: дедуктивный анализ, проверка эквивалентности, основанная на утверждениях верификация (ABV), проверка модели.

Дедуктивный анализ представляет собой математическое доказательство соответствия между верифицируемым проектом и спецификацией. Проект и спецификация должны быть представлены в терминах определенной формальной логики. Используемые средства автоматизированного доказательства теорем (theorem prover): HOL, Mizar, PVS, Coq, Isabelle, Alfa, ACL2. Метод применялся для отладки алгоритмов вещественной арифметики (умножение, деление, извлечение квадратного корня) процессора AMD-K7 (ACL2) [1] и сопроцессора вещественной арифметики процессора Intel Itanium (HOL-light) [2].

Основная идея ABV заключается в формализации знаний о работе проекта и применении их при моделировании в автоматическом режиме. Для контроля правильности работы проекта используются утверждения, написанные на языках PSL и SVA. Возможна автоматическая генерация утверждений при помощи DesignPSL. Существуют как свободно распространяемые библиотеки утверждений (OVL), так и коммерческие (QVL, CheckerWare). Библиотечные элементы содержат блоки для тестирования простых структур (счетчики, буферы, константы) и для сложных протоколов (шины AMBA, PCI, USB, интерфейс DDR) [3]. Библиотеки используются также при моделировании, эмуляции или для ПЛИС-прототипов. Для ПЛИС-прототипов возможно комбинирование проверки утверждений и задания режима работы программным тестом.

Проверка эквивалентности. Утверждение о работоспособности модели проекта основывается на доказательстве эквивалентности другой модели. Например, эквивалентности схемы на уровне транзисторов и RTL-модели. Инструментами являются Conformal, Formality, FormalPro.

Проверка модели связана с формальной проверкой выполнения на модели свойств поведения проектируемого объекта, специфицированных на языке формальной логики. Метод может быть полностью автоматизирован. Проблемой является

комбинаторный взрыв числа состояний системы при увеличении числа компонентов. Для ее решения используются символьная верификация на основе бинарных решающих диаграмм (BDDs) или решатели задач выполнимости булевых формул (SAT-solvers). Инструментами являются SMV, Magellan, Forte, Murphy. Метод использовался для верификации протоколов когерентности кэш-памяти в Intel (Murphy) [4] и для графических процессоров NVIDA (Magellan) [5]. Для процессора Intel Core i7 проведена полная ФВ кластера EXE (Forte) [6]. При этом остались незамеченными пять проблем, три из которых были выявлены уже после изготовления кристалла. Проблемы связаны с некорректной спецификацией или с несвоевременным окончанием работ по ФВ [6].

В отличие от других подходов ФВ не обеспечивает равномерного улучшения качества результата тестирования от времени. Методология применяется в крупных компаниях с большими коллективами инженеров-верификаторов. Использование ФВ ограничено отладкой отдельных особо ответственных узлов. Главным недостатком является проверка абстрактной модели, а не реальной системы. Метод обеспечивает высокое качество верификации, но не гарантирует полное отсутствие ошибок.

Имитационное тестирование. В настоящее время имитационное тестирование, или моделирование, является самым распространенным подходом к верификации. Различают системную и модульную верификации. Модульная верификация выполняется для отдельного блока или для нескольких логически связанных блоков. Задачей системной верификации является проверка взаимодействия между компонентами, а следовательно, проверка устройства в целом [7].

Работа устройства моделируется при помощи программного RTL-симулятора. Существуют ПО с открытым исходным кодом (Icarus, GHDL, CVC) и бесплатные программы с ограниченной функциональностью (Active-HDL и ModelSim). Для рабочих мест на ПК применяются ModelSim, Active-HDL, Riviera-PRO, Silos, VeriLogger Extreme и др. Для предприятий предлагаются инструменты, обеспечивающие точное временное моделирование на вентильном уровне, что критически важно при передаче проектов СБИС на производство, например Incisive Enterprise Simulator, ModelSim/SE, Synopsys VCS. Проектирование на ПЛИС не требует дорогих симуляторов, используются инструменты ModelSim, Altera Quartus II и Xilinx ISE.

В процессе имитационного тестирования RTL-модель запускается в рамках тестовой системы (testbensh), для создания которой используются или языки программирования C, C++, или языки описания аппаратуры SystemC, Verilog, VHDL, или специальные языки верификации OpenVera, SystemVerilog. Тестовая система выполняет три основные задачи: генерация тестовых воздействий, проверка правильности поведения модели и оценка полноты тестирования. Применяются следующие методы создания тестовых воздействий: ручная разработка, случайная генерация, генерация на основе шаблонов, генерация на основе конечных автоматов. Анализ ошибок и оценка полноты тестирования выполняются на основе отчета о тестировании. Для проверки правильности поведения модели используются ко-симуляция (сравнение результатов теста с эталонной моделью), тесты со встроенной проверкой и формальные спецификации [8]. Программы-симуляторы обычно имеют возможность оценки полноты покрытия RTL-модели: количество выполненных строк кода, статистика прохождения ветвлений, полнота комбинаций аргументов.

Для создания тестовых стендов используются специальные методологии верификации. В 2010 г. независимая организация Accellera, занимающаяся разработкой стандартов, предложила методологию UVM (Universal Verification Methodology), созданную на основе OVM и включающую в себя AVM, URM, VMM и eRM. Методология UVM поддерживается тремя основными компаниями-разработчиками HDL-симуляторов: Cadence, Synopsys и Mentor.

В UVM применяются объектно-ориентированное программирование и моделирование на уровне транзакций (TLM). Для быстрого создания тестовых окружений и компонент верификации на языке System Verilog служит библиотека классов. Используются случайная генерация входных воздействий на основе ограничений и управляемая покрытием верификация. Предполагается повторное применение компонент верификации и тестовых окружений. UVM считается перспективной и широко распространенной методологией верификации.

В НИИСИ РАН методология UVM использовалась для отладки блока преобразования адресов TLB в составе микропроцессора [9]. Сообщается об использовании UVM для верификации контроллера прерываний APIC микропроцессора «Эльбрус 2S» в ЗАО «МЦСТ» и об использовании OVM для верификации модуля кэша второго уровня того же микропроцессора [7].

Также в НИИСИ РАН была разработана интегрируемая система INTEG, использующая случайную генерацию тестов по шаблонам. Система включает в себя графическую оболочку, графический редактор шаблонов, генератор случайных тестов, симулятор целевого процессора VMIPS (эталонная модель), симулятор RTL-модели (Verilog) и программу сравнения результатов. Система INTEG доказала свою полезность при верификации проекта микропроцессора с архитектурой MIPS64. Совершенствование разрабатываемых в НИИСИ РАН микропроцессоров, изменения в организации, сце-

нариях и инструментах тестирования обусловили разработку модернизированного программного комплекса INTEG2 [10].

В ИСП РАН для верификации микропроцессов и других программируемых устройств (сопроцессоров, контроллеров) на системном уровне разработан инструмент MicroTESK [11]. Генератор создает набор тестовых программ небольшого размера. Имеется поддержка автоматической генерации тестовых шаблонов на основе комбинаторных техник. MicroTESK использовался для отладки MIPS-64 совместимого микропроцессора, DSP-сопроцессора и арифметических сопроцессоров вещественных чисел. Для верификации на модульном уровне используются технология UniTESK и инструмент CTESK, осуществляющий автоматизированную разработку тестов на основе формальных спецификаций. CTESK был применен для отладки Verilog-моделей кэш-памяти второго уровня и буфера трансляции адресов MIPS64-совместимого микропроцессора. Инструмент C++TESK предполагает создание спецификации и всех компонент тестовой системы на языке Си++. Тестовые системы создаются на основе эталонных программных моделей различных уровней абстракции. Сообщается о применении рассматриваемого метода в виде библиотеки классов для девяти проектов верификации модулей промышленных микропроцессоров [12].

Бутылочным горлышком имитационного тестирования является низкая частота моделирования, десятки герц. Скорость моделирования отдельного блока на порядок выше скорости моделирования устройства в целом. Комбинаторная сложность тестов для современных *систем на кристалле* (СнК) высока, поэтому в литературе зачастую сообщается о тестировании только отдельных, наиболее ответственных блоков микропроцессоров. Ограничивающим фактором становится не только машинное время для прохождения тестов, но и время разработки полного набора тестов. Необходимы высокий уровень знаний UVM-инженеров в области разработки аппаратных средств, в области разработки ПО, а также знание методов верификации. Неполная документация способна снизить качество верификации. Недостатком случайной генерации тестов является малая вероятность нахождения сложных ошибок, для которых требуется привести модель устройства в особое состояние. Генерация тестов при обходе конечного автомата позволяет находить сложные ошибки, но требует адекватности используемой эталонной модели [12].

Достоинствами имитационного моделирования являются наличие метрик, позволяющих количественно оценивать тестовое покрытие, и возможность обнаружения непредсказуемых ошибок при выполнении случайных тестов. Автоматическая генерация входных воздействий и автоматическая проверка результатов тестирования снижают трудоемкость тестирования и обеспечивают высокий процент тестового покрытия.

Аппаратные ускорители. Для увеличения скорости моделирования на несколько порядков применяются аппаратные ускорители, или платформы эмуляции. Представленные на рынке ускорители реализованы на специальных процессорах, на заказных или на коммерческих ПЛИС (табл. 1). Все три архитектуры являются масштабируемыми, они позволяют размещать проекты любого размера, от IP-блоков до полных систем. Важным является тестирование взаимодействия на системном уровне, поддерживаемое аппаратными ускорителями. Частота зависит от многих факторов и составляет от единиц МГц до 100 МГц (для HAPS-80). Обычно поддерживаются методология UVM, верификация на основе утверждений, многопользовательская работа, модели уровня транзакций (TLM).

Современные СнК могут включать в себя несколько процессорных ядер и высокопроизводительные интерфейсы. Для управления используются операционная система и драйверы.

Таблица 1

Аппаратные ускорители и платформы для эмуляции

Table 1

Hardware accelerators and platforms for emulation

Платформа	Компания	Микросхемы	Количество вентилей в проекте
Palladium	Cadence	Специализированные процессоры	До 2 млрд
Protium	Cadence	ПЛИС Xilinx Virtex-7	До 100 млн
Veloce II	Mentor	Специализированные ПЛИС Crystal2	От 256 млн до 2 млрд
ZeBu Server-3	Synopsys	ПЛИС Xilinx Virtex-7	300 млн, масштабируется до 3 млрд
HAPS-80	Synopsys	ПЛИС Xilinx Virtex UltraScale	До 1,6 млрд
Prodigy Cloud Cube 32	S2C	ПЛИС Xilinx Virtex-7 (UltraScale, Kintex-7) или Altera Stratix IV	До 1,4 млрд

Эмуляция на основе процессоров реализована в ускорителях семейства Palladium. Аппаратная часть эмулятора состоит из массива специализированных процессоров, выполняющих булевы операции тестируемого проекта. Эмулятор включает в себя от 103 до 104 арифметико-логических устройств. Задача ПО состоит в разделении проекта между процессорами и в оптимальном планировании булевых операций в корректной временной последовательности. Поддержка стандартных интерфейсов реализована за счет SpeedBridges – карт адаптации частот эмулятора и периферийных устройств.

Основными преимуществами Palladium являются быстрое время компиляции и полная видимость проекта на полной скорости эмуляции. Palladium эффективен в режиме ICE, поддерживаемом набором карт SpeedBridges. Имеется возможность оценки питания проекта с целью уменьшения энергопотребления.

Главные недостатки Palladium – большие физические размеры и высокое энергопотребление по сравнению с ПЛИС-эмулятором эквивалентной вычислительной мощности [13]. Ограничена скорость в режиме Transaction-based acceleration. Также следует отметить высокую стоимость аппаратных эмуляторов.

Ускорители на основе специализированных ПЛИС обеспечивают стопроцентный доступ без компиляции пробников и быструю трассировку временных диаграмм. Недостатком является использование фермы рабочих станций для быстрой компиляции, скорость которых ниже, а физические размеры больше, чем у эквивалентного эмулятора на основе коммерческих ПЛИС.

Ускорители на основе коммерческих ПЛИС обладают наименьшими физическими размерами и потребляемой мощностью. Они достигают более высокой скорости выполнения. Недостатки заключаются в меньшей скорости компиляции по сравнению с другими двумя архитектурами, по крайней мере, для проектов в 10 млн логических элементов или меньше. Полная видимость проекта достигается в обмен на высокую скорость эмуляции.

Cadence сообщает об использовании Palladium XP II компанией Realtek для разработки и верификации проекта СнК. Отмечаются ускорение в 250 раз по сравнению с предыдущей методологией, улучшение качества верификации, экономия времени за счет повторного использования более чем 90 % предыдущих настроек среды моделирования, а также оценка потребляемой мощности в пределах 5 % от фактически измеренной мощности СнК.

Palladium XP использовался компанией Altair Semiconductor (выкуплена Sony в 2016 г.) для верификации и валидации проекта Internet of Thing. Сообщается об уменьшении времени цикла разработки на 20 % [14]. Выполнены разработка ПО и его проверка в аппаратном контексте.

Компания NVIDA применяла установку, состоящую из 16 шасси, под названием Tigris (Cadence) для эмуляции архитектуры GPU Fermi. Самая крупная установка Palladium XP под названием Indus использовалась для эмуляции архитектуры следующего поколения Kepler [15].

В НИИСИ РАН Palladium был использован для отладки микропроцессора 1890ВМ8Я. В настоящее время ведется тестирование проектируемого микропроцессора 1890ВМ108Я с использованием Protium.

Компания AMD использовала ZeBy для эмуляции графического процессора (GPU), имеющего 240 млн вентилей. Было задействовано от 50 до 70 ПЛИС. Применен подход, основанный на транзакциях. Проверялось функционирование GPU в среде IBM PC-совместимой ЭВМ. Различные материнские платы IBM-PC моделировались при помощи виртуальной машины VirtualBox. Производилась загрузка ОС Linux, драйвера, выполнялись операции чтения/записи памяти эмулируемого GPU. Также были выполнены 20 базовых аппаратных тестов для GPU [16].

Платформа эмуляции Veloce была использована Barefoot Networks для проверки высокопроизводительного сетевого коммутатора Tofino 6.5 ТБ/сек. Важным было использование инструмента Veloce Virtual LAB Ethernet, включающего в себя генератор и монитор пакетов Ethernet (EPGM) [17].

HAPS-70 использовался для верификации двухъядерного микропроцессора Baikal-T1 компании «Байкал электроникс». Микропроцессор работает на частоте 1,2 ГГц, использует 32-битные ядра MIPS Warrior P5600, имеет в своем составе ряд высокоскоростных периферийных устройств, таких как SATA, PCI Express, и Ethernet 10 Гбит/с. Тестирование проводилось в несколько этапов. Сообщается о загрузке ОС Linux и запуске одного ядра на частоте 25 МГц [18].

Достоинствами аппаратных ускорителей являются короткое время старта работ по верификации и хорошие возможности для отладки.

Недостаток состоит в высокой стоимости при избыточных возможностях для некоторых задач, таких, например, как тестирование контроллера периферийного интерфейса. Частота моделирования существенно ниже по сравнению со специальными платами для отладки ПЛИС-прототипов.

ПЛИС-прототипы. Высокая скорость эмуляции, в сотни МГц, достигается при помощи модулей на основе ПЛИС. Можно использовать как готовые, так и специально разработанные отладочные платы. Возможно непосредственное встраивание ПЛИС-прототипа в реальную систему, например, путем установки в разъем PCI-e IBM-PC совместимой ЭВМ. Отладочная плата содержит различные устройства: генераторы тактовых сигналов, приемопередатчики Ethernet, высокопроизводительное ОЗУ, видеоинтерфейсы.

В НИИСИ РАН отладочные платы разрабатываются и изготавливаются специально в количестве нескольких экземпляров. Такой подход позволяет максимально приблизить их характеристики к будущим модулям за счет установки соответствующих компонент. При этом начало тестирования задерживается до момента получения плат. Стоимость таких плат ниже по сравнению с аппаратными ускорителями. Возможность параллельной разработки системного ПО (драйверов) и микросхемы снижает время выхода на рынок. Разнообразие отладочных плат на рынке свидетельствует об их активном применении.

Существуют три подхода к отладке проектов на ПЛИС: использование встроенного логического анализатора, использование внешнего контрольно-измерительного оборудования и их комбинирование.

Встроенный логический анализатор предполагает размещение на ПЛИС аппаратных средств отладки, предназначенных для отслеживания внутренних состояний в моменты срабатывания триггеров, устанавливаемых разработчиками. Обычно анализатор встраивается в ПЛИС за счет добавления одного или нескольких IP-ядер. Для размещения логики встроенного анализатора и буферов памяти требуется от 5 % до 10 % ресурсов ПЛИС. Через порт JTAG ядро анализатора можно сконфигурировать и динамически управлять им. Этот порт используется и для вывода данных. Встроенный логический анализатор позволяет отслеживать любые внутренние сигналы в реальном времени. Для получения доступа к сигналам работа системы останавливается, цепочки сканирования (scan chains) вместе с захваченными данными передаются из ПЛИС. Затем полученная информация анализируется на IBM-PC с использованием программ с графическим интерфейсом или текстовой консоли. В настоящее время на рынке предлагаются различные встроенные логические анализаторы (табл. 2).

Отладка проектов на ПЛИС может производиться с использованием внешнего контрольно-измерительного оборудования, такого как цифровые осциллографы смешанных сигналов или логические анализаторы. Ведущие изготовители – компании Tektronix и Agilent. Примерами являются осциллографы смешанных сигналов серии MSO4000 и логические анализаторы серии TLA компании Tektronix.

Технологические платы с ПЛИС должны иметь специальные разъемы для подключения внешнего оборудования. Сложность отладки обусловлена ограниченным доступом к внутренним сигналам ПЛИС, взаимным влиянием сигналов на печатной плате. Многие устройства имеют ограниченное число выводов, меньшее, чем количество сигналов, требуемых для отладки. Поскольку возможности встроенных логических анализаторов ограничены, существуют программы, такие как Altera SignalProbe, которые позволяют выбирать нужные внутренние сигналы и перенаправлять их на свободные выводы ПЛИС.

Возможно комбинирование встроенных ядер отладки и внешнего контрольно-измерительного оборудования. Например, ядро Agilent Trace Core 2 (ATC-2) в составе программы Xilinx Chipscope Pro Analyzer специально разработано для подключения внешнего логического анализатора Agilent и обеспечивает доступ анализатора к внутренним сетям проекта на ПЛИС. Логический анализатор обеспечивает задание сложных условий захвата данных. Для сохранения данных используется большой объем ОЗУ, позволяющий отслеживать разнесенные во времени события. Возможно сопоставление сигналов проекта на ПЛИС с другими сигналами системы.

Программа FPGAView от компании First Silicon Solutions (FS2) используется с логическими анализаторами серии TLA или осциллографом смешанных сигналов от кампании Tektronix. Предоставляется возможность измерения различных внутренних сигналов ПЛИС Altera или Xilinx без необходимости перекомпиляции проекта. Автоматическое изменение имен каналов на имена выбранных внутренних сигналов облегчает интерпретацию результатов.

Встроенный логический анализатор не требует дополнительных выводов и использует для реализации своих функций внутренние ресурсы ПЛИС. Объем ОЗУ внутри ПЛИС ограничивает суммарное количество отладочной информации, являющееся компромиссом между количеством захватываемых сигналов и глубиной их отслеживания.

Внешнее контрольно-измерительное оборудование дороже встроенного логического анализатора, но обладает определенными преимуществами. Цифровой осциллограф позволяет запускаться от широкого спектра аналоговых и цифровых сигналов, а также от сигналов последовательных шин, захватывать их с различным разрешением по времени. Внешний логический анализатор открывает доступ к различным состояниям запуска и может захватывать очень длинные последовательности данных с высоким разрешением по времени. Внешний осциллограф позволяет захватывать до 10 млн точек, а логический анализатор до 256 млн. Име-

Таблица 2

Встроенные логические анализаторы

Table 2

Built-in Logic Analyzers

Анализатор	ПЛИС	Интеграция с САПР	Особенность	Количество каналов данных	Объем буфера
Altera SignalTap II	Altera	Quartus II	Частота выше 300 МГц	2 048	128 К
FS2 Logic Navigator	Actel	-	Интегрируется в проект на языке Verilog или VHDL	256	64 К
Xilinx Chipscope Pro	Xilinx	Quartus II	Интеграция с внешним логическим анализатором Agilent	4 096	128 К

ется преимущество в динамичности: не требуется останавливать работу системы для анализа захватываемых данных. Внешнее оборудование позволяет изучать временные диаграммы сигналов с разрешением меньше наносекунды, тогда как встроенный логический анализатор может захватывать данные только синхронно с внутренней тактовой частотой ПЛИС. Использование внешнего контрольно-измерительного оборудования имеет некоторые ограничения: получение доступа к новому набору сигналов требует перекомпиляции проекта, что отнимает время, может привести к изменению временных характеристик и скрыть искомые проблемы. Обычно число отладочных выводов мало, что ограничивает обзор и глубину анализа.

Эти ограничения можно преодолеть путем комбинирования отладочных ядер и внешнего контрольно-измерительного оборудования. Выбор конкретной методики зависит от особенностей проекта. Если отладка ограничивается функциональными проблемами внутри ПЛИС, то достаточно возможностей встроенного логического анализатора. Внешнее оборудование предпочтительнее при определении предельных значений временных характеристик, сопоставлении внутренней активности ПЛИС с работой других устройств на печатной плате.

Недостаток отладки проектов на ПЛИС состоит в отсутствии метрик покрытия RTL-кода и метрик покрытия функциональности. Тесты в основном находят ошибки, поиск которых ведется целенаправленно. Поэтому некоторые ошибки могут остаться незамеченными. Выявить их помогают случайные тесты. Считается, что утверждения System Verilog Assertions (SVA) сокращают время тестирования проекта на 50 %. При их использовании улучшается наблюдаемость проекта, упрощаются поиск и исправление недостатков.

Примером отладки ПЛИС-прототипа является эмуляция ядра x86 микропроцессора Intel Atom (см. [19]). Авторами предложена методология использования RTL-проекта и создания его ПЛИС-синтезируемой версии с применением стандартного инструментария. Использовалась специально разработанная отладочная плата на основе ПЛИС Xilinx Virtex-5. Наиболее сложным аспектом являлась механическая адаптация разъема Socket7 материнской платы и ПЛИС.

Отладка сводилась к регрессионному тестированию и сравнению проекта с существующим процессором Intel Pentium. Чтобы избежать ошибок, вносимых в RTL-код непреднамеренно, использовались три метода: RTL-моделирование, сравнение исполнения программ и онлайн-отладка. Для проверки функциональной полноты выполнялась загрузка стандартных ОС: MS-DOS, Linux, Windows XP и различных приложений. Также оценивалась производительность [19].

В НИИСИ РАН были разработаны ПЛИС-прототипы графического ускорителя и универсального микропроцессора с различными внешними интерфейсами. Графический 2D-ускоритель реализует аппаратную поддержку системы команд X Windows, принятую для ОС семейства UNIX. Ускоритель 2D является встроенным узлом микросхем 1890ВГ10Т и 1890ВГ16Т. Метод встречной оптимизации программ под аппаратуру служит повышению реальной (в отличие от пиковой) производительности, и наоборот, создаваемая аппаратура учитывает назначение и особенности программ пользователей. Метод встречного тестирования предполагает привлечение контрольных тестовых задач от потенциальных пользователей [20]. Предложен метод тестирования производительности и корректности микропроцессоров нацеленными тестовыми программами [21]. Данные методы предполагают использование ПЛИС-прототипов. Они применялись для микропроцессоров 1890ВМ5Ф, 1890ВМ6Я, 1890ВМ8Я и 1890ВМ9Я.

Заключение

Рассмотренные методы отладки и верификации проектов СБИС имеют достаточно разграниченные области применения.

Формальная верификация используется крупными компаниями для ответственных узлов, например, арифметических алгоритмов процессоров, сопроцессоров, кэш-памяти, графических процессоров. Требуется работа коллектива специалистов.

Имитационное моделирование широко распространено и обычно применяется для отладки отдельных блоков микросхем. Частота моделирования составляет десятки герц, что ограничивает область применения. Сильными сторонами являются наличие метрик покрытия, автоматическая генерация тестов, методологии OVM/UVM.

Аппаратные ускорители обеспечивают высокую скорость, обычно несколько МГц. Они позволяют производить тестирование проектов СнК на системном уровне. При этом обеспечивается полная видимость проекта. Недостаток состоит в больших размерах и высокой стоимости аппаратного ускорителя. Основная сфера применения – микропроцессоры, графические процессоры, сетевые приложения. Платы для создания ПЛИС-прототипов обладают наибольшей скоростью в сотни МГц, но меньшими отладочными возможностями. Размеры позволяют встраивать их в реальную систему, а стоимость сравнительно невелика.

Возможен комбинированный подход, включающий, например, имитационное моделирование отдельных блоков и использование ПЛИС-прототипов для проверки проекта в целом.

Литература

1. Russinoff D. A mechanically checked proof of IEEE compliance of the floating-point multiplication, division, and square root

algorithms of the AMD-K7* processor. Jour. of Computation and Mathematics, 1998, pp. 148–200. URL: http://citeseerx.ist.psu.edu/viewdoc/download?rep=rep1&type=pdf&doi=10.1.1.212.6294 (дата обращения: 07.04.2017).

2. Harrison J. Floating-point verification using theorem proving. URL: http://www.cl.cam.ac.uk/~jrh13/papers/sfm.pdf (дата обращения: 07.04.2017).

3. Лохов А.Л. Современные методы функциональной верификации цифровых HDL-проектов: методология ABV, библиотеки OVL и QVL // Современная электроника. 2010. № 1. С. 56–59.

4. Chou C.-T., Manna P.K., Park S. A simple method for parameterized verification of cache coherence protocols. 2004. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.101.5271&rep=rep1&type=pdf (дата обращения: 07.04.2017).

5. Synopsys' Magellan Deployed by NVIDIA to maximize verification productivity on next-generation graphics processing units. 2004. URL: http://news.synopsys.com/index.php?s=20295&item=122694 (дата обращения: 07.04.2017).

6. Kaivola R., Ghughal R., Narasimhan N., Telfer A., Whittemore J., Pandav S., Slobodova A., Taylor C., Frolov V., Reeber E., Naik A. Replacing testing with formal verification in intel coretmi7 processor execution engine validation. 2010. URL: http://is.muni.cz/el/1433/jaro2010/IA159/um/intel.pdf (дата обращения: 07.04.2017).

7. Ровнягин М.М., Лебедев М.С., Чудновский А.Л. Современные методы верификации и особенности их применения // Современные информационные технологии и ИТ-образование // VI Междунар. науч.-практич. конф.: сб. избр. тр.; [под ред. В.А. Сухомлина]. М.: Изд-во ИНТУИТ.РУ, 2011. С. 1009–1020.

8. Камкин А.С. Верификация микропроцессоров: борьба с ошибками и управление качеством // Электроника: НТБ. 2010. № 3. С. 98–104. URL: http://www.electronics.ru/journal/article/57 (дата обращения: 07.04.2017).

9. Барских М.Е., Аряшев С.И., Рогаткин Б.Ю. Современные методы функциональной верификации RTL-моделей блоков СБИС микропроцессора // Проблемы разработки перспективных микро- и наноэлектронных систем: сб. тр.; [под общ. ред. А.Л. Стемпковского]. М.: Изд-во ИППМ РАН, 2014. Ч II. С. 119–122. URL: http://www.mes-conference.ru/data/year2014/pdf/D066.pdf (дата обращения: 07.04.2017).

10. Захаров А.В., Хисамбеев И.Ш., Котович Н.В., Кравченко А.А., Осипов А.С., Кольцов П.П., Коганов М.А., Грибков И.В., Куцаев А.С. Развитие системы стохастического тестирования микропроцессоров INTEG // Программные продукты и системы. 2010. № 2. С. 14–23.

11. Камкин А.С., Коцыняк А.М., Смолов С.А., Сортов А.А., Татарников А.Д., Чупилко М.М. Средства функциональной верификации микропроцессоров // Тр. ИСП РАН. Т. 26 (1). С. 149–200. URL: http://docplayer.ru/271393-Sredstva-funkcionalnoy-verifikacii-mikroprocessorov.html (дата обращения: 07.04.2017).

12. Чупилко М.М. Динамическая верификация цифровой аппаратуры на основе формальных спецификаций: автореф. дис. канд. физ.-мат. наук. М.: ИСП РАН, 2012. 24 с.

13. Rizzatti L. Hardware emulation: three decades of evolution. Part III. Verification Horizons. 2015. URL: https://verificationacademy.com/verification-horizons/november-2015-volume-11-issue-3/hardware-emulation-three-decades-of-evolution-part-iii (дата обращения: 07.04.2017).

14. Altair semiconductor adopts cadence palladium XP platform for advanced IoT SoC development. 2015. URL: https://www.cadence.com/content/cadence-www/global/en_US/home/company/newsroom/press-releases/pr/2015/altair-semiconductor-adopts-cadence-palladium-xp-platform-for-advanced-iot-soc-development.html (дата обращения: 07.04.2017).

15. Mackey H. Sneak peek: inside NVIDIA'S emulation lab. 2011. URL: https://blogs.nvidia.com/blog/2011/05/16/sneak-peak-inside-nvidia-emulation-lab (дата обращения: 07.04.2017).

16. Ross A., Star A. Enabling greater reliability, scalability and flexibility of GPU emulation at AMD using a hybrid virtual-machine based approach. 2015. URL: http://www.techdesignforums.com/practice/technique/reliable-scalable-flexible-gpu-emulation-using-hybrid-virtual-machine-approach/ (дата обращения: 07.04.2017).

17. Mentor Graphics Veloce emulation platform helps barefoot networks verify the world's first fully programmable switch. Jour. Electronic Engineering. 2016. URL: http://www.eejournal.com/archives/news/20160719_06/ (дата обращения: 07.04.2017).

18. Osipenko P. Synopsys and Baikal Electronics SoC team applies Synopsys HAPS and AMBA transactors to accelerate the availability of prototypes and improve product quality. 2015. URL: https://www.synopsys.com/content/dam/synopsys/verification/prototyping/success-stories/baikal_ss.pdf (дата обращения: 07.04.2017).

19. Wang P.H., Collins J.D., Weaver C.T., Kuttanna B., Salamian S., Chinya G.N., Schuchman E., Schilling O., Doil T., Steibl S., Wang H. Intel atom processor core made FPGA-synthesizable. URL: http://www.cse.wustl.edu/~roger/565M.f12/p209-wang.pdf (дата обращения: 01.04.17).

20. Чибисов П.А. Встречное тестирование высокопроизводительных микропроцессоров: автореф. дис. канд. технич. наук. М.: ИСП РАН, 2013. 23 с.

21. Зубковская Н.В. Метод тестирования производительности и корректности микропроцессоров при помощи нацеленных тестовых программ: автореф. дис. канд. технич. наук М.: НИИСИ РАН, 2013. 24 с.

Software & Systems
DOI: 10.15827/0236-235X.119.401-408

Received 17.04.17
2017, vol. 30, no. 3, pp. 401–408

ANALYSIS OF MODERN METHODS FOR VLSI PROJECT TESTING AND VERIFICATION

D.I. Slinkin [1], *Head of Group, slin@nm.ru*

[1] *Federal State Institution "Scientific Research Institute for System Analysis of the Russian Academy of Sciences" (SRISA RAS), Nakhimovsky Ave. 36/1, Moscow, 117218, Russian Federation*

Abstract. VLSI development companies are keeping selected design flows and a testing process is a part of them. It is considered that from 60 % to 80 % of development teams' efforts are spent on VLSI project verification and debugging. Debugging of register-transfer level (RTL) models is an important phase. There is no universal way to solve this problem.

The article is devoted to the analysis of foreign and domestic publications on industrial VLSI project debugging. There are four main methodologies that are being considered: formal verification, simulation testing, using of hardware accelerators, prototyping based on programmable logic integrated circuits (FPGAs). For each of these methodologies there is information on an error detection method, existing software and hardware debugging tools. The paper analyzes some of their features, such as labor consumption, requirements for qualification and size of verification group, the cost of necessary tools and the availability of metrics to evaluate the test coverage. There are names of some industrial VLSI projects that used these methodologies, such as microprocessors, high-performance network switches, graphics processors. The paper mentions the tools used by the developers.

Special attention is paid to debugging FPGA projects. The following approaches are considered: using the built-in logic analyzer, external control and measuring equipment and their combination.

Finally, based on the practical experience the paper shows that four VLSI project debugging and verifying methodologies have different applications. It mentions the types of VLSI, which use one of these methodologies. Their advantages and disadvantages are briefly summarized.

Keywords: formal verification, simulation testing, hardware accelerators, FPGA prototypes, practical application.

References

1. Russinoff D. A mechanically checked proof of IEEE compliance of the floating-point multiplication, division, and square root algorithms of the AMD-K7* processor. *London Mathematical Society Journ. of Computation and Mathematics*. 1998, pp. 148–200. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?rep=rep1&type=pdf&doi=10.1.1.212.6294 (accessed April 7, 2017).

2. Harrison J. *Floating-Point Verification using Theorem Proving*. Available at: http://www.cl.cam.ac.uk/~jrh13/papers/sfm.pdf (accessed April 7, 2017).

3. Lokhov A. Advanced Methods for Functional Verification of HDL Projects: ABV Methods, OVL and QVL Libraries. *Sovremennaya elektronika* [Modern Electronics]. Moscow, 2010, no. 1, pp. 56–59 (in Russ.).

4. Chou C.-T., Manna P.K., Park S. *A Simple Method for Parameterized Verification of Cache Coherence Protocols*. Intel Corp. 2004. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.101.5271&rep=rep1&type=pdf (accessed April 7, 2017).

5. Synopsys' Magellan Deployed by NVIDIA to Maximize Verification Productivity on Next-Generation Graphics Processing Units. 2004. Available at: http://news.synopsys.com/index.php?s=20295&item=122694 (accessed April 7, 2017).

6. Kaivola R., Ghughal R., Narasimhan N., Telfer A., Whittemore J., Pandav S., Slobodova A., Taylor C., Frolov V., Reeber E, Naik A. *Replacing Testing with Formal Verification in Intel CoreTMi7 Processor Execution Engine Validation*. 2010. Available at: http://is.muni.cz/el/1433/jaro2010/IA159/um/intel.pdf (accessed April 7, 2017).

7. Rovnyagin M.M., Lebedev M.S., Chudnovsky A.L. Modern methods of verification and their applications features. *Sb. izbran. tr. VI Mezhdunar. nauch.-praktich. konf.* "*Sovremennye informatsionnye tekhnologii i IT-obrazovanie*" [Proc. 6th Int. Science and Practice Conf. "Modern IT and IT-education"]. V.A. Sukhomlin (Ed.). Moscow, 2011, pp. 1009–1020 (in Russ.).

8. Kamkin A.S. Microprocessor Verification. Combating Errors and Quality Control. *Elektronika: NTB* [Electronics: STB]. Moscow, 2010, no. 3, pp. 98–104 (in Russ.).

9. Barskikh M.E., Aryashev S.I., Rogatkin B.Yu. Modern methods of functional verification RTL-models blocks for VLSI microprocessor. *Problemy razrabotki perspektivnykh mikro- i nanoelektronnykh sistem: sb. tr.* [Proc. "Development Issues of Advanced Micro and Nanoelectronic Systems"]. A.L. Stempkovsky (Ed.). Moscow, 2014, pp. 119–122 (in Russ.).

10. Zakharov A.V., Khisambeev I.Sh., Kotovich N.V., Kravchenko A.A., Osipov A.S., Koltsov P.P., Koganov M.A., Gribkov I.V., Kutsaev A.S. The development of the system for microprocessor random testing INTEG. *Programmnye produkty i sistemy* [Software & Systems]. Tver, 2010, no. 2, pp. 14–23 (in Russ.).

11. Kamkin A.S., Kotsynyak A.M., Smolov S.A., Sortov A.A., Tatarnikov A.D., Chupilko M.M. Tools for Functional Verification of Microprocessors. *Tr. ISP RAN* [Proc. of ISP RAS]. Moscow, 2014, vol. 26, iss. 1, pp. 149–200 (in Russ.).

12. Chupilko M.M. *Dinamicheskaya verifikatsiya tsifrovoy apparatury na osnove formalnykh spetsifikatsy* [Dynamic Verification of Digital Equipment Based on Formal Specifications]. PhD Author's Thesis. Moscow, 2012, 24 p.

13. Rizzatti L. Hardware Emulation: Three Decades of Evolution - Part III. *Verification Horizons*. Mentor Graphics Verification Academy. 2015, vol. 11, iss. 3. Available at: https://verificationacademy.com/verification-horizons/november-2015-volume-11-issue-3/hardware-emulation-three-decades-of-evolution-part-iii (accessed April 7, 2017).

14. *Altair Semiconductor Adopts Cadence Palladium XP Platform for Advanced IoT SoC Development*. San Jose, Calif., 2015. Available at: https://www.cadence.com/content/cadence-www/global/en_US/home/company/newsroom/press-releases/pr/2015/altair-semiconductor-adopts-cadence-palladium-xp-platform-for-advanced-iot-soc-development.html (accessed April 7, 2017).

15. Mackey H. *Sneak Peek: Inside NVIDIA'S Emulation Lab*. 2011. Available at: https://blogs.nvidia.com/blog/2011/05/16/sneak-peak-inside-nvidia-emulation-lab (accessed April 7, 2017).

16. Ross A., Star A. *Enabling greater reliability, scalability and flexibility of GPU emulation at AMD using a hybrid virtual-machine based approach*. 2015. Available at: http://www.techdesignforums.com/practice/technique/reliable-scalable-flexible-gpu-emulation-using-hybrid-virtual-machine-approach/ (accessed April 7, 2017).

17. Mentor Graphics Veloce Emulation Platform Helps Barefoot Networks Verify the World's First Fully Programmable Switch. *Electronic Engineering*. 2016. Available at: http://www.eejournal.com/archives/news/20160719_06/ (accessed April 7, 2017).

18. Osipenko P. Synopsys and Baikal Electronics SoC Team Applies Synopsys HAPS and AMBA Transactors to Accelerate the Availability of Prototypes and Improve Product Quality. 2015. Available at: https://www.synopsys.com/content/dam/synopsys/verification/prototyping/success-stories/baikal_ss.pdf (accessed April 7, 2017).

19. Wang P.H., Collins J.D., Weaver C.T., Kuttanna B., Salamian S., Chinya G.N., Schuchman E., Schilling O., Doil T., Steibl S., Wang H. *Intel Atom Processor Core Made FPGA-Synthesizable*. Available at: http://www.cse.wustl.edu/~roger/565M.f12/p209-wang.pdf (accessed April 7, 2017).

20. Chibisov P.A. *Vstrechnoe testirovanie vysokoproizvoditelnykh mikroprotsessorov* [Counter Testing of High-Performance Microprocessors]. PhD Author's Thesis. Moscow, 2013, 23 p.

21. Zubkovskaya N.V. *Metod testirovaniya proizvoditelnosti i korrektnosti mikroprotsessorov pri pomoshchi natselennykh testovykh program* [Method for Testing the Performance and Correctness of Microprocessors Using Targeted Test Programs]. PhD Author's Thesis. Moscow, 2013, 23 p.