

ПРЕИМУЩЕСТВА ФОРМАЛЬНОЙ ПАРАМЕТРИЧЕСКОЙ ВЕРИФИКАЦИИ

РЕЙК БРИНКМАНН (RAIK BRINKMANN), вице-президент, OneSpin Solutions

В статье описаны основные особенности формальной параметрической верификации. Особое внимание уделено анализу RTL-кода на наличие неохваченных участков, что позволяет в дальнейшем избежать неопределенностей при встраивании IP-блока.

Увеличение использования объектов интеллектуальной собственности (IP-блоков) является жизненно необходимым фактором для повышения продуктивности проектирования больших и сложных систем на кристалле (СнК). Увеличение эффективности разработки СнК должно сопровождаться быстрым повышением качества верификации. Однако пока этого не происходит. К тому же, качество верификации IP-блоков оставляет желать лучшего, вследствие чего часто приходится проводить повторную верификацию. Из-за недостаточно тщательной или поздно проведенной верификации может потребоваться повторное проектирование.

Статья посвящена самому современному виду верификации — параметрической и способам устранения неточности интеграции за счет автоматической генерации параметров покрытия без использования симуляции. При формальной параметрической верификации производится анализ покрытия не только сценариев, но и наблюдений — важный момент, который упущен в большинстве других подходов. Метод может использоваться как в рамках существующего процесса верификации, так и отдельно.

ТИПИЧНЫЕ ПРОБЛЕМЫ

Перед тем как использовать IP-блок, разработчик должен достоверно знать его качество. Обычно список вопросов выглядит следующим образом:

- насколько тщательно проведена верификация RTL-моделей;
- при каких условиях и ограничениях проводилась верификация;
- насколько точно эти условия и ограничения соответствуют реальной среде, в которой будет работать IP-блок в дальнейшем;
- все ли требуемые функции реализованы;
- реализованы ли они правильно?

Как правило, вопросы и неопределенности всегда одни и те же, независимо от производителя IP-блока. Они возникают даже и в проверенном

блоке, если он был модифицирован. Еще сложнее обстоит дело, если блок конфигурируемый и предназначен для различных применений.

НЕДОСТАЮЩЕЕ ЗВЕНО: ПОКРЫТИЕ НАБЛЮДЕНИЯ

Как правило, при верификации IP-блока возникают одни и те же вопросы. Чтобы установить, соответствует ли реализованный функционал IP-блока требуемому, используются точки функционального покрытия. Однако этот метод нерационален и очень затратен по времени. Для повышения качества и скорости верификации лучше использовать стандартное структурное покрытие, например, покрытие линий, утверждений, блоков, выражений, ответвлений и автоматов. В то же время стандартное структурное покрытие позволяет оценить только покрытие сценариев, т.е. определить, какие части RTL-кода были выполнены в ответ на воздействие, и достаточно ли предусмотрено воздействий?

Если в ответ на воздействие возникает сбой, его необходимо выявить. За это отвечает покрытие наблюдения, которое показывает, какие части кода были верифицированы, а какие нет. По стандартным параметрам, характеризующим покрытие, невозможно оценить покрытие наблюдения. Покрытие сценариев является необходимым, но не достаточным критерием оценки покрытия наблюдения. Тщательный анализ покрытия наблюдения обязателен, чтобы удостовериться в том, что верификация проводится тщательно и всесторонне. Мутационный анализ позволяет избежать этой проблемы, однако и он не является полноценным решением, поскольку в нем используются предустановленные модели отказа и он опирается на симуляцию.

Следовательно, если даже при верификации достигнуто полное 100-% покрытие сценариев (например, каждая строка кода была выполнена хотя бы один раз), это не дает оснований полагать, что достигнуто желаемое

покрытие наблюдения, если оно вообще обеспечено. Неспособность многих типов параметрической верификации охарактеризовать покрытие наблюдений является серьезным поводом для возникновения неточностей.

По этой же причине динамическая верификация не позволяет четко определить, какая часть кода RTL была проверена, а какая еще должна пройти верификацию. Тем не менее, разработчики часто используют оба этих метода оценки покрытия сценариев, а также временную диаграмму изменения частоты обнаружения ошибок, чтобы принять решение о завершении стадии верификации. Как бы то ни было, отсутствие единой меры оценки покрытия наблюдения приводит к тому, что это решение является больше субъективным, нежели аналитическим.

ФОРМАЛЬНЫЕ ИСТОРИЧЕСКИЕ ОГРАНИЧЕНИЯ

Прежде формальные методы использовались меньше. С одной стороны, всесторонний формальный анализ должен давать исчерпывающие ответы на упомянутые выше вопросы. В частности формальные методы предоставляют все необходимое для оценки покрытия наблюдения и сценариев.

Одна из причин заключается в том, что инструменты формального анализа непросты в использовании и имеют ограниченные возможности, что делает формальный анализ функционального покрытия больших блоков неудобным и даже более сложным, чем тестирование. Соответственно, формальный метод применяется для верификации только тех частей кода, которые сложно достигнуть, например, для крайних случаев и специфических доменов, таких как протоколы связи. Выборочное использование формальной верификации позволяет избежать пошаговой модернизации испытательного стенда. Вместо использования прямых тестов или автоматических, генерируемых случайно, тестов пишется утверждение, которое выра-

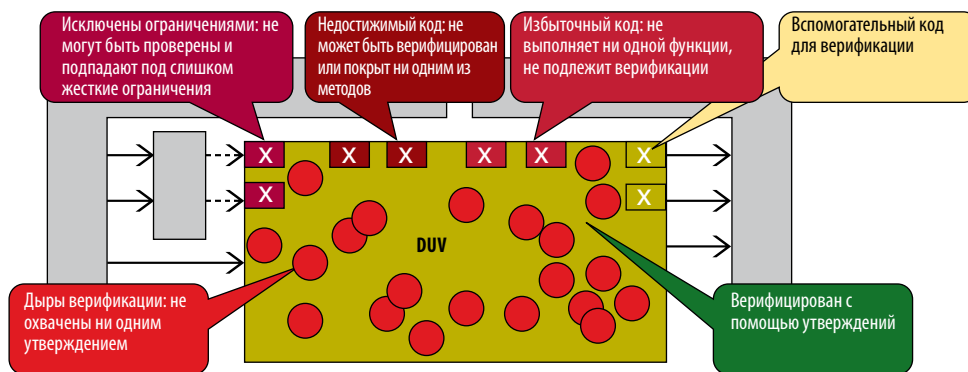


Рис. 1. Верификация IP-блока

жает требуемое поведение системы и описывает поведение, которого не должно быть. Результат такой верификации поведения не зависит от способа анализа (симуляция, утверждение или формальная верификация). Однако использование формального подхода в качестве ситуационной надстройки к симулятору уменьшает неопределенность интеграции IP.

В последние годы емкость формального анализа значительно возросла, а его использование существенно упростилось в частности за счет введения стандартных форматов утверждений, например, SVA — System Verilog Assertion. Чтобы формальных методов для верификации целых IP-блоков было достаточно, чтобы ответить на типичные вопросы, формальный подход должен обеспечивать больше, чем просто расширение симуляции. Он должен представлять собой общую методику, применимую к целому блоку, которая при необходимости может использоваться отдельно либо в сочетании с динамической верификацией. Решение об окончании верификации должно приниматься аналитически по результатам анализа покрытия наблюдений и сценариев.

Для использования всех возможностей метода утверждений и проведения мониторинга качества и степени завершенности верификации целого IP-блока компания Accellera ввела несколько параметров измерения покрытия. Тем не менее, исторически сложилось так, что формальный метод не позволял производить их количественную оценку аналитически, за счет чего увеличивалась неточность интеграции блока. Чтобы заполнить этот пробел, был принят еще один параметр — плотность утверждений. Он характеризует количество утверждений, приходящихся на одну строку RTL-кода [2], однако не дает представления о качестве и эффективности утверждений, не позволяет выя-

вить избыточность. К тому же, по нему нельзя оценить степень завершенности процесса верификации или определить расположение пропущенных участков кода, а эта информация очень нужна. Частично проблему помогает решить измерение покрытия утверждений, однако это довольно трудоемкая задача, не дающая исчерпывающего ответа.

Чтобы избежать неопределенностей при формальной верификации IP, нужен структурный подход, который устанавливает четкое соответствие между набором утверждений, функциями, определенными в техническом задании, и планом верификации. Один из таких структурных подходов включает использование утверждений, которые выражают функционирование на уровне транзакций и операций. Эти операционные утверждения открывают доступ к процессу верификации и позволяют выделить те функции, которые еще не были проверены. Таким образом, формальный метод приближается к тестированию. Однако современная формальная методика имеет еще больший потенциал. Инструменты Accellera позволяют аналитически вычислять покрытие сценариев и наблюдения в автоматическом режиме, обеспечивая 100-% верификацию. Они позволяют выявить дыры в покрытии сценариев и в некотором смысле создают эталонную функциональную модель IP-блока.

ИЗМЕРЕНИЕ ПОКРЫТИЯ

Современная формальная параметрическая верификация автоматически обследует покрытие утверждений и находит неохваченные участки RTL-кода. Эти данные экспортируются в базу данных (Unified Coverage Data Base) и объединяются с результатами тестирования. Формальная параметрическая верификация без использования симуляции производит анализ

таких аспектов как покрытие утверждений, результат утверждений, свойства покрытия, покрытие безусловных команд и покрытие ответвлений.

Сочетание всесторонней формальной верификации и автоматической генерации параметров покрытия обеспечивает большую определенность при интеграции IP, чем многократные прогоны динамических тестов. Более того, она может применяться сразу после того, как закончен первый RTL-блок, т.е. намного раньше, чем прошло хотя бы частичное испытание на стенде.

На рисунке 1 показан IP-блок в процессе верификации. Перед началом верификации весь нефункциональный код должен быть исключен. Автоматически удаляются следующие части:

- функциональный код, подпадающий под ограничения (например, по внешним условиям эксплуатации);
- недостижимый код, который невозможно проверить формальными и иными видами верификации;
- излишний код, который не выполняет никакой функции;
- код верификации, который был написан для упрощения процесса.

Весь остальной функциональный код проходит верификацию. Измеряется объем проверенного и оставшегося кода, выявляются дыры верификации. После добавления недостающих утверждений процесс повторяется до тех пор, пока не будет достигнут 100-% охват кода RTL с точки зрения покрытия сценариев и наблюдения. Очевидно, что такой подход, когда недостающие утверждения вписываются в нужные места, более эффективен, чем оценка плотности утверждений. Кроме того, поскольку код, выполняемый при определенных условиях, исключается автоматически, удается существенно упростить процесс поиска избыточных ограничений. Напомним, что все это производит-

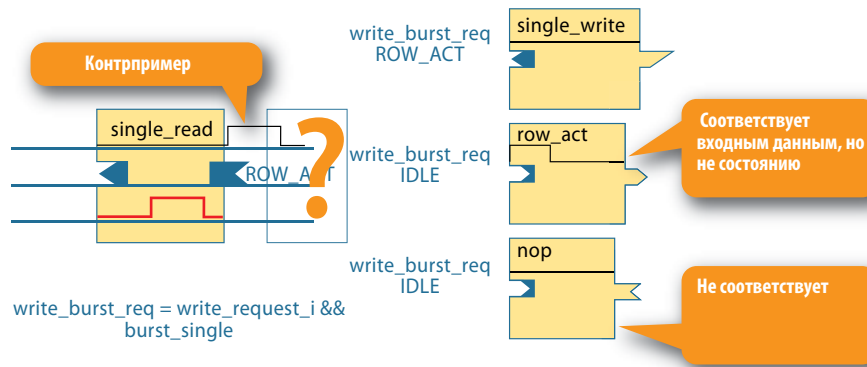


Рис. 2. Функциональная верификация

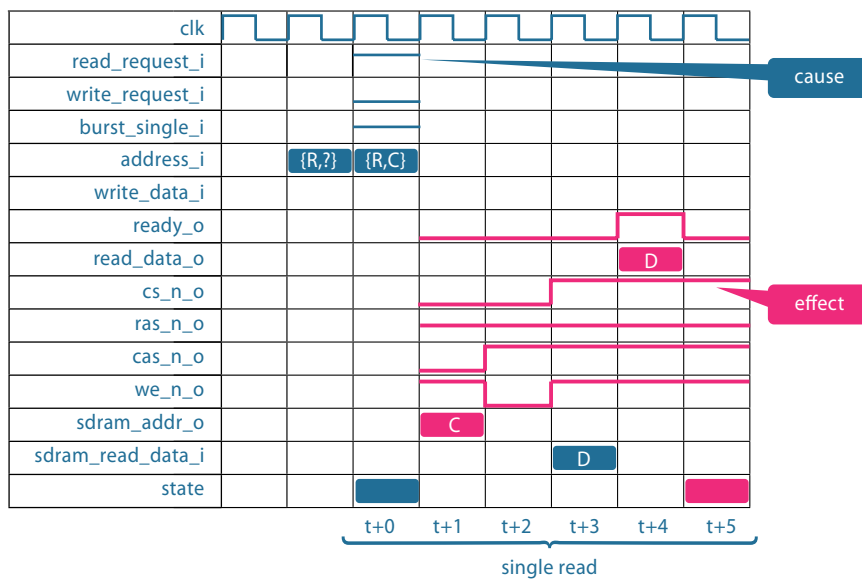


Рис. 3. Временная диаграмма чтения из памяти

ся на первых стадиях верификации, задолго до интеграции.

ПОЛНОТА РЕАЛИЗАЦИИ

Теперь, когда мы научились анализировать тщательность верификации RTL-кода, самое время определить, полностью ли реализованный IP-блок удовлетворяет заданию? Это задача функциональной верификации. Как говорилось ранее, функционал анализируется с помощью набора операционных утверждений, которые описывают желаемое поведение (концепцию проекта) на уровне технического задания. Другими словами, они работают на уровне операций чтения, записи и преобразования данных. Когда набор составлен, проводится формальный анализ всех возможных входных сценариев и ответных вариантов выходного поведения, чтобы выявить несоответствия или разрывы в наборе и закрыть их (см. рис. 2). На этом функциональную верификацию можно считать завершённой.

Исходя из рисунка 3, нужно определить, может ли контроллер DRAM выполнять функции, не покрытые операционными утверждениями. Набор состоит из операций `single_read`, `single_write`, `idle (nop)` и `row_activation`. Производится поиск таких конфигураций данных на входе `single_read`, который не соответствует ни одно утверждение из набора. Затем генерируется контрпример, показывающий, что нет операционного утверждения, охватывающего запрос `burst_write`. Это и есть дыра, которую теперь надо заполнить.

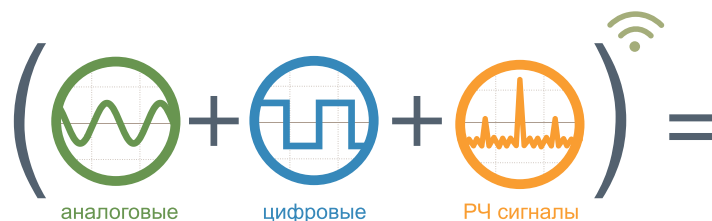
СОЗДАНИЕ ОПЕРАЦИОННЫХ УТВЕРЖДЕНИЙ

К сожалению, операционные утверждения не могут быть синтезированы автоматически. Их можно составить с помощью временных диаграмм из ТЗ, которые устанавливают взаимосвязи «воздействие-отклик», и библиотеки SystemVerilog, в которой содержатся основные определения и функции. Таким образом, между утверждениями

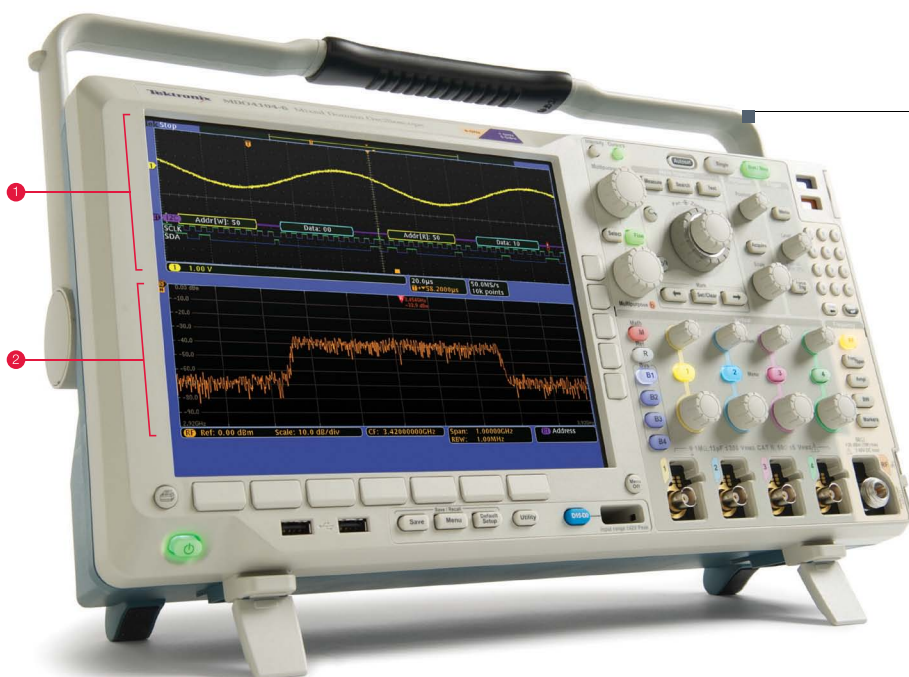
ми SVA и функциями IP есть взаимно-однозначное соответствие. Рассмотрим пример на рисунках 3 и 4. Временная диаграмма описывает выполнение операции чтения контроллером SDRAM. Воздействие и отклик показаны синим и красным соответственно.

Воздействие. Изначально устройство находится в неопределённом состоянии с активным рядом таблицы памяти. Сигнал запроса на чтение `read_request_i` имеет высокий уровень, в то время как `write_request_i` и `burst_single_i` имеют низкий уровень. Верхние разряды адреса `address_i`, отвечающие за ряд, не меняют состояния по сравнению с предыдущей операцией, т.е. операция `single_read` применяется к ряду, который уже активен. Младшие разряды могут меняться, пометим их буквой C. Через три такта после начала `single_read` на вход `sdram_read_data_i` поставим отметку D (см. рис. 3).

Отклик. Некоторые выходные величины определены во всем про-



Первый в мире комбинированный осциллограф



Комбинированный осциллограф MDO4000

- | | | | |
|--|-------------------|---|-------------------|
| 1 | Временная область | 2 | Частотная область |
| <ul style="list-style-type: none"> ■ 4 аналоговых канала <ul style="list-style-type: none"> — с полосой пропускания 500 МГц и 1 ГГц ■ 16 цифровых каналов ■ 1 радиочастотный канал <ul style="list-style-type: none"> — Модели с полосами частот 50 кГц–3 ГГц и 50 кГц–6 ГГц — Ширина полосы захвата до 3 ГГц — Уникальные функции радиочастотного анализа: автоматические маркеры, спектрограмма, графики зависимости параметров РЧ сигнала от времени, расширенная система запуска по РЧ сигналам ■ Система запуска и анализа параллельных протоколов передачи ■ Опциональная система запуска и анализа по последовательным протоколам передачи ■ Построен на платформе осциллографов MSO4000B | | | |

Представляем новую серию приборов Tektronix MDO4000 - революционная серия комбинированных осциллографов со встроенным анализатором спектра.

Комбинированный осциллограф MDO4000 навсегда изменит Ваши представления о тестировании и отладке: захват с полной временной корреляцией аналоговых, цифровых и радиочастотных сигналов, исследование системы одновременно во временной и частотной областях и наблюдение изменений РЧ сигнала в любой момент времени. Комбинированный осциллограф разработан для быстрого и эффективного решения самых сложных задач при разработке современных встраиваемых систем. Три сигнала. Две области. Всего один прибор.

Революция в области тестирования - осциллограф в действии, технические характеристики и многое другое www.tektronix.com/revolutionary.

master
Tool

Приглашаем на выставку «Новая электроника»,
17-19 апреля 2012 г., «Экспоцентр»,
пав. 1, стенд 625, ООО «Мастер-Тул»

Tektronix

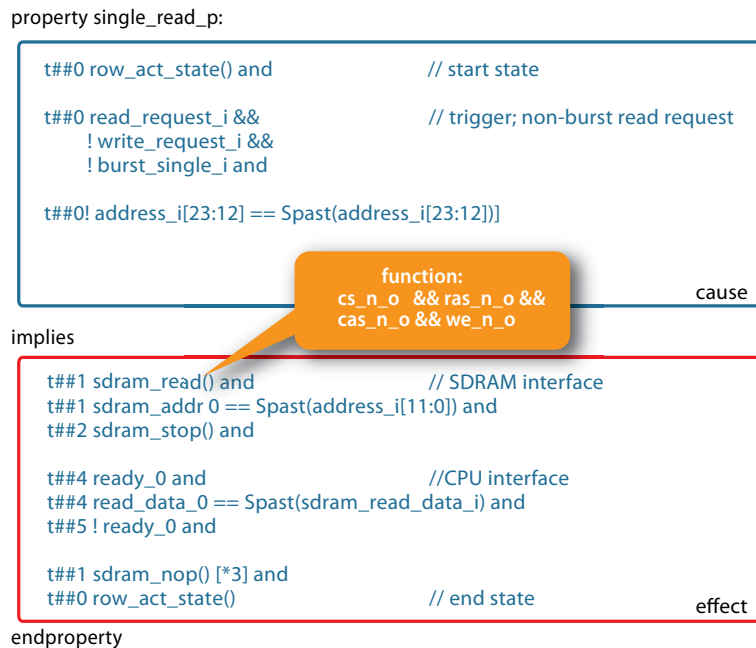


Рис. 4. Составление операционных утверждений

цессе работы t##1, ..., t##5. Например, данные cs_n_o, ras_n_o, которые были получены заранее (например, командой sdram_read()). Другие выходные величины, такие как read_data_o, имеют определенные значения только в некоторые моменты времени. Так, read_data_o он устанавливается, когда ready_o==1'b1. Когда для данного сигнала не определен ни один параметр на временной диаграмме, сигнал принимает произвольное значение. Это абстракция от исходной временной диаграммы, на которой показан один цикл, когда все сигналы имеют установленные значения в каждый заданный момент времени. Заметим, что для операции одиночного считывания сигналы воздействия и отклика накладываются, а это сложно описать в формате SVA, гораздо удобнее воспользоваться операционными утверждениями, например, используя t##0, t##1 и т.д. для обращения к разным временным точкам (см. рис. 4).

На рисунке 4 показано, как формулируется операционное утверждение из временной диаграммы. В первой части описано воздействие (cause), а ниже — отклик (effect). Пронумерованные моменты t могут выбираться по какому-то условию или задаваться сигналами. Временные точки можно отмечать иначе, например, по интервалам.

КОРРЕКТНОСТЬ ТЕХНИЧЕСКОГО ЗАДАНИЯ

Иногда само ТЗ может содержать ошибки или быть неполным. На сегодня

не существует методики поиска этих несоответствий. Некоторые упущения могут быть найдены с помощью утверждений. Иногда подозрение на неопределенность ТЗ возникает после получения неожиданных результатов тестирования.

КОНФИГУРИРУЕМЫЕ IP-БЛОКИ

В начале статьи мы упоминали, что верификация конфигурируемых IP-блоков гораздо сложнее. При верификации симуляцией или тестированием необходимо разрабатывать разные наборы тестов для каждой возможной конфигурации. Поскольку при формальной верификации с применением операционных утверждений анализируется, скорее, концепция, чем реализация, то должна быть возможность верификации нескольких конфигураций одновременно, за один прогон. Формальный подход позволяет проверять простые конфигурируемые блоки, например, шины с переменной глубиной или FIFO с переменной глубиной. Для более сложных блоков требуется доработка методики. Тем не менее, это реальная возможность для продуктивного устранения неопределенности интеграции конфигурируемого IP-блока.

ЗАКЛЮЧЕНИЕ

Формальную параметрическую верификацию можно использовать для обнаружения неохваченных участков кода и для оценки тщательности верификации RTL-модели. Полную и качественную верификацию обеспечивает

анализ покрытия (сценариев и наблюдений). Этот метод можно применять сразу, как только будут готовы первые RTL-блоки, т.е. намного раньше, чем тестирование на стенде. Весь процесс верификации может проводиться без симуляции.

Мы рассмотрели метод утверждений SVA, который позволяет полностью выразить концепцию и верифицировать внешние условия, в которых IP будет работать. Используя формальный анализ неохваченных участков, строится набор операционных утверждений, которые позволяют выявить, насколько разработанный IP-блок соответствует заданию и добиться 100-% функционального покрытия. В процессе составления этого набора могут быть найдены ошибки или упущения, заложенные в самом ТЗ.

Итак, рассмотренная методика универсальна и может применяться для верификации целого блока, причем как отдельно, так и в сочетании с динамической верификацией. Она позволяет закрыть некоторые неучтенные моменты, повысить качество верификации и избавиться от неопределенностей при интеграции IP.

ЛИТЕРАТУРА

1. *Functional coverage metrics — the next frontier by Leonard Drucker*.//EE Times, 2002//www.eetimes.com.
2. *Harry D., Krolnik, Adam C., Lacey, David J. Assertion-Based Design by Foster*//Springer 2004//http://www.springer.com.