

УДК 004.415.5
ББК 32.973
У36

Авторы:

Уилкокс Пол, Cadence Design Systems
Романов Александр Юрьевич, д. т. н., проф. НИУ ВШЭ

Рецензенты:

Тарасов И. Е., д. т. н., проф. РТУ МИРЭА
Саенко В. И., к. т. н., доц. института радиотехники
и инфокоммуникационных технологий ГУАП

Уилкокс П., Романов А. Ю.

У36 Профессиональная верификация: Руководство по продвинутой функциональной верификации / пер. с англ. А. Ю. Романова. – М.: ДМК Пресс, 2025. – 250 с.: ил. – (Книжная полка Истового инженера).

ISBN 978-5-93700-398-0

Книга, которую вы держите в руках, продолжает серию «Книжная полка истового инженера», которая издается при поддержке компании YADRO. Это издание подготовлено к публикации Московским институтом электроники и математики им. А. Н. Тихонова НИУ ВШЭ совместно с издательством «ДМК Пресс».

Данная книга представляет собой учебник по теоретическим основам продвинутой функциональной верификации и содержит лучшие практики, используемые в настоящее время. В ней подробно описана унифицированная методология верификации (UVM) и раскрыты такие темы, как функциональный виртуальный прототип, функциональное покрытие, утверждения, формальная верификация, тестбенчи, косимуляция, эмуляция, аппаратное ускорение верификации и т. д.

Каждая глава книги предваряется краткими тезисами и конспектом, а в конце завершается контрольными вопросами, что делает ее отличным базовым учебником для преподавания верификации в вузах и для самостоятельного обучения.

УДК 004.415.5
ББК 32.973

First published in English under the title Professional Verification. A Guide to Advanced Functional Verification by Paul Wilcox. This edition has been translated and published under licence from Springer Nature Switzerland AG. Springer Nature Switzerland AG takes no responsibility and shall not be made liable for the accuracy of the translation.

Все права защищены. Любая часть этой книги не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами без письменного разрешения владельцев авторских прав.

Материал, изложенный в данной книге, многократно проверен. Но, поскольку вероятность технических ошибок все равно существует, издательство не может гарантировать абсолютную точность и правильность приводимых сведений. В связи с этим издательство не несет ответственности за возможные ошибки, связанные с использованием книги.

ISBN 978-1-4020-7875-0 (англ.)

ISBN 978-5-93700-398-0 (рус.)

© Springer Nature Switzerland AG, 2013
© НИУ ВШЭ, 2025
© Коллектив авторов
© Перевод, оформление, издание,
ДМК Пресс, 2025

Содержание

От издательства	10
Об авторах	11
Рецензии на книгу	13
Благодарности	15
Предисловие от автора русского издания	16
Глава 1. Введение	18
Конспект	18
Введение	19
Перенимаем опыт	20
Для кого эта книга?.....	21
Контрольные вопросы.....	22
Глава 2. Проблемы верификации	23
Конспект	23
Проблемы верификации	24
Пропущенные ошибки – результат невезения?.....	25
Жажда скорости	27
Делать больше с меньшими ресурсами	28
Фрагментарный процесс разработки и верификации	29
Контрольные вопросы.....	31
Глава 3. Продвинутая функциональная верификация	33
Конспект	34
Продвинутая функциональная верификация	35
Верификация как отдельная задача.....	35
Координация верификации с другими задачами разработки.....	37
Верификация как многопоточный процесс	38
Верификация не бывает стопроцентной	40
Методология важнее инструментов	40
Каждому проекту – своя верификация.....	41
Контрольные вопросы.....	42
Глава 4. Успешная верификация	43
Конспект	43
Успешная верификация	44

Тайм-менеджмент	45
Управление ресурсами	47
Процессы верификации	48
Подходы к верификации	51
Контрольные вопросы	52
Глава 5. Профессиональная верификация	54
Конспект	55
Профессиональная верификация	55
Понятие профессиональной верификации	56
Ценность верификации	57
Стоимость продвинутой верификации	58
Верификация: задача второго сорта	59
Изменение текущего восприятия	61
Контрольные вопросы	63
Глава 6. Унифицированная методология верификации	64
Конспект	65
Унифицированная методология верификации	66
Что делает методологию единой?	67
Повышение скорости и эффективности	68
Ключевые понятия	69
Обзор методологии UVM	78
Контрольные вопросы	83
Глава 7. Проектирование на системном уровне унифицированной методологии верификации	84
Конспект	85
Проектирование на системном уровне унифицированной методологии верификации	87
От чистого листа к функциональному виртуальному прототипу	88
Использование функционального виртуального прототипа	91
Разработка функционального виртуального прототипа	94
Верификация функционального виртуального прототипа	98
Контрольные вопросы	100
Глава 8. Цифровые подсистемы управления	101
Конспект	102
Цифровые подсистемы управления	104
Шаг 1: планирование верификации	106
Шаг 2: обеспечение качественных входных данных	109
Шаг 3: выполнение	113
Шаг 4: проверка готовности блоков	120
Контрольные вопросы	121

Глава 9. Алгоритмические цифровые подсистемы	123
Конспект	124
Алгоритмические цифровые подсистемы	124
Шаг 1: верификация алгоритма.....	125
Шаг 2: разработка тестбенчей.....	128
Шаг 3: верификация окончательной реализации.....	129
Шаг 4: интеграция и проверка готовности проекта	131
Контрольные вопросы.....	131
Глава 10. Аналоговые/высокочастотные подсистемы	133
Конспект	134
Аналоговые/высокочастотные подсистемы	135
Методология Cadence ACD.....	136
Сходящееся проектирование	138
Маршрут проектирования согласно методологии ACD	140
Системные требования	142
Оценка осуществимости дизайна	142
Требования к интегральной схеме.....	142
Стратегия моделирования.....	143
Поведенческое моделирование. Моделирование на верхнем уровне.....	146
Проектирование на уровне модулей	146
Калибровка моделей	146
Стратегия физического проектирования	147
Планирование кристалла и предварительная трассировка верхнего уровня.....	147
Обновление трассировки.....	147
Извлечение RC-параметров	147
Анализ уровня кристалла	148
Завершение разработки чипа	148
Контрольные вопросы.....	148
Глава 11. Интеграция и системная верификация	149
Конспект	150
Интеграция и системная верификация	151
Системная интеграция	152
Системная верификация	155
Контрольные вопросы.....	160
Глава 12. Проектирование на системном уровне	162
Конспект	163
Проектирование на системном уровне	164
Проблемы, решаемые с помощью функционального виртуального прототипа.....	164
Верификация и разработка программного обеспечения	166
Абстракция	171

Абстракция верификации	173
Моделирование на уровне транзакций	175
Контрольные вопросы	176
Глава 13. Инструменты формальной верификации	178
Конспект	179
Инструменты формальной верификации	180
Когда надо использовать формальную верификацию?	180
Формальные методы верификации	182
Контрольные вопросы	191
Глава 14. Разработка тестбенчей	193
Конспект	194
Разработка тестбенчей	195
Компромиссы	196
Унифицированные тестбенчи	201
Верификационные тесты	207
Контрольные вопросы	213
Глава 15. Продвинутые тестбенчи	214
Конспект	215
Продвинутые тестбенчи	215
Утверждения	216
Покрытие тестами	223
Реактивные тестбенчи	228
Контрольные вопросы	231
Глава 16. Аппаратная верификация	232
Конспект	233
Аппаратная верификация	234
Ускоренная совместная верификация	235
Интеграция процесса совместной верификации в среду проектирования	241
Контрольные вопросы	242
Глоссарий	244
Предметный указатель	247

Об авторах

Пол Уилкоккс – директор по методологии инженерии в компании Cadence Design Systems, где работает с 2002 года. До этого он работал в Cisco Systems, StratumOne Communications, 0-In Design Automation и Sun Microsystems. Является обладателем патентов на разработки в области проектирования и верификации ASIC. Пол Уилкоккс получил степень бакалавра по электронной инженерии в Northeastern University и степень MBA в San Jose State University.

Романов Александр Юрьевич (<https://www.hse.ru/staff/a.romanov>) – автор русской адаптации данной книги, д. т. н., профессор Московского института электроники и математики им. А. Н. Тихонова Национального исследовательского университета «Высшая школа экономики» (МИЭМ НИУ ВШЭ). С 2014 г. работает в МИЭМ НИУ ВШЭ, где возглавляет лабораторию САПР (<https://miem.hse.ru/edu/ce/cadsystem>), специализирующуюся на проектной деятельности, а также разработке цифровых систем на ПЛИС/микроконтроллерах, робототехнических комплексов, аппаратных реализаций систем искусственного интеллекта, многопроцессорных систем, систем удаленного доступа к лабораторному оборудованию и т. д. Является автором более 200 научных статей.

Помогали делать книгу:

Тимошук Алина, Воронина Елизавета, Нефедов Никита, Чебоненко Варвара, Перов Георгий – студенты МИЭМ НИУ ВШЭ, которые помогали готовить русскую версию данной книги.

Консультанты:

Курт Джонсон – директор группы по маркетингу специализированных интегральных схем в компании Cadence Design Systems, где он возглавлял разработку среды проектирования AMS, создав ее с нуля. Также Курт Джонсон занимал должность главного технического архитектора и специалиста по стратегическому планированию в области специализированных интегральных схем в Cadence IC Methodology Services. До этого он работал в таких компаниях, как Qualcomm, Western Digital, Teledyne Systems и Motorola. Курт Джонсон получил степень бакалавра по электронной инженерии в Purdue University.

Рэй Тернер – старший менеджер по производству программных продуктов, совместимых с ускорителем Incisive Palladium Candance, и систем моделирования интегральных схем, которые входят в платформу верификации Incisive Verification Platform. До того как присоединиться к Cadence, он 7 лет работал в компании P-CAD менеджером по маркетингу программных продуктов по автоматизации проектирования электроники (EDA). Рэй Тернер имеет

18 лет опыта в управлении продажами программных продуктов по автоматизации проектирования электроники и 14 лет в разработке аппаратного и программного обеспечения, а также интегральных схем в телекоммуникационной и аэрокосмической отраслях, в области разработки тестового оборудования (ATE) и в микроэлектронной отрасли. Он получил степень бакалавра по электронной инженерии в Loyola University в Лос-Анджелесе. Также Рэй имеет патенты на ранние разработки в области цифровой обработки сигналов и автор двух книг по инженерии.

Рецензии на книгу

Разработка современных интегральных схем – один из наиболее трудоемких и наукоемких процессов в настоящее время. Его сложность определяется тем, что проектирование схемы происходит по длинному многоступенчатому маршруту, на котором надо учитывать множество различных факторов, принимать компромиссные решения и в который может быть вовлечено множество команд разработки, выполняющих проектирование на разных уровнях абстракции. Вероятность ошибок и их цена очень высоки. Поэтому верификация является одним из важнейших элементов проектирования интегральных схем.

При этом учебных материалов по теме очень мало. Часто разработчики ограничиваются максимум небольшими тестбенчами и утверждениями, а тема верификации системно в вузах не преподается. Рассматриваемый учебник «Профессиональная верификация: Руководство по продвинутой функциональной верификации» – важный шаг в сторону исправления данной ситуации. В нем в систематизированном и подробном виде даны основы верификации – ключевые понятия и подходы. Также читателям будет интересно узнать историю вопроса и почему верификации надо уделять внимания не меньше, чем самой разработке.

Данная книга хороша еще тем, что в начале каждой из глав даны выжимки информации, что позволяет прочитать и акцентировать внимание на ключевые идеи и понятия, раскрываемые в главе. В конце каждой главы есть вопросы, которые помогут читателям провести рефлексию и убедиться, что они поняли весь материал. Это также хорошее подспорье для преподавателей.

В целом можно констатировать появление еще одного качественного издания, которое будет полезно инженерам и студентам, занимающимся разработкой интегральных схем.

Автор книг по разработке цифровых систем на основе ПЛИС Xilinx,
д. т. н., профессор федерального государственного бюджетного
образовательного учреждения высшего образования
«МИРЭА – Российский технологический университет»

Илья Евгеньевич Тарасов

Дорогие читатели!

Перед вами отличная, своевременная книга, затрагивающая важные вопросы верификации систем разного уровня сложности. В книге доступным языком объясняются довольно сложные понятия и технологии, что является несомненной заслугой авторов и переводчиков.

Для кого эта книга? Данное издание ориентировано прежде всего на контингент студентов технических специальностей. В то же время профессионал, занимающийся верификацией систем, также найдет в ней много для себя полезного и важного.

Исходная основа на английском написана П. Уилкоксом, известным специалистом по технологиям верификации сложных систем. Второй автор, А. Романов, известный специалист в области проектирования на ПЛИС, удачно дополнил книгу различным материалом, позволившим сделать ее удобным инструментом в учебном процессе.

Проблема верификации и тестирования при разработке систем тем важнее, чем сложнее разрабатываемая система. Для большинства людей представляется, что главное – разработать саму систему, а верификация и тестирование – это всего лишь вспомогательные операции. Но в наше время этап тестирования систем часто требует не меньше времени и усилий, чем сама разработка. Очень важно для многих систем обнаружить все ошибки до основного этапа тестирования и внедрения. Ошибки сборки иногда могут привести к значительным затратам на их исправление, непредсказуемым ситуациям и даже авариям. Культура проведения работ по верификации при разработке систем на кристалле, понимание полного цикла всех операций и особенностей реализации отдельных этапов – то, что определяет уровень профессионализма команды верификации. Описание того, чем и как должна заниматься команда верификации, вы как раз найдете в предлагаемой книге.

Отмечу еще один важный момент: своевременность появления русскоязычного варианта такой книги. Современное возрождение многих отраслей компьютерной индустрии и микроэлектроники в России требует наличия высококлассных специалистов. Задачи, которые стоят перед страной, имеют высочайшую важность и срочность. В этой связи доступное изложение сложных вопросов по верификации систем является очень актуальным и своевременным. Данная книга является полезной для вузов для обеспечения качественной и эффективной подготовки и повышения квалификации специалистов по разработке цифровых систем и системных программистов.

Книга хорошо структурирована и проста для понимания. Несомненно, и студент, и профессионал найдет в книге мудрые советы, примеры и технические решения для разных проблемных вопросов, связанных с тестированием и верификацией различных цифровых систем и интегральных схем. Также она полезна и инженерам в области разработки электроники для расширения своего кругозора.

Владимир Иванович Саенко, к. т. н.,
доцент института радиотехники и инфокоммуникационных технологий,
Санкт-Петербургский государственный университет
аэрокосмического приборостроения,
доцент кафедры теоретической и прикладной информатики,
Национальный исследовательский университет
«Московский физико-технический институт»

Благодарности

Исаак Ньютон однажды сказал: «Если я видел дальше [чем другие], то это благодаря тому, что стоял на плечах гигантов». Эта книга основана на опыте и упорном труде многих «гигантов» в области проектирования и верификации современных интегральных схем. Перечислить всех, кто сделал свой вклад в создание этой работы, было бы невозможно, но также было бы неправильно не признать их заслуги.

За всю свою карьеру я повстречал множество «гигантов», которые потратили свое время и проявили терпение, чтобы научить меня многому из того, что содержится в этом тексте. За это я хотел бы выразить благодарность друзьям и коллегам, с которыми я работал в Sun Microsystems, 0-In Design Automation, StratumOne Communications и Cisco Systems. Особая благодарность Уиллису Хендли, Дэвиду Каффину, Джеймсу Антонеллису, Кертису Уиддосу и Ричарду Хо.

Эта книга является результатом усилий многих людей в Cadence Design Systems, и я хотел бы поблагодарить следующих сотрудников за их вклад и работу по рецензированию текста: Андреаса Мейера, Гранта Мартина, Леонарда Друкера, Нейзазана Хана, Фу Хуина, Лизу Пайпер и всех членов команды по разработке методологии инженерии.

Я также хочу поблагодарить Линду Фогель за ее неустанное и профессиональное редактирование, а также Кристен Уиллетт, Кристин Лиетцке и Глорию Крейтман из группы маркетинговых коммуникаций Cadence.

Особую благодарность я хочу выразить Полу Эстрада за то, что дал мне возможность и время для подготовки этой книги, а также за веру в меня, когда я сам был готов сдать. Он мой лучший наставник и друг.

В завершение хочу выразить свою глубокую благодарность моим настоящим «гигантам» – моим родителям, Элеоноре и Гэри Уилкоккс, за безмерную любовь и поддержку, а также за то, что они привили мне любовь к знаниям.

Предисловие от автора русского издания

Дорогие друзья!

Книга, которую вы держите в руках, продолжает серию «Книжная полка истового инженера», которая издается при поддержке компании YADRO. Это издание подготовлено к публикации Московским институтом электроники и математики им. А. Н. Тихонова НИУ ВШЭ совместно с издательством «ДМК Пресс».

Если вы интересуетесь цифровой электроникой, разработкой на ПЛИС, проектированием на языках описания аппаратуры Verilog или VHDL, процессорной архитектурой RISC-V, то вы, скорее всего, уже знакомы с книгами по цифровому синтезу, такими как: Д. Харрис и С. Л. Харрис «Цифровая схемотехника и архитектура компьютера», «Цифровая схемотехника и архитектура компьютера: RISC-V», «Цифровая схемотехника и архитектура компьютера. Дополнение по архитектуре ARM»; «Цифровой синтез: RISC-V» (под ред. А. Ю. Романова), Ф. Бруно «Программирование FPGA для начинающих», С. Смит «Программирование на языке ассемблера RISC-V» и др. (воспользуйтесь QR-кодом, чтобы узнать об упомянутых книгах).



Эта книга была задумана давно. Дело в том, что в цепочке курсов по цифровому синтезу, которые читаются под моим руководством в МИЭМ, не хватало курса по верификации. Чтобы построить такой курс, нужен базовый учебник, на основе которого могут быть созданы лекции и практические работы. Но готовых материалов по верификации – как на английском, так и тем более на русском – очень мало. Если классическая электроника и проектирование на ПЛИС еще преподаются в профильных вузах, то верификации как полноценной дисциплине уделяется, как правило, очень мало внимания. Единственный полноценный курс по этой тематике, который мне известен, – это курс от YADRO «Верификация систем на кристалле» (<https://careers.yadro.com/practical-courses/verification/>), созданный совсем недавно.

Поэтому в рамках проектной деятельности МИЭМ (https://miem.hse.ru/project_office/project_activities) я собрал команду студентов и предложил вместе перевести классическую книгу Paul Wilcox «PROFESSIONAL VERIFICATION: A Guide to Advanced Functional Verification». Книга довольно старая и не содержит каких-то открытий. Но зато в ней системно излагаются основы верифи-

кации, история возникновения этого подхода, его особенности и суммируется опыт основателей этого направления: менеджмента компании Cadence.

Перевод книги занял почти полгода. Была разработан целый конвейер, когда главы переводились одним человеком, потом перекрестно рецензировались, проходили через меня, возвращались на доработку и т. д. У команды был создан словарь терминов, чтобы добиться однозначности и избавиться от субъективности при переводе. Все рисунки в книге были обновлены и перерисованы. Также мы добавили к каждой главе короткие тезисы и концепты. Если вам лень читать всю главу, главное можно прочесть там. В конце каждой главы были добавлены контрольные вопросы для самопроверки. Еще были созданы презентации и другие дополнительные материалы, которые вы свободно можете скачать в репозитории книги по ссылке: <https://github.com/RomeoMe5/ProfessionalVerificationBookRus>.

Таким образом, получилась по сути новая книга, адаптированная для русскоязычной аудитории и снабженная многими полезными дополнениями.

Что вы узнаете с помощью этой книги:

- что такое верификация и историю вопроса;
- какие проблемы есть у верификации;
- как добиться успеха при верификации и как делать так, чтобы верификация была «продвинутой» и «профессиональной»;
- как устроена унифицированная методология верификации (UVM) и для чего она нужна;
- как верифицируют дизайны разного вида от цифровых до аналоговых;
- как производится системная верификация;
- что такое формальная верификация;
- как разрабатываются тестбенчи;
- как производится аппаратное ускорение верификации
- и многое другое.

Книга предназначена в первую очередь для тех, кто уже умеет программировать на HDL, знаком с ПЛИС и VLSI и теперь хочет понять, что представляет собой профессиональная верификация.

В добрый путь!

Александр Юрьевич Романов,
д. т. н., профессор ДКИ МИЭМ НИУ ВШЭ,
преподаватель курсов «Проектирование систем на кристалле»,
«Системное проектирование цифровых устройств»
и «Верификация систем на кристалле»
г. Москва, Россия

P. S. Хочу выразить огромную благодарность ребятам, которые помогли мне подготовить эту книгу: **Ворониной Елизавете**, **Нефедову Никите**, **Чебо-ненко Варваре** за огромный труд по подготовке глав, **Перову Георгию** за работу над рисунками к книге и, наконец, **Тимощук Алине** за то, что она смогла объединить и организовать всю команду и как настоящий лидер довести проект до финала.

Глава 1

Введение

ТЕЗИСЫ

1. В случае возникновения большого количества ошибок единственный способ вернуть проект на правильный путь – перестать думать о том, как он должен работать, и начать думать о том, почему он может не работать.
2. Функциональная верификация современных сложных ИС требует профессионального подхода.
3. Растущие размеры и сложность проектов вместе с сокращением циклов разработки сделали устаревшие методы верификации неэффективными.
4. Если инженеры будут изучать опыт продвинутых команд, анализируя их успехи и ошибки, то они получают бесценный опыт для решения собственных задач по верификации.
5. В современной индустрии не существует идеальной команды по верификации – каждая из них встречается с уникальным набором проблем.

Конспект

Первая глава посвящена введению и личному опыту автора этой книги.

Пол Уилкоккс – директор по методологии инженерии в компании Cadence Design Systems, где работает с 2002 года, имеет степень бакалавра по электронной инженерии в Northeastern University и степень MBA в San Jose State University.

После большого количества времени, проведенного за рутинной работой, у него появилась возможность спроектировать ключевую часть важного чипа. Он придерживался всех этапов проекта, и разработка шла гладко, но незадолго до окончания работ начали поступать сообщения об ошибках: симуляции начали завершаться ошибками даже после внесения необходимых корректировок. Сроки сдачи проекта были под угрозой. Пол Уилкоккс осознал, что разработанный тестбенч был не способен выявить проблемные места

дизайна из-за сложности и единственный способ исправить ситуацию – это попытаться понять, в каких ситуациях проект может не работать. Таким образом, автору этой книги пришло полное понимание значимости функциональной верификации.

Проблема тестирования разрабатываемого дизайна распространена в индустрии интегральных схем. Необходимо производить проверку выполнения кода, благодаря чему функциональная верификация стала неотъемлемой частью процесса разработки. В данной книге описан продвинутый подход к процессу верификации.

Многие команды разработки считают функциональную верификацию сложной из-за используемых неэффективных и устаревших методов. Если команды по верификации начнут перенимать опыт друг у друга, они смогут научиться решать возникающие проблемы.

Несмотря на то что основная аудитория этой книги – инженеры по верификации, она может быть также полезна исследователям, студентам инженерных специальностей, руководителям проекта, системным дизайнерам и т. д. В книге освещены основные вопросы верификации и решения проблем, возникающих при верификации. Она поможет научиться эффективно решать конкретные задачи благодаря собранному в ней опыту продвинутых команд верификации.

Введение

Размышления о возможных неудачах

После многих лет выполнения, как мне казалось, рутинной работы в области тестирования, разработки программных инструментов и верификации у меня наконец-то появилась возможность спроектировать ключевую часть важного чипа. Я разработал подробную спецификацию и строго придерживался всех этапов проекта. Моя разработка превосходила целевые показатели производительности, а первоначальный макет выглядел впечатляюще. Но за две недели до окончания работ стали поступать сообщения об ошибках. Тестирование с помощью случайной регрессионной верификации успешно проходило несколько недель, пока некоторые параметры не нарушились. Внезапно мой блок начал терять или неправильно упорядочивать операции, и все симуляции стали проваливаться. Я обнаружил на первый взгляд очень редкую ошибку, но на следующий день симуляции снова начали давать сбой. Я внес необходимые корректировки, но и после этого ошибки продолжали появляться. В результате меня вызвали к менеджеру проекта. Сроки сдачи оказались под угрозой срыва из-за меня.

Той ночью по дороге домой я размышлял о том, что же могло пойти не так. Ведь я следовал всем правилам проектирования, разработав сложную схему с меньшими размерами и большей производительностью, чем требовалось.

Разработанная мной структура данных поддерживала множество продвинутых функций, и мы даже запатентовали ее. Размышляя над этим, я пришел к выводу, что добавленная мной сложность привела к появлению новых побочных эффектов, которые не могли быть выявлены разработанным мною тестбенчем (testbench). Я осознал, что единственный способ вернуть проект на правильный путь – это перестать думать о том, как он должен работать, и начать думать о том, почему он может не работать. Именно в этот момент я начал понимать истинную суть функциональной верификации.

Но я не единственный, кто сталкивался с подобным. Большинству инженеров в индустрии интегральных схем (ИС) приходилось иметь дело с такими проблемами. К счастью, функциональная верификация эволюционирует из второстепенной задачи в неотъемлемую часть процесса разработки. Это развитие происходит не благодаря предвидению и тщательному планированию, а из необходимости. Команды, занимающиеся функциональной верификацией, вынуждены справляться с возрастающей сложностью и увеличивающимися размерами устройств, а также быстро меняющимися стандартами, растущими требованиями к производительности и интеграцией множества функций в единые системы. Функциональная верификация современных сложных ИС нанометрового масштаба требует профессионального подхода.

В этой книге верификация рассматривается с профессиональной стороны. В ней детально описаны лучшие практики, используемые продвинутыми командами отрасли. Цель книги не в том, чтобы представить исследования в новых областях верификации или предложить пошаговое руководство по конкретному инструменту или языку. Вместо этого изложение построено вокруг описания продвинутого процесса верификации с опорой на практики современных компаний, чтобы вы могли применить эти знания в своей работе.

Перенимаем опыт

Каждому, кто хоть в какой-то степени связан с разработкой современных интегральных схем, хорошо известны многочисленные проблемы, возникающие при функциональной верификации. Растущие размеры и сложность проектов наряду с сокращением циклов разработки сделали устаревшие методы верификации неэффективными. Команды, занимающиеся верификацией, вынуждены проводить еще больше тестов, чтобы обеспечить высокий уровень качества с самого начала, сталкиваясь при этом с нехваткой ресурсов и устаревшими инструментами. Для многих команд функциональная верификация представляет собой постоянную и практически недостижимую цель, к которой они стремятся, не имея достаточной подготовки и необходимых сил.

Где же командам по верификации искать решения проблем, с которыми они сталкиваются ежедневно? Значительная часть новых исследований в этой области сосредоточена на математических методах и системных под-

ходах, которые требуют кардинальных изменений как в самой верификации, так и в дизайне и инфраструктуре. Инженерные школы, как правило, не включают функциональную верификацию в свои учебные программы. Поэтому единственным источником, к которому можно обратиться за знаниями, остаются другие команды по верификации. Изучая опыт продвинутых команд, анализируя их успехи и ошибки, инженеры могут получить бесценный опыт для решения собственных задач по верификации.

В современной индустрии не существует универсальной идеальной команды по верификации – каждая из них встречается с уникальным набором проблем. Поэтому очень важно изучать лучшие практики различных команд, чтобы выявить наиболее значимые для себя аспекты. Компания Cadence Design Systems активно сотрудничает с наиболее продвинутыми командами по верификации, изучая их опыт и накапливая знания. Эта книга основана на собранном опыте. Она содержит широкий круг полезной информации и методов, которые помогут вам эффективно решать ваши конкретные задачи.

Для кого эта книга?

Если вы исследователь или студент инженерной специальности, эта книга станет для вас полезным обзором современных продвинутых техник, применяемых успешными командами по верификации. При этом авторы предполагают, что у вас уже есть базовые знания о текущих проблемах и процессах в этой области. В приложении А «Ресурсы» вы найдете список отличных материалов по основам и методам функциональной верификации.

Если вы руководитель проекта в области разработки интегральных схем, книга будет полезна для вас в плане понимания методологий и путей улучшения процессов верификации. Если вы архитектор, системный разработчик, интегратор или разработчик программного обеспечения либо инструментов, скорее всего, верификация не является вашим основным приоритетом. Но понимание процесса верификации в целом и его интеграция в вашу сферу работы могут оказаться весьма полезными. Часто инженеры по проектированию проверяют свои собственные блоки или блоки других разработчиков. Современные разработчики на уровне регистровых передач (RTL) осознают важность функциональной верификации и ее роль в процессе разработки. Эта книга содержит описание продвинутых методов верификации, которые могут быть полезны на протяжении всей вашей карьеры.

Основная аудитория этой книги – инженеры по верификации – профессионалы, которые ежедневно сталкиваются с задачами верификации самых крупных и сложных интегральных схем. Эта книга является ценным источником знаний для понимания полного процесса верификации и того, как продвинутые команды в индустрии достигают своих целей.

Книга сосредоточена на практическом применении современных методов верификации с использованием лучших практик. Главы 1–5 книги посвящены основным вопросам верификации и обсуждению профессии инженера

по верификации. В главах 6–11 описывается, как преодолевать проблемы, возникающие при верификации, и как различные практики объединяются в единую методологию. В главах 12–16 книги детально рассмотрено множество тем, которые были затронуты в предыдущих главах.

Контрольные вопросы

1. В чем состоит суть функциональной верификации?
2. В чем заключаются причины эволюции функциональной верификации?
3. Что помогает командам по верификации находить решения проблем, с которыми они сталкиваются?
4. Почему важно изучать лучшие практики различных команд по верификации?
5. Кому нужны знания о процессе верификации?

Глава 2

Проблемы верификации

ТЕЗИСЫ

1. Задача команд верификации заключается в том, чтобы наиболее эффективно и результативно находить все ошибки в кратчайшие сроки, используя минимальное количество ресурсов.
2. Все усилия, вложенные в этапы разработки архитектуры, проектирования, реализации и верификации, могут оказаться напрасными из-за одной пропущенной функциональной ошибки.
3. Как бы тщательно ни был проработан процесс верификации, невозможно учесть и проверить каждый сценарий.
4. Лучший способ найти скрытые ошибки – использовать новые или более эффективные инструменты и процессы.
5. Верификация больших проектов – это управление рисками.
6. Поиск сложных ошибок – это искусство, удача и упорная работа.
7. Нехватка квалифицированных кадров, вычислительных мощностей и инструментов верификации снижает эффективность процесса верификации.
8. Повышение скорости и эффективности верификации не всегда дается так легко, как устранение ошибок.
9. Причина фрагментации процесса разработки – отсутствие целостного подхода.
10. Лишь сквозная (унифицированная) методология, объединяющая все этапы маршрута проектирования – от разработки до финальной реализации, способна устранить фрагментацию, что значительно повысит скорость и эффективность проектирования.

Конспект

Основными проблемами верификации являются пропущенные ошибки, нехватка времени и ограниченность ресурсов. Из-за одной функциональной ошибки могут оказаться напрасными все этапы выполнения проекта.

Чтобы устранить источники функциональных ошибок, необходимо применять улучшенные инструменты и процессы верификации. Говоря о нехватке времени, нельзя однозначно ответить, когда следует закончить разработку проекта, так как такое решение обычно принимается на основе комбинации опыта, данных и интуиции, но если быстрее провести нужные проверки, то останется больше времени для оценки готовности проекта. Стоит понимать, что выявление ошибок полезно тогда, когда это ускоряет процесс. Кроме того, для обнаружения редких ошибок часто требуется множество циклов симуляции. Под нехваткой ресурсов понимается нехватка квалифицированных специалистов и инструментов верификации, решающих более чем одну задачу. Для того чтобы справиться с нехваткой ресурсов, нужно стремиться к повторному использованию уже полученных результатов, правильно организовывать процессы и создавать универсальные блоки, подходящие для любых проектов. Задача команд по верификации состоит в создании сред, которые могут быстро проверять существующие блоки.

Независимо от того, с чем связаны проблемы с верификацией, их решение требует тщательного анализа и выбора правильной стратегии. Скорость проекта может снизиться из-за фрагментации процесса верификации, причиной которой является отсутствие целостного подхода. С помощью единого подхода к верификации можно повысить скорость и эффективность проектирования.

Проблемы верификации

Пропущенные ошибки, нехватка времени и ограниченность ресурсов

У каждой команды разработчиков время от времени возникают проблемы в области верификации, будь то из-за размера или количества проектов, сложности дизайна или особенностей используемого процесса верификации. Команды разработчиков постоянно ищут способы решения этих проблем, но перед ними часто возникают новые и еще более сложные задачи. В основном текущие проблемы верификации можно разделить на три категории: пропущенные ошибки, нехватка времени и ограниченность ресурсов. Наиболее серьезной проблемой является неспособность выявить все ошибки в процессе верификации. Когда у членов команды достаточно времени и ресурсов, они могут с высокой степенью уверенности подтвердить работоспособность продукта. Но выделить достаточное количество времени для полной верификации – задача не из легких. Чтобы решить эту проблему, команды верификации пытаются увеличивать используемые ресурсы, улучшать вычислительную производительность тестбенчей и внедрять новые процессы, дабы сократить количество времени, необходимого для успешного завершения верификации. Тем не менее цена ресурсов порой настолько высока, что создается впечатление, что для своевременной успешной верификации требуется слишком много средств.

Задача команд верификации заключается в том, чтобы максимально эффективно и результативно находить все ошибки в кратчайшие сроки, используя минимальное количество ресурсов. Рассмотрим каждую из основных трех проблем, описанных выше.

Пропущенные ошибки – результат невезения?

Приоритетной задачей команд верификации всегда является выявление ошибок. Все усилия, вложенные в этапы разработки архитектуры, проектирования, реализации и верификации, могут оказаться напрасными из-за одной пропущенной функциональной ошибки. Чем позже обнаруживается ошибка, тем дороже ее исправление для всех участников процесса. Вера компании в свои процессы проектирования и верификации часто выражается в количестве запланированных исправлений, которые учитываются в графиках и бюджетах. Некоторые команды заранее планируют и включают в бюджет необходимость доработок, принимая во внимание то, что часть ошибок может остаться невыявленной в ходе верификации. Другие же команды настаивают на достижении успеха с первой попытки, считая повторные корректировки лишь запасным планом действий.

Независимо от того, насколько быстрым или продвинутым является процесс разработки, возникновение ошибок в конечном продукте может вызвать задержки или отмену проекта. Недавнее исследование компании Collette International Research, посвященное проектированию IC/ASIC и проблемам функциональной верификации, показало, что наиболее распространенными источниками функциональных ошибок в конечных продуктах являются ошибки в проектировании, неверные или неполные спецификации, а также внесенные в спецификации изменения. Каждая из этих проблем может быть решена с помощью улучшенных инструментов и процессов верификации. Ошибки проектирования могут варьироваться от простых опечаток до неверной реализации сложных функций или свойств. Использование инструментов для повышения качества программного кода и разработки тестбенчей, тщательно проверяющих проект, увеличивает шансы на обнаружение таких ошибок. Неполные или неверные спецификации часто являются результатом недостаточной дисциплины и слабой организации процессов на стадии разработки. Верификация может решить эту проблему путем внедрения исполняемых спецификаций и проведения тщательных процессов рецензирования. Ошибки в коммуникации, как правило, возникают, когда объединяются части проекта, разработанные разными инженерами или командами. Верификация может устранить подобные проблемы путем аккуратной интеграции всех компонентов вместе и всестороннего тестирования итоговой системы.



Рис. 1 ❖ Причины пропущенных ошибок
из исследования компании Collette International Research

Как правило, каждый проект содержит одну-две «ошибки на миллион». Как бы тщательно ни был проработан процесс верификации, невозможно учесть и проверить каждый сценарий работы схемы. В конечном итоге если ошибка обнаруживается, то она может быть настолько редкой, что ее возникновение требует невероятной последовательности событий. Инженеры часто утешают себя мыслью, что предугадать такую ошибку просто невозможно. Но когда в каждом проекте находятся две-три такие «ошибки на миллион», необходимо искать решение этой проблемы.

Сложность нахождения таких скрытых ошибок в крупных проектах заключается в том, что существует больше возможных сценариев, чем времени на их тестирование. Команды верификации пытаются справиться с этими проблемами с помощью продвинутых автоматизированных методов, таких как формальная верификация (formal verification) или тестбенчи с ограниченной рандомизацией (constrained random testbench). Автоматизация процессов увеличивает вероятность того, что нужная последовательность событий для обнаружения ошибки произойдет. Тем не менее, как и в случае любой другой ошибки, лучший способ найти скрытые ошибки – использовать новые или более эффективные инструменты и процессы.

Жажда скорости

Сокращение сроков вывода продукта на рынок заставляет команды ускорять процесс разработки интегральных схем, несмотря на увеличение их сложности и масштабов. Это часто приводит к проблеме преждевременного завершения процесса верификации.

Один из самых частых вопросов в разработке: «Когда мы закончим?» Верификация крупных проектов – это управление рисками. Преждевременное завершение и выпуск проекта могут привести к возникновению серьезных ошибок, которые могут быть обнаружены на более поздних этапах, например в производстве. С другой стороны, если слишком затянуть, можно потерять конкурентное преимущество на рынке. Нет универсального ответа на вопрос, когда процесс верификации можно считать полностью завершенным, и в каждом проекте это может решаться по-разному. Команды разработки применяют различные метрики, такие как уровень покрытия тестами (coverage) или скорость выявления ошибок (bug rate), но они не всегда дают полную картину. В большинстве случаев решение принимается на основе комбинации опыта, данных и интуиции.

Хотя точные сроки завершения верификации проекта трудно предсказать, команды разработки обычно хорошо понимают, когда процесс еще далек от завершения. Если остается множество тестов и проверок, вопрос «готов ли проект?» даже не возникает. Проблема возникает тогда, когда основные проверки завершены, но остаются сомнения по поводу готовности проекта. В таких ситуациях вопрос сводится к скорости проведения верификации. Чем быстрее удастся провести нужные проверки, тем больше времени останется для анализа и финальной оценки готовности проекта перед дедлайном.

Позднее обнаружение простых ошибок вызывает особенно большое разочарование. На поздних стадиях верификации такие ошибки могут привести к необходимости перезапускать множество смежных процессов, включая синтез и системную интеграцию. Чем раньше выявляются ошибки, тем лучше, но это действительно полезно только тогда, когда это ускоряет весь процесс разработки. В некоторых случаях исправление обнаруженных ошибок занимает так много времени, что общее время верификации не уменьшается, а иногда даже увеличивается.

Примерно 98 % ошибок в проекте можно обнаружить за счет стандартных методик, но на оставшиеся 2 % обычно уходит больше всего времени и усилий. Поиск сложных ошибок – это отчасти искусство, отчасти удача и, конечно, упорная работа. Даже опытные инженеры по верификации сталкиваются с серьезными трудностями. Ошибки часто маскируются под другими проблемами, что затрудняет их выявление, и могут требовать множества циклов симуляции для их воспроизведения. Средства проектирования или отладки часто имеют неверные представления о работе дизайнера, поэтому они могут не обратить внимания на истинную причину ошибки. При отладке сложных

дизайнов может показаться, что ошибка перемещается по всему дизайну – она может начинаться в одном блоке, проследившись в других блоках, пока не обнаруживается там, где ее никто и не думал искать. После выявления неправильной операции может потребоваться еще много циклов, чтобы воссоздать ошибку с помощью инструментов отладки.

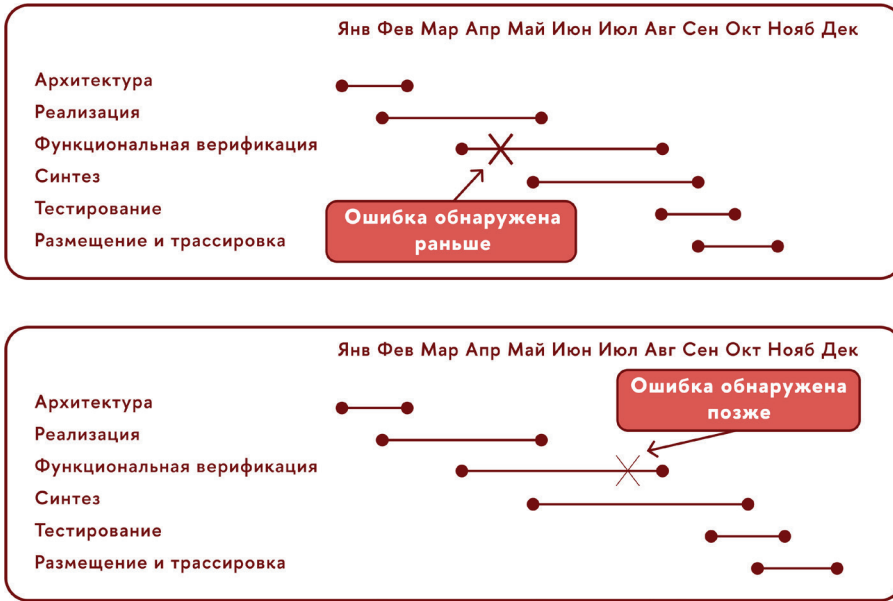


Рис. 2 ❖ Задержки, вызванные поздним обнаружением ошибки

Делать больше с меньшими ресурсами

При достаточном количестве времени и ресурсов опытные команды по верификации могут достичь поставленных целей. Но ресурсы ограничены, и команды верификации вынуждены справляться с задачами в более сжатые сроки и с минимальными затратами.

Процесс верификации требует значительных ресурсов, в том числе и человеческих. Найти опытных инженеров по верификации сложно, так как учебные заведения обычно этому не учат. Поэтому компании либо нанимают редких специалистов, либо обучают новичков. Это приводит к узкой специализации инженеров, которые выполняют отдельные задачи по верификации, не обладая знаниями для комплексной работы. Нехватка квалифицированных кадров снижает общую эффективность процесса верификации.

Также ресурсами являются вычислительные мощности и инструменты верификации. В прошлом ускорение процесса верификации часто достигалось за счет увеличения вычислительных ресурсов. В наше время, несмотря на то что компьютеры стали дешевле, лицензии на специализированные инст-

рументы могут быть очень дорогими. Многие инструменты верификации решают только одну специфическую задачу, и их использование ограничивается определенными этапами проекта, что снижает эффективность их применения.

В программной разработке распространен принцип: «Написал один раз – используй многократно». В верификации часто работает обратный подход: «Пиши часто – используй один раз». Много времени тратится на повторную разработку одних и тех же данных для разных инструментов. Команды по верификации стремятся к повторному использованию информации, что позволяет значительно сократить затраты. Но чтобы модели и компоненты можно было применять многократно, требуется больше времени на их разработку. Если это оправдано, то такое вложение в долгосрочной перспективе окупается. Команды верификации должны правильно организовывать процессы, чтобы обеспечить многократное использование кода без лишних затрат.

В наше время повторное использование блоков дизайна стало стандартной практикой для крупных проектов. Часто проще адаптировать существующий блок, чем создавать его с нуля. Но такие блоки могли быть проверены другой командой, использовавшей другие методы. Нужно ли повторно проверять этот блок? Стоит ли доверять предыдущей верификации? Эти вопросы могут вынудить команду верификации провести дополнительные проверки, чтобы исключить риск ошибок.

Для эффективного повторного использования старых блоков дизайна без необходимости их полной проверки команды разработки должны создавать универсальные блоки, подходящие для любых проектов. Команды по верификации, в свою очередь, должны создавать среды, которые могут быстро проверять существующие блоки в разном системном окружении. Эффективность, достигаемая за счет повторного использования моделей, должна сопровождаться аналогичным ростом эффективности процесса верификации.

Фрагментарный процесс разработки и верификации

Независимо от того, связаны ли проблемы верификации с пропущенными ошибками, низкой скоростью или недостаточной эффективностью тестирования, их решение требует внимательного анализа и правильной стратегии. Ошибки обычно возникают из-за неверного выбора инструментов и методов тестирования либо их неправильного применения. Решить эти проблемы можно, определив их первопричины и применив подходящие техники, которые описаны в главах 12–16 этой книги.

Скорость и эффективность не всегда также легко улучшить, как устранить ошибки. Одной из главных причин снижения скорости и эффективности является фрагментация процесса верификации. В наше время этот процесс

часто разделен на несколько изолированных этапов, которые используют разные данные, методы и подходы, что приводит к дублированию усилий, несовместимости инструментов и трате времени.



Рис. 3 ❖ Типичный процесс верификации

Фрагментация типичного процесса верификации очевидна. Каждая задача решается в отдельной среде со своими уникальными инструментами и моделями. Передача информации между этапами – вертикальное повторное использование (vertical reuse) – практически отсутствует. Та же самая информация воссоздается заново на каждом этапе, но ее использование часто заканчивается после завершения задачи. Это приводит к «информационному застою» (information rot), из-за которого поздние изменения становится труднее внедрить.

Фрагментация проявляется и при переходе от проекта к проекту. Немногие компании могут применить одну и ту же схему верификации ко всем проектам. Даже проекты с общими IP-блоками часто требуют новых процессов верификации. Хотя повторное использование IP-блоков в разных проектах возможно, применить наработки для верификации чаще всего не удается.

Современные команды проектирования обычно представляют собой группы разработчиков, объединенных в цепочки, что тоже приводит к фрагментации. Когда разработчики одних блоков проекта предоставляют их другим, каждый новый этап обычно требует создания собственной среды верификации для одних и тех же блоков.

Причина фрагментации – отсутствие целостного подхода. Вместо того чтобы строить единый процесс, команды разработки решают отдельные задачи

по мере их появления. Добавляются новые инструменты и техники, которые не учитывают всю картину, что создает «острова автоматизации». Развитие языков описания аппаратуры (HDL) и унификация этапов разработки привели к тому, что скорость разработки часто превышает возможности верификации. Под давлением сроков команды верификации решают текущие задачи, не думая о долгосрочных улучшениях, что еще больше увеличивает фрагментацию.

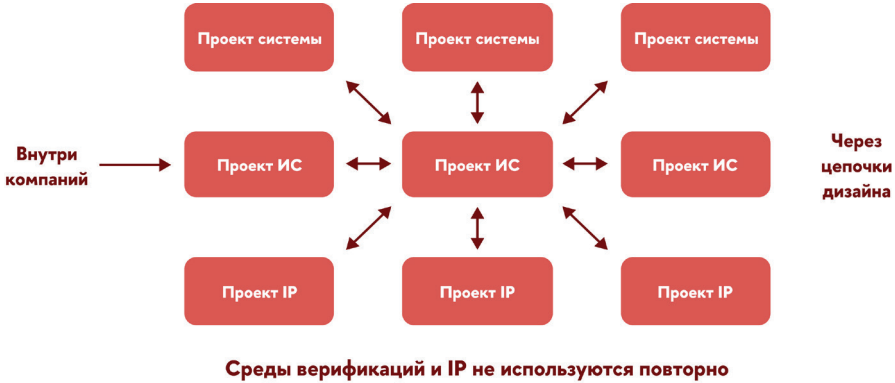


Рис. 4 ❖ Фрагментация в проектной цепочке

Ни один инструмент или метод верификации в одиночку не решит проблему фрагментации. Более того, внедрение новых подходов без учета общей схемы процесса проектирования может только усилить ее. Необходима сквозная методология, которая объединит все этапы маршрута проектирования – от разработки до финальной реализации. Только целостный подход к верификации способен уменьшить фрагментацию, что значительно повысит скорость и эффективность проектирования.

Главы 6–11 книги посвящены унифицированной методологии верификации, основанной на лучших практиках продвинутых команд верификации. Она позволяет совместить различные инструменты и методы вместе для решения ключевых проблем верификации.

Контрольные вопросы

1. Как команды решают проблему нехватки времени для полной верификации?
2. Как верификация может решить проблемы проектирования и коммуникации?
3. Как команды верификации могут справиться с проблемой поиска «ошибок на миллион»?

4. В каком случае действительно полезно раннее выявление ошибок в проекте?
5. Почему некоторые ошибки трудно обнаружить в процессе верификации?
6. Почему команды верификации должны стремиться к повторному использованию старых блоков?
7. Что такое фрагментация верификации?
8. Между кем и чем возникает фрагментация?
9. Каковы причины возникновения фрагментации процесса разработки?
10. Как устранить фрагментацию в проекте?