

Оглавление

Введение	0-1
Глава 1. Основы комбинационной логики. Маршрут разработки цифровых схем	1-1
Глава 2. Виртуальная лаборатория DESim	2-1
Глава 3. Удаленная лаборатория: когда отладочной платы нет под рукой	3-1
Глава 4. Основы последовательностной логики. Управление энергопотреблением цифровой схемы	4-1
Глава 5. Шифраторы и дешифраторы. Скорость работы комбинационных блоков	5-1
Глава 6. Мультиплексор, демультимплексор и селектор. Построение иерархических модулей	6-1
Глава 7. Сумматор, компаратор, устройство сдвига и АЛУ. Повышение скорости арифметических операций	7-1
Глава 8. Последовательностная логика. Счетчики и сдвиговые регистры	8-1
Глава 9. Память: регистровый файл и стек	9-1
Глава 10. Конечные автоматы: основы	10-1
Глава 11. Использование конечных автоматов для связи с периферийными устройствами	11-1
Глава 12. Конвейерная обработка данных	12-1
Глава 13. Софт-процессор RISC-V: основы микроархитектуры	13-1
Глава 14. Систолический массив	14-1
Глава 15. Тестирование больших проектов	15-1
Приложение А. Истовый инженер	А-1
Приложение Б. Инженерный хакатон SoC Design Challenge	Б-1
Приложение В. Школа синтеза цифровых схем	С-1
Приложение Г. Комьюнити FPGA-Systems	Г-1
Предметный указатель	У-1

Александр Романов, Юрий Панчул

Цифровой синтез: RISC-V

Введение

Предисловие А. Ю. Романова

История создания книги, которую вы держите в руках

История создания книги «**Цифровой синтез: RISC-V**» традиционно (для таких проектов) не проста. После успеха переводной русскоязычной версии учебника «**Цифровая схемотехника и архитектура компьютера**» стало понятно, что необходимо его продолжение в виде расширенного практического курса, ориентированного на **Verilog** и обеспечивающего возможность выполнения практических задач на дешевых отладочных платах. Ясно было также и то, что одному человеку (и даже коллективу университетской кафедры) создать такой курс – непосильная задача, притом что издание книги требовало финансирования. У истоков идеи подготовки такой книги стоял **Юрий Панчул**, сумевший объединить для ее воплощения множество преподавателей и инженеров со всего мира. Идея была поддержана **Евгением Аврамовичем Круком**, руководителем Московского института электроники и математики Национального университета «Высшая школа экономики» (**МИЭМ НИУ ВШЭ**). Несмотря на экономический уклон НИУ ВШЭ, **МИЭМ** в прошлом был отдельным техническим вузом, осуществлявшим подготовку студентов в области электроники и математики. Сейчас же, опираясь на необходимые технические и человеческие ресурсы, **МИЭМ** является одним из ведущих центров компетенций, в том числе и по цифровому синтезу. Со стороны **МИЭМ** к работе подключился я (**Романов Александр Юрьевич**), руководитель учебной лаборатории САПР. Далее началась работа по подготовке книги, сбору и редактированию глав от разных авторов, рецензированию и разработке дополнительных материалов.

Так родилась книга «**Цифровой синтез: практический курс**», ставшая настоящим событием в инженерном сообществе и достигшая статуса бестселлера издательства «ДМК Пресс». Выход этой книги дал начало изданию целой серии оригинальных книг и переводов зарубежных бестселлеров под брендом «**Книжная полка Истового инженера**», поддержку которых осуществляет компания **YADRO**.

Прошло четыре года, многое поменялось, и назрела необходимость в создании новой книги:

- широчайшее распространение приобрела архитектура RISC-V – она практически полностью заняла нишу MIPS, которой была посвящена базовая книга, и сильно подвинула ARM;
- вышла книга «**Цифровая схемотехника и архитектура компьютера: RISC-V**», адаптированная под новую архитектуру;
- ряд глав базовой книги устарел, и у авторов накопился внушительный список необходимых исправлений;
- сам практический курс, отработанный на нескольких поколениях студентов, был улучшен и приведен в соответствие современным потребностям – были разработаны новые средства, позволяющие изучать материалы курса, не имея отладочной платы, с помощью виртуальных и удаленных стендов.

Все вышеперечисленное и стало основанием для создания книги «**Цифровой синтез: RISC-V**», которую вы держите в руках.

Чем эта книга отличается от других?

Следует отметить, что данная книга – отдельное произведение, значительно отличающееся от предыдущего издания. Часть глав подверглась небольшим изменениям, другие были заменены, устаревшие главы – удалены. Были добавлены новые главы и приложения, охватывающие следующую проблематику: новые технологии виртуальных и удаленных стендов; простое ядро с архитектурой RISC-V; разработку средств аппаратного искусственного интеллекта на основе систолических массивов; новые возможности для продолжения обучения читателей, освоивших материалы книги.

Особенностью книги является то, что во всех главах каждый пример кода сопровождается листингом и тестбенчем, которые находятся в дополнительных материалах (https://github.com/RomeoMe5/ddlm_riscv). Это позволяет читателю легко использовать уже готовые исходные коды, проводить моделирование и прототипирование на учебной плате с ПЛИС, а также модифицировать работающие примеры программ. Появилась также возможность выполнять отладку кода на виртуальном или удаленном стенде.

Несколько слов об отладочных платах: поскольку участники команды, разрабатывавшей книгу, находились в разных по уровню дохода населении странах, одна из целей курса состояла в том, чтобы сделать его максимально доступным для людей даже самого скромного достатка. Изначально книга ориентирована на плату **DE10-Lite** от компании **Terasic** на основе **ПЛИС MAX 10K** производства **Altera** (после покупки компанией Intel некоторое время называлась **Intel FPGA**, пока ей снова не было возвращено исходное имя). Выбор платы обусловлен ее относительно небольшой стоимостью (около \$82 по академической цене) и доступностью, а также тем, что она обладает достаточной периферией, функционалом, емкостью ресурсов и даже может быть интегрирована с платформой **Arduino**. Платы на основе **ПЛИС Altera**, популярные в России и ближнем зарубежье, имеются во многих академических организациях. При этом исходные коды примеров могут быть легко перенесены и на другие платы.

Поскольку данный проект ориентирован на широкое сообщество, была реализована возможность выполнять все примеры на виртуальном стенде **DESIm**, обеспечивающем значительную часть функционала платы **Terasic DE1-SoC**, но при этом не требующем наличия самой платы. Также в отдельной главе рассказывается, как можно подключиться к **удаленной лаборатории** на базе **МИЭМ НИУ ВШЭ**. Это дает возможность работать с различными реальными платами в дистанционном режиме.

Для выполнения работ не требуется никакого платного программного обеспечения, так как **Quartus Prime Lite Edition + Modelsim / GTKWave / DESIm** распространяются свободно. Таким образом, чтобы приступить к изучению цифрового синтеза, достаточно иметь только эту книгу; при желании увидеть результаты не только моделирования, но и прототипирования понадобится отладочная плата на основе **ПЛИС** или интернет-соединение для подключения к **удаленной лаборатории МИЭМ НИУ ВШЭ**.

Содержание книги

Еще одна особенность книги – это то, что авторы представили ее главы в виде отдельных независимых разделов: то есть можно изучать тот раздел, который необходим сейчас, а обращаться к другим главам – по необходимости. Главы книги создавались, как правило, одним или двумя участниками большого авторского коллектива, поэтому каждая глава имеет свой неповторимый оригинальный стиль (подвергшийся, впрочем, единому научному редактированию); в случае если какая-то из глав по каким-либо причинам будет восприниматься читателями тяжело, есть надежда на то, что другие будут читаться ими намного легче.

Поскольку книга задумана как практикум, ее подразделы сопровождаются заданиями для самостоятельной проработки. В конце каждой главы приводятся вопросы и упражнения, позволяющие преподавателям встраивать данный материал в любой учебный курс, а читателям книги – закрепить новые знания, самостоятельно выполнив предлагаемые задания. Если вы университетский преподаватель, вы можете связаться со мной (Романовым Александром Юрьевичем, a.romanov@hse.ru) и получить дополнительные материалы к книге, в том числе презентации.

Рассмотрим кратко по главам содержание книги.

Глава 1. Основы комбинационной логики. Маршрут разработки цифровых схем

Глава знакомит читателя с типичным циклом разработки цифровой системы на примере проектирования простой комбинационной схемы, которая содержит всего несколько логических вентилях. Сначала демонстрируется, как описать цифровую схему с помощью графического редактора. Далее та же схема проектируется с использованием языка описания аппаратуры. После этого демонстрируются этапы моделирования и прототипирования.

Глава 2. Виртуальная лаборатория DESim

В этой главе рассказывается о том, как можно использовать инструменты виртуализации плат ПЛИС на примере приложения **DESim**. Его можно использовать для симуляции схем, разработанных на языках **VHDL** и **Verilog** (включая **SystemVerilog**). Приложение **DESim** обеспечивает графический интерфейс пользователя (GUI), имитирующий работу с периферией отладочной платы **DE1-SoC**.

Глава 3. Удаленная лаборатория: когда отладочной платы нет под рукой

В этой главе описана организация и строение удаленной лаборатории, реализованной в **МИЭМ НИУ ВШЭ**. Читатели книги по запросу могут воспользоваться удаленными стендами лаборатории и получить удаленный доступ к различным отладочным платам для отработки глав этой книги или реализации собственных проектов.

Глава 4. Основы последовательностной логики.

Управление энергопотреблением цифровой схемы

Данная глава посвящена разработке последовательностных устройств. Рассматриваются защелки и триггеры на примерах их реализаций на языке **Verilog**. Гла-

ва позволяет понять, чем комбинационная логика отличается от последовательностной и как можно управлять энергопотреблением цифровых устройств на этапе их проектирования.

Глава 5. Шифраторы и дешифраторы. Скорость работы комбинационных блоков

В этой главе приводятся примеры реализаций таких важных комбинационных блоков, как шифраторы и дешифраторы, а также дается методика оценки временных характеристик цифровых блоков и их оптимизации.

Глава 6. Мультиплексор, демультиплексор и селектор. Построение иерархических модулей

В главе на примере разработки различных вариантов реализаций таких комбинационных блоков, как мультиплексор, демультиплексор и селектор, демонстрируется иерархический подход к проектированию цифровых устройств. Также вводятся понятие параметризации модулей и конструкция **generate**. В конце главы демонстрируются некоторые приемы использования мультиплексоров на практических примерах.

Глава 7. Сумматор, компаратор, устройство сдвига и АЛУ. Повышение скорости арифметических операций

В главе рассматриваются примеры всевозможных реализаций комбинационной арифметики (сумматоров, компараторов, устройств сдвига и АЛУ на их основе). Отдельный раздел посвящен повышению скорости арифметических блоков на этапе их проектирования.

Глава 8. Последовательностная логика. Счетчики и сдвиговые регистры

В данной главе изложение возвращается к последовательностной логике и особенностям ее разработки на **Verilog** (блокирующие/неблокирующие присваивания, понятие защелок и т. д.) на примере разработки счетчиков и сдвиговых регистров. В конце главы даны примеры организации взаимодействия цифровых систем с простыми периферийными модулями.

Глава 9. Память: регистровый файл и стек

Эта глава посвящена различным вариантам реализации памяти (регистровая память, однопортовая/многопортовая память, стек, очередь и т. д.). В конце главы приводится небольшой пример проектирования на **HDL** памяти с привязкой к библиотекам фабрик-производителей **ASIC**.

Глава 10. Конечные автоматы: основы

В главе приводятся основные понятия и приемы для проектирования конечных автоматов. Иллюстрируются особенности проектирования конечных автоматов Мили и Мура и рассматриваются наиболее оптимальные случаи их использования. Также демонстрируется использование специальных инструментов для проектирования и анализа конечных автоматов.

Глава 11. Использование конечных автоматов для связи с периферийными устройствами

Эта глава расширяет тему проектирования конечных автоматов путем формализации академического подхода к проектированию автоматов; также демонстри-

ругуются и другие автоматы – например, на основе счетчика. Глава обосновывает применение конечных автоматов в проектировании цифровых устройств как ключевого блока управления ими.

Глава 12. Конвейерная обработка данных

Глава посвящена описанию конвейерного подхода к обработке данных и особенностям разработки на **Verilog**. Приводится сравнение комбинационного, многотактного и конвейерного подходов на примере разработки арифметического блока; описываются дополнительные приемы повышения эффективности конвейерных схем.

Глава 13. Софт-процессор RISC-V: основы микроархитектуры

Глава посвящена пошаговому проектированию очень простого софт-микропроцессора с последующим запуском на нем нескольких несложных программ, разработанных на ассемблере. Для этого используется **schoolRISCV** – ядро, представляющее собой упрощенный вариант процессора Сары Харрис.

Глава 14. Систолический массив

Глава призвана сформировать понимание современного состояния разработки аппаратного обеспечения вычислений, в частности для задач **машинного обучения**. Систолический массив интересен как одна из наиболее применяемых и успешных концепций в этой области. В главе рассматривается базовая реализация систолического массива, а также пример его применения для задач **машинного обучения** (линейная классификация).

Глава 15. Тестирование больших проектов

Глава посвящена возможным способам тестирования систолического массива, требующего большого количества ресурсов ПЛИС, чем есть в наличии на отладочной плате. Описан способ тестирования масштабного проекта с использованием **DESIm**. Рассмотрены компиляция проекта под плату **DE1-SoC**, требования по различным ресурсам, а также тестирование на плате. В конце главы показана возможность использования **удаленной лаборатории МИЭМ НИУ ВШЭ** для тестирования больших проектов.

Приложение А. Истовый инженер

Данное приложение является кратким экскурсом в контент, который доступен на портале **Истовый инженер** (engineer.yadro.com).

Приложение Б. Инженерный хакатон SoC Design Challenge

Приложение посвящено хакатону **SoC Design Challenge**, который ежегодно проводится компанией **YADRO** в партнерстве с НИУ МИЭТ и другими вузами. Здесь рассказывается об особенностях мероприятия и основных треках. Приводятся примеры заданий и их решений.

Приложение В. Школа синтеза цифровых схем

Приложение посвящено Школе синтеза цифровых схем – месту, где можно закрепить и расширить знания, полученные в результате изучения этой и других книг из серии «Истовый инженер»; школа проводится на ежегодной основе и обеспечивает более глубокое изучение проектирования процессорных ядер и их верификации.

Приложение Г. Комьюнити FPGA-Systems

Приложение посвящено людям – русскоязычному комьюнити, – организовавшимся вокруг разработки на ПЛИС, проектирования ASIC, создания новых процессорных ядер и других тематик цифрового синтеза. В нем рассказывается о том, где найти единомышленников, помощь и даются ответы на вопросы по теме, а также приведены ссылки на множество полезных ресурсов, которые доступны для всех, кто хочет продолжить свое развитие в области цифрового синтеза.

Таким образом, данный практикум по **Verilog** и **ПЛИС** дополняет и объединяет теоретические курсы по цифровой логике, языкам описания аппаратуры, компьютерной архитектуре и микроархитектуре. Практикум также подготавливает студентов к работе с промышленными процессорными ядрами, к созданию специализированных вычислителей (например, ускорителей нейросетей) и **VLSI** по проектированию массовых микросхем **ASIC**. Он будет полезен разработчикам аппаратно-программных систем, а также прикладным математикам, интересующимся алгоритмами **САПР**.

Выражаю уверенность в том, что данный практикум (так же как и учебник «**Цифровая схемотехника и архитектура компьютера: RISC-V**» авторов Дэвида и Сары Харрис) станет надежной основой для курсов по цифровой электронике во многих университетах России и стран СНГ и поспособствует появлению нового поколения инженеров, которые смогут занять лидирующие позиции в среде разработчиков промышленной электроники.

Александр Юрьевич Романов,
к. т. н., доцент МИЭМ НИУ ВШЭ,
преподаватель курсов
«Проектирование систем на кристалле»,
«Системное проектирование цифровых устройств»,
«Системы искусственного интеллекта» и др.
г. Москва, Россия

Предисловие Ю. Панчула

Verilog – не просто один из редких языков, а обязательный инструмент современного разработчика электроники

До сих пор встречаются люди, которые считают, что Verilog (или его современная версия SystemVerilog) – просто один из редких языков программирования, а ПЛИС – устройство для очень специальных применений вроде обработки сигнала с радиотелескопа. В действительности же Verilog и ПЛИС – вход во всю современную цифровую электронику. Это так, поскольку подавляющее большинство цифровых микросхем, разработанных за последние 30 лет, используют технологию компиляции (синтеза) схем из языков описания аппаратуры, главный из которых – Verilog. Огромное количество инженеров, которые сейчас разрабатывают микросхемы в Apple, Intel и других электронных компаниях, во время учебы в таких университетах, как Беркли и MIT, прошли через лабораторные работы с применением учебных отладочных плат на ПЛИС. Такого рода практические занятия позволяют наработать опыт в технологии проектирования на уровне регистровых передач (Register Transfer Level, RTL), которая используется для создания массовых микросхем внутри популярных цифровых устройств вроде Apple iPhone.

Учебник, который вы держите в руках, – важный шаг на пути построения экосистемы разработки современной электроники в России и в других странах бывшего СССР. России предстоит пройти тот же путь, который прошли Япония, Южная Корея, Тайвань и который сейчас проходит Китайская Народная Республика. На этом пути необходимо создать большое количество групп разработчиков разной специализации, готовых слаженно работать вместе. Таких разработчиков нужно выращивать из сегодняшних студентов:

- некоторые из студентов после окончания университета будут специализироваться в разработке микроархитектуры процессоров, сетевых устройств и других логически сложных блоков. Они будут или сами использовать Verilog, или создавать модели устройств, основываясь на понимании того, как работает технология RTL;
- другие студенты будут специализироваться на физическом уровне проектирования. Им придется решать проблемы физического уровня, возникающие при превращении логического графа схемы в план расположения дорожек и транзисторов на пластине кремния при ее фабричном производстве. Хотя эти инженеры не будут вести разработку на Verilog сами, им нужно понимать основы того, как работает в пространстве и времени граф, который они раскладывают;
- третья группа студентов будет специализироваться на создании программ автоматизированного проектирования (САПР), которые помогают разработчикам аппаратуры. Компании, разрабатывающие такие программы, образуют целую небольшую индустрию автоматизации проектирования (Electronic Design Automation, EDA). В этой индустрии востребованы математически мыслящие инженеры, умеющие решать алгоритмически сложные задачи, которые возникают в программах синтеза, размещения и трассировки схем, автоматического доказательства их свойств и проверки их эквивалентности высоко-

уровневым моделям. Этим инженерам также необходимо понимать основы цифрового синтеза и методологию разработки кода на языках проектирования аппаратуры (HDL);

- знать основы логического проектирования электроники необходимо и создателям аппаратно-программных систем. В компьютерах и встраиваемых системах XX века аппаратура и программы были довольно сильно разделены. В XXI веке, когда повышение скорости процессоров за счет простого уменьшения размера транзисторов зашло в тупик, началось быстрое развитие специализированных вычислителей. Сначала появились графические процессоры для вычислений трехмерной графики (Graphics Processing Unit – GPU), затем их архитектуру обобщили для специализированных высокопроизводительных вычислений (General-Purpose GPU – GPGPU). Сейчас бурно развиваются ускорители нейронных сетей, чипы для машинного зрения и специализированные чипы для датацентров. Создателям всех этих устройств необходимо понимать и программную, и аппаратную стороны вычислений.

Основания для создания этой книги

Данная книга – «Цифровой синтез: RISC-V» – создана совместными усилиями преподавателей и инженеров из нескольких университетов и компаний не только из России, но и всего мира. Этот практикум – один из нескольких образовательных проектов, нацеленных на подъем электроники в странах постсоветского пространства, которые все вместе можно рассматривать как осознанную стратегию. К таким проектам относятся, например:

- симулятор MIPT-MIPS¹, созданный базовой кафедрой Intel в МФТИ;
- совместный курс компьютерной архитектуры и ПЛИС, созданный ВМК МГУ в партнерстве с европейскими университетами;
- курс по интернету вещей, созданный в российском отделении Samsung в партнерстве с российскими университетами;
- учебное софт-процессорное RISC-V ядро schoolRISC-V², разработанное Станиславом Жельнио во время работы в IVA Technologies.

Предтечами создания практикума стали три проекта:

- перевод вводного учебника Дэвида Харриса и Сары Харрис «Цифровая схемотехника и архитектура компьютера». Этот перевод сделала в 2015 году группа из сорока с лишним преподавателей российских и зарубежных университетов, русских сотрудников компаний в Silicon Valley (включая MIPS, AMD, Synopsys, Apple и NVidia) и российских компаний (включая НИИСИ, МЦСТ, Модуль). Начинание поддержали британская компания Imagination Technologies, образовательное отделение РОСНАНО, а также российское издательство «ДМК Пресс». Эта книга закрыла брешь в теоретической части преподавания языков описания аппаратуры и микроархитектуры, связала их с основами цифровой логики и программированием. В 2022 году вышло продолжение этой книги под назва-

¹ <https://mipt-ilab.github.io/mipt-mips/>.

² <https://github.com/zhelnio/schoolRISCV>.

нием «Цифровая схемотехника и архитектура компьютера: RISC-V», адаптированное под новую процессорную архитектуру RISC-V. Данный практикум можно рассматривать как расширенное продолжение и дополнение к базовому теоретическому курсу, ориентированное на практическое применение;

- семинары MIPSfpga, организованные в 2015–2017 годах Imagination Technologies в партнерстве с российскими и зарубежными университетами (МГУ, НИУ ВШЭ (МИЭМ), МИФИ, МФТИ, МИЭТ, ИТМО, Самарский университет, ТГУ, КПИ и КНУ, AlmaU). MIPSfpga – это базовая конфигурация разработанного на Verilog промышленного процессора MIPS interAptiv UP, различные варианты которого используют такие компании, как Microchip Technology, Broadcom и Байкал Электроникс.

Проекты, где студенты соединяют с MIPSfpga свои собственные блоки и синтезируют для ПЛИС простые системы на кристалле, позволяют им работать с тем же кодом, с которым работают инженеры в промышленности. Главный недостаток MIPSfpga состоит в том, что он слишком сложен для начальной демонстрации основных принципов микроархитектуры. Данный практикум содержит целую главу, посвященную проекту schoolRISC-V, который значительно проще, чем MIPSfpga. При этом schoolRISC-V позволяет студенту понять базовое устройство процессоров, работу процессорного конвейера и прерываний;

- цикл популярных семинаров Nanometer ASIC, организованный РОСНАНО, МИСиС, КПИ и Imagination Technologies, состоялся в 2016 году. Автор этих материалов – Чарльз Данчек, преподаватель Университета Калифорнии Санта-Круз и бывший инженер Intel. К ASIC относятся практически все микросхемы, которые применяются в массовых электронных устройствах. Приложение, где описывается процесс разработки заказных микросхем (ASIC, Application Specific Integrated Circuits), находится в предыдущей книге «Цифровой синтез: практический курс», а из этой был исключен по причине слишком большого объема.

Юрий Панчул,
разработчик блоков процессорных ядер в MIPS Technologies,
чипа для магистрального маршрутизатора в Juniper Networks
и мобильного GPU в Samsung Advanced Computing Lab.

Саннивейл, Калифорния

Благодарности

Прежде всего мы благодарим читателей всех изданий книги, приславших свои отзывы и замечания, мы постарались их учесть.

Особая благодарность Евгению Аврамовичу Круку и Веронике Борисовне Прохоровой. Без их поддержки и руководства книга не смогла бы выйти.

Мы благодарим Виктора Владимировича Романова и Василия Владимировича Бурова за внимательную вычитку книги и полезные замечания.

Мы также благодарны Юрию Александровичу Романову за помощь в переводе и вычитке глав, которые были изначально написаны на английском языке.

Рукопись книги, которую вы сейчас читаете, подготовил и прочитал большой коллектив авторов, много раз прочитал научный редактор, один корректор и еще несколько рецензентов. Книга апробирована на четырех потоках студентов, но, безусловно, она не идеальна. Мы будем очень признательны тем внимательным читателям, которые обнаружат в данном издании какие-либо ошибки или опечатки и сообщат о них главному редактору на e-mail: a.romanov@hse.ru или dmkpress@gmail.com (книги постоянно перепечатываются, и в каждом новом тираже все найденные ошибки и недочеты исправляются).

Мы благодарим тех, кто прислал замечания и найденные ошибки:

Андрей Сидоренко

Никита Игоревич Гонтарев

Анатолий Лернер

Полина Акимова

Глеб Баканов

Полина Подкопаева

Дарья Галкова

Александр Чеботарев

Отдельная благодарность Семену Разенкову, студенту ИПМКН ТГУ, который прочитал обе книги: «Цифровой синтез: практический курс» и «Цифровой синтез: RISC-V» – и прислал больше 100 различных замечаний и исправлений ошибок.

Рецензия П. Н. Бибилу

С увеличением степени интеграции и появлением больших и сверхбольших интегральных схем (БИС и СБИС) изменилось проектирование интегральных схем, которые позже были названы схемами малой степени интеграции. Появление программируемых логических интегральных схем (ПЛИС) сопровождалось унификацией языков описания проектов цифровых систем, сначала появился язык VHDL, затем Verilog, были созданы системы моделирования и синтеза структур ПЛИС. ПЛИС используются в самых различных областях, применяются также в качестве прототипов при создании заказных СБИС и систем на кристалле. Важными проблемами в проектировании цифровых систем являются в настоящее время синтез схем как можно меньшей площади, функциональная верификация, решаемая чаще всего на основе моделирования, и сокращение энергопотребления.

Книга «Цифровой синтез: RISC-V» посвящена этим основным проблемам проектирования с целью подготовки на современном уровне специалистов – проектировщиков цифровых систем. Сначала изучаются Verilog-модели комбинационных логических элементов и элементов памяти, типовых устройств и их моделирование, затем – синтез. Уделяется внимание методологическим аспектам и пониманию превращения функциональных и алгоритмических Verilog-моделей в логические схемы и структуры ПЛИС. На практических примерах показывается решение проблем увеличения быстродействия схем, в том числе и на основе конвейеризации, приводятся конкретные примеры применения ПЛИС в качестве вычислительных и управляющих блоков цифровой аппаратуры. Кроме маршрутов проектирования, в различных программных пакетах дается информация об установке таких пакетов программ. Однако материалы книги не ограничиваются лишь моделями цифровых схем и изучением систем для их моделирования и синтеза. Возможность почувствовать применение реальных микросхем и проверить реакции ПЛИС на входные воздействия обеспечивается реализацией студенческих проектов на отладочных платах либо применением средств интернета для удаленной работы с такими платами и стендами.

Изложение материалов хорошо структурировано, имеются контрольные вопросы, задания для самостоятельной работы, по всем изучаемым темам реализован принцип «от простого к сложному». Последние главы книги предлагают задания по достаточно сложным проектам создания софт-процессора RISC-V и решению задач, возникающих при аппаратной реализации алгоритмов в области искусственного интеллекта. Книга может составлять основу многих курсов по подготовке студентов технических вузов и университетов, может быть полезной для тех специалистов, которые не изучали проектирование цифровых систем на ПЛИС, но переходят к их использованию в своей практической работе.

Бибилу П. Н.,
автор более 20 книг по логическому синтезу и программированию на VHDL,
д. т. н., профессор, заведующий лабораторией логического проектирования
Объединенного института проблем информатики
Национальной академии наук Беларуси

Рецензия А. Д. Иванникова

RISC-V – самая популярная архитектура последние несколько лет, доступная всем желающим для изучения и построения собственного процессора. С одной стороны, она является открытой, а с другой – к ее развитию прикладывают усилия значительное количество компаний, состоящих в RISC-V альянсе. Открытость архитектуры дает возможность энтузиастам запускать и дорабатывать ее для собственных разработок.

Но мало понимать архитектуру на уровне блок-схемы и знать, как она работает, нужно еще уметь ее реализовать. Один из способов – это использование ПЛИС. Чтобы разрабатывать проекты на ПЛИС, нужно: знать HDL-язык, владеть средой проектирования для ПЛИС, уметь выполнять моделирование HDL-кода, запускать прототип «в железе» и т. д. Вот почему важно не только дать читателю теоретические аспекты, которые очень хорошо описаны в книге Д. Харрис и С. Харрис «Цифровая схемотехника и архитектура компьютера», но и проверить теорию на практике с помощью отладочной платы. Эта книга – именно то, что нужно: она воплощает в себе баланс между теоретической частью и ее практической реализацией.

В книге последовательно рассмотрены основы проектирования цифровых схем, начиная с простых логических элементов, плавно переходя к более сложным структурам, таким как конечные автоматы, и в конечном итоге подводя читателя к описанию и построению простого процессорного ядра с архитектурой RISC-V. Каждый этап обучения сопровождается практикой работы со средствами проектирования и моделирования цифровых схем. Как итог читатель получит достаточно начальных знаний и практики для дальнейшего полноценного изучения мира RTL-проектирования.

Ключевое преимущество книги «Цифровой синтез: RISC-V» состоит в том, что она предоставляет читателю свободу инструментальных средств. Примеры из книги можно свободно скачать на Github и потом легко выполнить как на своей недорогой отладочной плате, так и с помощью средств виртуализации или удаленной лаборатории. То есть читателю не требуется покупать дорогую и специфическую плату – любой пример из книги можно реализовать в аппаратуре удаленно или на собственном компьютере.

Иванников А. Д.,
д. т. н., профессор, Главный научный сотрудник
Института проблем проектирования в микроэлектронике
Российской академии наук

Рецензия И. Е. Тарасова

Цифровая электроника – одно из наиболее интересных и перспективных направлений деятельности технических специалистов. Современная ситуация в промышленности позволяет изучать ее не только теоретически, рассчитывая на работу в крупной организации, но и создавая на практике устройства в условиях дома или небольшой лаборатории с помощью программируемых логических интегральных схем (ПЛИС). Возможности, которые предоставляет эта элементная база, приближают схемотехнику к работе программиста, но в то же время между этими сферами деятельности существуют важные отличия. Поэтому изучение языков описания аппаратуры, таких как Verilog и VHDL, не следует приравнивать к изучению программирования. В этой связи техническая литература, объясняющая не только синтаксические конструкции Verilog, но и принципы построения цифровых схем, представляет большой интерес и открывает перед разработчиками широкие возможности создания новых устройств.

Данная книга предоставляет читателю интересный практический материал, позволяющий в короткие сроки освоить проектирование цифровых устройств с минимальными затратами ресурсов. Поскольку используемые средства разработки позволяют моделировать схемы без привлечения отладочных плат, читатель может попробовать освоить материал без приобретения дополнительного оборудования. Структура книги удачно выстроена и сохраняет баланс между изложением теоретических сведений и практическими примерами, которые позволяют использовать ее как руководство к действию, закрепляя полученные навыки работой в САПР.

В целом можно констатировать появление еще одного качественного издания, созданного группой российских авторов с учетом современного состояния цифровой электроники.

Илья Евгеньевич Тарасов,

автор книг по разработке цифровых систем на основе ПЛИС Xilinx,
д. т. н., профессор Российского технологического университета МИРЭА

Сергей Иванец, Александр Романов

Цифровой синтез: RISC-V

**Глава 1. Основы комбинационной логики.
Маршрут разработки цифровых схем**

Содержание

1.1	Краткие теоретические сведения	1-4
1.2	Использование схемотехнического редактора	1-7
1.2.1	Установка пакета Quartus Prime	1-7
1.2.2	Создание проекта в схемотехническом редакторе	1-11
1.2.3	Создание файла в схемотехническом редакторе	1-16
1.2.4	Использование схемотехнического редактора (Schematic editor)	1-18
1.2.5	Разработка более сложной схемы	1-22
1.3	Компиляция проекта	1-24
1.3.1	Использование RTL Viewer	1-25
1.4	Назначение выводов. Компиляция проекта	1-25
1.5	Конфигурирование ПЛИС	1-27
1.6	Разработка схемы с использованием языка описания аппаратуры. Симуляция	1-30
1.6.1	Загрузка и установка ModelSim Starter Edition	1-31
1.6.2	Загрузка и установка пакетов Icarus Verilog и GTK Wave	1-33
1.7	HDL-модуль и его описание	1-34
1.8	Тестбенч и его описание	1-36
1.8.1	Симуляция с использованием ModelSim	1-38
1.8.2	Симуляция с использованием Icarus Verilog и GTKWave	1-39
1.9	Создание описания схемы на языке Verilog HDL. Синтез схемы	1-40
1.10	Упражнения	1-43
1.10.1	Основное задание	1-43
1.10.2	Контрольные вопросы	1-44

Глава посвящена основам цифрового дизайна и знакомит с логическими вентилями – основными элементами цифровых систем. Вначале описывается проектирование простой схемы, содержащей всего несколько логических вентиляей, с помощью графического редактора. Далее спроектирована та же схема, но с использованием языка описания аппаратуры (**Hardware Description Language, HDL**). Спроектированная схема проверяется с помощью симулятора – специальной программы для тестирования цифровых схем. Для того чтобы увидеть, как работает схема «в железе», программируется микросхема **ПЛИС (Программируемая логическая интегральная схема, Field-Programmable Gate Array, FPGA)**. Выполнив все описанные шаги, вы познакомитесь с типичным циклом разработки цифровой системы.

Требования к аппаратным и программным средствам

Для выполнения практических работ вам понадобится следующее программное и аппаратное обеспечение:

- персональный компьютер с установленной операционной системой Windows (виртуальная машина с ОС Windows не подойдет), x64, 8GB RAM, USB port;
- пакет **Quartus Prime 17.0¹** и старше;
- пакет **ModelSim Altera Edition**;
- программы **Icarus Verilog** и **GTKWave²**.

Программы **Quartus** и **ModelSim** являются платными, но они имеют и студенческие бесплатные версии, которые могут быть свободно скачаны с сайта производителя **ПЛИС Altera**.

Также в данном практикуме используется отладочная плата компании **Terasic DE10-Lite³**. Она содержит микросхему **ПЛИС** компании **Altera MAX10⁴ (10M50DAF484C7G)**. В папке doc дополнительных материалов к настоящей главе (https://github.com/RomeoMe5/ddlm_riscv, lab_01/doc) размещены инструкция к данной отладочной плате и ее электрическая схема (эти же документы могут быть бесплатно загружены с сайта компании **Terasic**).

Хотя в этом практикуме используется отладочная плата с микросхемой **MAX10** от компании **Intel FPGA**, концепции и методологии, которые вы узнаете при выполнении работ, могут быть использованы и при работе с **ПЛИС** от других производителей, например **Xilinx**. Однако следует учитывать то, что инструменты для проектирования и микросхемы быстро развиваются, и последние версии **САПР** компании **Xilinx (Vivado Design Suite)⁵** больше не поддерживают схематический редактор, а только разработку на основе языков описания аппаратуры.

¹ <http://dl.altera.com/?edition=lite> (в некоторых регионах ссылка может быть недоступна).

² Инсталлятор можно найти в папке **pkg** материалов к данной главе.

³ <http://de10-lite.terasic.com/>.

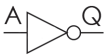

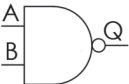
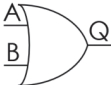

⁴ <https://www.altera.com/products/fpga/max-series/max-10/overview.html>.

⁵ <https://www.xilinx.com/support/answers/53764.htm>.

1.1 Краткие теоретические сведения

Рассмотрим цикл разработки **комбинационного устройства**, схема которого будет содержать логические вентили – **И**, **ИЛИ**, **НЕ**, а также **исключающее ИЛИ**. Особенностью комбинационных схем является то, что они выполняют только заданную логическую функцию над входными сигналами, но не сохраняют их значения. В следующей главе также рассмотрены **последовательностные** устройства, которые содержат элементы для хранения значений, и их состояние поэтому может зависеть не только от текущего набора входных сигналов, но и от предыстории. Логические вентили являются теми основными «кирпичиками», с помощью которых строятся все остальные элементы цифровых систем – от простых элементов, таких как дешифратор или триггер, до самых сложных – процессоров и систем на кристалле (**system-on-chip, SoC**).

Для построения устройства на основе логических элементов необходимо определить логические функции, которые описывают требуемые логические операции. Ниже показаны основные логические элементы, их обозначения, уравнения и таблицы истинности.

Вентиль	Символ	Уравнение	Таблица истинности															
НЕ (NOT)		$Q = \overline{A}$	<table border="1"> <thead> <tr> <th>A</th> <th>Q</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </tbody> </table>	A	Q	0	1	1	0									
A	Q																	
0	1																	
1	0																	
И (AND)		$Q = A \cdot B = A \& B$	<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>Q</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	A	B	Q	0	0	0	0	1	0	1	0	0	1	1	1
A	B	Q																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
И-НЕ (NAND)		$Q = \overline{A \cdot B} = \overline{A \& B}$	<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>Q</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	A	B	Q	0	0	1	0	1	1	1	0	1	1	1	0
A	B	Q																
0	0	1																
0	1	1																
1	0	1																
1	1	0																
ИЛИ (OR)		$Q = A + B$	<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>Q</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	A	B	Q	0	0	0	0	1	1	1	0	1	1	1	1
A	B	Q																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
ИЛИ-НЕ (NOR)		$Q = \overline{A + B}$	<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>Q</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	A	B	Q	0	0	1	0	1	0	1	0	0	1	1	0
A	B	Q																
0	0	1																
0	1	0																
1	0	0																
1	1	0																

Конец ознакомительного фрагмента.

Приобрести книгу можно

в интернет-магазине

«Электронный универс»

e-Univers.ru