

Оглавление

Введение	0-1
Глава 1. Основы комбинационной логики. Маршрут разработки цифровых схем	1-1
Глава 2. Основы последовательностной логики. Управление энергопотреблением цифровой схемы	2-1
Глава 3. Шифраторы и дешифраторы. Скорость работы комбинационных блоков	3-1
Глава 4. Мультиплексор, демультиплексор и селектор. Построение иерархических модулей	4-1
Глава 5. Сумматор, компаратор, устройство сдвига и АЛУ. Повышение скорости арифметических операций	5-1
Глава 6. Последовательная логика. Счетчики и сдвиговые регистры	6-1
Глава 7. Память: регистровый файл и стек	7-1
Глава 8. Конечные автоматы: основы	8-1
Глава 9. Использование конечных автоматов для связи с периферийными устройствами	9-1
Глава 10. Конвейерная обработка данных	10-1
Глава 11. Софт-процессор: основы микроархитектуры	11-1
Приложение А. Путь вперед: от устройств на базе FPGA к массовому рынку ASIC для популярных гаджетов	A-1
Приложение Б. История успеха победы российской команды на международном конкурсе Innovate FPGA от Intel	B-1

Цифровой синтез: практический курс

Введение

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

До сих пор встречаются люди, которые считают, что **Verilog** – 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 25 лет, использует технологию компиляции (синтеза) схем из языков описания аппаратуры, главный из которых – **Verilog**. Огромное число 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 использованием 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.

Данный учебник – 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 необходимо 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 писать 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 работать 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 высокоуровневым моделям. Этим инженерам также необходимо пони-

мать основы цифрового синтеза и программирования 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, сейчас бурно развиваются ускорители нейронных сетей, чипы для машинного зрения и даже специализированные вычислители криптовалют. Создателям всех этих устройств необходимо понимать и программную, и аппаратную стороны вычислений.

Обоснования для создания этой книги

Данная книга – «Цифровой синтез: практический курс» – создана совместными усилиями преподавателей и инженеров из нескольких университетов и компаний не только из России, но и из Украины и США. Этот практикум – один из нескольких образовательных проектов, нацеленных на подъем электроники в странах постсоветского пространства, которые все вместе можно уже рассматривать как осознанную стратегию.

К таким проектам относятся, например:

- Симулятор **MIPT-MIPS**¹, 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 **MIPS** ядро **schoolMIPS**², разработанное Станиславом Жельнио из 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 украинских 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. Данный практикум можно рассматривать как расши-

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

² <https://github.com/MIPSfpga/schoolMIPS>.

ренное продолжение и дополнение к этому курсу, ориентированное на практическое применение.

- Семинары **MIPSfpga**, организованные в 2015–2017 годах Imagination Technologies в партнерстве с российскими, украинскими и казахскими университетами (МГУ, НИУ ВШЭ (МИЭМ), МИФИ, МФТИ, МИЭТ, ИТМО, Самарский университет, Томский ТГУ, украинские КПИ и КНУ, казахский 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. К сожалению, **MIPSfpga** слишком сложен для начальной демонстрации основных принципов микроархитектуры. Данный практикум содержит целую главу, посвященную проекту **schoolMIPS**, который значительно проще, чем **MIPSfpga**. При этом **schoolMIPS** 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. Мистер Данчек написал приложение к данному практикуму. В нем рассказано, как студент, выполнивший упражнения на **ПЛИС**, сможет в будущем перейти к разработке заказных микросхем (**ASIC, Application Specific Integrated Circuits**). К **ASIC** относятся практически все микросхемы, которые применяются в массовых электронных устройствах.

Юрий Панчул,
разработчик процессорных ядер MIPS и ускорителя нейросетей Wave,
Саннивейл, Калифорния

История создания данной книги

История создания книги «Цифровой синтез: практический курс» традиционно для таких проектов не проста. После успеха переводной версии учебника «Цифровая схемотехника и архитектура компьютера» на русском языке стало ясно, 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, при том что издание книги требовало финансирования. У истоков идеи написания данной книги стоял Юрий Панчул, сумевший объединить для ее воплощения множество преподавателей и инженеров из России, Украины и США. Вторым фактором, который помог появиться книге, – это знакомство Юрия Панчула (в рамках семинаров Nanometer ASIC) со мной, Александром Романовым, руководителем Учебной лаборатории Систем автоматизированного проектирования Московского института электроники и математики Национального университета «Высшая школа экономики» (УИ САПР НИУ ВШЭ). Несмотря на экономический уклон НИУ ВШЭ, МИЭМ в прошлом был отдельным техническим вузом, готовившим студентов в области электроники и математики. Сейчас же, опираясь на необходимые технические и людские ресурсы, МИЭМ является одним из ведущих центров компетенций, в том числе и по цифровому синтезу. Идея создания учебника получила поддержку на уровне руководства в лице Евгения Аврамовича Крука, после чего началась работа по написанию книги, сбору и редактированию глав от разных авторов, рецензированию и разработке дополнительных материалов.

Чем эта книга отличается от других?

Особенность данной книги в том, что во всех главах каждый пример кода сопровождается листингом и тестбенчем, которые находятся в дополнительных материалах к книге (<https://github.com/RomeoMe5/DDLM>). Это позволяет читателю легко использовать уже готовые исходные коды, проводить моделирование, прототипирование на учебной плате с ПЛИС и модифицировать работающие примеры программ.

Немного про отладочные платы: поскольку команда, разрабатывавшая книгу, находилась территориально в разных местах, одна из целей курса состояла в том, чтобы сделать его максимально доступным для людей разного достатка. Изначально книга ориентирована на плату **De10-Lite** от компании Terasic на основе **ПЛИС MAX 10K** производства **Intel FPGA** (в прошлом – компания Altera). Выбор платы обусловлен ее относительно небольшой стоимостью (около \$55 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 Intel FPGA (Altera)** популярны в России и ближнем зарубежье и 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, в ближайшее время планируется миграция примеров кодов на другие попу-

лярные отладочные платы на основе ПЛИС: **Mapxход, Terasic De1-SoC, Terasic De10-Standard, Digilent Nexys 3/4** и др. Для выполнения работ не требуется какого-либо платного программного обеспечения, так как **Quartus Prime Lite Edition + Modelsim / GTKWave** 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 какая-либо отладочная плата на основе ПЛИС.

Содержание книги

Еще одна особенность книги – то, что авторы представили ее главы в виде отдельных независимых разделов. То есть можно изучать тот раздел, который необходим сейчас, и обращаться к другим главам по необходимости. Каждую главу в основном писали один-два автора, после чего редактировали и рецензировали другие люди. Вот почему каждая глава имеет свой неповторимый оригинальный стиль, приведенный, впрочем, к единому оформлению и терминологии. Таким образом, в случае если одна глава воспринимается читателем тяжело, то, возможно, другая будет читаться им 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.

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. Основы последовательностной логики. Управление энергопотреблением цифровой схемы

Данная глава посвящена разработке последовательностных устройств. Рассматриваются защелки и триггеры на примерах их различных реализаций на языке **Verilog**. Глава позволяет понять, чем комбинационная логика отличается от последовательностной и как можно управлять энергопотреблением цифровых устройств на этапе их проектирования.

Глава 3. 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 4. 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 5. 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 6. Последовательностная логика. Счетчики и сдвиговые регистры

В данной главе изложение возвращается к последовательностной логике и особенностям ее разработки на **Verilog** (блокирующие/неблокирующие присвоения, 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 7. 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 8. 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 9. Использование конечных автоматов для связи с периферийными устройствами

Эта глава расширяет тему проектирования конечных автоматов путем формализации академического подхода к проектированию автоматов; также демонстрируются и другие автоматы, например на основе счетчика. Глава обосновывает применение конечных автоматов в проектировании цифровых устройств как ключевого блока управления ими.

Глава 10. Конвейерная обработка данных

Глава посвящена описанию конвейерного подхода к обработке данных и особенностям разработки на **Verilog**. Приводится сравнение комбинационного, мультитактного и конвейерного подходов на примере разработки арифметического

блока; описываются дополнительные приемы повышения эффективности конвейерных схем.

Глава 11. Софт-процессор: основы микроархитектуры

Данная глава объединяет в себе необходимые знания из предыдущих разделов этой книги при проектировании HDL-реализации относительно простого, но при этом реально функционирующего, одноктактного софт-процессорного ядра с архитектурой MIPS. В главе даются понятия микроархитектуры, тракта данных, устройства управления и т. д.; демонстрируется процесс добавления новых инструкций и расширения процессорного ядра и его блоков.

Приложение А. Путь вперед: от FPGA-устройств к массовому рынку ASIC для популярных гаджетов

Приложение дает общее представление об этапах проектирования чипов ASIC, начиная с идеи специализированного чипа и заканчивая его конечной реализацией на кристалле. Это первый шаг в подготовке инженеров, прошедших обучение проектированию на ПЛИС, к применению имеющихся у них навыков в разработке ASIC для конкретных приложений.

Приложение Б. История успеха победы российской команды на международном конкурсе InnovateFPGA от Intel

Это приложение – небольшое интервью с участником российской команды, занявшей второе место на одном из основных международных конкурсов по проектированию на ПЛИС – InnovateFPGA от Intel.

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 курсов VLSI по проектированию массовых микросхем ASIC. Он будет полезен разработчикам аппаратно-программных систем, а также прикладным математикам, интересующимся алгоритмами САПР.

Выражаем надежду на то, что практикум станет такой же надежной основой курсов по цифровой электронике для большого количества университетов России и стран СНГ, какой уже стал переводной учебник «Цифровая схемотехника и архитектура компьютера» авторов Дэвида и Сары Харрис. В результате этого в России появится новое поколение инженеров, способных объединять лучшие мировые практики с изобретательностью, присущей народам наших стран; так Россия вместе со своими соседями займет достойное место в мировой промышленной электронике.

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

Данный учебник позволяет первично познакомиться с цифровым дизайном через изучение языка Verilog и правильных design-style и best practices

Прочтение этой книги вызвало у меня воспоминания об относительно далеком прошлом. В те времена, будучи студентом факультета технической информатики Мангеймского университета в Германии, я имел первый опыт программирования на языке **Verilog** и загрузки программного кода в ПЛИС. Самой сложной задачей была реализация программы для рисования, и самые продвинутые студенты рисовали круги. Помнится, что студенты одной из групп смогли разработать элементарный процессор и запрограммировать отрисовку кругов на ассемблере для этого процессора. Эти студенты стали теперь профессорами в университетах Германии.

Купив в 1990 году компанию Gateway Design Automation, в которой работал изобретатель языка **Verilog** Фил Мурби, компания Cadence Design Systems стала правообладателем **Verilog**. Этот язык изначально разрабатывался для описания микросхем для цифрового симулятора. Со временем **Verilog** также стал использоваться для синтеза, то есть превращения описания микросхемы на более абстрактном уровне **Register-Transfer-Logic RTL** в менее абстрактный **Gate-Level**, где описываются вентили, которые предоставляют конкретные изготовители микросхем (такие как: TSMC, UMC – Тайвань, GlobalFoundries – США). Этих изготовителей принято называть foundries; российские foundries – Ангстрем и Микрон в Зеленограде.

Verilog настолько успешен, что существуют несколько языков, которые имеют с **Verilog** общее название, но используются для разных целей. Например, **Verilog-A** – для моделирования аналоговых микросхем, **Verilog-AMS** – для моделирования цифроаналоговых микросхем, **SystemVerilog** – для верификации больших цифровых микросхем.

Конкретно этот учебник позволяет первично ознакомиться с цифровым дизайном, но охватывает при этом более широкие аспекты – обучает не только самому языку, но правильному **design-style** и **best practices**, что чрезвычайно важно для недопущения ошибок при подготовке к синтезу. Даже самое умное ПО не может синтезировать оптимальный **PPA** (**power, performance, area** – три самых важных показателя качества микросхемы), если исходный код (даже будучи грамматически правильным) не соблюдал определенные **design-styles**. Таким образом, данный учебник обладает двумя весомыми преимуществами: во-первых, он включает в себя базовую информацию, поиск которой требует, как правило, больших затрат времени и сил (проблемы выбора подходящего и, по возможности, бесплатного ПО и совместимой ПЛИС, проблемы их установки/настройки), и во-вторых, может использоваться для самообучения как студентами, так и научными сотрудниками, поскольку снабжен всеми необходимыми материалами от подготовленных исходных кодов программ для моделирования и синтеза до презентаций к каждой главе.

Компания Cadence (сотрудником которой я являюсь уже в течение 16 лет) выпускает программное обеспечение, применяемое на самых различных стадиях

проектирования и разработки комплексных микросхем, а также целых систем (приборы с печатными платами, **RF**-компонентами и несколькими системами на кристалле (**SoC**) в одной упаковке). Хотя ПО компании Cadence существенно облегчает процесс проектировки систем на кристалле с миллиардами элементов, именно талантливые и хорошо обученные инженеры являются главным капиталом компаний – разработчиков микросхем. Поэтому Cadence в высокой степени заинтересована в обучении следующего поколения инженеров-микроэлектронщиков, так как видит в них и будущих разработчиков, и будущих клиентов. В 2007 году была создана Cadence Academic Network (группа внутри компании Cadence), поддерживающая связи с ведущими университетами и предоставляющая им академические лицензии для доступа к ПО. Следует отметить, что данное программное обеспечение используется разработчиками самых передовых компаний мира, и Cadence Academic Network распространяет учебные материалы для обучения этим комплексным программам.

Компания Cadence приветствует появление такой нужной книги на российском рынке и будет рада контактам с читателями, усвоившими азы разработки цифровых микросхем и готовыми к дальнейшему обучению. Хотелось бы пожелать всем читателям этого учебника успехов и новых познаний.

Антон Клотц,
University Program Manager
Cadence Design Systems

Точно так же, как до этого в России стала массовой профессия программиста, данный курс поможет сделать массовой профессией электронного инженера – разработчика цифровой аппаратуры любой сложности

История развития мировой вычислительной техники насчитывает уже более 70 лет и является прямым отражением той жестокой конкурентной борьбы сверхдержав и их союзников, которую они ведут за контроль над мировой экономикой, финансовыми и товарными потоками с целью получения долгосрочных преимуществ перед остальными странами. Кроме того, разработка быстродействующих компьютеров является неотъемлемой частью технологического соперничества нынешних сверхдержав – США и Китая. Галопирующее развитие и широкое внедрение массовых коммуникаций и встроенных микрокомпьютеров в контексте интернета вещей приводит к тотальной цифровизации экономики.

Технологическая зависимость в области электронной компонентной базы может вызвать катастрофические последствия в случае разного рода санкций и ограничений на поставки со стороны зарубежных партнеров.

Несмотря на то что цифровая электронная инженерия была невостребованной в российской промышленности и устойчиво деградировала с 1990-х гг., в настоящее время наличие собственных инженерных компетенций в данной области на массовом уровне становится условием выживания любой страны, претендующей даже на частичный технологический суверенитет.

Учебное пособие «Цифровой синтез: практический курс» как комплекс практических работ является очередным этапом в создании массовой школы цифровой электронной инженерии на всем пространстве СНГ. Этому предшествовало издание профильных учебников и проведение серий семинаров, нацеленных на создание практических работ по различным разделам проектирования цифровых схем и микропроцессоров, о чем подробно написано Юрием Панчулом в предисловии.

Различные лабораторные работы по проектированию и синтезу цифровых схем на языках регистровых передач для описания аппаратуры стали необходимой частью подготовки электронных инженеров – проектировщиков микросхем для массовых электронных изделий. Через подобный практикум проходят будущие создатели смартфонов, разработчики автомобильной электроники и процессоров для различных встроенных применений, включая электронику для космических зондов. В США известным примером такого практического курса является 6.111 из MIT¹, а также различные варианты лабораторных заданий с **Verilog/VHDL** и **ПЛИС/FPGA** предусмотрены учебными программами практически для всех, кому преподают электронику, включая студентов местных университетов в небольших штатах.

Авторы надеются, что данный курс поможет сделать массовой профессией электронного инженера – разработчика цифровой аппаратуры любой сложности. Точно так же, как до этого в России стала массовой профессией программиста или

¹ <http://web.mit.edu/6.111/volume2/www/f2018/index.html>.

программного инженера – разработчика систем и приложений. Основами языков регистровых передач или **Register Transfer Level (RTL)** нужно владеть не только самим разработчикам, но и представителям смежных профессий – верификаторам, специалистам по физическому проектированию, а также алгоритмистам, которые пишут инструментальные программы для разработчиков электроники. Программисты встроенных систем и программ искусственного интеллекта тоже получают пользу от понимания того, как работают классические процессоры, GPU и нейроускорители.

Данное пособие создано усилиями международного авторского коллектива специалистов из ведущих университетов и ИТ-компаний. Помимо традиционных основ проектирования цифровой техники, авторы ввели в курс дополнительные проекты, которых не хватало в имеющихся курсах. Они включают развернутое объяснение принципов конвейерных вычислений, введение в микроархитектуру процессоров, а также детальные разъяснения о том, как студент может использовать свой опыт, полученный от лабораторных работ на ПЛИС, в своей дальнейшей карьере. Полученные практические навыки обеспечивают статус разработчика массовых изделий на специализированных полужаказных микросхемах **ASIC (Application Specific Integration Circuits)** и крупных систем на кристалле (СнК), которых так остро не хватает в отечественной индустрии.

Выражаю уверенность в том, что это пособие имеет очень хороший потенциал для его последующего перевода на английский язык и массового использования в кооперации с компаниями, разрабатывающими инструментарий электронной инженерии. Такими партнерами могут стать крупные компании Cadence, Synopsys и Mentor Graphics.

С пожеланиями успехов авторам и пользователям данного учебного пособия,

Тимур Турсунович Палташев,
доктор технических наук, профессор,
руководитель академических проектов,
Radeon Technology Group,
Корпорация Advanced Micro Devices.
17 марта 2020 года

Книга «Цифровой синтез: практический курс» является, по сути, продолжением классического учебника по проектированию микроэлектроники «Цифровая схемотехника и архитектура компьютера» Дэвида Харриса и Сары Харрис, новая редакция которого (на русском языке) вышла всего несколько лет назад

Рецензируемая книга представляет собой расширенный практический курс, ориентированный на **Verilog** и обеспечивающий возможность выполнения практических задач на широкодоступных отладочных платах **FPGA**. Именно такой подход позволяет эффективно организовать подготовку квалифицированных разработчиков для отечественной микроэлектроники. Следует отметить, что данное издание, подготовленное международным авторским коллективом русскоязычных специалистов из ведущих университетов и ИТ-компаний, обладает всем необходимым потенциалом для последующего перевода на английский язык и дальнейшего использования в университетской среде.

История развития систем автоматизированного проектирования берет свое начало в 70–80-х годах прошлого века. К тому времени сложность систем, возрастающая по мере увеличения количества транзисторов на микросхеме по закону Мура, стала такой, что ручное проектирование схемотехники микроэлектронных устройств становилось практически невозможным. Проектирование аппаратуры повторило историю развития языков программирования: если первые вычислительные системы программировались на уровне кодов, то уже в 50-е годы приходилось приступать к разработке систем автоматизации программирования и, соответственно, языков программирования высокого уровня. По тому же пути (несколько позднее) пошли и разработчики аппаратуры – для проектирования новых вычислительных систем использовалось программное обеспечение, разработанное для уже существующих компьютеров. Но если для проектирования печатных плат устройств уже в середине 80-х годов существовали специализированные САПР, то к промышленному проектированию схемотехники микроэлектронных изделий удалось приступить только в начале 90-х (при том что сам **Verilog** был разработан в середине 80-х). Однако первоначально он был ориентирован на описание и моделирование логических схем; применение его для синтеза на уровне логических элементов и гейтов было реализовано лишь с ростом популярности основанных на нем средств моделирования и отладки.

Знание **Verilog** и использование его обширного инструментария с целью проектирования цифровых систем сегодня является абсолютно необходимым навыком для любого инженера-электронщика (кроме разве что специалистов в аналоговой и силовой электронике, где количество элементов весьма ограничено, зато чрезвычайно важны их физические характеристики). В микроэлектронике ситуация прямо противоположная – все цифровые системы строятся из огромного (как правило, исчисляемого миллионами, а иногда десятками и сотнями миллионов) количества физически одинаковых элементов. Поэтому для инженера-схемотехника необходимы инструменты иерархической абстракции, позволяющие манипулировать логикой крупных модулей и строить из них системы, не опускаясь на уровень отдельных транзисторов и имея при этом возможность анализировать их поведение во времени. Именно такую возможность и предоставляет **Verilog**

с его инструментарием, зачастую поддерживающий и другие языки описания аппаратуры (в частности, **VHDL**).

В течение достаточно длительного времени одной из основных проблем отладки и тестирования микроэлектронной аппаратуры являлись технологические процессы производства микросхем. Ситуация коренным образом изменилась с появлением в середине 80-х и широким распространением в 90-е годы прошлого века технологии **FPGA** (**ПЛИС** в русскоязычной литературе) – программируемых логических матриц, состоящих из сотен тысяч и миллионов одинаковых индивидуально программируемых гейтов. В то же время именно **Verilog** и другие языки описания аппаратуры сделали эффективным проектирование и использование систем на **FPGA**. И если в 90-е годы они использовались в основном в телекоммуникационном и сетевом оборудовании, то сегодня трудно представить большую цифровую систему, не использующую **FPGA** в качестве акселераторов тех или иных процессов. Эта технология оказалась чрезвычайно удачным компромиссом между эффективностью, сравнимой с чисто аппаратными решениями (хотя и уступающей им), и гибкостью, простотой использования, характерной для программного обеспечения.

Таким образом, **FPGA** стали активно применяться в тестировании блоков схемотехники или целых цифровых систем, ориентированных на дальнейшую реализацию в микроэлектронном исполнении (**ASIC**). Хотя до сих пор не существует полностью автоматизированного процесса переноса **Verilog**-программы из **FPGA**-прототипа на технологический процесс конкретного изготовителя, этот процесс существенно повышает эффективность и снижает стоимость отладки аппаратных решений микроэлектроники. Для упрощения процесса прототипирования на **FPGA** все основные производители стали выпускать готовые отладочные платы, включающие (кроме самой микросхемы **FPGA**) все необходимое окружение, в том числе стандартные интерфейсы с компьютером; отпала нужда в специализированных программаторах, логических анализаторах и другом инженерном оборудовании стадии отладки.

В то же время с массовым распространением микросхем **FPGA**, отладочных плат на их основе и радикальным падением стоимости за счет эффекта масштаба стало возможным использовать их для учебных задач. Современные инженерные курсы в области микроэлектроники, как правило, используют **FPGA** в качестве основного инструмента проверки и тестирования программ на **Verilog** даже при ориентации на создание в дальнейшем **ASIC** и **SoC** (систем на кристалле).

Предлагаемый в книге «Цифровой синтез: практический курс» практикум по инструментам и технологиям цифрового синтеза схемотехники полностью охватывает все основные разделы цифровой схемотехники, а также используемый при проектировании бесплатный инструментарий. Покрывается практически весь материал, представленный в учебнике «Цифровая схемотехника и архитектура компьютера» Дэвида Харриса и Сары Харрис. Важной особенностью рецензируемой книги является то, что во всех главах каждый пример кода сопровождается исходным кодом и тестбенчем, которые находятся в дополнительных материалах к книге.

В заключение хотелось бы отметить, что необходимость в подготовке квалифицированных кадров в области проектирования микроэлектроники является вполне объективной – востребованность таких кадров будет только расти в связи с массовым распространением устройств интернета вещей и переводом все большего объема функционала на системы на кристалле. Если в последние десятилетия основным фокусом при обучении специалистов по цифровым системам была подготовка программистов и (позднее) аналитиков данных, то в ближайшие годы будут все более востребованы инженеры со специализацией на стыке программирования и аппаратуры, специалисты по киберфизическим системам и электронике; в нашей стране их готовят в ограниченном количестве учебных заведений, при этом ощущается существенный дефицит опыта и литературы. Представляется важным, чтобы МИЭМ НИУ ВШЭ стал одним из лидеров этого направления в российской академической среде; таким образом, издание рецензируемой книги в «ДМК Пресс» – ведущем издательстве, специализирующемся на выпуске компьютерной и радиотехнической литературы, – является важным шагом в этом направлении.

Игорь Рубенович Агамирзян,
вице-президент НИУ ВШЭ,
профессор факультета компьютерных наук НИУ ВШЭ,
канд. физ.-мат. наук

Сергей Иванец, Александр Романов

Цифровой синтез: практический курс

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

Содержание

1.1.	Краткие теоретические сведения	1-4
1.2	Использование схмотехнического редактора	1-7
1.2.1	Установка пакета Quartus Prime	1-7
1.2.2	Создание проекта в схмотехническом редакторе	1-11
1.2.3	Создание файла в схмотехническом редакторе	1-16
1.2.4	Использование схмотехнического 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 GTK Wave	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

Конец ознакомительного фрагмента.

Приобрести книгу можно

в интернет-магазине

«Электронный универс»

e-Univers.ru