

Оглавление

Об авторах	11
Предисловие от главного редактора русского перевода	13
Предисловие	16
РАЗДЕЛ 1. ВВЕДЕНИЕ В FPGA.....	21
Глава 1. Введение в FPGA и Xilinx Vivado	23
Технические требования	23
Аппаратура.....	24
Программное обеспечение	24
Что такое ASIC?	24
Почему ASIC или FPGA?	25
Как компания создает программируемое устройство, используя ASIC ...	27
Базовые логические элементы	27
Более сложные операции	30
Знакомство с FPGA	31
Изучение Xilinx Artix-7 и устройств 7-й серии	33
Знакомство с набором инструментов Vivado и отладочными платами ...	37
Знакомство с Vivado	40
Выполнение примера.....	46
Программирование платы.....	55
Выводы	56
Вопросы.....	56
Задание повышенной сложности	57
Дополнительное чтение.....	57
РАЗДЕЛ 2. ВВЕДЕНИЕ В ПРОЕКТИРОВАНИЕ, МОДЕЛИРОВАНИЕ И СИНТЕЗ НА VERILOG RTL	59
Глава 2. Комбинационная логика.....	61
Технические требования.....	61
Создание модулей SystemVerilog.....	61
Создание многократно используемого кода с помощью параметров.....	62
Знакомство с типами данных	63
Представление встроенных типов данных.....	63
Создание массивов	64
Обращение к элементам массива.....	65
Присвоение значений элементам массива	66
Работа с цепями с тремя состояниями	66

Работа со знаковыми и беззнаковыми числами	67
Добавление битов к сигналу с помощью операции конкатенации	68
Преобразование знаковых и беззнаковых чисел	68
Создание типов, определяемых пользователем	69
Доступ к сигналам при использовании значений перечисляемых типов... ..	69
Упаковка кода с помощью функций	70
Создание комбинационной логики	70
Операторы присваивания.....	71
Принятие решений: if-then-else	73
Сравнение значений	73
Операторы if с уникальностью или приоритетностью.....	74
Оператор выбора case	75
Использование пользовательских типов данных	76
Создание структур.....	76
Создание объединений	77
Проект 1. Создание комбинационной схемы	77
Testbench	78
Моделирование с помощью целевого тестирования.....	80
Моделирование с использованием рандомизированного тестирования ...	80
Моделирование с использованием ограниченной рандомизации	80
Реализация детектора ведущей единицы с использованием оператора case	80
Управление реализацией с помощью generate	81
Проектирование многоразового детектора ведущей единицы с помощью цикла for	83
Реализация сумматора/вычитателя (adder/subtractor)	85
Сложение.....	85
Вычитание.....	86
Умножение	86
Объединяем все вместе	87
Добавление защелки	89
Выводы	89
Вопросы.....	89
Задание повышенной сложности	90
Дополнительное чтение.....	90

Глава 3. Подсчет нажатий на кнопку **91**

Технические требования.....	91
Что такое последовательностный элемент?	91
Синхронизация проекта	91
Базовый регистр	93
Создание триггеров.....	93
Когда использовать always@() для генерации триггера.....	95
Использование неблокирующих присваиваний	96
Регистры в Artix 7	97
Как удерживать состояние схемы с помощью входа разрешения тактового сигнала.....	98
Сброс триггера	99

Проект 2. Подсчет нажатий на кнопку	100
Семисегментный индикатор	100
Обнаружение нажатия на кнопку.....	103
Проблемы, возникающие из-за асинхронных сигналов	104
Использование асинхронного сигнала напрямую	105
Проблема с нажатием кнопок	106
Разработка безопасной реализации	107
Переход на десятичное представление.....	109
Знакомство с PLA.....	110
Что насчет симуляции?	114
Подробное изучение синхронизации	115
Зачем использовать несколько тактовых сигналов?	115
Двухступенчатый синхронизатор	115
Синхронизация управляющих сигналов	116
Передача данных	117
Выводы	118
Вопросы.....	118
Задание повышенной сложности	119
Дополнительное чтение.....	119
Глава 4. Разработка калькулятора.....	120
Технические требования	121
Реализация первого конечного автомата	121
Разработка последовательного конечного автомата.....	121
Разделение комбинационной и последовательностной логики в конечном автомате.....	122
Разработка интерфейса калькулятора	123
Проектирование конечного автомата Мура	124
Реализация конечного автомата Мили	126
Практическое проектирование конечных автоматов	126
Проект 3. Создание простого калькулятора.....	127
Инкапсуляция для повторного использования.....	127
Проектирование модуля верхнего уровня иерархии.....	129
Изменение тактовой частоты с помощью PLL или MMCM	130
Разработка блока деления.....	134
Построение конечного автомата невосстанавливающего делителя	134
Моделирование делителя	138
Определение размера промежуточного остатка.....	138
Проект 4. Управление перекрестком с помощью светофоров	139
Определение графа состояний	140
Отображение состояний светофоров	140
Выводы	142
Вопросы.....	142
Задание повышенной сложности	143
Задание еще более высокой сложности	143
Дополнительное чтение.....	143

Глава 5. Ресурсы FPGA, и как их использовать..... 144

Технические требования.....	144
Проект 5. Слушать и учиться.....	145
Что такое цифровой PDM-микрофон?.....	145
Моделирование работы микрофона.....	148
Встроенная память.....	150
Захват аудиоданных.....	154
Проект 6. Использование датчика температуры.....	157
Обработка данных.....	158
Сглаживание данных.....	159
Более глубокое погружение в FIFO.....	160
Ограничения.....	163
Генерация FIFO.....	163
Выводы.....	165
Вопросы.....	165
Дополнительное чтение.....	166

Глава 6. Математика, параллелизм и конвейеризация 167

Технические требования.....	168
Числа с фиксированной точкой.....	168
Проект 7. Использование чисел с фиксированной точкой для обработки данных с датчика температуры.....	169
Использование арифметики чисел с фиксированной точкой для очистки времени запуска.....	170
Преобразование температуры с помощью арифметики с фиксированной точкой.....	172
A как насчет чисел с плавающей точкой?.....	174
Сложение и вычитание с плавающей точкой.....	176
Умножение с плавающей точкой.....	176
Обратное значение для числа с плавающей точкой.....	176
Более практичная библиотека операций с плавающей точкой.....	176
Краткий обзор потокового интерфейса AXI.....	177
Проект 8. Обновление проекта датчика температуры до конвейерной реализации с плавающей точкой.....	179
Преобразование чисел из представления с фиксированной точкой в формат с плавающей точкой.....	179
Математические операции с плавающей точкой.....	181
Преобразование формата с плавающей точкой в формат с фиксированной точкой.....	182
Моделирование.....	183
Параллельные конструкции.....	185
ML, AI и массовый параллелизм.....	185
Параллельное проектирование – небольшой пример.....	186
Выводы.....	187
Вопросы.....	187
Задание повышенной сложности.....	188
Дополнительное чтение.....	188

РАЗДЕЛ 3. ВЗАИМОДЕЙСТВИЕ С ВНЕШНИМИ КОМПОНЕНТАМИ 189

Глава 7. Введение в AXI 191

Технические требования.....	191
Потоковая передача AXI.....	192
Проект 9. Создание IP-блоков для Vivado с использованием потоковых интерфейсов AXI.....	192
Потоковый интерфейс для семисегментного индикатора.....	193
Разработка IP ADT7420.....	198
Ядро t_temp.....	198
IP-интегратор.....	198
Отладка проекта с помощью IP-интегратора.....	206
Интерфейсы AXI4 (AXI full и AXI-Lite).....	207
Разработка IP-блоков – AXI-Lite, AXI full и AXI Stream.....	209
Добавление неупакованного IP-блока в IP-интегратор.....	212
Выводы.....	214
Вопросы.....	214
Дополнительное чтение.....	215

Глава 8. Много данных? MIG и DDR2 216

Технические требования.....	216
Проект 10. Подключение внешней памяти.....	217
Память DDR2.....	218
Генерация контроллера DDR2 с помощью Xilinx MIG.....	219
Установка параметров интерфейса AXI.....	223
Настройка параметров памяти.....	223
Настройка параметров FPGA.....	224
Модификация проекта для использования на плате.....	232
Другие типы внешней памяти.....	236
Память SRAM с четырехкратной скоростью передачи данных (Quad Data Rate, QDR).....	236
HyperRAM.....	236
SPI RAM.....	236
Выводы.....	237
Вопросы.....	237
Задача повышенной сложности.....	238
Дополнительное чтение.....	238

Глава 9. Лучший способ отображения – VGA..... 239

Технические требования.....	239
Проект 11. Основы работы с VGA.....	240
Определение регистров.....	243
Разработка простого интерфейса AXI-Lite.....	244
Генерация сигналов синхронизации для VGA.....	245
Отображение текста.....	251
Запрос памяти.....	253

Тестирование контроллера VGA	257
Проверка ограничений	257
Выводы	259
Вопросы	259
Задание повышенной сложности	260
Дополнительное чтение	260
Глава 10. Свести все воедино	261
Технические требования	261
Изучение интерфейса клавиатуры	262
Проект 12. Работа с клавиатурой	267
Моделирование работы интерфейса PS/2	270
Проект 13. Сводим все воедино	272
Отображение кодов клавиш PS/2 на экране VGA	272
Отображение показаний датчика температуры	275
Отображение аудиоданных	277
Выводы	280
Вопросы	281
Задание повышенной сложности	281
Дополнительное чтение	281
Глава 11. Темы повышенной сложности	282
Технические требования	282
Изучение более продвинутых конструкций SystemVerilog	282
Взаимодействие компонентов с использованием конструкции под названием «интерфейс»	282
Использование структур	285
Метки блоков	286
Цикл for	287
Цикл do...while	287
Выход из цикла с помощью оператора disable	288
Пропуск фрагментов кода с помощью оператора continue	288
Использование констант	289
Некоторые продвинутые конструкции языка SystemVerilog для верификации	289
Знакомство с очередями SystemVerilog	289
Продвинутое использование системной функции \$display	291
Утверждения	292
Использование \$error или \$fatal при синтезе проекта	292
Другие проблемы, и как их избежать	293
Выведение однобитных проводов	293
Несоответствие ширины шин	294
Повышение или понижение приоритетности сообщений Vivado	294
Обработка timing closure	295
Конвейеризация	297
Выводы	301
Вопросы	302
Дополнительное чтение	303

Об авторах

АВТОР

Фрэнк Бруно – опытный инженер-разработчик высокопроизводительных систем, специализирующийся на FPGA и имеющий некоторый опыт работы с ASIC. Работал в таких компаниях, как Cruise, SpaceX, Allston Trading и Number Nine. В настоящее время работает инженером по разработке на FPGA в компании Cruise.

РЕЦЕНЗЕНТ

Джордж Калдис получил степень бакалавра электротехники в Северо-Восточном университете и имеет более чем 30-летний опыт работы с FPGA. Является президентом GK-Digital LLC, консалтинговой компании по проектированию FPGA. Реализовал множество проектов FPGA для различных приложений – от беспроводных и проводных сетей до высокочастотного трейдинга и тестового оборудования.

РЕДАКТОРЫ РУССКОГО ПЕРЕВОДА

Романов Александр Юрьевич – главный научный редактор русского перевода данной книги, доцент 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. В 2015 г. защитил диссертацию в Институте проблем проектирования в микроэлектронике РАН (г. Зеленоград), является автором более 150 научных статей, патентов и книг. Более подробно об учебном процессе в лаборатории можно узнать из интервью: <https://miem.hse.ru/news/364316102.html>.

Юрий Всеволодович Ревич – научный редактор русского перевода – инженер-электронщик с многолетним стажем. Занимался автоматизацией производств, конструированием измерительных приборов для изучения океана и другими научными исследованиями. Работал редактором в периодических изданиях по IT-направлению. Автор многих статей и ряда популярных книг по электронике, среди которых «Занимательная электроника», «Практическое программирование микроконтроллеров Atmel AVR», сборник «Информационные технологии в СССР» и др.

Смехов Дмитрий Геннадьевич – консультант русского перевода – инженер-разработчик на ПЛИС с многолетним стажем. Занимается разработкой и верификацией проектов на основе ПЛИС Xilinx и Altera (Intel FPGA). Является сертифицированным инструктором компании Xilinx, проводит тренинги по темам PCI Express, Vitis AI, Versal ACAP. Работал в таких компаниях, как АО «ИнСис», КТЦ «Инлайн Групп», IRQ.

Предисловие от главного редактора русского перевода

Дорогие друзья!

Книга, которую вы держите в руках, открывает серию переводов зарубежных изданий по проектированию цифровых систем, которую готовят к выходу в свет компания «ДМК Пресс» и МИЭМ НИУ ВШЭ при поддержке группы компаний YADRO (yadro.com).

Актуальность подобных книг несомненна. Очевидно, что после 24 февраля 2022 г. развитие отечественной электроники сталкивается с новыми вызовами: жесткие торговые ограничения, запрет на пользование интеллектуальными продуктами (в том числе САПРами; например, такими как Quartus), а также и множество иных санкционных действий оказывают серьезное давление на высокотехнологичные отрасли в России. Единственный путь, который позволит не оказаться в стороне от прогресса, – это предпринимать консолидированные усилия ведущими компаниями и организациями по развитию отечественной электроники и в первую очередь по подготовке высококвалифицированных кадров. Серьезный барьер на этом направлении – существенная нехватка качественных учебных материалов на русском языке.

Впрочем, благодаря усилиям многих людей в последние годы уже наметилась тенденция к ее преодолению. Прошли те времена, когда для многих практически единственными доступным источником русскоязычных материалов был известный сайт Наливателя – Егорова Антона (<http://www.naliwator.narod.ru/>), – где преимущественно собраны машинные переводы руководств по проектированию на FPGA Altera. Появилась и много раз переиздана знаменитая книга Харрисов «Цифровая схемотехника и архитектура компьютера», вышел к ней сборник практических работ по Verilog «Цифровой синтез: практический курс» (под ред. А. Ю. Романова и Ю. В. Панчула), издана в русском переводе книга Д. Томаса «Логическое проектирование и верификация систем на System-Verilog», вышло множество других изданий. (О наличии материалов по FPGA на русском языке можно узнать на «Книжном FPGA-стриме #42» с М. Коробковым и С. Иванцом <https://www.youtube.com/watch?v=XaYGfNlhX0c>.) Тем не менее недостаток в русскоязычных материалах по различным аспектам проектирования цифровых систем все еще ощущается.

Компания «ДМК Пресс» в сотрудничестве с МИЭМ НИУ ВШЭ выбрала стратегию, обеспечивающую (кроме подготовки собственных изданий по цифровому синтезу) создание серии переводов наиболее популярных зарубежных книг по данной проблематике. Настоящее издание – Ф. Бруно «Программирование FPGA для начинающих» – является первым в этой серии. (В обозримом будущем планируется также выход в свет перевода книги С. Сазерленда

«RTL-моделирование на SystemVerilog для моделирования и синтеза: применение SystemVerilog для проектирования ASIC и FPGA».)

Следует отметить, что процесс создания переводных изданий по проектированию цифровых систем является достаточно непростой задачей: несмотря на свою популярность, оригинальные англоязычные книги имеют множество недостатков – частые повторения, опечатки, неточности, значительные стилистические погрешности. Бытующее американское изложение от первого лица при переложении на русский язык выглядит неестественным и является чуждым русской научной речи. Оказалось, что зарубежные авторы технической литературы в погоне за прибылью часто пишут «на поток», не сильно утруждаясь стилистикой. Поэтому при редактировании переводных материалов научно-технического содержания редактору приходится решать многочисленные задачи, связанные с обеспечением выполнения основной функции научного стиля речи – точной передачи логической информации в переведенном тексте, исключающей выражение субъективных, личных эмоций автора и его отступления от стилистических норм. Таким образом, отредактированный русский перевод в его окончательном виде – это скорее переизложение начального материала, обеспечивающее его читабельность для русскоязычной аудитории; фактически это книга, написанная заново.

О чем эта книга?

При выборе изданий для перевода в первую очередь преследовалась цель восполнить очевидные пробелы в массиве существующей русскоязычной литературы, при этом важным критерием отбора являлась популярность книги. Привлечение ряда экспертов (в том числе Ю. Панчула) позволило составить список наиболее интересных книг. Выбор книги «Программирование FPGA для начинающих» обусловлен тем, что она рассказывает о проектировании на FPGA компании Xilinx, демонстрационные примеры в ней реализованы на дешевых и доступных платах Nexys A7 и сопровождаются исходными кодами.

В целом абсолютному новичку эта книга поначалу может показаться несколько сложной, но если проявить немного усидчивости и вникнуть в исходные коды первых примеров, то ее восприятие заметно облегчится. В принципе, большая часть курсов по SystemVerilog, в отличие от других классических языков программирования, так и построена – обучаемого сразу погружают в несколько больших примеров, где он ведет разработку, часто копируя или модифицируя уже готовый код, постепенно изучая особенности языка (разработка программ, описывающих аппаратуру, гораздо сложнее, и постепенно изучать операторы языка нет никакого смысла).

Данная книга хороша для тех, кто раньше работал, например, в Quartus с продукцией фирмы Intel FPGA (т. е. имеет некоторый базис) и хочет расширить свой кругозор, познакомившись с Vivado и FPGA от Xilinx.

Если же у читателя имеется опыт работы в Vivado, эта книга все равно может быть полезной, потому что содержит ряд примеров работы с различной периферией (VGA, датчик температуры, микрофон, PS/2-клавиатура), а также обеспечивает знакомство с организацией проектов на основе интерконнекта AXI.

Книга, несомненно, понадобится студентам вузов (МИЭТ, ИТМО и др.), изучающим SystemVerilog, а также более продвинутым разработчикам, которые хотели бы получить практический опыт работы с реальными проектами.

Научный редактор будет очень признателен тем внимательным читателям, которые обнаружат в данном издании какие-либо ошибки или опечатки и сообщат о них на e-mail a.romanov@hse.ru. (Книги постоянно перепечатываются, и в каждом новом тираже все найденные ошибки и недочеты исправляются.)

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

Предисловие

Готовьтесь повеселиться. Автор этой книги разрабатывает ASIC и FPGA¹ уже 30 лет и каждый день сталкивается с новыми вызовами и волнениями, поскольку продвигает технологии к разработке новых приложений. За свою карьеру автор разработал ASIC, которые обеспечивали работу военных самолетов; графику, работающую на высокопроизводительных рабочих станциях и обычных ПК; технологию для питания следующего поколения программно-определяемых радиосистем; а также участвовал в проекте по обеспечению сети Интернет через космос на всем земном шаре. Часть этого опыта представлена в данной книге.

Для кого эта книга

Эта книга предназначена для тех, кто хочет узнать о технологии FPGA и о том, как ее можно использовать в своих проектах. Предполагается, что читатель ничего не знает о цифровой логике, поэтому книга начинается с представления базовых логических элементов и их функций, а заканчивается разработкой полноценной системы на кристалле. Некоторые знания в области программирования или аппаратного обеспечения полезны, но не обязательны. Если вы сможете установить программу, подключить USB-кабель и следовать пошаговым инструкциям, вы узнаете много нового.

Что включает в себя эта книга

Глава 1. Введение в FPGA и Xilinx Vivado. В главе объясняется, что такое ASIC и FPGA и как установить Xilinx Vivado и создать небольшой проект.

Глава 2. Комбинационная логика. В главе описано, как разработать с нуля полноценный модуль на SystemVerilog для выполнения некоторых базовых операций, чтобы продемонстрировать, как использовать комбинационную логику в собственных проектах. Глава также знакомит с разработкой testbench² на примере создания testbench с самопроверкой.

¹ ASIC (application-specific integrated circuit, «интегральная схема специального назначения») – интегральная схема, специализированная для решения конкретной задачи. FPGA (field-programmable gate array, «программируемая пользователем вентильная матрица») – разновидность программируемых логических интегральных схем, ПЛИС. Существуют и другие разновидности программируемых схем, но FPGA, как самые распространенные и универсальные, фактически стали синонимом ПЛИС. ASIC, подобно FPGA, разрабатываются на типовой основе, но затем отдаются в производство, в то время как FPGA выпускается в виде полуфабриката, который доводится до нужной функциональности программными методами непосредственно перед применением. Подробнее об ASIC и FPGA рассказано в главе 1. – *Прим. ред.*

² Testbench (дословно «испытательный стенд») – тестирующая программа или программно-аппаратный комплект, созданный для испытания запрограммированной в FPGA функциональности. В русскоязычной профессиональной среде прижился оригинальный англоязычный термин, потому в этой книге он приводится без перевода. – *Прим. ред.*

Глава 3. Подсчет нажатий на кнопку. В этой главе к проекту на основе комбинационной логики из предыдущей главы добавляются последовательностные элементы (встроенная память данных). В главе рассказывается о возможностях Artix-7 и других устройств FPGA для хранения данных, приводится пример простого проекта по подсчету нажатий на кнопку. Также пересказывается про использование тактовых генераторов и синхронизации, того немногого, что может полностью разрушить проект, если сделано неправильно.

Глава 4. Разработка калькулятора. В главе демонстрируется, как при создании более сложных схем неизбежно возникает необходимость отслеживать состояние устройства. В этой главе на примере классического инженерного устройства – контроллера светофора – рассказывается о конечных автоматах. Практическая часть главы посвящена улучшению калькулятора за счет разработки делителя с использованием схем с состояниями.

Глава 5. Ресурсы FPGA, и как их использовать. В этой главе после быстрого погружения в проектирование довольно сложных схем делается шаг назад и рассмотрены более подробно некоторые ресурсы FPGA. Чтобы продемонстрировать использование этих ресурсов в демонстрационных проектах, используются некоторые периферийные устройства (микрофон PDM и датчик температуры I2C), которые есть на отладочной плате.

Глава 6. Математика, параллелизм и конвейеризация. В этой главе более подробно рассмотрено использование чисел с фиксированной и плавающей точкой, также конвейерное проектирование и параллелизм для повышения производительности.

Глава 7. Введение в AXI¹. В главе рассказывается о том, как компания Xilinx приняла стандарт AXI для сопряжения своих IP² и разработала инструмент IP integrator для простого соединения IP-блоков с помощью графического интерфейса. В этой главе показано, как использовать AXI-интерфейс, на примере интеграции датчика температуры в проект с помощью IP integrator.

Глава 8. Много данных? MIG и DDR2. В этой главе рассказывается, как в Artix-7 обеспечен достаточный объем памяти и что делать, если требуется обеспечить доступ к мегабайтам или гигабайтам оперативной памяти. На отладочной плате размещена память DDR2. В главе рассказывается, как использовать Xilinx Memory Interface Generator для реализации интерфейса DDR2 и провести его моделирование, а также тестирование на плате.

Глава 9. Интерфейс VGA. В главе рассказывается о том, как устроен интерфейс VGA, и приводится простой способ отображения текста. До этого для вывода информации в проектах использовались светодиоды и семисегментный индикатор, что накладывало значительные ограничения. С помощью VGA становится возможно отображать аудиоданные и текст.

¹ AXI (Advanced eXtensible Interface) – стандарт высокопроизводительного интерфейса, разработанного фирмой ARM для связи между устройствами на одном кристалле. – *Прим. ред.*

² Сокращение IP (расшифровывающееся просто как intellectual property, «интеллектуальная собственность») означает в контексте FPGA специализированные области кристалла (IP-ядра, IP-блоки), добавленные для облегчения программирования некоторых распространенных функций. Об использовании IP речь идет на протяжении всей книги (см., например, главы 3,4,6, особенно подробно – в главе 7). – *Прим. ред.*

Глава 10. Свести все воедино. Глава посвящена добавлению в итоговый проект дополнительных периферийных устройств. В итоге благодаря подключению клавиатуры с помощью PS/2 будет создан комплексный проект, использующий VGA для отображения данных с микрофона и датчика температуры.

Глава 11. Темы повышенной сложности. В этой главе рассмотрены некоторые концепции SystemVerilog, которые были пропущены в других главах, но которые могут оказаться полезными. Продемонстрированы более продвинутые методы тестирования, и разобраны некоторые проблемы, которые могут возникать при проектировании, и способы их предотвращения.

КАК ПОЛУЧИТЬ МАКСИМАЛЬНУЮ ПОЛЬЗУ ОТ ЭТОЙ КНИГИ

Эта книга не предполагает наличия знаний о FPGA, логическом проектировании или программировании. Для того чтобы начать изучение, понадобится компьютер с операционной системой Windows или Linux. В первой главе содержатся инструкции по установке необходимого программного обеспечения.

Программное обеспечение / аппаратное обеспечение, рассматриваемые в этой книге	Требования к ОС
Xilinx Vivado 2020.1	Windows 10 или Linux (Centos 7.4-7.7 или Ubuntu 18.04 или 20.04)
Nexys A7 board	Windows 10 или Linux (Centos 7.4-7.7 или Ubuntu 18.04 или 20.04)

Если вы используете цифровую версию этой книги, рекомендуется набирать код самостоятельно или получить доступ к коду через репозиторий GitHub (ссылка доступна в следующем разделе). Это поможет избежать возможных ошибок, связанных с копированием и вставкой кода.

СКАЧАТЬ ФАЙЛЫ ПРИМЕРОВ КОДА

Файлы кодов примеров для этой книги можно загрузить из своей учетной записи на сайте www.packt.com. Если книга куплена в другом месте, можно посетить сайт поддержки www.packtpub.com/support и зарегистрироваться, чтобы получить файлы по электронной почте.

Файлы кодов примеров можно загрузить, выполнив следующие действия.

1. Авторизуйтесь или зарегистрируйтесь на сайте www.packt.com.
2. Выберите вкладку **Support**.
3. Перейдите по ссылке **Code Downloads**.
4. Введите название книги в поле поиска и следуйте инструкциям на экране.

После загрузки файла следует его разархивировать с помощью последней версии:

- WinRAR/7-Zip для Windows;
- Zipeg/iZip/UnRarX для Mac;
- 7-Zip/PeaZip для Linux.

Архив с кодами для книги также размещен на GitHub по адресу: <https://github.com/PacktPublishing/Learn-FPGA-Programming>.

В случае обновления кода он будет обновлен на существующем репозитории GitHub.

По адресу <https://github.com/PacktPublishing/> также размещены примеры кодов и проектов для других книг из обширного каталога Packt.

СКАЧАТЬ ЦВЕТНЫЕ ИЗОБРАЖЕНИЯ

PDF-файл с цветными изображениями скриншотов/вейвформ, используемых в этой книге, находится по адресу:

http://www.packtpub.com/sites/default/files/downloads/9781789805413_ColorImages.pdf.

ИСПОЛЬЗУЕМЫЕ ОБОЗНАЧЕНИЯ

В этой книге используется ряд обозначений в тексте.

Код в тексте: моноширинный шрифт обозначает кодовые служебные слова в тексте, имена переменных, операторы языка, цитаты из кода. Например, «сигнал тактовой частоты `sys_clk_i`».

Имена папок, имена файлов, расширения файлов, имена путей, URL-адреса, пользовательский ввод, некоторые названия модулей и ники в Twitter также выделяются моноширинным шрифтом: «файл `logic_ex.xpr`».

Блок кода задается следующим образом:

```
always @(posedge CK) begin
    stage = D;
    Q = stage;
end
```

Чтобы обратить внимание на определенную часть блока кода, соответствующие строки или элементы выделяются жирным шрифтом:

```
module dff (input wire D, CK, output logic Q);
    initial Q = 1;
    always_ff @(posedge CK) Q <= D;
endmodule
```

Любой ввод или вывод командной строки записывается следующим образом:

```
`timescale 1ps/100fs
```

Жирный шрифт: обозначает новый термин, важное слово или слова, которые читатель должен найти в интерфейсе среды разработки. Например, слова в меню или диалоговых окнах отображаются в тексте следующим образом: «В окне проекта нажмите правой кнопкой мыши и выберите **Add Module**».

Подсказки и важные замечания

Выглядят таким образом.

Контакты для связи

Отзывы читателей всегда приветствуются.

Обратная связь общего характера: если у вас есть вопросы по любому аспекту этой книги, укажите название книги в теме сообщения и напишите по адресу customercare@packtpub.com.

Ошибки: мы будем очень признательны тем внимательным читателям, которые обнаружат в русском переводе этой книги какие-либо ошибки или опечатки и сообщат о них на e-mail: a.romanov@hse.ru.

Пиратство: если вы встретите в интернете незаконные копии наших произведений в любой форме, мы будем благодарны, если вы сообщите нам адрес местонахождения или название сайта. Пожалуйста, свяжитесь с нами по адресу copyright@packt.com со ссылкой на материал.

Если вы заинтересованы в том, чтобы стать автором: если у вас есть тема, в которой вы разбираетесь, и вы заинтересованы в том, чтобы участвовать в создании книги, посетите сайт authors.packtpub.com.

Отзывы

Пожалуйста, оставьте отзыв. Если вы прочитали и использовали эту книгу, почему бы не оставить отзыв на сайте, на котором вы ее приобрели? Потенциальные читатели могут ознакомиться с вашим непредвзятым мнением и использовать его для принятия решения о покупке, мы в Packt можем понять, что вы думаете о наших продуктах, а наши авторы могут увидеть ваш отзыв о своей книге. Спасибо!

Для получения дополнительной информации о компании Packt посетите сайт packt.com.

Благодарности

Конюхов Владислав Игоревич
Вадим Селезнев
Pavel Gurev
Алексей Яковлев

Отдельная благодарность (внесли значительный вклад в исправления):

Александр Щербенко (телеграмм [@AlexSevere](https://t.me/AlexSevere),
e-mail: khy_alexandr@mail.ru)

Раздел 1

Введение в FPGA

В этом разделе вы получите представление о том, что такое Field Programmable Gate Array (FPGA), что за технология лежит в ее основе, а также познакомитесь с архитектурой Artix-7.

В эту часть книги входит:

- Глава 1 «Введение в FPGA и Xilinx Vivado».

Глава 1

Введение в FPGA и Xilinx Vivado

В данной главе мы рассмотрим **Field Programmable Gate Array (FPGA)** (программируемая логическая интегральная схема, ПЛИС) и технологию, лежащую в их основе. Эта технология позволяет таким компаниям, как Xilinx, производить перепрограммируемые микросхемы по технологии **Application Specific Integrated Circuit (ASIC)**. Затем мы узнаем, как использовать FPGA на примере решения простой задачи. Если вы хотите ускорить математически сложные вычисления, как в задачах машинного обучения или искусственного интеллекта, или просто хотите сделать несколько проектов для развлечения, таких как ретро-вычисления или воспроизведение устаревших видеоигровых машин (https://github.com/MiSTer-devel/Main_MiSTer/wiki), эта книга станет началом вашего путешествия. Сейчас самое время, чтобы погрузиться в эту область, пусть даже только в качестве хобби. Платы для разработки дешевы и многочисленны, и поставщики начали предоставлять свои инструменты бесплатно для недорогих и небольших проектов.

В этой книге мы собираемся реализовать несколько примеров проектов, которые познакомят вас с разработкой на FPGA, а кульминацией станет проект, способный управлять монитором VGA.

К концу этой главы вы получите хорошее представление о FPGA и ее компонентах.

Основные темы, которые мы рассмотрим в этой главе:

- что такое ASIC;
- как создаются FPGA;
- что входит в состав FPGA;
- как использовать инструменты Xilinx Vivado для проектирования, тестирования и реализации проектов на FPGA.

ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ

Для работы с примерами в этой главе вам потребуется следующее аппаратное и программное обеспечение.

Аппаратура

В отличие от языков программирования SystemVerilog является языком описания аппаратуры, и для того, чтобы действительно увидеть результаты реализации проектов по этой книге, вам понадобится плата FPGA для загрузки проектов. Для целей этой книги рекомендуется использовать одну из двух плат для разработки, которые легко доступны. Можно использовать и другую плату, если она у вас уже есть. Но некоторые ресурсы платы могут быть не идентичны, или вам может потребоваться изменить файл ограничений (xdc), чтобы получить доступ к ресурсам другой платы.

- Информация о Nexys A7: <https://store.digilentinc.com/nexys-a7-fpga-trainer-board-recommended-for-ece-curriculum/>.
- Информация об обучающей плате Basys 3 Artix-7 FPGA: <https://store.digilentinc.com/basys-3-artix-7-fpga-trainer-board-recommended-for-introductory-users/>.

Nexys A7 предпочтительнее, поскольку она имеет внешние интерфейсы, которые будут обсуждаться в последующих главах и дадут вам опыт взаимодействия с внешним оборудованием. Рекомендуется использовать версию 100T на тот случай, если вы будете амбициозны и захотите изучить больше, поскольку разница в цене относительно невелика, и у нее вдвое больше ресурсов. За исключением памяти DDR, на плате Basys 3 можно реализовать большинство проектов, хотя для некоторых из них может потребоваться приобретение интерфейсных плат PMOD.

Программное обеспечение

Для работы вам потребуется следующее программное обеспечение:

- <https://www.xilinx.com/products/design-tools/vivado.html>;
- файлы кода для всех примеров из этой главы можно найти в репозитории GitHub этой книги по адресу <https://github.com/PacktPublishing/Learn-FPGA-Programming/tree/master/CH1>.

Что такое ASIC?

Интегральные схемы специального назначения (ASIC) являются фундаментальными строительными блоками современной электроники – вашего ноутбука или ПК, телевизора, мобильного телефона, цифровых часов, практически всего, чем вы пользуетесь ежедневно. Это также фундаментальный строительный блок, на основе которого создается рассматриваемая нами FPGA. Если коротко, ASIC – это специально созданная микросхема, разработанная с использованием того же языка и методов, которые мы рассмотрим в этой книге.

FPGA появились благодаря тому, что технология создания ASIC следовала закону Мура (*Gordon E. Moore, Cramming more components onto integrated circuits, Electronics, Volume 38, Number 8, https://newsroom.intel.com/wp-content/uploads/sites/11/2018/05/moores-law-electronics.pdf*) – идее о том, что количество транзисторов в чипе удваивается каждые 2 года. Это позволило создавать очень дешевую электронику при массовом производстве изделий, содержащих ASIC, и также привело к распространению более дешевых FPGA.

Конец ознакомительного фрагмента.

Приобрести книгу можно

в интернет-магазине

«Электронный универс»

e-Univers.ru