

Содержание

<i>Отзывы на книгу «Цифровая схемотехника и архитектура компьютера. RISC-V»</i>	13
<i>Об авторах</i>	14
<i>Предисловие к русскому изданию</i>	15
<i>Предисловие от редактора русского перевода</i>	22
<i>Предисловие</i>	24

Глава 1 От нуля до единицы	31
1.1. План игры.....	31
1.2. Искусство управления сложностью.....	32
1.2.1. Абстракция.....	33
1.2.2. Конструкторская дисциплина.....	35
1.2.3. Три базовых принципа.....	36
1.3. Цифровая абстракция.....	38
1.4. Системы счисления.....	40
1.4.1. Десятичная система счисления.....	40
1.4.2. Двоичная система счисления.....	41
1.4.3. Шестнадцатеричная система счисления.....	43
1.4.4. Байт, полубайт и «весь этот джаз».....	45
1.4.5. Сложение двоичных чисел.....	46
1.4.6. Знак двоичных чисел.....	47
1.5. Логические элементы.....	53
1.5.1. Логический элемент НЕ.....	53
1.5.2. Буфер.....	54
1.5.3. Логический элемент И.....	54
1.5.4. Логический элемент ИЛИ.....	54
1.5.5. Другие логические элементы с двумя входными сигналами.....	55
1.5.6. Логические элементы с количеством входов больше двух.....	56
1.6. За пределами цифровой абстракции.....	57
1.6.1. Напряжение питания.....	57
1.6.2. Логические уровни.....	57
1.6.3. Допускаемые уровни шумов.....	58
1.6.4. Передаточная характеристика.....	59
1.6.5. Статическая дисциплина.....	60
1.7. КМОП-транзисторы.....	62
1.7.1. Полупроводники.....	63
1.7.2. Диоды.....	64
1.7.3. Конденсаторы.....	64
1.7.4. <i>n</i> -МОП- и <i>p</i> -МОП-транзисторы.....	65
1.7.5. Логический элемент НЕ на КМОП-транзисторах.....	69
1.7.6. Другие логические элементы на КМОП-транзисторах.....	69
1.7.7. Передаточный логический элемент.....	72
1.7.8. Псевдо- <i>n</i> -МОП-логика.....	72
1.8. Потребляемая мощность.....	73
1.9. Краткий обзор главы 1 и того, что нас ждет впереди.....	75
Упражнения.....	77
Вопросы для собеседования.....	89

Глава 2	Разработка комбинационной логики	91
2.1.	Введение	91
2.2.	Логические функции	95
2.2.1.	Терминология	95
2.2.2.	Дизъюнктивная форма	96
2.2.3.	Конъюнктивная форма	98
2.3.	Булева алгебра	99
2.3.1.	Аксиомы	100
2.3.2.	Теоремы одной переменной	100
2.3.3.	Теоремы с несколькими переменными	102
2.3.4.	Доказательство теорем булевой алгебры	104
2.3.5.	Упрощение логических уравнений	105
2.4.	От логики к логическим элементам	106
2.5.	Многоуровневая комбинационная логика	110
2.5.1.	Минимизация аппаратных затрат	111
2.5.2.	Перемещение инверсии	112
2.6.	Что такое X и Z?	115
2.6.1.	Недопустимое значение: X	115
2.6.2.	Третье состояние: Z	116
2.7.	Карты Карно	118
2.7.1.	Думайте об овалах	119
2.7.2.	Логическая минимизация на картах Карно	120
2.7.3.	Безразличные переменные	124
2.7.4.	Карты Карно: подведение итогов	124
2.8.	Базовые комбинационные блоки	125
2.8.1.	Мультиплексоры	125
2.8.2.	Дешифраторы	129
2.9.	Временные характеристики	131
2.9.1.	Задержка распространения и задержка реакции	131
2.9.2.	Импульсные помехи	136
2.10.	Заключение	139
	Упражнения	140
	Вопросы для собеседования	147
Глава 3	Разработка последовательностной логики	149
3.1.	Введение	149
3.2.	Зашелки и триггеры	150
3.2.1.	RS-триггер	151
3.2.2.	D-зашелка	154
3.2.3.	D-триггер	155
3.2.4.	Регистр	156
3.2.5.	Триггер с функцией разрешения	156
3.2.6.	Триггер с функцией сброса	158
3.2.7.	Разработка триггеров и защелок на транзисторном уровне	159
3.2.8.	Сравнение защелок и триггеров	160
3.3.	Разработка синхронных логических схем	161
3.3.1.	Некоторые проблемные схемы	161
3.3.2.	Синхронные последовательностные схемы	163
3.3.3.	Синхронные и асинхронные схемы	166
3.4.	Конечные автоматы	166
3.4.1.	Пример разработки конечного автомата	167
3.4.2.	Кодирование состояний	173

3.4.3. Автоматы Мура и Мили	176
3.4.4. Декомпозиция конечных автоматов	180
3.4.5. Восстановление конечных автоматов по электрической схеме	182
3.4.6. Конечные автоматы: подведение итогов	185
3.5. Синхронизация последовательностных схем	185
3.5.1. Динамическая дисциплина	187
3.5.2. Временные характеристики системы.....	188
3.5.3. Расфазировка тактовых сигналов	194
3.5.4. Метастабильность.....	197
3.5.5. Синхронизаторы.....	199
3.5.6. Вычисление времени разрешения	201
3.6. Параллелизм.....	205
3.7. Заключение.....	209
Упражнения	210
Вопросы для собеседования.....	218

Глава 4 Языки описания аппаратуры 221

4.1. Введение	221
4.1.1. Модули	222
4.1.2. Происхождение языков SystemVerilog и VHDL	222
4.1.3. Моделирование и синтез	224
4.2. Комбинационная логика.....	226
4.2.1. Битовые операторы	227
4.2.2. Комментарии и пробелы	229
4.2.3. Операторы сокращения	230
4.2.4. Условное присваивание	230
4.2.5. Внутренние переменные.....	233
4.2.6. Приоритет	235
4.2.7. Числа.....	235
4.2.8. Z-состояние и X-состояние.....	237
4.2.9. Манипуляция с битами.....	239
4.2.10. Задержки.....	239
4.3. Структурное моделирование.....	241
4.4. Последовательностная логика	245
4.4.1. Регистры.....	245
4.4.2. Регистры со сбросом.....	245
4.4.3. Регистры с сигналом разрешения	248
4.4.4. Группы регистров	249
4.4.5. Защелки.....	250
4.5. И снова комбинационная логика	251
4.5.1. Операторы case	254
4.5.2. Условный оператор (if)	256
4.5.3. Таблицы истинности с незначащими битами.....	259
4.5.4. Блокирующие и неблокирующие присваивания.....	260
4.6. Конечные автоматы.....	264
4.7. Типы данных	268
4.7.1. SystemVerilog.....	268
4.7.2. VHDL.....	269
4.8. Параметризованные модули.....	272
4.9. Тестбенч	275
4.10. Заключение.....	280
Упражнения	281
Упражнения для SystemVerilog	287

Упражнения для VHDL.....	289
Вопросы для собеседования.....	291

Глава 5 Цифровые функциональные узлы 293

5.1. Введение	293
5.2. Арифметические схемы	294
5.2.1. Сложение	294
5.2.2. Вычитание	302
5.2.3. Компараторы	303
5.2.4. Арифметико-логическое устройство.....	304
5.2.5. Схемы сдвига и циклического сдвига.....	309
5.2.6. Умножение	310
5.2.7. Деление	312
5.2.8. Дополнительная литература.....	313
5.3. Представление чисел	313
5.3.1. Числа с фиксированной запятой.....	314
5.3.2. Числа с плавающей запятой	315
5.4. Функциональные узлы последовательностной логики	319
5.4.1. Счетчики	319
5.4.2. Сдвиговые регистры	321
5.5. Матрицы памяти	324
5.5.1. Обзор матриц памяти	324
5.5.2. Динамическое ОЗУ (DRAM)	328
5.5.3. Статическое ОЗУ (SRAM).....	328
5.5.4. Площадь и задержки.....	329
5.5.5. Регистровые файлы	330
5.5.6. Постоянное запоминающее устройство.....	330
5.5.7. Реализация логических функций с использованием матриц памяти.....	332
5.5.8. Языки описания аппаратуры и память.....	333
5.6. Матрицы логических элементов	336
5.6.1. Программируемые логические матрицы	336
5.6.2. Программируемые пользователем вентильные матрицы	338
5.6.3. Схемотехника матриц.....	345
5.7. Заключение.....	346
Упражнения	347
Вопросы для собеседования.....	357

Глава 6 Архитектура 359

6.1. Предисловие	359
6.2. Язык ассемблера	362
6.2.1. Инструкции	362
6.2.2. Операнды: регистры, память и константы	364
6.3. Программирование	370
6.3.1. Порядок выполнения программы.....	371
6.3.2. Арифметические / логические инструкции.....	371
6.3.3. Ветвление программ.....	374
6.3.4. Условные операторы	377
6.3.5. Циклы.....	378
6.3.6. Массив	381
6.3.7. Вызовы функций.....	385
6.3.8. Псевдокоманды	398
6.4. Машинный язык	400

6.4.1. Инструкции типа <i>R</i>	401
6.4.2. Инструкции типа <i>I</i>	403
6.4.3. Инструкции типа <i>S/B</i>	404
6.4.4. Инструкции типа <i>U/J</i>	407
6.4.5. Кодирование констант.....	408
6.4.6. Режимы адресации.....	409
6.4.7. Расшифровываем машинные коды.....	411
6.4.8. Могущество хранимой программы.....	412
6.5. Камера, мотор! Компилируем, асемблируем и загружаем.....	413
6.5.1. Карта памяти.....	414
6.5.2. Директивы ассемблера.....	416
6.5.3. Компиляция.....	419
6.5.4. Трансляция.....	421
6.5.5. Компоновка.....	424
6.5.6. Загрузка.....	426
6.6. Добавочные сведения.....	426
6.6.1. Порядок байтов.....	426
6.6.2. Исключения.....	427
6.6.3. Команды для чисел со знаком и без знака.....	431
6.6.4. Команды для работы с числами с плавающей запятой.....	433
6.6.5. Сжатые инструкции.....	434
6.7. Эволюция архитектуры RISC-V.....	436
6.7.1. Базовые наборы команд и расширения RISC-V.....	436
6.7.2. Сравнение архитектур RISC-V и MIPS.....	437
6.7.3. Сравнение архитектур RISC-V и ARM.....	438
6.8. Живой пример: архитектура x86.....	439
6.8.1. Регистры x86.....	440
6.8.2. Операнды x86.....	440
6.8.3. Флаги состояния.....	442
6.8.4. Команды x86.....	442
6.8.5. Кодировка команд x86.....	444
6.8.6. Другие особенности x86.....	446
6.8.7. Архитектура x86: подведение итогов.....	447
6.9. Заключение.....	448
Упражнения.....	449
Вопросы для собеседования.....	462

Глава 7 Микроархитектура 465

7.1. Введение.....	465
7.1.1. Архитектурное состояние и система команд.....	466
7.1.2. Процесс разработки.....	466
7.1.3. Микроархитектуры RISC-V.....	469
7.2. Анализ производительности.....	470
7.3. Однотактный процессор.....	472
7.3.1. Пример программы.....	473
7.3.2. Однотактный тракт данных.....	473
7.3.3. Однотактный блок управления.....	482
7.3.4. Дополнительные команды.....	485
7.3.5. Анализ производительности.....	488
7.4. Многотактный процессор.....	490
7.4.1. Многотактный тракт данных.....	491
7.4.2. Многотактное устройство управления.....	497
7.4.3. Дополнительные команды.....	509

7.4.4. Анализ производительности.....	512
7.5. Конвейерный процессор.....	515
7.5.1. Конвейерный тракт данных	518
7.5.2. Конвейерное устройство управления	520
7.5.3. Конфликты	520
7.5.4. Анализ производительности.....	531
7.6. Разрабатываем процессор на HDL	533
7.6.1. Однотактный процессор	535
7.6.2. Универсальные строительные блоки	539
7.6.3. Тестбенч.....	542
7.7. Улучшенные микроархитектуры.....	547
7.7.1. Длинные конвейеры	548
7.7.2. Микрокоманды	549
7.7.3. Предсказание условных переходов.....	550
7.7.4. Суперскалярный процессор.....	552
7.7.5. Процессор с внеочередным выполнением команд	555
7.7.6. Переименование регистров	558
7.7.7. Многопоточность	560
7.7.8. Мультипроцессоры.....	561
7.8. Живой пример: эволюция микроархитектуры RISC-V	565
7.9. Заключение.....	569
Упражнения	571
Вопросы для собеседования.....	579

Глава 8 Системы памяти 581

8.1. Введение	581
8.2. Анализ производительности систем памяти	586
8.3. Кеш-память	588
8.3.1. Какие данные хранятся в кеш-памяти?.....	589
8.3.2. Как найти данные в кеш-памяти?.....	590
8.3.3. Какие данные заместить в кеш-памяти?	599
8.3.4. Улучшенная кеш-память	600
8.4. Виртуальная память.....	604
8.4.1. Трансляция адресов.....	607
8.4.2. Таблица страниц.....	609
8.4.4. Защита памяти	612
8.4.5. Стратегии замещения страниц	612
8.4.6. Многоуровневые таблицы страниц.....	613
8.5. Заключение.....	616
Эпилог	616
Упражнения	617
Вопросы для собеседования.....	624

Глава 9 Ввод/вывод во встраиваемых системах 626

9.1. Введение	626
9.2. Отображение ввода/вывода в пространство памяти.....	627
9.3. Ввод/вывод во встраиваемых системах	629
9.3.1. Плата RED-V	629
9.3.2. Система на кристалле FE310-G002.....	631
9.3.3. Цифровой ввод/вывод общего назначения.....	634
9.3.4. Драйверы устройств ввода/вывода.....	638
9.3.5. Последовательный ввод/вывод.....	642

9.3.6. Таймеры.....	659
9.3.7. Аналоговый ввод/ вывод	661
9.3.8. Прерывания.....	669
9.4. Другие внешние устройства микроконтроллера	674
9.4.1. Символьные ЖК-дисплеи	674
9.4.2. VGA-монитор	678
9.4.3. Беспроводная связь Bluetooth	684
9.4.4. Управление двигателями.....	686
9.5. Заключение.....	698

Приложение А. Реализация цифровых систем 699

A.1. Введение.....	699
A.2. Логические микросхемы серии 74xx	700
A.2.1. Логические элементы	700
A.2.2. Другие логические функции	701
A.3. Программируемая логика	703
A.3.1. PROM	704
A.3.2. Блоки PLA.....	705
A.3.3. FPGA	705
A.4. Заказные специализированные интегральные схемы	708
A.5. Работа с документацией	709
A.6. Семейства логических микросхем	714
A.7. Корпуса и монтаж интегральных схем.....	717
A.8. Линии передачи.....	721
A.8.1. Согласованная нагрузка	723
A.8.2. Нагрузка холостого хода.....	725
A.8.3. Нагрузка короткого замыкания.....	726
A.8.4. Рассогласованная нагрузка.....	726
A.8.5. Когда нужно применять модели линии передачи	729
A.8.6. Правильное подключение нагрузки к линии передачи	730
A.8.7. Вывод формулы для Z_0	731
A.8.8. Вывод формулы для коэффициента отражения.....	733
A.8.9. Линии передачи: подведение итогов	733
A.9. Экономика.....	735

Приложение В. Система команд RISC-V 738

Приложение С. Программирование на языке С 747

C.1. Введение.....	747
Краткий итог.....	749
C.2. Добро пожаловать в язык С.....	750
C.2.1. Структура программы на языке С	750
C.2.2. Запуск С-программы	751
Краткий итог.....	752
C.3. Компиляция.....	752
C.3.1. Комментарии.....	753
C.3.2. #define	753
C.3.3. #include.....	754
Краткий итог.....	755
C.4. Переменные.....	756
C.4.1. Базовые типы данных.....	756
C.4.2. Глобальные и локальные переменные	758

С.4.3. Инициализация переменных	759
Краткий итог.....	759
С.5. Операции.....	760
С.6. Вызовы функций.....	763
С.7. Управление последовательностью выполнения действий	765
С.7.1. Условные операторы.....	765
С.7.2. Циклы	767
Краткий итог.....	769
С.8. Другие типы данных	770
С.8.1. Указатели	770
С.8.2. Массивы	772
С.8.3. Символы.....	777
С.8.4. Строки символов.....	778
С.8.5. Структуры.....	780
С.8.6. Оператор typedef.....	781
С.8.7. Динамическое распределение памяти.....	783
С.8.8. Связные списки.....	784
Краткий итог.....	786
С.9. Стандартная библиотека языка С.....	786
С.9.1. <code>stdio</code>	787
С.9.2. <code>stdlib</code>	791
С.9.3. <code>math</code>	794
С.9.4. <code>string</code>	794
С.10. Компилятор и опции командной строки.....	795
С.10.1. Компиляция нескольких исходных с-файлов.....	795
С.10.2. Опции компилятора	795
С.10.3. Аргументы командной строки	796
С.11. Типичные ошибки	797
Дополнительная литература	801
Предметный указатель	803

Отзывы на книгу «Цифровая схемотехника и архитектура компьютера. RISC-V»

Харрис и Харрис детально описали устройство процессора RISC-V от электронных компонентов до микроархитектуры. Их ясные объяснения в сочетании с широким охватом темы дают полное представление как о цифровой схемотехнике, так и об архитектуре RISC-V. Это очень информативный и познавательный подход, поскольку у студентов есть отличная возможность запускать большие цифровые проекты на современных FPGA.

Дэвид А. Паттерсон, Калифорнийский университет в Беркли

Потрясающе, какие разнообразные знания авторы объединили в одной книге! По мере развития производства полупроводников значимость правильной разработки цифровых схем и компьютерной архитектуры будет только возрастать. Читатели найдут доступное и всестороннее рассмотрение обеих тем и после прочтения книги получают четкое понимание архитектуры набора команд RISC-V.

Эндрю Уотерман, SiFive

Мне доводилось видеть отличные учебники по цифровой схемотехнике и отличные учебники по компьютерным архитектурам – но этот учебник представляет собой и то, и другое! Он также уникален своей способностью формировать общую картину. Авторы начинают с азов, и это делает архитектуру RISC-V понятной. Упражнения к главам этой книги послужат отличным методическим ресурсом для университетских преподавателей.

Рой Кравиц, Государственный университет Портленда

Когда я впервые прочитал учебник по MIPS в 2008 году, то подумал, что это один из лучших учебников по компьютерной архитектуре. Я сразу начал использовать его в своих лекциях. Тринадцать лет спустя мне посчастливилось прочитать новое издание про RISC-V, и мое мнение осталось прежним: это отличная книга, очень понятная, исчерпывающая, с высоким образовательным потенциалом. Она полностью соответствует учебной программе, которую проходят студенты в области цифровой схемотехники и компьютерной архитектуры. Я с нетерпением жду возможности использовать этот учебник по архитектуре RISC-V в своих лекциях.

Даниэль Чавер Мартинес, Мадридский университет Комплутенсе

Об авторах

Дэвид Мани Харрис (David Money Harris) – доцент в колледже им. Харви Мадда (Harvey Mudd College). Получил ученую степень кандидата наук по электронике в Стэнфордском университете и степень магистра по электронике и информатике в Массачусетском технологическом институте (MIT). Перед Стэнфордом работал в компании Intel в качестве схемотехника и разработчика логики для процессоров Itanium и Pentium II. Впоследствии работал консультантом в Sun Microsystems, Hewlett-Packard, Evans & Sutherland и других компаниях.

Увлечения Дэвида включают в себя преподавание, разработку чипов и активный отдых на природе. В свободное от работы время занимается пешим туризмом, скалолазанием и альпинизмом. Особенно любит длинные прогулки с сыном Абрахамом, который родился, когда Дэвид начал работать над этой книгой. Дэвид имеет более десяти патентов и является автором трех других учебников по разработке чипов, а также двух путеводителей по горам Южной Калифорнии.

Сара Л. Харрис (Sarah L. Harris) – доцент в колледже им. Харви Мадда (Harvey Mudd College). Получила степени магистра и кандидата наук по электронике в Стэнфордском университете и степень бакалавра по электронике и вычислительной технике в университете Брайама Янга (Brigham Young University). Сара также работала в компаниях Hewlett-Packard, San Diego Supercomputer Center, Nvidia и исследовательском отделе компании Microsoft Research в Пекине.

Интересы Сары не ограничиваются преподаванием, изучением и разработкой новых технологий, она также любит путешествовать, увлекается виндсерфингом, скалолазанием и игрой на гитаре. Среди ее недавних начинаний можно отметить исследование в области интерфейсов, позволяющих разрабатывать цифровые электрические схемы простыми рисунками от руки, работу в качестве научного корреспондента для филиала Национального общественного радио (National Public Radio) и обучение кайтсерфингу. Сара говорит на четырех языках и собирается изучить еще несколько в ближайшем будущем.

Романов Александр Юрьевич – 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). В 2009 г. закончил магистратуру в Харьковском политехническом институте, работал в Киевском политехническом институте им. Сикорского. 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>.

Предисловие к русскому изданию

Вы держите в руках книгу, которая занимает на российском книжном рынке особое место. Если вы студент и хотите пройти собеседование в крупную электронную компанию на позицию проектировщика процессоров, нейроскорителей или сетевых микросхем, то самое лучшее, что вы можете сделать сейчас, — это прочитать данную книгу от корки до корки, одновременно выполняя упражнения на симуляторах и платах ПЛИС.

Когда мы говорим о собеседованиях, мы говорим о таких компаниях, как Apple, Intel, NVidia, а также о передовых российских проектировщиках процессоров Syntacore, «Элвис-НеоТек» и «Байкал Электроникс». В каждой из них вам дадут задания типа «напишите на доске дизайн простого арбитра на языке описания аппаратуры Verilog» или «объясните, как помогают производительности микропроцессора байпасы в его конвейере».

Конечно, мы не утверждаем, что изучение этого учебника гарантирует вам успех, но эта книга закладывает современную базу во всех областях, о которых вас будут спрашивать: цифровая логика и ее тайминг, арифметические блоки и конечные автоматы, архитектура (система команд) и микроархитектура (строение конвейера) процессора. С использованием того же самого языка SystemVerilog, который используют современные разработчики цифровых систем на рабочем месте (вам также могут встретиться блоки на языке VHDL, и он тоже есть в книге).

После этой книги вам, конечно, нужно будет сделать несколько учебных проектов и изучить по статьям в сети Интернет некоторые типы дизайнов, которых в книге нет (очереди FIFO, пересечение доменов тактовой частоты и т. д.). Совместно с этой книгой также рекомендуется читать еще одну — «Цифровой синтез: практический курс»¹. Она создана специально как дополнение к предыдущей версии книги Харрисов; в ближайшее время планируется ее переиздание, адаптированное под RISC-V. После этого вы будете готовы к бою. Никакая другая книга или комбинация книг на русском языке не поможет вам пройти эту начальную часть траектории эффективнее, чем «Цифровая схемотехника и архитектура компьютера: RISC-V» Дэвида Харриса и Сары Харрис.

Как возникла современная база проектирования

В 1980-е годы произошли две революции в проектировании цифровых микросхем. Первая революция была в маршруте проектирования. До конца 1980-х схемы рисо-

¹ Цифровой синтез: практический курс / под общ. ред. А. Ю. Романова, Ю. В. Панчула. М.: ДМК Пресс, 2020. https://dmkpress.com/catalog/electronics/circuit_design/978-5-97060-850-0/.

вали мышкой на экране, а с начала 1990-х их стали синтезировать из кода на языках описания аппаратуры Verilog и VHDL. Основные события:

- ▶ 1984 – Gateway Design Automation / Cadence изобретают язык описания аппаратуры Verilog;
- ▶ 1984 – Xilinx изобретает реконфигурируемые микросхемы ПЛИС / FPGA;
- ▶ 1986 – Optimal Solutions / Synopsys изобретают цифровой синтез;
- ▶ 1988–1992 – цифровой синтез внедряют в проектирование Apple, Sun, Nokia и др.;
- ▶ 1997–1999 – Lexra, MIPS, ARM начинают лицензировать процессорные ядра в виде IP-блоков (Intellectual Property – интеллектуальная собственность).

Вторая революция произошла в архитектуре и микроархитектуре процессоров. В 1970-х были популярны процессоры с двухуровневой организацией на основе так называемой технологии микропрограммирования. Команды процессора, видимые программисту, реализовывались на аппаратном уровне с помощью цепочек из слов (последовательностей битов в памяти) с сигналами контроля, так называемого микрокода. Такая организация позволяла создавать очень сложные системы команд, но ограничивала возможности по их параллельному выполнению.

В 1978 году группа исследователей в Стенфорде под руководством Джона Хеннесси задала себе вопрос: действительно ли нужны эти сложные команды, или их необходимость – просто маркетинговая иллюзия? Стенфордцы провели анализ большого количества пользовательских программ и пришли к выводу, что большинство используемых в программах команд – простые. И если тратить усилия не на усложнение цепочек микрокода, а на построение так называемого конвейера – структуры, в которой простые команды выполняются с перекрытием во времени, – то можно выполнять программы быстрее. Так появилась архитектура MIPS.

К похожим идеям пришла группа в Беркли под руководством Дэвида Паттерсона, которая в начале 1980-х создала архитектуры RISC I и RISC II, из которых выросла архитектура SPARC. В середине 1980-х появилась компания ARM, и за последующие десятилетия процессоры с новой организацией сначала завоевали рынок рабочих станций, а потом и бытовой электроники, сотовых телефонов и микроконтроллеров.

В конце 1980-х даже Intel, которая изначально делала процессоры на основе микрокода, стала вводить в Intel 486 конвейер, а к 1996 году построила процессор PentiumPro, в котором большинство команд на лету преобразовывались в простые команды, отправлявшиеся на конвейер в стиле RISC-процессоров. Хранимый в памяти микрокод остался только для сложных инструкций.

В начале 1990-х основатели концепции RISC-процессоров Джон Хеннесси и Дэвид Паттерсон опубликовали два учебника, которые стали бестселлерами:

- ▶ учебник начального уровня «Архитектура компьютера и проектирование компьютерных систем»
- ▶ и более сложный учебник «Компьютерная архитектура: количественный подход».

Эти учебники описывали архитектуру и микроархитектуру сначала на основе MIPS-образной архитектуры DLX, а потом стали использовать MIPS. К тому времени процессоры архитектуры MIPS уже использовались в компьютерах для голливудских спецэффектов, а потом и в домашней электронике.

В течение 1990-х американские университеты внедрили в учебный процесс книги Хеннесси и Паттерсона, курсы по языкам описания аппаратуры Verilog и VHDL, а также лабораторные работы на платах с микросхемами реконфигурируемой логики ПЛИС/FPGA, которые позволили строить студенческие процессоры без сложной процедуры заказа их на фабрике. Так выросло поколение студентов, которые разработали Apple iPhone, графические процессоры от NVidia, микросхемы для маршрутизаторов Cisco и Juniper и другие популярные устройства.

Что происходило в это время в России

Революции в цифровом синтезе и микроархитектуре процессоров по времени выпали на сложный период российской истории. Открытие советского рынка для иностранных компьютеров, коллапс СССР и недофинансирование вузов привели к тому, что в российском обществе перестали верить, что в России возможно проектирование конкурентоспособных чипов.

Долгое время группы разработчиков сохранялись только в компаниях, связанных с обороной и космосом, для проектирования чипов для космоса в таких организациях, как НИИСИ и НПЦ «Элвис». Российская команда, разработавшая процессор «Эльбрус», прототип которого при симуляции на Verilog показывал многообещающие результаты на вычислениях с плавающей запятой, попыталась в 2000 году получить финансирование у венчурных капиталистов Кремниевой долины, но вернулась в Россию.

В результате обучение компьютерной архитектуре во многих российских вузах стало описательным. Например, вузовские преподаватели стали использовать учебник Эндрю Таненбаума «Архитектура компьютеров», который был больше ориентирован на программистов, чем разработчиков процессоров. Что и понятно – Таненбаум получил известность как создатель операционной системы Minix, предшественницы Linux, а не разработчик процессора. Для микроархитектуры учебник использовал предыдущую технологическую базу (микрокод) и никак не был привязан к синтезу процессоров на языках описания аппаратуры. То есть студенты изучали системы команд и виды кеша для программистов, но не могли сделать процессор руками.

Учебники Паттерсона и Хеннесси были переведены на русский язык с большим опозданием, и в них не вошли приложения с описанием языков проектирования аппаратуры. Профессор Аркадий Поляков после работы в Кремниевой долине вернулся в Россию и издал в 2003 году учебник по Verilog, но в нем не было привязки к компьютерной архитектуре. Даже когда российские вузы делали лабораторные работы с ПЛИС, преподаватели часто выбирали разработку схемы с помощью рисования мышкой на экране, хотя в американских компаниях это перестали делать еще в начале 1990-х. В типичной вузовской методичке по цифровой электронике 2000-х годов шло качественное описание схем мультиплексоров и триггеров, а потом, пропустив

два уровня абстракции, студенты сразу изучали программирование микроконтроллеров. Не было учебника, который бы увязывал все эти элементы в одно целое.

История появления учебника «Цифровая схемотехника и архитектура компьютера»

Дэвид Харрис учился в MIT как раз тогда, когда произошла революция в маршруте проектирования конца 1980 – начала 1990-х годов. Вооруженный новыми методологиями, Дэвид пошел работать в Intel над процессором Pentium II. После этого защитил диссертацию в Стенфорде и стал преподавателем в Колледже Харви-Мадд в южной Калифорнии.

▶ http://pages.hmc.edu/harris/about/General_Resume.pdf.

Колледж Харви-Мадд не особо известен широкой публике, но находится среди топ-университетов по заработным платам выпускников, а также количеству выпускников, защищающих впоследствии диссертации. Еще Харви-Мадд известен проектами в области робототехники, которые они делают вместе с NASA. Иными словами, это практик высшего калибра.

▶ <https://www.monster.com/career-advice/article/colleges-that-get-most-pay-for-graduates>.

▶ <https://www.hmc.edu/about-hmc/2020/09/14/harvey-mudd-ranks-high-in-u-s-news-and-world-report-2021/>.

▶ <https://ti.arc.nasa.gov/news/ASR-hosts-Clinic-project/>.

Дэвид Харрис и его коллега Сара Харрис (они не родственники, а просто однофамильцы) в 2008 году написали первый вариант учебника, в котором в лаконичной и технически корректной форме изложили материал, который обычно входил в несколько учебников: цифровая логика, языки описания аппаратуры Verilog и VHDL, архитектура и микроархитектура компьютера, а также использование готовых чипов. Студенты получили возможность, используя только один учебник, начать с нуля, дойти до конструирования собственного небольшого процессора, реализующего подмножество архитектуры MIPS, а потом сравнить его работу с реальным микроконтроллером Microchip PIC32 на архитектуре MIPS.

Книга Харрисов появилась в России

В начале 2010-х годов в российской электронной индустрии наступило оживление. Зеленоградские компании «Элвис» и «Миландр» налаживали контакты с ARM и MIPS для лицензирования процессорных ядер, НИИСИ строил суперскалярное 64-битное MIPS-ядро, КМ211 разрабатывали процессоры для смарт-карт и налаживали контакты с тайваньской фабрикой TSMC. РОСНАНО финансировало проект компании «Элвис» в области умных камер и новую компанию «Байкал Электроникс».

Когда проблемы недостатка финансирования и изоляции российских компаний от международного рынка стали решаться, на первый план вышла проблема нехватки кадров. Хотя вузовские программы в МИЭТ и ИТМО старались поддерживать свои программы на уровне, компаниям приходилось обучать не только разработчиков схем на Verilog (на уровне RTL – Register Transfer Level), но и инженеров-верификаторов, которым нужно было создавать тесты и модели со знанием, что происходит в схеме, спроектированной на Verilog.

Поэтому когда в 2014 году появилась идея перевести на русский язык книгу Харрисов, ее поддержали сразу несколько человек и компаний. Преподаватели и аспиранты российских университетов МИФИ, ИТМО, ИТМиВТ, СПб ГУАП, украинских КНУ, КПИ, ХНУРЭ и ЧНТУ; сотрудники российских компаний МЦСТ, НИИСИ РАН, «Модуль», RusBITech, amperka.ru, Runtime Design Automation, «БиДжи»; русские инженеры американских и европейских компаний Imagination Technologies / MIPS Processors, AMD, Synopsys, Apple, eASIC, Cadence, NVidia, Marvell Semiconductor, университета Принстон – более 40 человек приняли участие в переводе, ревью, редактировании и корректировании как учебника, так и лекционных слайдов для него.

Перевод поддержала британская компания Imagination Technologies, которая в это время заключала сделки по лицензированию процессорных ядер MIPS и графических ядер PowerVR с российскими компаниями и была заинтересована в улучшении технического образования в России для налаживания долгосрочных бизнес-отношений с российскими партнерами. В издании книги также помогло eNano, образовательное отделение РОСНАНО, российского фонда, который вкладывал в микроэлектронные проекты.

После выхода первого онлайн-издания за дело взялось российское издательство «ДМК Пресс», которое выпустило второе издание Харрисов (использующее архитектуру MIPS) в бумажном виде, затем дополнение, которое применяет архитектуру ARM. Книга «Цифровая схемотехника и архитектура компьютера» стала настолько популярна, что ее начали использовать в ведущих российских вузах. Единственного, чего ей не хватало, это полноценного практического курса, который бы мог дополнить основной материал лабораторными работами. В 2019 г. такой курс был создан. Под эгидой МИЭМ НИУ ВШЭ была собрана большая команда преподавателей и разработчиков из СНГ и США, написавшие книгу «Цифровой синтез: практический курс» под редакцией А. Ю. Романова и Ю. В. Панчула. Книга хороша тем, что она раскрывает и дополняет материал книги Харрисов, а также поддержана репозитарием с исходными кодами всех примеров, приведенных в ней, и адаптирована под выполнение лабораторных работ на дешевых отладочных платах с ПЛИС.

И вот, наконец, ввиду все большего распространения архитектуры RISC-V, появилось новое издание книги «Цифровая схемотехника и архитектура компьютера».

Почему RISC-V?

Лицензируемые ядра RISC-процессоров совершили еще одну революцию в конце 1990 – начале 2000-х годов, когда ARM стал сердцем сотовых телефонов от Nokia и Ericsson, а MIPS стали использовать в телевизорах Sony, игровых приставках

и даже роботах. К компаниям ARM и MIPS присоединились несколько конкурентов, в частности ARC и Tensilica, которые образовали так называемую индустрию полупроводниковой интеллектуальной собственности, semiconductor IP, общим размером в несколько миллиардов долларов.

Помимо разработчиков центральных процессоров в эту индустрию вошли Imagination Technologies – компания, которая спроектировала графический процессор PowerVR для ранних Apple iPhone, затем разработчик процессора для обработки сигналов CEVA и уже в наше время компании, которые выпускают ускорители нейросетевых вычислений.

ARM и MIPS получали доход двумя способами:

- 1) продаж лицензий на процессорные ядра – фактически на использование сотни тысяч строк на Verilog, написанных инженерами ARM и MIPS, внутри систем на кристалле заказчика. Примерами таких компаний стали Microchip, которая лицензировала ядро MIPS M4K для микроконтроллеров PIC32, и ST Microelectronics, которая лицензировала ядра ARM Cortex M для линейки микроконтроллеров STM32;
- 2) продаж так называемой архитектурной лицензии – права на создание процессора собственной микроархитектуры. Инженеры компании-покупателя архитектурной лицензии создавали собственную микроархитектуру и могли разрабатывать код на Verilog сами, но их ядро делалось совместимым по архитектуре (системе команд) с ARM или MIPS. Последним примером такого лицензиата является компания Apple, которая создала свое ARM-совместимое ядро для системы на кристалле Apple M1.

Хотя разделение компаний на разработчиков IP-блоков и разработчиков систем на кристалле помогло развить индустрию в 1990–2000-е годы, не все в этой схеме было идеальным.

- ▶ Во-первых, многие компании были недовольны условиями и политикой лицензирования как ядер, так и архитектуры. Особенно сильное негативное впечатление на индустрию произвел судебный процесс MIPS против Lexra в 1999 году, в результате которого пионер IP-лицензирования компания Lexra обанкротилась из-за довольно мелкого нарушения патента на редко используемые инструкции невыравненного обмена с памятью (<https://www.eetimes.com/lexra-quits-ip-cores-business-in-deal-with-mips/>).
- ▶ Во-вторых, контроль архитектуры со стороны коммерческих компаний не нравился университетским исследователям. Хотя MIPS активно использовался в учебниках, а ARM давал гранты университетам, но ученые были недовольны перспективой получения писем от корпоративных юристов из-за какого-нибудь созданного ими экспериментального процессора.
- ▶ Наконец, во всех RISC-архитектурах скопились разные черты, которые когда-то казались хорошими идеями, но стали тормозом прогресса при усложнении процессоров, повышении частоты, введении микроархитектуры с внеочередным выполнением команд, переменной длины инструкций и предсказателями перехода. У SPARC такой чертой были регистровые окна, у MIPS – слоты отло-

женного ветвления, у ARM – условное выполнение инструкций. Нужна была ревизия мира RISC-процессоров.

И этой ревизией стала RISC-V – архитектура, созданная в 2010 году группой того же Дэвида Паттерсона из Университета Калифорнии в Беркли, который написал два учебника и стоял у истоков архитектуры SPARC. Группа RISC-V не только объединила опыт процессорных компаний за предыдущие 30 лет, но и вступила в партнерство с Linux Foundation и многими крупными компаниями – Google, AMD, Western Digital.

Когда вы используете архитектуру RISC-V для проектирования своего процессора, вам не нужно платить за архитектурную лицензию. При этом сами вы можете получать за свой процессор деньги: продавать его как IP-блок, систему на кристалле или производить на его основе чипы. Вы также можете решить сделать бесплатный процессор с открытым кодом на Verilog для исследователей – это тоже поощряется сообществом вокруг архитектуры RISC-V.

Сейчас RISC-V может сыграть большую роль в становлении российской электроники. Российские компании CloudBEAR и Syntacore (приобретенная компанией «Ядро») работают над процессорами собственной микроархитектуры, совместимыми по системе команд с архитектурой RISC-V. Это идеальная комбинация, которая позволяет разрабатывать свои процессоры и конкурировать по производительности, энергопотреблению и набору расширений с производителями на мировом рынке, одновременно сохраняя программную совместимость со всеми программами, которые создаются для экосистемы RISC-V во всем мире. К таким программам относятся компиляторы, операционные системы и прикладные программы – от программ для миниатюрных чипов для интернета вещей до мобильных устройств, автомобильной электроники, десктопов и суперкомпьютеров.

Подводя итог

Предыдущие издания учебника Харрисов уже помогли исправить серьезный дисбаланс в преподавании цифровой электроники в России, который возник еще в 1990-е годы. Книга также стала отправной точкой для создания курса лабораторных работ на ПЛИС под эгидой МИЭМ НИУ ВШЭ, онлайн-курсов от РОСНАНО и семинаров на ChipEXPO в Сколково. Новое же издание учебника Харрисов выходит как раз тогда, когда в России разворачиваются амбициозные проекты по созданию высокопроизводительных процессорных ядер, которые совместимы с открытой международной архитектурой RISC-V и при этом спроектированы в России.

Мы ожидаем, что читатели этой книги станут топ-разработчиками и бизнес-лидерами российской электронной промышленности и помогут ей занять место в мире, которое соответствует российским традициям достижений в математике, физике, атомных и космических технологиях.

Юрий Панчул,
инженер-проектировщик CPU, GPU и сетевых микросхем,
с опытом работы в MIPS Technologies, Imagination Technologies,
Juniper Networks и Samsung Advanced Computing Lab

Предисловие от редактора русского перевода

Дорогие читатели,
перед вами – уникальное издание.

После распада СССР в русскоязычной образовательной среде возник вакуум, интеллектуальный рынок быстро захватили иностранные САПРы, а на первых ролях оказалась западная электроника. В сфере образования курсы по цифровой электронике нередко сводились к локальным курсам под конкретные платы, существовавшие в том или ином университете, а во многих случаях (в том числе из-за отсутствия надлежащего оборудования) учебный процесс превращался в сугубо теоретическое изучение дисциплины. Об этом явлении совершенно справедливо написал Юрий Панчул: <https://habr.com/ru/post/589091/> («Почему книга Эндрю Таненбаума “Архитектура компьютера” вредна для образования»), чью точку зрения я полностью поддерживаю, поскольку сам учился по книге Таненбаума.

К счастью, в последующие годы картина начала меняться. Университеты стали богаче, появилась возможность приобретения необходимого оборудования, оно стало доступно и для личного пользования; началось оживление в российских компаниях, и обозначилась все большая потребность в специалистах по цифровой электронике. Все эти обстоятельства сформировали запрос на появление массовых учебных материалов на русском языке.

Звезды сошлись в 2016 году: для написания книги, по инициативе Юрия Панчула, удалось собрать вместе специалистов из ряда университетов и международных компаний, получить финансирование от Imagination technologies и найти понимание ведущего издательства в этой сфере – «ДМК Пресс». Основой для написания материала будущего издания стала великолепная книга D. M. Harris, S. L. Harris «Digital Design and Computer Architecture», де-факто являющаяся стандартом при изучении компьютерной архитектуры и цифрового синтеза во многих зарубежных университетах.

Так появилось первое издание книги «Цифровая схемотехника и архитектура компьютера». Несмотря на то что перевод был в некоторой степени аматорским и в первом издании обнаружилось некоторые ошибки и неточности, книга стала бестселлером и разошлась тиражом в не одну тысячу экземпляров. Поскольку перевод первого издания книги осуществлялся без моего участия, мною в учебном процессе использовалась ее английская версия. Но как только появился русский перевод, он был сразу внедрен в учебные курсы, и на нем выросло несколько поколений студентов.

Дальше – больше. «Цифровой синтез» издали в цветном варианте, потом вышло дополнение по архитектуре ARM, а также была выпущена отдельная книга, допол-

няющая основную: «Цифровой синтез: практический курс», которая представляет собой компьютерный практикум, построенный на дешевых и доступных платах ПЛИС, при этом был сделан акцент на изучении языка Verilog.

Следует отметить, что время не стоит на месте: архитектура MIPS, которой посвящена исходная книга, все больше теряет свои позиции и вытесняется RISC-V, объединяющей в себе новые подходы к проектированию RISC-процессоров и принципы открытой разработки. Таким образом, появилась насущная необходимость в переводе нового издания книги D. M. Harris, S. L. Harris «Digital Design and Computer Architecture. RISC-V Edition». Хотя новое издание в целом ряде глав пересекается с исходной книгой, другие главы, посвященные архитектуре RISC-V, – полностью новые. Можно было пойти при этом по одному из путей: либо выпустить дополнение к основной книге (как это было сделано для архитектуры ARM), либо перевыпустить книгу полностью. Чтобы не нарушать целостность произведения, было принято решение пойти по второму пути, попутно исправив допущенные ранее ошибки и тщательно переработав старые главы. Результат этого труда – перед вами.

Данная книга будет полезна всем студентам (таких вузов, как, например, МИЭТ или ИТМО), изучающим архитектуру компьютера и языки описания аппаратуры, а также всем разработчикам, которым необходимо понимать, как устроен микропроцессор / микроконтроллер или другая цифровая схема изнутри.

Александр Юрьевич Романов,

научный редактор книги,

к. т. н., доцент МИЭМ НИУ ВШЭ,

преподаватель курсов «Проектирование систем на кристалле»

и «Системное проектирование цифровых устройств»,

г. Москва, Россия

Предисловие

Эта книга уникальна тем, что описывает цифровую схемотехнику с точки зрения компьютерной архитектуры, начиная с двоичной логики и заканчивая проектированием микропроцессора.

Мы считаем, что проектирование микропроцессора является своеобразным обрядом посвящения для студентов инженерных и компьютерных специальностей. Внутренняя работа микропроцессора кажется почти магической для непосвященных, но при подробном объяснении оказывается простой и доступной для понимания. Проектирование цифровых схем само по себе является захватывающим предметом. Программирование на языке ассемблера позволяет понять внутренний язык, на котором говорит микропроцессор. Микроархитектура, в свою очередь, является тем связующим звеном, которое объединяет эти предметы воедино.

Первые две версии этого набирающего популярность учебника описывают архитектуры MIPS и ARM. MIPS – одна из исходных вычислительных архитектур с сокращенным набором команд (Reduced Instruction Set Computing, RISC), простая в изучении и применении. Значимость архитектуры MIPS сложно переоценить, поскольку она вдохновила разработчиков на создание последующих архитектур, включая RISC-V. Архитектура ARM стала очень популярной за последние несколько десятилетий благодаря своей эффективности и богатой экосистеме. Было продано более 50 млрд процессоров ARM, и более 75 % людей на планете используют продукты с этими процессорами.

В течение последнего десятилетия архитектура RISC-V становится все более значимой как с образовательной, так и с коммерческой точки зрения. Будучи широко распространенной компьютерной архитектурой с открытым исходным кодом, RISC-V сочетает простоту MIPS с гибкостью и функциональностью современных процессоров.

С познавательной точки зрения использование трех версий учебника – MIPS, ARM и RISC-V – полностью идентично. Архитектура RISC-V имеет ряд особенностей, включающих расширяемость и компактный формат представления инструкций, которые повышают ее эффективность, но немного увеличивают сложность. Три микроархитектуры также похожи, а архитектуры MIPS и RISC-V имеют много общего. Мы планируем переиздавать версии учебника про MIPS, ARM и RISC-V до тех пор, пока эти архитектуры востребованы рынком.

Особенности книги

Эта книга содержит ряд особенностей.

Одновременное использование языков SystemVerilog и VHDL

Языки описания аппаратуры (hardware description languages, HDL) находятся в основе современных методов проектирования сложных цифровых систем. К сожалению, разработчики делятся на две примерно равные группы, использующие два

Конец ознакомительного фрагмента.

Приобрести книгу можно

в интернет-магазине

«Электронный универс»

e-Univers.ru