

Оглавление

Похвальные отзывы на книгу «Цифровая схемотехника и архитектура компьютера»	15
Об авторах	17
Предисловие к изданию на русском языке	18
Благодарности участникам проекта	22
Предисловие	24
Особенности книги	24
Материалы в Интернете	26
Как использовать программный инструментарий в учебном курсе	27
Опечатки	28
Признательность за поддержку	28
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 вентиль НЕ	53
1.5.2. Буфер	54
1.5.3. Логический вентиль И	54
1.5.4. Логический вентиль 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. п-МОП- и р-МОП-транзисторы.....	65
1.7.5. Логический вентиль НЕ на КМОП-транзисторах	69
1.7.6. Другие логические вентили на КМОП-транзисторах	69
1.7.7. Передаточный логический вентиль	72
1.7.8. Псевдо п-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 Проектирование комбинационной логики 91

2.1. Введение	91
2.2. Булевы уравнения.....	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. Правда обо всем этом	104
2.3.5. Упрощение уравнений.....	105
2.4. От логики к логическим элементам	106
2.5. Многоуровневая комбинационная логика	110
2.5.1. Минимизация аппаратуры.....	111
2.5.2. Перемещение инверсии.....	112
2.6. Что за 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. Подводя итоги	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. Резюме	139
Упражнения	140
Вопросы для собеседования	147

Глава 3	Проектирование последовательной логики	149
3.1.	Введение	149
3.2.	Защелки и триггеры	150
3.2.1.	RS-триггер	151
3.2.2.	D-защелка	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.	Проектирование триггеров и защелок на транзисторном уровне	159
3.2.8.	Общий обзор	160
3.3.	Проектирование синхронных логических схем	161
3.3.1.	Некоторые проблемные схемы	161
3.3.2.	Синхронные последовательные схемы	163
3.3.3.	Синхронные и асинхронные схемы	166
3.4.	Конечные автоматы	166
3.4.1.	Пример проектирования 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.	Обзор конечных автоматов	185
3.5.	Синхронизация последовательных 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.	Резюме	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.	Симуляция 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 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. Операторы 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. Среда тестирования	275
4.10. Резюме	280
Упражнения	281
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. АЛУ	304
5.2.5. Схемы сдвига и циклического сдвига	306
5.2.6. Умножение	308
5.2.7. Деление	309
5.2.8. Дополнительная литература	311
5.3. Представление чисел	311
5.3.1. Числа с фиксированной точкой	311
5.3.2. Числа с плавающей точкой	312
5.4. Функциональные узлы последовательностной логики	317
5.4.1. Счетчики	317
5.4.2. Сдвигающие регистры	318
5.5. Матрицы памяти	321
5.5.1. Обзор	321
5.5.2. Динамическое ОЗУ (DRAM)	324
5.5.3. Статическое ОЗУ (SRAM)	325

5.5.4. Площадь и задержки.....	326
5.5.5. Регистровые файлы	327
5.5.6. Постоянное запоминающее устройство.....	327
5.5.7. Реализация логических функций с использованием матриц памяти.....	330
5.5.8. Языки описания аппаратуры и память.....	331
5.6. Матрицы логических элементов	332
5.6.1. Программируемые логические матрицы	333
5.6.2. Программируемые пользователем матрицы логических элементов	335
5.6.3. Схемотехника матриц.....	340
5.7. Резюме.....	342
Упражнения	343
Вопросы для собеседования	353

Глава 6 Архитектура **355**

6.1. Предисловие	355
6.2. Язык ассемблера	357
6.2.1. Инструкции	358
6.2.2. Операнды: регистры, память и константы	360
6.3. Машинный язык	367
6.3.1. Инструкции типа <i>R</i>	368
6.3.2. Инструкции типа <i>I</i>	369
6.3.3. Инструкции типа <i>J</i>	371
6.3.4. Расшифровываем машинные коды.....	371
6.3.5. Могущество хранимой программы	372
6.4. Программирование	373
6.4.1. Арифметические / логические инструкции.....	374
6.4.2. Переходы.....	378
6.4.3. Условные операторы	381
6.4.4. Зацикливаемся	383
6.4.5. Массивы.....	385
6.4.6. Вызовы функций.....	390
6.5. Режимы адресации.....	400
6.6. Камера, мотор! Компилируем, ассемблируем и загружаем	404
6.6.1. Карта памяти	404
6.6.2. Трансляция и запуск программы	406
6.7. Добавочные сведения	409
6.7.1. Псевдокоманды	410
6.7.2. Исключения.....	411
6.7.3. Команды для чисел со знаком и без знака.....	413
6.7.4. Команды для работы с числами с плавающей точкой	415
6.8. Живой пример: архитектура x86.....	417
6.8.1. Регистры x86	418
6.8.2. Операнды x86	418
6.8.3. Флаги состояния	420
6.8.4. Команды x86.....	421
6.8.5. Кодировка команд x86	423
6.8.6. Другие особенности x86	425

6.8.7. Оглядываясь назад	425
6.9. Резюме	426
Упражнения	427
Вопросы для собеседования	437
Глава 7 Микроархитектура	439
7.1. Введение	439
7.1.1. Архитектурное состояние и система команд	440
7.1.2. Процесс разработки	441
7.1.3. Микроархитектуры MIPS	443
7.2. Анализ производительности	444
7.3. Однотактный процессор	446
7.3.1. Однотактный тракт данных	446
7.3.2. Однотактное устройство управления	452
7.3.3. Дополнительные команды	456
7.3.4. Анализ производительности	458
7.4. Многотактный процессор	460
7.4.1. Многотактный тракт данных	460
7.4.2. Многотактное устройство управления	467
7.4.3. Дополнительные команды	474
7.4.4. Анализ производительности	478
7.5. Конвейерный процессор	479
7.5.1. Конвейерный тракт данных	482
7.5.2. Конвейерное устройство управления	484
7.5.3. Конфликты	484
7.5.4. Дополнительные команды	496
7.5.5. Анализ производительности	496
7.6. Пишем процессор на HDL	498
7.6.1. Однотактный процессор	499
7.6.2. Универсальные строительные блоки	504
7.6.3. Тестовое окружение	506
7.7. Исключения	510
7.8. Улучшенные микроархитектуры	513
7.8.1. Длинные конвейеры	514
7.8.2. Предсказание условных переходов	516
7.8.3. Суперскалярный процессор	518
7.8.4. Процессор с внеочередным выполнением команд	521
7.8.5. Переименование регистров	524
7.8.6. SIMD	525
7.8.7. Многопоточность	526
7.8.8. Симметричные мультипроцессоры	528
7.8.9. Гетерогенные мультипроцессоры	529
7.9. Живой пример: микроархитектура x86	532
7.10. Резюме	539
Упражнения	541
Вопросы для собеседования	546

Глава 8 Иерархия памяти и подсистема ввода-вывода	549
8.1. Введение	549
8.2. Анализ производительности систем памяти	554
8.3. Кэш-память	556
8.3.1. Какие данные хранятся в кэш-памяти?	557
8.3.2. Как найти данные в кэш-памяти?	558
8.3.3. Какие данные заместить в кэш-памяти?	567
8.3.4. Улучшенная кэш-память	569
8.3.5. Эволюция кэш-памяти процессоров MIPS	573
8.4. Виртуальная память	573
8.4.1. Трансляция адресов	576
8.4.2. Таблица страниц	578
8.4.3. Буфер ассоциативной трансляции	580
8.4.4. Защита памяти	582
8.4.5. Стратегии замещения страниц	583
8.4.6. Многоуровневые таблицы страниц	584
8.5. Системы ввода-вывода	586
8.6. Ввод-вывод во встроенных системах	588
8.6.1. Микроконтроллер PIC32MX675F512H	589
8.6.2. Цифровой ввод-вывод общего назначения	594
8.6.3. Последовательный ввод-вывод	596
8.6.4. Таймеры	610
8.6.5. Прерывания	612
8.6.6. Аналоговый ввод-вывод	614
8.6.7. Другие внешние устройства микроконтроллера	621
8.7. Интерфейсы ввода-вывода персональных компьютеров	644
8.7.1. USB	646
8.7.2. PCI и PCI Express	647
8.7.3. Память DDR3	648
8.7.4. Сеть	648
8.7.5. SATA	649
8.7.6. Подключения к ПК	649
8.8. Живой пример: системы памяти и ввода-вывода семейства x86	652
8.8.1. Системы кэш-памяти процессоров семейства x86	652
8.8.2. Виртуальная память x86	655
8.8.3. Программируемый ввод-вывод x86	656
8.9. Резюме	656
Эпилог	657
Упражнения	658
Вопросы для собеседования	665
Приложение А Реализация цифровых систем	667
А.1. Введение	667
А.2. Логические микросхемы серии 74xx	668
А.2.1. Логические элементы	668
А.2.2. Другие логические функции	669

А.3. Программируемая логика.....	671
А.3.1. PROM.....	672
А.3.2. Блоки PLA.....	673
А.3.3. FPGA.....	673
А.4. Заказные специализированные интегральные схемы.....	676
А.5. Работа с документацией.....	677
А.6. Семейства логических элементов.....	682
А.7. Корпуса и монтаж интегральных схем.....	685
А.8. Линии передачи.....	690
А.8.1. Согласованная нагрузка.....	691
А.8.2. Нагрузка холостого хода.....	693
А.8.3. Нагрузка короткого замыкания.....	694
А.8.4. Рассогласованная нагрузка.....	695
А.8.5. Когда нужно применять модели линии передачи.....	697
А.8.6. Правильное подключение нагрузки к линии передачи.....	698
А.8.7. Вывод формулы для Z_0	700
А.8.8. Вывод формулы для коэффициента отражения.....	701
А.8.9. Подводя итог.....	702
А.9. Экономика.....	704

Приложение В Инструкции архитектуры MIPS 707

Приложение С Программирование на языке Си 713

С.1. Введение.....	713
С.2. Добро пожаловать в язык Си.....	716
С.2.1. Структура программы на языке СИ.....	716
С.2.2. Запуск Си-программы.....	717
С.3. Компиляция.....	718
С.3.1. Комментарии.....	719
С.3.2. #define.....	719
С.3.3. #include.....	720
С.4. Переменные.....	721
С.4.1. Базовые типы данных.....	722
С.4.2. Глобальные и локальные переменные.....	724
С.4.3. Инициализация переменных.....	725
С.5. Операции.....	726
С.6. Вызовы функций.....	729
С.7. Управление последовательностью выполнения действий.....	731
С.7.1. Условные операторы.....	731
С.7.2. Циклы.....	733
С.8. Другие типы данных.....	736
С.8.1. Указатели.....	736
С.8.2. Массивы.....	738
С.8.3. Символы.....	743
С.8.4. Строки символов.....	744
С.8.5. Структуры.....	745

С.8.6. Оператор typedef.....	747
С.8.7. Динамическое распределение памяти.....	748
С.8.8. Связные списки.....	749
С.9. Стандартная библиотека языка С.....	752
С.9.1. stdio.....	753
С.9.2. stdlib.....	757
С.9.3. math.....	759
С.9.4. string.....	760
С.10. Компилятор и опции командной строки.....	760
С.10.1. Компиляция нескольких исходных С-файлов.....	761
С.10.2. Опции компилятора.....	761
С.10.3. Аргументы командной строки.....	762
С.11. Типичные ошибки.....	762
Литература для дальнейшего изучения	769
Дополнительная информация	771
Предметный указатель	772

Похвальные отзывы на книгу «Цифровая схемотехника и архитектура компьютера»

Авторы книги вывели преподавание предмета на качественно иной уровень, создав более доступный для понимания и наглядный учебник, чем «Устройство и проектирование компьютеров» («Computer Organization and Design»), и описав в нем в деталях, как спроектировать микропроцессор архитектуры MIPS с помощью языков SystemVerilog и VHDL. Текст окажется особенно полезным для студентов, которые в процессе обучения столкнутся с разработкой больших цифровых систем на современных ПЛИС.

Дэвид А. Паттерсон, Калифорнийский Университет в Беркли¹

Книга дает свежий взгляд на старую дисциплину. Многие учебники напоминают неухоженные заросли кустарника, но авторы данного учебника сумели отстричь засохшие ветви, сохранив основы и представив их в современном контексте. Эта книга поможет студентам справиться с техническими испытаниями завтрашнего дня.

Джим Френзел, Университет Айдахо

Книга написана в информативном приятном для чтения стиле. Материал представлен на хорошем уровне для введения в проектирование компьютеров и содержит множество полезных диаграмм. Комбинационные схемы, микроархитектура и системы памяти изложены особенно хорошо.

Джеймс Пинтер-Люк, Колледж им. Дональда Маккенны, Клермонт

Харрис и Харрис написали очень ясную и легкую для понимания книгу. Упражнения хорошо разработаны, а примеры из реальной практики являются замечательным дополнением. Длинные и вводящие в заблуждение объяснения, часто встречающиеся в подобных книгах, здесь отсутствуют. Очевидно, что авторы посвятили много времени и усилий созданию доступного текста. Я настоятельно рекомендую книгу.

Пейи Чжао, Университет Чепмена

Харрис и Харрис написали первую книгу, которая успешно совмещает проектирование цифровых систем и архитектуру компьютеров. Книга – долгожданное учебное пособие, в котором подробно рассматривается проектирова-

¹ Дэвид Паттерсон – соавтор вышеупомянутого учебника «Устройство и проектирование компьютеров». – Прим. перевод

ние цифровых систем и в фантастических деталях объясняется архитектура MIPS. Я настоятельно рекомендую эту книгу.

Джеймс Э. Стайн, Мл., Университет Оклахомы

Это великолепная книга. Авторы органично связывают все важные в проектировании микропроцессоров элементы – транзисторы, схемы, логические элементы, конечные автоматы, память, арифметические блоки – и получают компьютерную архитектуру. Этот текст является незаменимым руководством для понимания, как последовательно разрабатывать сложные системы.

Джеха Ким, Рамбус Инк

Это очень хорошо написанная книга, которая будет полезна как молодым инженерам, изучающим предмет впервые, так и опытным инженерам, которые смогут использовать ее в качестве справочника. Я настоятельно рекомендую ее.

А. Утку Дирил, Корпорация Энвидиа

Об авторах

Дэвид Мани Харрис (David Money Harris) – доцент в колледже им. Харви Мадда (Harvey Mudd College). Он получил ученую степень кандидата наук по электронике в Стэнфордском университете и степень магистра по электронике и информатике в Массачусетском технологическом институте (MIT). Перед Стэнфордом он работал в компании Интел (Intel) в качестве схемотехника и разработчика логики для процессоров Итаниум и Пентиум 2 (Itanium and Pentium II). Впоследствии он работал консультантом в Сан Майкросистемз (Sun Microsystems), Хьюлетт-Паккард (Hewlett-Packard), Эванс энд Сазерленд (Evans & Sutherland) и других компаниях.

Увлечения Дэвида включают в себя преподавание, разработку чипов и активный отдых на природе. В свободное от работы время он занимается пешим туризмом, скалолазанием и альпинизмом. Особенно он 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 проектированию чипов, а также двух путеводителей по горам Южной Калифорнии.

Сара Л. Харрис (Sarah L. Harris) – доцент в колледже им. Харви Мадда (Harvey Mudd College). Она 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 Хьюлетт-Паккард, Суперкомпьютерном Центре Сан-Диего (San Diego Supercomputer Center), Энвидиа (Nvidia) и исследовательском отделе компании Майкрософт (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 исследования в области интерфейсов, позволяющих проектировать 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.

Предисловие к изданию на русском языке

История развития вычислительной техники в СССР насчитывает практически столько же лет, как и в США, так как разработка быстродействующих компьютеров являлась неотъемлемой частью технологического соперничества двух сверхдержав. Вскоре после разработки первого американского компьютера общего назначения ENIAC (1943–1947), в СССР была разработана МЭСМ (Малая электронная счетная машина, 1947–1950), самый быстрый компьютер в континентальной Европе того времени.

С развалом СССР для вычислительной техники наступили трудные времена, когда всем новым государствам она оказалась практически не нужна. Рыночные реформаторы считали, что все проблемы можно решить закупками на мировом рынке и современная электронная промышленность является излишней обузой для страны в условиях перестройки экономики.

Компьютерная и электронная инженерия оказались невостребованными в отечественной промышленности и устойчиво деградировали с 1990-х и до середины 2000-х годов, когда руководителям страны стала очевидной невозможность решения проблем информатизации страны за счет импорта без угрозы полной потери технологического суверенитета, совмещенной с возможным подрывом национальной безопасности.

Разработка собственных архитектур электронно-вычислительных машин (ЭВМ) и микропроцессоров были практически остановлены, и немногочисленные выжившие конструкторско-технологические структуры занимались, в основном, клонированием микропроцессоров ведущих мировых производителей в пределах выделенных весьма ограниченных бюджетов. Чудесным исключением можно назвать микропроцессоры с отечественной архитектурой «Эльбрус», которые разрабатывают и развивают ОАО «ИНЭУМ им. И. С. Брука» и фирма «МЦСТ».

Разумеется, это привело к сужению сферы применения современной компьютерной инженерии в рамках этого круга предприятий, которым были доступны лицензии на производство и соответствующие инструментальные средства. Кроме того, такое состояние сказалось на вкладе отечественной электроники в мировую индустрию. Стал актуальным вопрос: есть ли в России электроника?

Если сравнивать с Россией 1913 года, то электроника в стране, безусловно, была. Но если сравнивать с мировой индустрией – то ее практически не было. Общий объем производства электроники в России в 2008 году составлял восемь миллиардов долларов (данные ассоциации производителей электронной продукции РФ), то есть всего 0.4% от мирового рынка, объем которого составлял более двух триллионов долларов. Население России (142 млн.) составляло тогда 2.14% населения планеты, то есть уровень развития электроники был в пять раз ниже «порога самоуваже-

ния нации» ($2.14/0.4 = 5.35$). И это происходило при наличии кадрового корпуса инженеров и работающей высшей инженерной школе.

Начиная с середины 2000-х годов начали издаваться большие учебники по компьютерной архитектуре (архитектуре микропроцессоров), которые являлись переводами популярных американских или европейских университетских учебников. Например, издательство «Питер» уже издало, по крайней мере, два таких учебника:

- ▶ Паттерсон Д., Хеннеси Дж. Архитектура компьютера и проектирование компьютерных систем. 4-е изд. – СПб.: Питер, 2012. – ISBN 978-5-459-00291-1;
- ▶ Таненбаум Э., Остин Т. Архитектура компьютера. 6-е изд. – СПб.: Питер, 2014. – ISBN 978-5-496-00337-7 (а также предыдущие издания).

К общим недостаткам этих популярных книг относится их описательный характер по отношению к архитектуре микропроцессоров, также как и невозможность спроектировать и построить собственный микропроцессор, базируясь на изучении материала, представленного в этих учебниках. К недостаткам русскоязычных изданий относятся ошибки в переводе специальной терминологии и частая потеря технического смысла в предложениях, также как и относительно небольшой тираж в 2000 экземпляров, обусловленный ограничениями приобретенной издательством лицензии.

Издательство «Техносфера» при поддержке ОАО «ИНЭУМ им. И. С. Брука» готовит перевод еще одного учебника:

- ▶ Хеннеси Д. Л., Паттерсон Д. Компьютерная архитектура. Количественный подход / перевод с англ. под ред. к.т.н. А. К. Кима.

Это хорошо известный и популярный учебник для старших курсов и магистерских программ, но он начинается именно там, где заканчивается «Архитектура компьютера и проектирование компьютерных систем» Паттерсона и Хеннеси. Понимание и освоение материала из этого учебника практически невозможно без освоения материала из предшествующих книг.

Отечественная учебная литература по архитектурам ЭВМ и микропроцессоров представлена книгами:

- ▶ Жмакин А. П. Архитектура ЭВМ. 2-е изд. – СПб.: БХВ-Петербург, 2010. – ISBN 978-5-9775-0550-5;
- ▶ Орлов С. А., Цилькер Б. Я. Организация ЭВМ и систем. 2-е изд. – СПб.: Питер, 2011. – ISBN: 978-5-49807-862-5.

Эти учебники в еще большей степени носят описательный характер способов построения различных микропроцессоров на основе зарубежных публикаций и могут рассматриваться как справочники по данному предмету, хотя содержат далеко не всю информацию об архитектурах. Они также непригодны для практической разработки и построения собственного микропроцессора и нуждаются в специальном лабораторном практикуме. Кроме этих книг, имеются многочисленные методические пособия, изданные в разных университетах, курсы и конспекты лекций по предмету, доступные через Интернет. К сожалению, ни один из них не может использоваться в качестве массового учебника для студентов, как младших, так и старших курсов.

Прогресс полупроводниковых технологий и инструментальных средств проектирования цифровых систем в 1990–2000-е годы вывел на первый план языки описания аппаратуры System Verilog и VHDL, которые практически вытеснили традиционное схемотехническое проектирование электронных устройств, включая блоки микропроцессоров. Создание сложных систем на кристалле, объединяющих несколько различных типов микропроцессоров, стало возможным только при использовании средств проектирования, моделирования и верификации ультра-больших интегральных схем, поставляемых фирмами Cadence, Synopsys и Mentor Graphics.

Кроме того, появились доступные по цене средства моделирования и макетирования в виде конструкторских плат, использующих FPGA (Field Programmable Gate Array), называемых в России ПЛИС (Программируемая логическая интегральная схема). Такие платы могут быть приобретены любым учебным центром или даже частным лицом. Используя языки описания аппаратуры и предоставленные производителем FPGA инструментальные средства от вышеупомянутых фирм, любой грамотный студент в состоянии самостоятельно спроектировать и построить сложную цифровую систему, включая микропроцессор.

К сожалению, массовых учебных пособий, позволяющих изучать архитектуру микропроцессоров с практическим уклоном, в том числе и студентам младших курсов, до последнего времени просто не было. В том числе и в США, где основным учебником оставался вышеупомянутый учебник Паттерсона и Хеннесси «Архитектура компьютера и проектирование компьютерных систем», который впервые был издан в начале 1990-х годов, когда современные инструментальные средства еще не были массово доступны.

Поэтому изданный в 2012 году во второй раз учебник Дэвида Харриса и Сары Харрис «Digital Design and Computer Architecture» стал крайне популярным в США, включая Калифорнийский университет в Беркли, в котором работает профессором Д. Паттерсон. С одной стороны, он базируется на классическом материале из учебника Паттерсона и Хеннесси, с другой стороны, добавляет все уровни проектирования цифровых блоков на языках описания аппаратуры System Verilog и VHDL, позволяющей практическую реализацию методов построения блоков микропроцессоров на платах FPGA. Кроме того, он содержит интенсивный практикум программирования на языке ассемблера популярного микропроцессора MIPS. Все детали подготовки этой книги и мотивацию вы можете прочитать в предисловии авторов к американскому изданию.

Традиционный подход к русскоязычному изданию такого учебника привел бы к аналогичным результатам, полученным издательством «Питер» – малому тиражу и ограниченной доступности для студентов младших курсов и техникумов, где эта книга могла бы принести наибольшую пользу. Поэтому, для того, чтобы сделать русскоязычное издание этого учебника массово доступным, его нужно было сделать в электронной версии, загружаемой через Интернет без оплаты или за минимальную плату. Кроме того, нужно было найти спонсора, который бы оплатил стоимость лицензии у издательства «Elsevier» и подготовку перевода этой книги наряду с необходимой версткой и подготовкой текста к электронному изданию. Решение всех этих вопросов выглядело крайне проблематичным.

Дальнейшие события, связанные с этим проектом, можно отнести к категории мистического совпадения многих случайностей, которые предопределили дальнейшую судьбу проекта. Первым и главным фактором стала личная инициатива Юрия Панчула, разработчика микропроцессоров MIPS, инициатора перевода на русский язык второго издания учебника «Digital Design and Computer Architecture», который он активно пропагандировал в университетах США, России и Украины.

До включения компании MIPS в структуру британской корпорации Imagination Technologies перспективы издания учебника на русском языке были достаточно прозрачны ввиду отсутствия значимых бизнес-проектов MIPS в России. Поэтому вторым фактором был приход британских управленцев в подразделение MIPS в составе Imagination Technologies, которые начали интенсивную экспансию на российский рынок при активной помощи Юрия Панчула и российской компании «Наутех». Сложилась ситуация, когда появилась реальная необходимость в массовом учебнике, в котором в деталях рассматривается архитектура микропроцессоров MIPS и его практическое применение для построения современных систем на кристалле для растущего российского рынка.

Третьим фактором стал приход в Imagination Technologies нового менеджера по мировым образовательным программам Роберта Оуэна (Robert Owen), который предложил спонсировать электронное издание русского перевода учебника и лицензировать право на перевод у издательства «Elsevier». Главным условием являлось бесплатное распространение русскоязычной версии учебника с образовательного портала Imagination Technologies.

Четвертым фактором, сделавшим возможным успех проекта, стал энтузиазм и пассионарность русскоязычных инженеров мировой электронной индустрии как в США, так и в России, взявшихся за перевод глав и разделов учебника методом краудсорсинга при активной координации Юрия Панчула. Перевод и редактирование многих глав и разделов, а также полная верификация перевода и синхронизация технических терминов были бы невозможны без активного участия профессоров, доцентов, стажеров и аспирантов из университетов России и Украины, также присоединившихся к проекту.

Пятым фактором можно назвать активную помощь издательских структур и сотрудников Imagination Technologies и корпорации Роснано, которая также активно включилась в проект по изданию учебника в рамках поддержки развития электронной инженерии и разработки наноэлектроники в России.

Шестым и последним фактором явилась заинтересованность издательства «ДМК Пресс» выпустить бумажную версию книги. При этом макет был полностью переверстан из планшетного (электронного) варианта в печатный по образцу оригинала, выпущенного издательством «Elsevier». Также для печатной версии книги были переведены на русский язык все рисунки и программные листинги.

Надеемся, что этот проект получит дальнейшее продолжение в виде создания портала поддержки развития электронной инженерии в странах русскоязычного мира. Такой портал необходим для консолидации учебных и научных ресурсов, которые помогут университетам и колледжам всегда «быть на острие прогресса» электронных технологий и инструментальных средств.

Благодарности участникам проекта

Перевод второго издания учебника Дэвида Харриса и Сары Харрис «Digital Design and Computer Architecture» на русский язык был осуществлен в рекордные сроки – всего за четыре месяца. Перевод и редактирование выполнила команда из полусотни энтузиастов, заинтересованных в том, чтобы в русскоязычных странах (России, Украине, Беларуси, Казахстане и других) возникла твердая основа преподавания современной цифровой электроники на основе системного подхода, с одновременным введением в разработку аппаратуры и низкоуровневого программного обеспечения.

В группу переводчиков вошли преподаватели российских и украинских университетов, сотрудники институтов Российской академии наук, инженеры ведущих российских, американских и западноевропейских компаний. Это позволило воспроизвести устойчивую терминологию, пригодную не только для этого проекта, но и для будущих книг по цифровой схемотехнике, языкам описания аппаратуры, компьютерной архитектуре и микроархитектуре, разработке систем на кристалле, использованию программируемых логических интегральных схем и микроконтроллеров.

Инициаторами проекта выступили:

- ▶ Юрий Панчул, старший инженер по разработке аппаратуры компании Imagination Technologies, группа разработки микропроцессора MIPS I6400;
- ▶ Тимур Палташев, старший менеджер группы компьютерной графики компании Advanced Micro Devices;
- ▶ Роберт Оуэн, консультант по образовательным программам, менеджер мировых образовательных программ Imagination Technologies.

Хотелось бы особо отметить следующих участников, которые отличились объемом, скоростью и качеством перевода:

- ▶ Валерий Казанцев, старший инженер по применению процессорных ядер ARC компании Synopsys, Санкт-Петербург, Россия;
- ▶ Александр Барабанов, доцент кафедры компьютерной инженерии факультета радиофизики, электроники и компьютерных систем Киевского национального университета имени Тараса Шевченко;
- ▶ Группа переводчиков в Самарском государственном аэрокосмическом университете имени академика С. П. Королёва (СГАУ), руководитель группы – декан радиотехнического факультета Илья Александрович Кудрявцев.

Также хотелось бы поблагодарить коллектив компании АНО «eNano», созданной фондом инфраструктурных образовательных программ ОАО «РОСНАНО», за высококачественную работу по форматированию и верстке книги. «eNano» организовали оперативный и слаженный процесс работы над изданием, который существенно облегчил выход книги в свет. Окончательную версию книги помог создать отдел Creative Services компании Imagination Technologies.

31 декабря 2014 года

Тимур Палташев, Advanced Micro Devices, Sunnyvale, California, USA

Юрий Панчул, Imagination Technologies, Santa Clara, California, USA

Роберт Оуэн, Imagination Technologies, Hertfordshire, United Kingdom

Переведено командой из компаний и университетов России, Украины, США и Великобритании

Александр Барабанов	доцент Киевского КНУ
Александр Биргер	на пенсии, ранее в Cadence Design Systems
Александр Леденев	Apple, OS X
Александр Телешов	Модуль
Александра Богданова	МИФИ
Алексей Евгеньевич Платунов	профессор ИТМО
Алексей Лавров	аспирант Принстона
Алексей Фрунзе	Imagination Technologies
Андрей Лихолит	eASIC
Андрей Терехов	директор НИИ информационных технологий СПбГУ
Анна Степашкина	Самарский СГАУ
Антон Моисеев	НГТУ им Р. Е. Алексеева, ФИВТ МФТИ
Валерий Казанцев	Synopsys, процессоры ARC
Виктория Ведица	Imagination Technologies
Владимир Рытиков	Apple
Владимир Серяпов	RusBITech
Владимир Хаханов	декан Харьковского НУ радиоэлектроники
Григорий Жихарев	МЦСТ
Денис Хартиков	NVidia
Дмитрий Миронов	Runtime Design Automation
Екатерина Степанова	Самарский СГАУ
Иван Графский	выпускник МИФИ
Илья Александрович Кудрявцев	декан Самарского СГАУ
Константин Евтушенко	Модуль
Константин Петров	НИИСИ РАН
Константин Пылаев	Компания БиДжи
Леонид Брухис	Synopsys, группа по эмуляторам
Леонид Егошин	Imagination Technologies
Линк Джепсон	Imagination Technologies
Максим Горбунов	НИИСИ РАН
Максим Матуско	МИФИ
Максим Парфенов	Marvell Semiconductor, документация
Михаил Барских	НИИСИ РАН
Нина Захарчук	корректор
Павел Валерьевич Кустарев	доцент ИТМО
Петр Чибисов	НИИСИ РАН
Роберт Оуэн	Imagination Technologies
Руслан Тихонов	amperka.ru
Сергей Аряшев	заведующий отделом НИИСИ РАН
Сергей Чураев	AMD, ИТМО
Симон Атанасян	Imagination Technologies
Тимур Палташев	старший менеджер AMD, ИТМО
Эдуард Стародубцев	доцент ИТМО
Юрий Панчул	Imagination Technologies, MIPS processors
Юрий Холопов	кафедра МФТИ в ИТМиВТ
Юрий Шейнин	СПб ГУА

Предисловие

Зачем публиковать еще одно учебное пособие по цифровой схемотехнике и архитектуре компьютера? Несколько дюжин книг по цифровым системам уже опубликованы и активно используются. Также существует несколько хороших изданий по компьютерной архитектуре, в том числе и классические учебники Паттерсона и Хеннесси (Patterson & Hennessy). Предлагаемая книга уникальна тем, что в ней подробно и доступно описан процесс проектирования цифровых систем с точки зрения компьютерной архитектуры, начиная с двоичных цифр «0» и «1» и заканчивая проектированием микропроцессора MIPS.

В течение многих лет в колледже Harvey Mudd мы использовали различные издания книги Паттерсона и Хеннесси «*Computer Organization and Design*»¹. В их книге нам особенно нравится описание архитектуры и микроархитектуры MIPS ввиду того, что MIPS является коммерчески успешной микропроцессорной архитектурой, но при этом остается достаточно простой и позволяет студентам младших курсов самостоятельно построить микропроцессор. Так как этот курс не имеет специальных требований по предварительным знаниям, первая половина семестра посвящена основам проектирования цифровых систем, которые не рассматриваются в книге Паттерсона и Хеннесси. Другие университеты тоже испытывали потребность в учебном пособии, которое сочетало бы цифровую схемотехнику и компьютерную архитектуру. Поэтому мы и взяли на себя обязательство подготовить такую книгу.

Мы считаем, что проектирование микропроцессора является своеобразным образом 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.

Данная книга подходит как для сжатого односеместрового курса «Введение в цифровую схемотехнику и архитектуру компьютера», так и для более углубленного двухсеместрового курса, позволяющего студентам посвятить больше времени освоению материала и проведению лабораторных работ. Курс может преподаваться «с нуля» и не требует предварительной подготовки. Материал, содержащийся в книге, обычно преподается на втором или третьем курсе университетов, но и заинтересованные первокурсники тоже смогут освоить его.

Особенности книги

Эта книга содержит ряд особенностей.

¹ Четвертое издание этой книги переведено на русский язык в 2012 году издательством «Питер» под названием «Архитектура компьютера и проектирование компьютерных систем» – *Прим. перевод.*

Одновременное использование языков SystemVerilog и VHDL

Языки описания аппаратуры (hardware description languages, HDL) находятся в центре современных методов проектирования сложных цифровых систем. К сожалению, разработчики делятся на две примерно равные группы, использующие два разных языка – SystemVerilog и VHDL. Языки описания аппаратуры рассматриваются в **главе 4**, сразу после глав, посвященных проектированию комбинационных и последовательностных логических схем. Затем языки HDL используются в **главе 5** и **главе 7** для разработки цифровых блоков большего размера и процессора целиком. Тем не менее, **главу 4** можно безболезненно пропустить, если изучение языков HDL не входит в программу.

Эта книга уникальна тем, что использует одновременно и SystemVerilog, и VHDL, что позволяет читателю освоить проектирование цифровых систем сразу на двух языках. В **главе 4** сначала описываются общие принципы, применимые к обоим языкам, а затем вводится синтаксис и приводятся примеры использования этих языков. Этот двуязычный подход облегчает преподавателю выбор языка HDL, а читателю позволит перейти с одного языка на другой как во время учебы, так и в профессиональной деятельности.

Архитектура и микроархитектура классического процессора MIPS

Главы 6 и 7 посвящены изучению архитектуры MIPS и написаны на основе учебника Паттерсона и Хеннесси «Computer Organization and Design». Архитектура MIPS является идеальной в том смысле, что это реальная архитектура, на которой основаны миллионы выпускаемых ежегодно микросхем, и в то же время она проста для изучения. Кроме того, сотни университетов по всему миру разрабатывают учебные курсы, лабораторные работы и различные инструменты именно для этой архитектуры.

Живые примеры

В **главах 6, 7 и 8** в качестве примера также рассматривается архитектура, микроархитектура и иерархия памяти процессоров Intel x86. В качестве другого примера в **главе 8** описываются периферийные устройства микроконтроллеров PIC32 компании Microchip. Эти живые примеры показывают, как описанные в данных главах концепции применяются в реальных микросхемах, которые широко используются в персональных компьютерах и бытовой электронике.

Доступное описание высокопроизводительных архитектур

Глава 7 содержит краткий обзор современных высокопроизводительных микроархитектур: суперскалярной, с внеочередным выполнением команд, многопоточной и многоядерной. Материал изложен в доступной для первокурсников форме и показывает, как можно расширить микроархитектуры, описанные в книге, чтобы получить современный процессор.

Упражнения в конце глав и вопросы для собеседования

Лучшим способом изучения цифровой схемотехники является разработка устройств. В конце каждой главы приведены многочисленные упражнения. За упражнениями следует набор вопросов для собеседования, которые наши коллеги обычно задают студентам, претендующим на работу в отрасли. Эти вопросы предлагают читателю взглянуть на задачи, с которыми соискателям придется столкнуться в ходе собеседования при трудоустройстве. Решения упражнений доступны через веб-сайт книги и специальный веб-сайт для преподавателей. Более подробная информация приведена в следующем разделе.

Материалы в Интернете

Дополнительные материалы для этой книги доступны на веб-сайте по адресу textbooks.elsevier.com/9780123944245. Этот веб-сайт доступен всем читателям и содержит:

- ▶ Решения нечетных упражнений.
- ▶ Ссылки на профессиональные средства автоматизированного проектирования (САПР) компаний Altera® и Synopsys®.
- ▶ Ссылка на QtSpim (также известен как SPIM), симулятор MIPS.
- ▶ HDL-код для процессора MIPS.
- ▶ Полезные советы по использованию САПР Altera Quartus II.
- ▶ Полезные советы по использованию среды разработки Microchip MPLAB IDE.
- ▶ Слайды лекций в формате PowerPoint.
- ▶ Образцы учебных и лабораторных материалов для курса.
- ▶ Список опечаток.

Также существует специальный веб-сайт для преподавателей, зарегистрировавшихся на textbooks.elsevier.com, который содержит:

- ▶ Решения всех упражнений.
- ▶ Ссылки на САПР компаний Altera® и Synopsys® (Synopsys предлагает инструментарий Synplify® Premier университетам, удовлетворяющим определенным требованиям, в упаковке по 50 лицензий. Для получения подробной информации об университетских программах Synopsys зайдите на специальный веб-сайт для преподавателей).
- ▶ Рисунки из текста в форматах JPG и PPT.

Дополнительная информация по использованию инструментов Altera, Synopsys, Microchip и QtSpim в вашем курсе приведена в следующем разделе, там же находится информация о материалах для лабораторных работ.

Как использовать программный инструментарий в учебном курсе

Quartus II Web

Quartus II Web Edition является бесплатной версией профессиональной САПР Quartus™ II, предназначенной для разработки на FPGA. Это позволяет студентам проектировать цифровые устройства в виде принципиальных схем или на языках SystemVerilog и VHDL. После создания схемы или кода устройства студенты могут симулировать их поведение с использованием САПР ModelSim™ – Altera Starter Edition, которая доступна вместе с Altera Quartus II Web Edition. Quartus II Web Edition также включает в себя встроенный логический синтезатор, поддерживающий как SystemVerilog, так и VHDL.

Разница между Web Edition и Subscription Edition заключается в том, что Web Edition поддерживает только подмножество наиболее распространенных FPGA производства Altera. Разница между ModelSim – Altera Starter Edition и коммерческих версий ModelSim заключается в том, что Starter Edition искусственно снижает производительность симуляции для проектов, содержащих больше 10 тысяч строк HDL-кода.

Microchip MPLAB IDE

Интегрированная среда разработки Microchip MPLAB является инструментом для программирования микроконтроллеров PIC и доступна для свободного скачивания в сети. MPLAB объединяет написание программы, компиляцию, моделирование и отладку в едином интерфейсе. Она включает в себя компилятор языка C и отладчик, позволяющий студентам разрабатывать C-код и ассемблерные программы, компилировать их, а также загружать и запускать их на микроконтроллере PIC.

Дополнительные инструменты: Synplify Premier и QtSpim

Synplify Premier и QtSpim являются дополнительными инструментами, которые могут быть использованы в этом курсе.

САПР Synplify Premier является средой для синтеза и отладки цифровых систем на FPGA и CPLD. В комплекте есть HDL-Analyst, уникальный графический инструмент анализа HDL-кода, который автоматически генерирует принципиальную схему из исходного HDL-кода и позволяет в реальном времени видеть соответствие определенных конструкций языка и частей принципиальной схемы. Это очень полезно в процессе обучения и отладки.

Synopsys предлагает инструментарий Synplify® Premier университетам, удовлетворяющим определенным требованиям, в упаковке по 50 лицензий. Для получения подробной информации об университетских программах Synopsys зайдите на специальный веб-сайт для преподавателей (textbooks.elsevier.com/9780123944245).

QtSpim, также известный как SPIM – это симулятор, который исполняет код на языке ассемблера MIPS. Студенты должны сохранить свой ассемблерный код как текстовый файл и запустить его с помощью QtSpim. В QtSpim отображаются

команды, содержимое памяти и значения регистров. Ссылки на руководство пользователя и примеры файлов доступны на веб-сайте книги (textbooks.elsevier.com/9780123944245).

Лабораторные работы

Веб-сайт книги содержит ссылки на ряд лабораторных работ, которые охватывают все темы, начиная от проектирования цифровых систем и заканчивая архитектурой компьютера. Из лабораторных работ студенты узнают, как использовать САПР Quartus II для описания своих проектов, их симулирования, синтеза и реализации. Лабораторные работы также включают темы по программированию на языке C и языке ассемблера с использованием среды проектирования MPLAB IDE.

После синтеза студенты могут реализовать свои проекты, используя платы Altera DE2. Эта мощная и относительно недорогая плата доступна для заказа на веб-сайте www.altera.com. Плата содержит микросхему FPGA, которую можно сконфигурировать для реализации студенческих проектов. Мы предоставляем лабораторные работы, которые описывают, как реализовать различные блоки на плате DE2 с использованием Quartus II Web Edition.

Для выполнения лабораторных работ студенты должны будут загрузить и установить САПР Altera Quartus II Web Edition и Microchip MPLAB IDE. Преподаватели могут также установить эти САПР в учебных лабораториях. Лабораторные работы включают инструкции по разработке проектов на плате DE2. Этап практической реализации проекта на плате можно пропустить, однако мы считаем, что он имеет большое значение для получения практических навыков.

Мы протестировали лабораторные работы на ОС Windows, но инструменты доступны и для ОС Linux.

Опечатки

Все опытные программисты знают, что любая программа сложная программа непременно содержит ошибки. Так же происходит и с книгами. Мы старались выявить и исправить все ошибки и опечатки в этой книге. Тем не менее, некоторые ошибки могли остаться. Список найденных ошибок будет опубликован на веб-сайте книги.

Пожалуйста, присылайте найденные ошибки по адресу iup@imgtec.com. Первый человек, который сообщит об ошибке и предоставит исправление, которое мы используем в будущем издании, будет вознагражден премией в \$1!

Признательность за поддержку

Прежде всего, мы благодарим Дэвида Паттерсона и Джона Хеннесси за разработку новаторских микроархитектур MIPS, описанных в их учебнике «Computer Organization and Design». В течение многих лет мы использовали различные издания этой книги в преподавании. Мы высоко ценим поддержку, оказанную Дэвидом и Джоном при написании этой книги, а также их великодушное разрешение на использование этих микроархитектур в нашем учебнике.

Отдельное спасибо Дуэйну Бибби (Dwyane Bibby), нашему любимому художнику-оформителю, который долго и упорно трудился, чтобы проиллюстрировать разработку цифровых систем. Мы также высоко ценим энтузиазм Нэйти МакФаддена (Nate McFadden), Тодда Грина (Todd Green), Дэниэла Миллера (Danielle Miller), Робина Дэя (Robyn Day) и остальных членов команды издательства Morgan Kaufmann, которые сделали возможным появление этой книги.

Мы хотели бы поблагодарить Мэтью Уоткинса (Matthew Watkins), который помог написать раздел о гетерогенных многопроцессорных системах в Главе 7. Мы также благодарны Крису Парксу (Chris Parks), Карлу Пирсону (Carl Pearson) и Джонатану Чай (Johnathan Chai), которые проверили коды и разработали оглавление для второго издания.

Огромный вклад в улучшение качества книги внесли многочисленные рецензенты: Джон Барр (John Barr), Джэк Брайнер (Jack V. Briner), Эндрю Браун (Andrew S. Brown), Карл Баумгартнер (Carl Baumgaertner), Утку Дирил (A. Utku Diril), Джим Френцель (Jim Frenzel), Джаэха Ким (Jaeha Kim), Филлип Кинг (Phillip King), Джеймс Пинтер-Лаки (James Pinter-Lucke), Амир Рот (Amir Roth), Джерри Ши (Z. Jerry Shi), Джеймс Стайн (James E. Stine), Люк Тэсье (Luke Teyssier), Пейуй Чжао (Peiyi Zhao), Зак Доддс (Zach Dodds), Натаниэл Гай (Nathaniel Guy), Эшвин Кришна (Aswin Krishna), Волней Педрони (Volnei Pedroni), Карл Ванг (Karl Wang), Рикардо и анонимный рецензент.










Мы также очень признательны нашим студентам в колледже Harvey Mudd, которые дали полезные отзывы на черновики этого учебника. Отдельного упоминания заслуживают Мэтт Вайнер (Matt Weiner), Карл Уолш (Carl Walsh), Эндрю Картер (Andrew Carter), Кейси Шиллинг (Casey Schilling), Элис Клифтон (Alice Clifton), Крис Эйкон (Chris Acon) и Стивен Браунер (Stephen Brawner).

И, конечно же, мы благодарим наши семьи за их любовь и поддержку.



От нуля до единицы

- 1.1. План игры
 - 1.2. Искусство управления сложностью
 - 1.3. Цифровая абстракция
 - 1.4. Системы счисления
 - 1.5. Логические элементы
 - 1.6. За пределами цифровой абстракции
 - 1.7. КМОП-транзисторы
 - 1.8. Потребляемая мощность
 - 1.9. Краткий обзор главы 1 и того, что нас ждет впереди
- Упражнения
- Вопросы для собеседования

Прикладное ПО	
Операционные системы	
Архитектура	
Микро-архитектура	
Логика	
Цифровые схемы	
Аналоговые схемы	
Пп приборы	
Физика	

1.1. План игры

За последние тридцать лет микропроцессоры буквально изменили наш мир до неузнаваемости. Сегодняшний ноутбук обладает большей вычислительной мощностью, чем большой компьютер недавнего прошлого, занимавший целую комнату. Внутри современного автомобиля представительского класса можно обнаружить около пятидесяти микропроцессоров. Именно прогресс в области микропроцессорной техники сделал возможным появление сотовых телефонов и Интернета, значительно продвинул вперед медицину и радикально изменил тактику и стратегию современной войны. Объем продаж мировой полупроводниковой промышленности вырос с 21 миллиарда долларов в 1985 году до 300 миллиардов долларов в 2011 году, причем микропроцессоры составили львиную долю этих продаж. И мы убеждены, что микропроцессоры важны не только с технической, экономической и социальной точек зрения, но и стали одним из самых увлекательных изобретений в истории челове-

ства. Когда вы закончите чтение этой книги, вы будете знать, как спроектировать и построить ваш собственный микропроцессор, а навыки, полученные на этом пути, пригодятся вам для разработки и многих других цифровых систем.

Мы предполагаем, что у вас уже есть базовые знания по теории электричества, некоторый опыт программирования и искреннее желание понять, что происходит под капотом компьютера. В этой книге основное внимание уделяется разработке цифровых систем, то есть систем, которые используют для своей работы два уровня напряжения, представляющих единицу и нуль. Мы начнем с простейших цифровых логических элементов – вентилях (*digital logic gates*), которые принимают определенную комбинацию единиц и нулей на входе и трансформируют ее в другую комбинацию единиц и нулей на выходе. После этого мы с вами научимся объединять эти простейшие логические элементы в более сложные модули, такие как сумматоры и блоки памяти. Затем мы перейдем к программированию на языке ассемблера – родном языке микропроцессора. И в завершение, из кирпичиков логических элементов мы с вами соберем полноценный микропроцессор, способный выполнять ваши программы, написанные на языке ассемблера.

Огромным преимуществом цифровых систем над аналоговыми является то, что необходимые для их построения блоки чрезвычайно просты, поскольку оперируют не непрерывными сигналами, а единицами и нулями.

Построение цифровой системы не требует запутанных математических расчетов или глубоких знаний в области физики. Вместо этого, задача, стоящая перед разработчиком цифровых устройств, заключается в том, чтобы собрать сложную работающую систему из этих простых блоков. Возможно, микропроцессор станет первой спроектированной вами системой, настолько сложной, что ее невозможно целиком удержать в голове. Именно поэтому одной из тем, проходящих красной нитью через эту книгу, является искусство управления сложностью системы.

1.2. Искусство управления сложностью

Одной из характеристик, отличающих профессионального инженера-электронщика или программиста от дилетанта, является систематический подход к управлению сложностью многоуровневой системы. Современные цифровые системы построены из миллионов и миллиардов транзисторов. Человеческий мозг не в состоянии предсказать поведение подобных систем путем составления уравнений, описывающих движение каждого электрона в каждом транзисторе системы, и последующего решения этой системы уравнений. Для того, чтобы разработать удачный микропроцес-

сор и не утонуть при этом в море избыточной информации, необходимо научиться управлять сложностью разрабатываемой системы.

1.2.1. Абстракция

Критически важный принцип управления сложностью системы – *абстракция*, подразумевающая исключение из рассмотрения тех элементов, которые в данном конкретном случае несущественны для понимания работы этой системы. Любую систему можно рассматривать с различных уровней абстракции. Политику, участвующему в выборах, например, нет нужды учитывать все детали окружающего его мира, ему достаточно абстрактной иерархической модели страны, состоящей из населенных пунктов, областей и федеральных округов. В области может быть несколько населенных пунктов, а федеральный округ включает в себя разные области. Если политик борется за пост президента, то его, скорее всего, интересует то, как проголосует федеральный округ в целом, при этом ему не обязательно знать, какое количество голосов он наберет в каждом конкретном населенном пункте этого округа. Для политика федеральный округ – это его уровень абстракции. С другой стороны, бюро переписи населения обязано знать количество жителей в каждом городе или поселке страны и потому должно оперировать на самом низком уровне абстракции данной системы – на уровне населенных пунктов.

На **Рис. 1.1** показаны уровни абстракции, типичные для любой электронной компьютерной системы вместе со строительными блоками, характерными для каждого уровня абстракции этой системы. На самом низком уровне абстракции находится физика, изучающая движение электронов. Поведение электронов описывается квантовой механикой и системой уравнений Максвелла.

Рассматриваемая нами современная электронная система состоит из полупроводниковых устройств (*devices*), таких как транзисторы (а когда-то это были электронные лампы). Каждое такое устройство имеет четко определенные точки соединения с другими подобными устройствами. Эти точки мы будем называть *контактами* (в англоязычной литературе используется термин *terminal*). Любое электронное устройство может быть представлено абстрактной математической моделью, описывающей изменяющуюся во времени взаимозависимость тока и напряжения. Такие же изменения тока и напряжения можно наблюдать на экране осциллографа, если подключить осциллограф к контактам реального устройства. Данный под-

Прикладное ПО		Прикладные программы (ПО)
Операционные системы		Драйверы устройств
Архитектура		Регистры команд (инструкций)
Микро-архитектура		Управление потоками
Логика		Сумматоры Память
Цифровые схемы		Элементы И Элементы НЕ
Аналоговые схемы		Усилители Фильтры
Пп приборы		Транзисторы Диоды
Физика		Электроны

Рис. 1.1 Уровни абстракции электронной вычислительной системы

Каждая глава этой книги начинается с иконок (см. Рис. 1.1), символически изображающих уровни абстракции электронной системы, которые мы перечислили выше. Иконка темно-синего цвета указывает на тот уровень абстракции, которому уделяется главное внимание в этой конкретной главе. Иконки более светлого оттенка синего указывают на другие уровни абстракции, также затронутые в этой главе.

ход означает, что, если рассматривать систему на уровне устройств, функции которых однозначно определены, то можно не учитывать поведение электронов внутри отдельных устройств этой системы.

Следующий уровень абстракции — это *аналоговые схемы* (analog circuits), в которых полупроводниковые устройства соединены таким образом, чтобы они образовывали функциональные компоненты, такие как усилители, например. Напряжение на входе и на выходе аналоговой цепи изменяется в непрерывном диапазоне.

В отличие от аналоговых цепей, *цифровые схемы* (digital circuits), такие как логические вентили, используют два строго ограниченных дискретных уровня напряжения. Один из этих дискретных уровней — это логический нуль, другой — логическая единица. В разделах этой книги, посвященных разработке цифровых схем и устройств, мы будем использовать простейшие цифровые схемы для построения сложных цифровых модулей, таких как сумматоры и блоки памяти.

Микроархитектурный уровень абстракции, или просто *микроархитектура* (microarchitecture), связывает логический и архитектурный уровни абстракции. Архитектурный уровень абстракции, или *архитектура* (architecture), описывает компьютер с точки зрения программиста. Например, архитектура Intel x86, используемая микропроцессорами большинства персональных компьютеров (*ПК*), определяется набором инструкций и регистров (памяти для временного хранения переменных), доступным для использования программистом. Микроархитектура — это соединение простейших цифровых элементов в логические блоки, предназначенные для выполнения команд, определенных какой-то конкретной архитектурой. Отдельно взятая архитектура может быть реализована с использованием различных вариантов микроархитектур с разным соотношением цены, производительности и потребляемой энергии, и такое соотношение зачастую выбирается как баланс между этими тремя факторами. Процессоры Intel Core i7, Intel 80486 и AMD Athlon, например, используют одну и ту же архитектуру x86, но реализованную с использованием трех разных микроархитектурных решений.

Теперь мы перемещаемся в область программного обеспечения. *Операционная система* (operating system) управляет операциями нижнего уровня, такими как доступ к жесткому диску или управление памятью. И, наконец, программное обеспечение использует ресурсы операционной системы для решения конкретных задач пользователя.

Именно принцип *абстрагирования от маловажных деталей* позволяет вашей бабушке общаться с внуками в Интернете, не задумываясь о квантовых колебаниях электронов или организации памяти компьютера.

Предмет этой книги – уровни абстракции от цифровых схем до компьютерной архитектуры. Работая на каком-либо из этих уровней абстракции, полезно знать кое-что и об уровнях абстракции, непосредственно сопряженных с тем уровнем, где вы находитесь. Программист, например, не сможет полностью оптимизировать код без понимания архитектуры процессора, который будет выполнять эту программу. Инженер-электронщик, разрабатывающий какой-либо блок микросхемы, не сможет найти компромисс между быстродействием и уровнем потребления энергии транзисторами, ничего не зная о той цифровой схеме, где этот блок будет использоваться. Мы надеемся, что к тому времени, когда вы закончите чтение этой книги, вы сможете выбрать уровень абстракции, необходимый для успешного выполнения любой стоящей перед вами задачи, и оценить влияние ваших инженерных решений на другие уровни абстракции в разрабатываемой вами системе.

1.2.2. Конструкторская дисциплина

Конструкторская дисциплина – это преднамеренное ограничение самим конструктором выбора возможных вариантов разработки, что позволяет работать продуктивнее на более высоком уровне абстракции. Использование взаимозаменяемых частей – это, вероятно, самый хорошо знакомый всем нам пример практического применения конструкторской дисциплины. Одним из первых примеров использования взаимозаменяемых деталей и узлов стала унификация при производстве кремнёвых ружей. До начала 19-го века такие ружья производились вручную и в штучном порядке. Высококвалифицированный оружейный мастер тщательно подтачивал и подгонял комплектующие, произведенные несколькими не связанными друг с другом ремесленниками. Конструкторская дисциплина для обеспечения взаимозаменяемости деталей и узлов произвела революцию в оружейной промышленности. Ограничение ассортимента комплектующих деталей до стандартного набора с жестко установленными допусками для каждой детали позволило собирать и ремонтировать ружья гораздо быстрее и использовать при этом менее квалифицированный персонал. Оружейный мастер перестал тратить свое время на разрешение проблем, связанных с нижними уровнями абстракции, такими как доводка какого-то конкретного ствола или исправление формы отдельного взятого приклада.

В контексте данной книги соблюдение конструкторской дисциплины в виде максимального использования цифровых схем играет очень важную роль. В цифровых схемах используются дискретные значения напряжения, в то время как в аналоговых схемах напряжение изменяется непрерывно. Таким образом, цифровые схемы, которые можно рассматривать как подмножество аналоговых цепей, в некотором смысле уступают по своим характеристикам более широкому классу аналоговых

цепей. Однако цифровые цепи гораздо проще проектировать. Ограничивая использование аналоговых схем и по возможности заменяя их цифровыми, мы можем легко объединять отдельные компоненты в сложные системы, которые, в конечном итоге, для большинства приложений превзойдут по своим параметрам системы, построенные на аналоговых цепях.

Примером тому могут служить цифровые телевизоры, компакт-диски (CD) и мобильные телефоны, которые уже практически полностью вытеснили своих аналоговых предшественников.

Капитан Мериуззер Льюис — один из руководителей знаменитой экспедиции Льюиса и Кларка на северо-запад США, был, пожалуй, одним из самых ранних сторонников взаимозаменяемости. В 1806 году в своем дневнике, касаясь унификации деталей кремневых ружей того времени, он написал следующее:

«Ружья Дрюера и сержанта Прайора одновременно вышли из строя. На ружье Дрюера сломался ударно-спусковой механизм, и мы заменили его на новый. У ружья сержанта Прайора был сломан курковый винт, вместо которого мы поставили запасной курковый винт, заранее изготовленный специально для ударно-спускового механизма этого ружья на мануфактуре Харперс Фейри, где это оружие и было произведено. Если бы не предусмотрительность, заключающаяся в том, что мы заранее позаботились о запасных частях для ружей, и не мастерство Джона Шилдса, выполнившего всю работу, то большинство ружей нашей экспедиции к этому времени было бы полностью непригодно для какого-либо использования. И я имею полное право записать в своем дневнике, что, к счастью для нас, все наше оружие находится в прекрасном состоянии».

См. «История экспедиции Льюиса и Кларка» в четырех томах под редакцией Элиота Куэса. Первое издание: Харпер, Нью-Йорк, 1893; переиздание: Довер, Нью-Йорк (3 тома), 3:817.

1.2.3. Три базовых принципа

В дополнение к абстрагированию от несущественных деталей и конструкторской дисциплине разработчики электронных систем используют еще три базовых принципа для управления сложностью системы: иерархичность, модульность конструкции и регулярность. Эти принципы применимы как к программному обеспечению, так и к аппаратной части компьютерных систем.

- ▶ *Иерархичность* — принцип иерархичности предполагает разделение системы на отдельные модули, а затем последующее разделение каждого такого модуля на фрагменты до уровня, позволяющего легко понять поведение каждого конкретного фрагмента.
- ▶ *Модульность* — принцип модульности требует, чтобы каждый модуль в системе имел четко определенную функциональность и набор интерфейсов и мог быть легко и без непредвиденных побочных эффектов соединен с другими модулями системы.
- ▶ *Регулярность* — принцип регулярности требует соблюдения единообразия при проектировании отдельных модулей системы. Стандартные модули общего назначения, например, такие как блоки питания, могут использоваться многократно, во много раз снижая количество модулей, необходимых для разработки новой системы.

Для иллюстрации трех базовых принципов вновь воспользуемся аналогией из оружейного производства. Нарезное кремневое ружье было одним из самых сложных устройств массового применения в начале 19-го века. Используя принцип иерархичности, мы можем разделить его на три главных модуля, как показано на **Рис. 1.2**: ствол, ударно-спусковой механизм и приклад с цевьем.

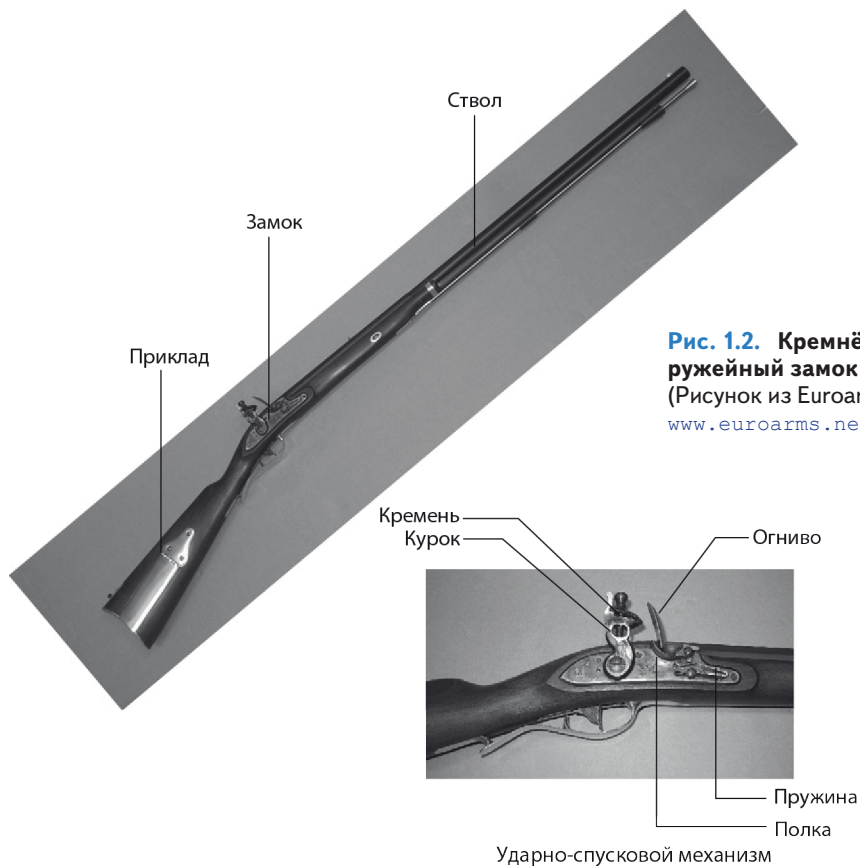


Рис. 1.2. Кремнёвый ружейный замок
(Рисунок из Euroarms Italia
www.euroarms.net © 2006 г.)

Ствол – это длинная металлическая труба, через которую при выстреле выбрасывается пуля. Ударно-спусковой механизм производит выстрел. Деревянные приклад и цевье соединяют воедино остальные части ружья и обеспечивают стрелку надежное удержание оружия при выстреле. В свою очередь, ударно-спусковой механизм включает в себя спусковой крючок, курок, кремль, огниво и пороховую полку. Каждый из этих компонентов также может рассматриваться как следующий иерархический уровень и может быть разделен на более мелкие детали.

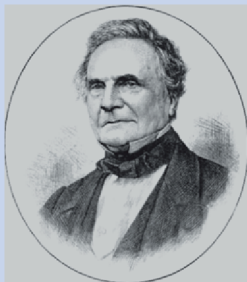
Принцип модульности требует, чтобы каждый компонент выполнял четко определенную функцию и имел интерфейс. Функция приклада и цевья – служить базой для установки ствола и ударно-спускового механизма. Интерфейс для приклада и цевья – это их длина и расположение крепёжных элементов, таких как винты или шурупы. Ствол ружья, изготовленного с соблюдением принципа модульности конструкции, может быть установлен на приклады и цевья от разных производителей, если все соединяемые части имеют правильную длину и подходящие

крепёжные элементы. Функция ствола – разогнать пулю до необходимой скорости и придать ей вращение, чтобы увеличить точность стрельбы (примечание переводчика: кремнёвые ружья не были нарезными и использовали круглые пули). Принцип модульности требует также, чтобы при соединении модулей не возникало никаких побочных эффектов: конструкция приклада и цевья не должна препятствовать функционированию ствола.

Принцип регулярности учит тому, что взаимозаменяемые детали – это хорошая идея. При соблюдении принципа регулярности поврежденный ствол может быть с легкостью заменен на идентичный. Стволы могут изготавливаться на поточной линии с гораздо большей экономической эффективностью, чем в случае штучного производства.

В данной книге мы будем постоянно возвращаться к этим трем базовым принципам: иерархичности, модульности и регулярности.

1.3. Цифровая абстракция



Чарльз Бэббидж
1791–1871

Чарльз Бэббидж родился в 1791 году. Закончил Кембриджский университет и женился на Джорджиане Витмур. Он изобрел Аналитическую Машину – первый в мире механический компьютер. Чарльз Бэббидж также изобрел предохранительную решетку для локомотивов, спидометр и универсальный почтовый тариф. Ученый также очень интересовался отмычками для замков и почему-то ненавидел уличных музыкантов. (Портрет любезно предоставлен Fourmilab Швейцария, www.fourmilab.ch).

Большинство физических величин изменяется непрерывно. Например, напряжение в электрическом проводе, частота колебаний или распределение массы – все это параметры, изменяющиеся непрерывно. Цифровые системы, с другой стороны, представляют информацию в виде дискретно меняющихся переменных с конечным числом строго определённых значений.

Одной из наиболее ранних цифровых систем стала Аналитическая Машина Чарльза Бэббиджа, которая использовала переменные с десятью дискретными значениями. Начиная с 1834 года и до 1871 года¹ Бэббидж разрабатывал и пытался построить этот механический компьютер. Шестеренки Аналитической Машины могли находиться в одном из десяти фиксированных положений, а каждое такое положение было промаркировано от 0 до 9 подобно механическому счетчику пробега автомобиля. **Рис. 1.3** показывает, как выглядел прототип Аналитической Машины. Каждый ряд шестеренок такой машины обрабатывал одну цифру. В своем механическом компьютере Бэббидж использовал 25 рядов шестеренок таким образом, чтобы машина обеспечивала вычисления с точностью до 25-го знака.

В отличие от машины Бэббиджа большинство электронных компьютеров использует двоичный (бинарный) код. В случае двоичного кода высокое напряжение – это

¹ А большинству из нас кажется, что обучение в университете – это так долго!

единица, а низкое напряжение – нуль, поскольку гораздо легче оперировать двумя уровнями напряжения, чем десятью.

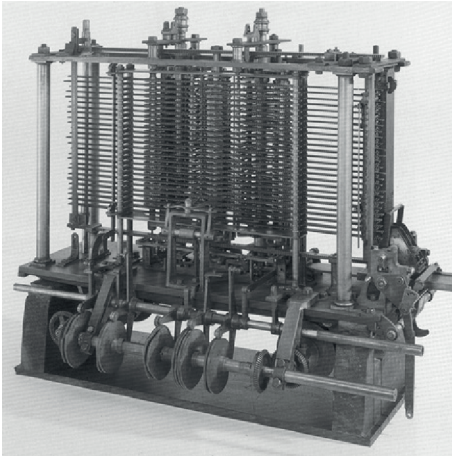


Рис. 1.3. Аналитическая машина Беббиджа в год его смерти (1871) (Изображение любезно предоставлено музеем Науки и общества)

Объем информации D , передаваемый одной дискретной переменной, которая может находиться в N различных состояниях, измеряется в единицах, называемых *битами*, и вычисляется по следующей формуле:

$$D = \log_2 N \text{ бит} \quad (1.1)$$

Двоичная переменная передает $\log_2 2 = 1$ – один бит информации. Теперь вам, вероятно, понятно, почему единица информации называется битом. *Bit* (*бум*) – это сокращение от английского *binary digit*, что дословно переводится как *двоичный разряд*. Каждая шестеренка в машине Бэббиджа содержит $\log_2 10 = 3,322$ бит информации, поскольку она может находиться в одном из $2^{3,322} = 10$ уникальных положений. Теоретически непрерывный сигнал может передавать бесконечное количество информации, поскольку может принимать неограниченное число значений. На практике, однако, шум и ошибки измерения ограничивают информацию, передаваемую большинством непрерывных сигналов, диапазоном от 10 бит до 16 бит. Если же измерение уровня сигнала должно быть произведено очень быстро, то объём передаваемой информации будет еще ниже (в случае 10 бит, например, это будет только 8 бит).

Предмет этой книги – цифровые схемы, использующие двоичные переменные нуль и единицу. Джордж Буль разработал систему логики, использующую двоичные переменные, и эту систему сегодня называют его именем – *булева логика*. Булевы переменные могут принимать значения *ИСТИНА* (*TRUE*) или *ЛОЖЬ* (*FALSE*). В электронных компьютерах положительное напряжение обычно представляет единицу, а нулевое напряжение представляет нуль. В этой книге мы будем использовать понятия единица (1), ИСТИНА (*TRUE*) и ВЫСОКОЕ (*HIGH*) как



Джордж Буль
1815–1864

Джордж Буль родился в семье небогатого ремесленника. Родители Джорджа не могли оплатить его формального образования, поэтому он осваивал математику самоучкой. Несмотря на это, Булю удалось стать преподавателем Королевского колледжа в Ирландии. В 1854 году Джордж Буль написал свою работу Исследование законов мышления, которая впервые ввела в научный оборот двоичные переменные, а также три основных логических оператора И, ИЛИ, НЕ (AND, OR, NOT). (Портрет любезно предоставлен Американским физическим институтом).

синонимы. Аналогичным образом мы будем использовать нуль (0), ЛОЖЬ (FALSE) и НИЗКОЕ (LOW) как взаимозаменяемые термины.

Преимущества *цифровой абстракции* заключаются в том, что разработчик цифровой системы может сосредоточиться исключительно на единицах и нулях, полностью игнорируя, каким образом булевы переменные представлены на физическом уровне. Разработчика не волнует, представлены ли нули и единицы определенными значениями напряжения, вращающимися шестернями или уровнем гидравлической жидкости. Программист может продуктивно работать, не располагая детальной информацией об аппаратном обеспечении компьютера. Однако, понимание того, как работает это аппаратное обеспечение, позволяет программисту гораздо лучше оптимизировать программу для конкретного компьютера.

Как вы могли видеть выше, один-единственный бит не может передать большого количества информации. Поэтому в следующем разделе мы рассмотрим вопрос о том, каким образом набор битов можно использовать для представления десятичных чисел. В последующих главах мы также покажем, как группы битов могут представлять буквы и даже целую программу.

1.4. Системы счисления

Мы все привыкли работать с десятичными числами. Однако, в цифровых системах, построенных на единицах и нулях, использование двоичных или шестнадцатеричных чисел зачастую более удобно. В данном разделе мы рассмотрим системы счисления, использованные в этой книге.

1.4.1. Десятичная система счисления

Еще в начальной школе нас всех научили считать и выполнять различные арифметические операции в *десятичной* (decimal) системе счисления. Такая система использует десять арабских цифр: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 – столько же, сколько у нас пальцев на руках. Числа больше 9 записываются в виде строки цифр. Причем, цифра, находящаяся в каждой последующей позиции такой строки, начиная с крайней правой цифры, имеет «вес», в десять раз превышающий «вес» цифры, находящейся в предыдущей позиции. Именно поэтому десятичную систему счисления называют системой по основанию (*base*) 10. Справа налево «вес» каждой позиции увеличивается следующим образом: 1, 10, 100, 1000 и т. д. По-

зицию, которую цифра занимает в строке десятичного числа, называют разрядом или декадой.

Чтобы избежать недоразумений при одновременной работе с более чем одной системой счисления, основание системы обычно указывается путем добавления цифры позади и чуть ниже основного числа: 9742_{10} . **Рис. 1.4** показывает, для примера, как десятичное число 9742_{10} может быть записано в виде суммы цифр, составляющих это число, умноженных на «вес» разряда, соответствующего каждой конкретной цифре.

Колонка тысяч	Колонка сотен	Колонка десятков	Колонка единиц	$=$	9×10^3	$+$	7×10^2	$+$	4×10^1	$+$	2×10^0
					Девять тысяч		Семь сотен		Четыре десятки		Две единицы

Рис. 1.4. Представление десятичного числа

N -разрядное десятичное число может представлять одну из 10^N цифровых комбинаций: 0, 1, 2, 3, ... $10^N - 1$. Это называется диапазоном N -разрядного числа. Десятичное число, состоящее из трех цифр (разрядов), например, представляет одну из 1000 возможных цифровых комбинаций в диапазоне от 0 до 999.

1.4.2. Двоичная система счисления

Одиночный бит может принимать одно из двух значений, 0 или 1. Несколько битов, соединенных в одной строке, образуют *двоичное* (binary) число. Каждая последующая позиция в двоичной строке имеет вдвое больший «вес», чем предыдущая позиция, так что двоичная система счисления – это система по основанию 2. В двоичном числе «вес» каждой позиции увеличивается (так же, как и в десятичном – справа налево) следующим образом: 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768, 65536 и т.д. Работая с двоичными числами, очень полезно для сохранения времени запомнить значения степеней двойки до 2^{16} .

Произвольное N -разрядное двоичное число может представлять одну из 2^N цифровых комбинаций: 0, 1, 2, 3, ... $2^N - 1$. В **Табл. 1.1** собраны 1-битные, 2-битные, 3-битные и 4-битные двоичные числа и их десятичные эквиваленты.

Пример 1.1. ПРЕОБРАЗОВАНИЕ ЧИСЕЛ ИЗ ДВОИЧНОЙ СИСТЕМЫ СЧИСЛЕНИЯ В ДЕСЯТИЧНУЮ

Преобразовать двоичное число 10110_2 в десятичное.

Решение: Нужные преобразования представлены на **Рис. 1.5**.

Рис. 1.5.
Преобразование
двоичного числа
в десятичное число

Колонка единиц
Колонка двоек
Колонка четверок
Колонка восьмерок
Колонка шестнадцати

$$101110_2 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 22_{10}$$

Одна шестнадцать
Нет восемь
Одна четыре
Одна двойка
Нет единиц

Табл. 1.1. Таблица двоичных чисел и их десятичный эквивалент

1-битные двоичные числа	2-битные двоичные числа	3-битные двоичные числа	4-битные двоичные числа	Десятичные эквиваленты
0	00	000	0000	0
1	01	001	0001	1
	10	010	0010	2
	11	011	0011	3
		100	0100	4
		101	0101	5
		110	0110	6
		111	0111	7
			1000	8
			1001	9
			1010	10
			1011	11
			1100	12
			1101	13
			1110	14
			1111	15

Пример 1.2. ПРЕОБРАЗОВАНИЕ ЧИСЕЛ ИЗ ДЕСЯТИЧНОЙ СИСТЕМЫ СЧИСЛЕНИЯ В ДВОИЧНУЮ

Преобразовать десятичное число 84_{10} в двоичное.

Решение: Определите, что должно стоять в каждой позиции двоичного результата: 1 или 0. Вы можете делать это, начиная с левой или правой позиции.

Если начать слева, найдите наибольшую степень 2, меньше или равную заданному числу (в примере такая степень – это 64). $84 > 64$, поэтому ставим 1 в позиции, соответствующей 64. Остается $84 - 64 = 20$, $20 < 32$, так что в позиции 32 надо поставить 0, $20 > 16$, поэтому в позиции 16 ставим 1. Остается $20 - 16 = 4$. $4 < 8$, поэтому 0 в позиции 8. $4 \geq 4$ – ставим 1 в позицию 4. $4 - 4 = 0$, поэтому будут 0 в позициях 2 и 1. Сбрав все вместе, получаем $84_{10} = 1010100_2$.

Если начать справа, будем последовательно делить исходное число на 2. Остаток идет в очередную позицию. $84/2 = 42$, поэтому 0 в самой правой позиции. $42/2 = 21$, 0 во вторую позицию. $21/2 = 10$, остаток 1 идет в позицию, со-

ответствующую 4. $10/2 = 5$, поэтому 0 в позицию, соответствующую 8. $5/2 = 2$, остаток 1 в позицию 16. $2/2 = 1$, 0 в 32 позицию. Наконец, $1/2 = 0$ с остатком 1, который идет в позицию 64. Снова, $84_{10} = 1010100_2$

1.4.3. Шестнадцатеричная система счисления

Использование длинных двоичных чисел для записи и выполнения математических расчетов на бумаге утомительно и чревато ошибками. Однако длинное двоичное число можно разбить на группы по четыре бита, каждая из которых представляет одну из $2^4 = 16$ цифровых комбинаций. Именно поэтому зачастую бывает удобнее использовать для работы систему счисления по основанию 16, называемую *шестнадцатеричной* (hexadecimal). Для записи шестнадцатеричных чисел используются цифры от 0 до 9 и буквы от А до F, как показано в Табл. 1.2 В шестнадцатеричном числе «вес» каждой позиции меняется следующим образом: 1, 16, 16^2 (или 256), 16^3 (или 4096) и т. д.

Интересно, что термин *hexadecimal* (шестнадцатеричный) введен в научный обиход корпорацией IBM в 1963 году и является комбинацией греческого слова *hexi* (шесть) и латинского *decem* (десять). Правильнее было бы использовать латинское же слово *sexa* (шесть), но термин *sexadecimal* воспринимался бы несколько неоднозначно.

Табл. 1.2. Шестнадцатеричная система счисления

Шестнадцатеричная цифра	Десятичный эквивалент	Двоичный эквивалент
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

Пример 1.3. ПРЕОБРАЗОВАНИЕ ШЕСТНАДЦАТЕРИЧНОГО ЧИСЛА В ДВОИЧНОЕ И ДЕСЯТИЧНОЕ

Преобразовать шестнадцатеричное число $2ED_{16}$ в двоичное и десятичное.

Решение: Преобразование шестнадцатеричного числа в двоичное и обратно — очень простое, так как каждая шестнадцатеричная цифра прямо соответ-

ствует 4-разрядному двоичному числу. $2_{16} = 0010_2$, $E_{16} = 1110_2$ и $D_{16} = 1101_2$, так что $2ED_{16} = 001011101101_2$. Преобразование в десятичную систему счисления требует арифметики, показанной на **Рис. 1.6**.

$$2ED_{16} = 2 \times 16^2 + E \times 16^1 + D \times 16^0 = 749_{10}$$

Две
Четырнадцать
Тринадцать
двести
шестнадцать
единиц
пятьдесят
шесть
шесть

Рис. 1.6. Преобразование шестнадцатеричного числа в десятичное число

Пример 1.4. ПРЕОБРАЗОВАНИЕ ДВОИЧНОГО ЧИСЛА В ШЕСТНАДЦАТЕРИЧНОЕ

Преобразовать двоичное число 1111010_2 в шестнадцатеричное.

Решение: Повторим еще раз, это просто. Начинаем справа. 4 наименее значимых бита $1010_2 = A_{16}$. Следующие биты $111_2 = 7_{16}$. Отсюда $1111010_2 = 7A_{16}$.

Пример 1.5. ПРЕОБРАЗОВАНИЕ ДЕСЯТИЧНОГО ЧИСЛА В ШЕСТНАДЦАТЕРИЧНОЕ И ДВОИЧНОЕ

Преобразовать десятичное число 333_{10} в шестнадцатеричное и двоичное.

Решение: Как и в случае преобразования десятичного числа в двоичное, можно начать как слева, так и справа.

Если начать слева, найдите наибольшую степень шестнадцати, меньшую или равную заданному числу (в нашем случае это $16^2 = 256$). Число 256 содержится в числе 333 только один раз, поэтому в позицию с «весом» 256 мы записываем единицу. Остается число $333 - 256 = 77$. Число 16 содержится в числе 77 четыре раза, поэтому в позицию с «весом» 16 записываем четверку. Остается $77 - 16 \times 4 = 13$. $13_{10} = D_{16}$, поэтому в позицию с «весом» 1 записываем цифру D. Итак, $333_{10} = 14D_{16}$, это число легко преобразовать в двоичное, как мы показали в примере 1.3: $14D_{16} = 101001101_2$.

Если начинать справа, будем повторять деление на 16. Каждый раз остаток идет в очередную колонку. $333/16 = 20$ с остатком $13_{10} = D_{16}$, который идет в самую правую позицию. $20/16 = 1$ с остатком 4, который идет в позицию с «весом» 16. $1/16 = 0$ с остатком 1, который идет в позицию с «весом» 256. В результате опять получаем $14D_{16}$.

Предметный указатель

Символы

#define, 719–720
#include, 720–721. *См. также* Стандартные библиотеки
0, 54. *См. также* LOW, OFF
1, 54. *См. также* HIGH, ON
32-битный тракт, 534
32-разрядный микропроцессор, 525
4004 ИС микропроцессора, 532, 533
74xx серии логика, 668–671
элементы,
 2:1 мультиплексор (74157), 670
 3:8 дешифратор (74138), 670
 4:1 мультиплексор (74153), 670
И (7408), 669
ЗИ (7411), 669
4И (7421), 669
счетчик (74161, 74163), 670
Триггер (7474), 668, 669
И-НЕ (7400), 669
ИЛИ-НЕ (7402), 669
НЕ (7404), 669
ИЛИ (7432), 669
регистр (74377), 670
Буфер с 3-м сост. (74244), 670
Исключ. ИЛИ (7486), 669

А

add, 358
addi, 366
addiu, 414
addu, 414
Advanced Micro Devices (AMD), 356, 444, 531, 535
Altera FPGA, 335–339
ALUControl, 447, 455
ALUOp, 453–455
ALUResult, 448
ALUSrc, 455
ALUSrcA, 469
ALUSrcB, 469
AMAT. *См.* Среднее время доступа к памяти
AMD. *См.* Advanced Micro Devices
and, 375
andi, 375–376
ASCII. *См.* Американский стандартный код для обмена информацией
ASIC. *См.* Заказные интегральные схемы

В

BCD. *См.* двоично-кодированная десятичная система
beq, 378–379
Bit cells, 322–327
 DRAM, ДОЗУ, 324–325
 ROM, ПЗУ, 326–327
 SRAM, СОЗУ, 325

BlueSMiRF silver, модуль, 632, 632
Bluetooth, беспроводная связь, 632–633
 BlueSMiRF silver, модуль, 632
 классы, 632
 PIC32 с ПК соединение, 633
bne, 378–379
Branch equal (beq)
 машинный код, 402
 реализация в процессоре, 451–452, 466–467, 471–472
Branch, 454
BTA. *См.* Целевой адрес ветвления
Bugs, баги, 225
 в коде на C, 762–768

С

C, программирование, 713–768
 типичные ошибки. *См.* Типичные ошибки компиляции. *См.* Компиляция
 условные операторы. *См.* Условные операторы операторы управления. *См.* Операторы управления
 типы данных. *См.* Типы данных
 вызовы функции. *См.* Вызовы функции
 циклы. *См.* Циклы
 операторы. *См.* Операторы
 запуск, 716
 простая программа, 715–716
 стандартные библиотеки. *См.* Стандартные библиотеки
 переменные. *См.* Переменные
Casez, case?, в HDL, 259
CISC. *См.* Компьютер со сложным набором команд
CLB. *См.* Конфигурируемые логические блоки
CPA. *См.* Сумматор с распространяющимся переносом (CPA)
CPI. *См.* число тактов на команду, тактов на команду
Cyclone IV FPGA, 334–339

Д

D защелка. *См.* защелка
D триггер. *См.* триггеры
DC. *См.* Постоянный ток
DDR, 326
DDR3. *См.* память с удвоенной скоростью обмена
DE-9 кабель, 606
Delaymicros функция, 611
DeleteUser функция, 749
Dice, Кристалл, 65
DIMM. *См.* модуль памяти с двумя рядами выводов
DIP. *См.* корпуса с двумя рядами выводов
div, 378
divu, 414
Do/while циклы, в C, 734–735
Double, тип в C, 722–723

DRAM. *См.* Динамическое оперативное запоминающее устройство

Е

EEPROM. *См.* электрически стираемое программируемое ПЗУ
EFLAGS регистр, 414
EPC. *См.* Exception program counter
EPROM. *См.* стираемые программируемые ПЗУ
exit, 758

F

FDIV. *См.* Деление с плавающей точкой
Field programmable gate arrays (FPGAs), 335–339, 531, 601, 628, 650, 676–678
 работа на кабель VGA, 627
 в интерфейсе SPI, 600–602
Float, тип в C, 720–723
 форматы печати, 753–755
Fog, циклы, 384–385, 387, 735
FPGAs. *См.* Field programmable gate arrays
FPU. *См.* Блок вычислений с плавающей запятой
FSK. *См.* Частотная манипуляция
Funct, поле, 368, 710–711
Genwaves функция, 619
GPIO. *См.* Общего назначения В/В

H

HDL для, 502
HDL. *См.* Язык описания схем, SystemVerilog, and VHDL
HIGH, 57. *См. также* 1, ON
H-мост, управление, 635

I

IA-32 архитектура. *См.* x86
IA-64, 425
If операторы
 в C, 731
 в HDL, 256–259
 в ассемблере MIPS, 381–382
If/else операторы, 381, 741
 в C, 730–732
 в HDL, 256–259
 в ассемблере MIPS, 381
IM. *См.* Память инструкций, команд
InitTimerInterrupt функция, 614
Institute of Electrical and Electronics Engineers (IEEE), 314
Intel x86. *См.* x86
Intel. *См.* x86
IOE. *См.* Элементы ввода/вывода
IoD, 461, 468
IPC. *См.* Инструкций за такт
IR. *См.* Инструкций регистр
IRWrite, 461, 468
ISR. *См.* Программы обработки прерывания
I-типа инструкции, 369–370

J

j, 380–381
jal, 391
Java, 388. *См. также* Языки
jr, 380–381, 392
JTA. *См.* Целевой адрес перехода
Jump, MIPS, 379–380
Jump, воплощение в процессоре, 457–458, 476–480
J-типа инструкции, 369

L

LAB. *См.* Блок логических матриц
Land grid array, корпус LGA, 645
lb, загрузить байт. *См.* Загрузки
lbu, загрузить байт без знака. *См.* Загрузки
lh, загрузить полуслово. *См.* Загрузки
lhu, загрузить полуслово без знака. *См.* Загрузки
LIFO. *См.* Компилятор и опции командной строки
LRU. *См.* Least recently used replacement
lsb. *См.* Least significant bit
LSB. *См.* Least significant byte
lui, загрузка константы в старшие, 377
LUT. *См.* Таблица преобразования
LVCMOS. *См.* Низковольтная КМОП-логика
LVTTL. *См.* Низковольтная TTL-логика
lw, загрузка слова. *См.* Загрузки

M

main, функция в C, 717
Malloc, функция, 748
math.h, библиотека C, 759–760
MCM. *См.* Многокристальный модуль
MemtoReg, 450, 468
MemWrite, 449, 468
mfc0. *См.* Загрузка из регистра сопроцессора 0,
Microchip ICD3, 593
Microchip In Circuit Debugger 3 (ICD3), 593
MIPS команды, 355–427, 274–277
 ветвления, переходы. *См.* Ветвления, Переходы
 форматы
 F-тип, 711–712
 I-тип, 369, 369–371
 J-тип, 371, 371
 R-тип, 368–369
 умножение и деление, 378, 414
 opcodes, 709–710
 R-типа поля funct, 710–711
MIPS одноктактный HDL, 498–510
 универсальные строительные блоки, 504–507
 контроллер, 501
 тракт данных, 503
 тестовая среда, 506–510
MIPS процессоры. *См.* MIPS многотактный процессор, MIPS конвейерный процессор, и MIPS одноктактный процессор
 HDL для. *См.* одноктактный MIPS, HDL
MIPS регистры,

- регистры сопроцессора 0, 412, 511–513
 счётчик команд, 373, 441–442
 регистровый файл, 441–442
 набор регистров, 360–362
- MIPS.** *См. также* Архитектура и микроархитектура архитектуры, 356, 589
 команды для работы с числами с плавающей точкой, 415, 415–416
 набор инструкций, 456
 микроархитектура
 многотактный. *См.* MIPS многотактный процессор
 конвейерный. *См.* MIPS конвейерный процессор
 одноктактный. *См.* MIPS одноктактный процессор
 микропроцессор, 511, 524, 526–527
 память данных, 443
 память инструкций, команд, 443
 счетчик инструкций, команд, 443
 регистровый файл, 443
 элементы памяти, 443
 системные функции (processor control), 413
 набор регистров, 362
 ср. с архитектурой x86, 411,8
- MOSFET.** *См.* Полевые транзисторы металл-оксид-полупроводник
- MPSSE.** *См.* Multi-Protocol Synchronous Serial Engine
- MSB.** *См.* наиболее (самый) значимый байт
msb. *См.* наиболее (самый) значимый бит
- MSI chips.** *См.* Средней степени интеграции
- MTBF.** *См.* Среднее время наработки на отказ
- mul,** умножение, результат 32 бита, 378
mult, умножение, результат 64 бита, 378
- Multi-Protocol Synchronous Serial Engine (MPSSE),** 651, 651
- multu,** 414
- Мих.** *См.* Мультиплексоры
- myDAQ,** 650
- N**
- nor,** 411
- nor,** 375
- n-МОП-транзисторы,** 65–68, 66–67
- O**
- OFF,** 58. *См. также* 0, LOW
- ON,** 58. *См. также* 1, HIGH
- or,** 375
- ori,** 375–376
- ОТР.** *См.* Однократно программируемая логика
- P**
- Parity gate.** *См.* XOR
- PC.** *См.* Счетчик команд или Персональный компьютер
- PCB.** *См.* Печатная плата
- PCI express (PCIe),** 645
- PCI.** *См.* Peripheral Component Interconnect
- PCSrc,** 466, 467–468, 468
- PCWrite,** 464, 468
- Peripheral Component Interconnect (PCI),** 647
- PIC32** микроконтроллер (PIC32MX675F512H), 589–593. *См. также* Встроенные системы В/В
- PLAs.** *См.* Programmable logic arrays
- Plastic leaded chip carriers (PLCCs),** 686
- PLCCs.** *См.* Plastic leaded chip carriers
- PLDs.** *См.* Programmable logic devices
- PLL.** *См.* Петля фазовой синхронизации
- POS.** *См.* Конъюнктивная форма
- Power processor element (PPE),** 530
- PPN.** *См.* Номер физической страницы
- printf,** 750–752
- Programmable logic arrays (PLA),** 107, 332–334, 673–675
 реализация на транзисторах, 342
- Programmable logic devices (PLD),** 673
- PROMs.** *См.* Программируемые ПЗУ
- p-МОП-транзисторы,** 65–68, 66
- R**
- rand,** 757–758
- ReadData,** 448
- RegDst,** 451, 454, 468
- RegWrite,** 448, 455, 468, 484, 485
- RF.** *См.* Регистровый файл
- ROM.** *См.* Постоянное запоминающее устройство
- RPM.** *См.* обороты в минуту
- RS-триггеры,** 151–153, 152
- RS-232,** 605–606
- S**
- SATA.** *См.* Serial ATA
- sb,** сохранить байт. *См.* Загрузки
- scanf,** 755
- SCK.** *См.* Serial Clock
- SDI.** *См.* Serial Data In
- SDO.** *См.* Serial Data Out
- SDRAM.** *См.* Синхронное динамическое ОЗУ
- Serial ATA (SATA),** 649
- Serial Clock (SCK),** 597
- Serial Data In (SDI),** 597
- Serial Data Out (SDO),** 597
- Serial Peripheral Interface (SPI),** 596–603
 связь между PIC32 и FPGA, 600
 порты
 Serial Clock (SCK), 597
 Serial Data In (SDI), 597
 Serial Data Out (SDO), 597
 поля регистра, 598
 ведомое устройство и его временная диаграмма, 601
 сигналы, 597
- set if less than (slt),**
 схема, 306
 на ассемблере MIPS, 384–385
- set if less than immediate (slti),** 414

- set if less than immediate unsigned (sltiu), 415
 set if less than unsigned (sltu), 414
 SFR. *См.* Регистры специального назначения
 sh, Загрузка полуслова. *См.* Загрузки
 Short path, 126, 130
 SIMD. *См.* Single instruction multiple data
 SIMD-блоки, 518, 525, 537
 sll, 376
 sllv, 377
 slt, установить если меньше, 384–385
 SLT. *См.* установить если меньше
 slti, 414
 sltiu, 414
 sltu, 414
 SOP. *См.* Дизъюнктивная форма
 SPECINT2000, 478
 SPI. *См.* Serial Peripheral Interface
 Spinstepper функция, 643
 SPIxCON, 598
 SRAM. *См.* Статическое ОЗУ
 srand, 757–758
 srl, 376
 srlv, 376–377
 stdio.h, С библиотека, 752–758. *См. также*
 Стандартные библиотеки
 stdlib.h, С, 757–759. *См. также* Стандартные
 библиотеки
 string.h, С библиотека, 760
 sub, 358
 subu, 414
 sw, сохранить слово, 364–365. *См. также*
 Сохранения
 SystemVerilog, 221–280. *См. также* Языки
 описания схем
 доступ к части шины, 239, 243
 плохой синхронизатор с блокирующими
 присваиваниями, 263
 манипуляции битами, 239
 блокирующие и неблокирующие
 присваивания, 252–253, 260–263
 case операторы, 254–255, 258
 комбинационная логика на, 226–243, 251–262,
 272–275
 комментарии, 227
 условное присваивание, 230–232
 типы данных, 268–272
 дешифраторы, 255–256, 275
 задержки (при), 240
 КА делителя на 3, 264–265
 конечные автоматы (КА), 262–267
 КА Мили, 267
 КА Мура, 262, 266
 полный сумматор, 234
 always/process использующий, 253
 с неблокирующими присваиваниями, 262
 история, 224
 операторы if, 256–259
 внутренние сигналы, 232–234
 инверторы, 227, 252
 защелки, 250
 логические вентили, 228–229
 мультиплексоры, 230–232, 241–244, 272–273
 умножитель, 272
 числа, 236–237
 операторы, 236
 параметризованные модули, 272–275
 N:2N декодер, 274
 N-bit мультиплексоры, 273–274
 N-входовой вентиль И, 275
 схема приоритетов, 258
 с пренебрежениями, 259
 операторы сокращения, 230–231
 регистры, 245–249
 с сигналом разрешения, 248
 со сбросом, 245–247
 последовательная логика, 245–250, 264–268
 дешифратор семисегментного индикатора 255
 симуляция и синтез, 224–226
 структурные модели, 241–244
 синхронизатор, 250
 среда тестирования, тестовое окружение 275–279,
 506–507
 самопроверка, 277
 простой, 276
 с файлом тестовых векторов, 279–280
 буфер с третьим состоянием (тристабильный),
 237
 таблицы истинности с неопределенными и
 плавающими входами, 238, 239
 Z-состояние и X-состояние, 237–239, 259
- T**
 Thin Quad Flat Pack (TQFP), 590
 Thin small outline package (TSOP), 686
 TLB. *См.* Буфер ассоциативной трансляции
 TSOP. *См.* Thin small outline package
 typedef, 747–748
- U**
 UART. *См.* Universal Asynchronous Receiver
 Transmitter
 Universal Serial Bus (USB), 330, 606, 646–647
 USB 1.0, 645
 USB 2.0, 645
 USB 3.0, 645
 USB. *См.* Universal Serial Bus
 USB-подключение, 652–653
 FTDI, 651
 модуль UM232H, 651
- V**
 Vanity Fair (Кэрролл), 118
 VCC, 58. *См. также* Напряжение питания
 VDD, VDD, 57, 58. *См. также* Напряжение
 питания
 Verilog. *См.* SystemVerilog
 Very High Speed Integrated Circuits (VHSIC), 224.

- См. также* VHDL
- VGA. *См.* монитор VGA
- VHDL. *См.* VHSIC Hardware Description Language
- VHSIC Hardware Description Language (VHDL), 222–224
- доступ к частям шин, 239, 244
 - плохой синхронизатор с блокирующими присваиваниями, 263
 - манипуляции битами, 239
 - блокирующие и неблокирующие присваивания, 252–253, 260–263
 - операторы case, 254–254, 259
 - комбинационная логика, 226–240, 251–262, 272–275
 - комментарии, 227
 - словное присваивание, 230–231
 - типы данных, 268–272
 - дешифраторы, 255–256, 274
 - задержки (при моделировании), 241
 - делитель на 3, КА, 264–265
 - конечные автоматы (FSMs), 264–268
 - Мили, КА, 264, 267–268
 - Мура, КА, 264, 266
 - полный сумматор, 234
 - с always / process, 253
 - с неблокирующими присваиваниями, 261
 - история, 224
 - операторы if, 256
 - внутренние сигналы, 232–234
 - инверторы, 228, 253
 - зашелки, 250
 - логические элементы, ЛЭ, 226–228
 - мультиплексор, 231–233, 242–246, 273–274
 - умножитель, 272
 - числа, 235–236
 - операторы, 235
 - параметризованные модули, 272–275
 - дешифратор $N:2^N$, 274
 - N -битный мультиплексор, 273, 274
 - N -входовой элемент И, 275, 275
 - схема приоритетов, 258
 - операторы сокращения, 230–231
 - работа с незначимыми битами, 259
 - операторы сокращения, 230–231
 - регистры, 245–250
 - разрешение, 248
 - со сбросом, 245–247
 - последовательностная логика, 245–250, 262–266
 - Дешифратор семисегментного индикатора, 255
 - симуляция и синтез, 224–226
 - структурное моделирование, 241–244
 - синхронизатор, 250
 - среда тестирования, тестовое окружение, 275–279, 506–507
 - самопроверка, 277
 - простая, 276
 - с файлом тестовых векторов, 278–279
 - тристабильный буфер, 237
 - таблицы истинности с неопределенными и плавающими входами 237, 238
 - Z-состояние и X-состояние, 237–239, 259
- VHSIC. *См.* Very High Speed Integrated Circuits
- Video Graphics Array (VGA). *См.* монитор VGA
- VSS, 58
- W**
- WAR. *См.* Запись после чтения
- WAW. *См.* Запись после записи
- Wi-Fi, 646
- X**
- X. *См.* Конфликт, Безразличность
- x86
- архитектура, 417–425
 - условия ветвлений, 422
 - кодировка команд, 421–425, 423
 - команды, 414–417, 415
 - режимы адресации памяти, 419
 - операнды, 418–421
 - регистры, 418
 - флаги состояния, 420
 - ср. с MIPS, 418
 - системы кэша, 652–655
 - система памяти, развитие, 652–653
 - микроархитектура, 532–539
 - развитие, 532–534
 - программируемый ввод-вывод, 656
 - регистры, 418
 - виртуальная память, 655
 - защищенный режим, 655
 - реальный режим, 655
- Xilinx FPGA, 335–337
- хог, 375
- хогі, 375–376
- Z**
- Z. *См.* Плавающий
- A**
- A/Ц преобразование, 614–616
- регистры, 616
- Абстракция, 33–34
- цифровая. *См.* Цифровая абстракция
- Автоматы Мили, 167, 167, 177
- Таблица переходов и выходов, 178
 - Диаграммы переходов, 177
 - временные диаграммы, 179
- Адрес. *См. также* Память
- физический, 576–582
 - трансляция, 576–582
 - виртуальная, 573. *См. также* Виртуальная память
- Адресация относительно счетчика команд, 401–402
- Аккумулятор, 424
- Активный низкий, 116–117

- АЛУ. См. Арифметико-логическое устройство
Амдал, Джин, 557
- Американский стандартный код для обмена информацией (ASCII), 388, 389, 723, 743–744
- Анализ производительности, 444–446. См. также Среднее время доступа в память многотактный процессор MIPS, 478–480 конвейерный процессор MIPS, 496–498 сравнение процессоров, 497 одноктактный процессор MIPS, 458–460
- Аналитическая машина, 38, 39
- Аналоговый ввод-вывод, 614–620
А / Ц преобразование, 614,2–615,3
Ц / А преобразование, 615,3–619,7
Широтно-импульсная модуляция (ШИМ), 620–621
- Аналого-Цифровые преобразователи (АЦП), 614–616
- Анод, 64
- Аппаратное подтверждение связи, 605
- Аргументы командной строки, 762–763
- Аргументы, 391–392, 399–400, 730
передача по ссылке, 737
передача по значению, 737
- Арифметика
операции С, 726–728
схемы, 293–310
HDL, операторы, 235
MIPS, инструкции, 373–377
пакованная, 525
- Арифметико / логическое устройство (АЛУ), 304–308, 447
реализация, 305
в процессоре MIPS, 452–455. См. также ALUControl, ALUOp
- Архитектура компьютера, Хеннесси и Паттерсон, 514
- Архитектура, 355–427, 708–712
MIPS
режимы адресации, 400–402
язык ассемблера, 358–366, 708–711
команды, 708–712
машинный язык, 367–372
операнды, 360–367
x86, 417–426
- Архитектурное состояние, 373, 440–442
- Асимметричные многопроцессорные системы.
См. Гетерогенные многопроцессорные системы
- Асинхронная последовательная связь, 604.
См. также Универсальный асинхронный приемопередатчик
- Асинхронно сбрасываемые триггеры,
определение 158
HDL, 246–248
- Асинхронные схемы, 163–166
- Ассемблер, 407–408, 761
- Ассоциативность
в Булевой алгебре, 102, 103
в кэше, 557, 563–565
- АЦП. См. Аналого-цифровой преобразователь
- Б**
- Баг в команде деления с плавающей точкой (FDIV), 224
- Базовая адресация, 401
- Базовый адрес, 363–364, 370, 385–387, 390
- Байпас, обход, 486–488. См. также Конфликты Байпас, через, 486. См. также Байпас, обход
- Байт, 45–46, 388–389. См. также Символ наименее значимые, 45–46 наиболее значимые, 45–46
- Байтовое смещение, 561
- Беззнаковый умножитель, 272
- Безразлично (X), 110, 124–126, 259
- Безусловные переходы, 379–380
- Беспроводная связь, Bluetooth, 632–633
- Биполярная система подачи сигналов, 606
- Биполярные транзисторы, 62
- Биполярный шаговый двигатель, 640, 641–642
AIRPAX LB82773-M1, 642, 643
прямой ток двигателя, 642
- Бистабильная ячейка, 150
- Бит для знака, 48
- Бит достоверности (V), 560
- Бит изменения (D), 572
- Бит использования (U), 568
- Бит, 39
изменения, D, 572
наименее значимый бит, 45, 46
наиболее значимый бит, 45, 46
знак, 47
использования, U, 568
достоверности, 560
- Битовые операторы, 227–229
- Блок вычислений с плавающей запятой (FPU), 316, 536
- Блок вычислений с плавающей запятой, 536
- Блок логических матриц (LAB), 336
- Блок обнаружения конфликтов, 487–498
- Блок, Строка, смещение, 565–566
- Блок, 557
- Блока, строки длина, (b), 557, 565–566
- Блокирующие и неблокирующие присваивания, 252–253, 258–262
- Блокирующие и неблокирующие присваивания, 252–253, 258–262
- Булева алгебра, 99–105
аксиомы, 100
Упрощение уравнений, 105–106
теоремы, 100–104
- Булева логика, 40. См. также Булева алгебра, Логические элементы
- Булевы теоремы, 100–104
ассоциативность, 102
склеивание, 102
дополнительность, 101

- согласованность, 102, 103
- поглощение, 102
- Де Моргана, 102–103
- дистрибутивность, 102
- идемпотентность, 101
- идентичность, 101
- инволюция, 101
- нулевой элемент, 101
- Булевы уравнения, 95–99
 - конъюнктивная нормальная форма (POS), 98
 - дизъюнктивная нормальная форма (SOP), 96–98
- Буль, Джордж, 40
- Буфер ассоциативной трансляции (TLB), 580–581
- Буфер с тремя состояниями, 115–116, 237
 - HDL для, 237–238
 - мультиплексор, 125–126, 134–136
- Буферы, 51
 - отсутствие, 159
 - с тремя состояниями, 115–116
- В**
- В/В. См. Системы ввода/вывода
- Ввод-вывод, отображенный в память
 - дешифратор адреса, 587
 - обмен с устройствами В/В, 587–588
 - аппаратные средства, 587
- Ведомая защелка 155. См. также D-триггер
- Ведущий, защелка, 155
- Векторный процессор, 518
- Вентили, ЛЭ
 - И, 54, 56, 172
 - буфер, 54
 - многоходовой, 56–57
 - И-НЕ, 54, 69
 - ИЛИ-НЕ, 54–55, 151, 172
 - НЕ, 53
 - ИЛИ, 54
 - транзисторный уровень. См. Транзисторы
 - Исключающее ИЛИ-НЕ, 56
 - Исключающее ИЛИ, 55
- Вентиль-приемник, 57
- Ветвления, буфер целевых адресов, 517
- Ветвления, Переходы
 - условные, 378–379
 - безусловные (jump), 379–380
- Взятие дополнения до двух, 49–50
- Виртуальная память, 554, 573–583
 - трансляция адреса, 576–580
 - сравнение терминов кэша, 575
 - защита памяти, 582
 - многоуровневые таблицы страниц, 584–585
 - страничная ошибка, 575
 - номер страницы, 576
 - смещение страницы, 576
 - таблица страниц, 579–580
 - страницы, 575
 - стратегии замещения, 583
 - буфер ассоциативной трансляции (TLB), 581–582
 - стратегии записи, 571–572
 - x86, 652. См. также x86
- Виртуальные страницы, 575
- Виртуальный адрес, 575
 - пространство, 582
- Виртуальный номер страницы (VPN), 577
- Витмур, Джорджиана, 38
- Вложенные операторы if/else, 733
- Внеочередное выполнение, 525
- Внешние устройства микроконтроллера, 621–644
 - Bluetooth беспроводная связь, 632–633
 - символьный ЖК дисплей, 622–625
 - управление, 625–625
 - параллельный интерфейс, 623
 - управление двигателями, 633–634
 - VGA монитор, 625–631
- Временная локальность, 551, 557–558, 561, 567
- Временной параллелизм, 206–207
- Временные регистры, 361, 396–397
- Временные характеристики
 - комбинационной логики, 130–137
 - задержка. См. Задержка распространения, Задержка реакции
 - импульсные помехи. См. Импульсные помехи последовательных схем, 185–204
 - анализ. См. Временные характеристики, анализ
 - расфазировка тактовых сигналов. См. Расфазировка тактовых сигналов динамическая дисциплина, 187–188
 - метастабильность. См. Метастабильность время разрешения. См. Время разрешения временные характеристики системы. См. Временные характеристики, анализ
 - анализ, 185–200
 - расчет времени цикла. См. Ограничение времени предустановки
 - ограничение времени удержания. См. Ограничение времени удержания
 - ограничение макс. задержки. См. Ограничение времени предустановки, 29
 - ограничение мин. задержки. См. Ограничение времени удержания
 - многотактный процессор, 476–477
 - конвейерный процессор, 498
 - ограничение времени предустановки. См. Ограничение времени предустановки
 - однотактный процессор, 458–459
 - с расфазировкой. См. Расфазировка тактовых сигналов
- Временный регистр ассемблера (\$at), 405, 2
- Время выборки, 615
- Время выборки. См. Время захвата,
- Время выполнения, 445
- Время разрешения, 198–199
 - вычисление, 201–204 См. также Метастабильность
- Время цикла. См. Период тактового сигнала

- Встроенные системы В/В (ввод/вывод), 588–643
 аналоговый В/В, 614–620
 А/Ц преобразование, 615–617
 Ц/А преобразование, 617–619
 цифровой В/В, 595–597
 общего назначения В/В (GPIO), 595–597
 прерывания, 612–614
 ЖКИ. *См.* Жидко-кристаллические индикаторы
 микроконтроллера, внешние устройства
 621–642
 двигатели. *См.* Двигатели
 микроконтроллер PIC32, 589–593
 последовательный В/В, 596–609. *См. также*
 Последовательный В/В
 таймеры, 610–611
 VGA монитор. *См.* VGA монитор, 625
- Выборочного присваивания сигнала операторы,
 232
- Вызов функции factorial, 397–398
 стэк при, 398
- вызовы процедур. *См.* вызовы функций
- Вызовы функций, 390–398, 729–730
 аргументы, 390–391, 729
 листовые, 397
 соглашения об именовании, 731
 нелитовые, 392
 Оберегаемые и необерегаемые регистры, 396–399
 прототипы, 730–731
 рекурсивные, 398–399
 return, 392–393, 729
 стек, применение, 393–399. *См. также* Стэк
 без входов или выходов, 392, 729,7
- Высокого уровня языка программирования, 357,
 714
 компиляция, ассемблирование и загрузка,
 404–409
 трансляция в ассемблер, 361
- Высокопроизводительные микропроцессоры, 513
- Вычитание, 50, 302, 358
 Команды для чисел со знаком и без знака,
 413–414
- Вычитатель, 302–303
- Г**
- Генерация задержки счетчиками, 611–612
- Гетерогенные мультипроцессоры, 529–531
- Главная память, 554
- Главный дешифратор, 452–457
 HDL для, 502
- Глобальный указатель (\$gp), 405
- Головка считывания-записи, 574
- Гомогенные мультипроцессоры, 528
- Гонки, 163–164, 164
- Граничные коэффициенты передачи (Точки
 единичного усиления), 60
- Графики сигналов, 225
 с задержками, 241
- Графические процессоры (GPU), 532
- Графические ускорители, 539
 Грэй, Фрэнк, 118
- Д**
- Датчик угла поворота, 637, 637
- двигатели постоянного тока, 634–638, 635
 Н-мост, 635, 636
 датчик угла поворота, 634–637
- Двигатели
 постоянного тока, 634–638
 Н-мост, 636
 серво, 635, 638–640
 шаговые, 634, 640–644
- Двоичное в десятичное, преобразование, 40,
 40–41
- Двоичное в шестнадцатиричное, преобразование,
 42
- Двоичное кодирование, 170–171, 175–178
 для счетчика-делителя на 3, 176–178
 для КА светофора, 168–169
- Двоичное сложение, 46–47
- Двоично-кодированная десятичная система
 (BCD), 315
- Двоичные числа со знаком, 47–52
- Двоичные числа
 со знаком, 47–51
 без знака, 40–42
- Двухбитный динамический предсказатель
 переходов, 517
- Двухуровневая логика, 110
- Де Морган, Август, 103
- Де Моргана теорема, 103–104
- Де Моргана, Теорема, 103
- Декодер АЛУ, 452–454
- Декомпозиция конечных автоматов 180–182
- Деление с плавающей точкой (FDIV), 316
- Деление
 схемы, 309–310
 инструкция MIPS, 378
 инструкции MIPS со знаком и без знака, 414
- Делитель на 3 счетчик
 проектирование, 173–175
 HDL для, 264–265
- Делитель, 309–310
- Десятичная система счисления, 40
- Десятичного числа в двоичное, преобразование,
 42
- Десятичного числа в шестнадцатеричное,
 преобразование, 45
- Дешифратор семисегментного индикатора,
 121–125
 HDL для, 255–256
 с безразличиями, 124–125
- Дешифраторы
 определение, 128–129
 HDL для
 поведенческий, 255–256
 параметризованный, 274
 логика, реализация на, 129–130

- семисегментный. *См.* Семисегментного индикатора дешифратор
- Дешифрация, стадия, 481–483
- Дизъюнктивная форма (SOP), 96–98
- Динамические предсказатели переходов, 516
- Динамическое оперативное запоминающее устройство (DRAM), 325, 326, 549–553, 648
- Динамическое потребление энергии, 74
- Динамическое распределение памяти (malloc, free), 748–749
- карта памяти MIPS, 404
- Диоды, 64
- p-n*-переход, 64
- Директивы ассемблера, 407
- Дискретно меняющиеся переменные, 38
- Дисциплина динамическая, 187–199. *См. также* Временные характеристики
- Дисциплина динамическая, 187–196. *См. также* Временные характеристики статическая, 187–196. *См. также* Допускаемые уровни шумов
- Диэлектрик, 65
- Длинные конвейеры, 514–515
- Доля промахов, 555–557
- и время доступа, 555
- Дополнение нулями, 306, 371, 375–376, 415
- Дополнительности теорема, 101
- Дополнительный код, числа, 48–50
- Допускаемые уровни шумов, 58–61, 58
- расчет, 59
- Драйвер устройства, 587–588, 608
- Е**
- Емкость, кэша, 556–557
- Ж**
- Жесткий диск, 554–555. *См. также* Накопитель с жестким диском
- Жидко-кристаллические дисплеи (LCD), 622–625
- ЖКИ. *См.* Жидко-кристаллические индикаторы
- З**
- Загрузка из регистра сопроцессора 0 (mfc0), 413, 511–513. *См. также* Исключения
- Загрузки, 415
- базовая адресация при, 401
- загрузить байт (lb or lbu), 365, 389–390, 415
- загрузить полуслово (lh or lhu), 415
- загрузить слово (lw), 364–367
- Задержка распространения, 130–134. *См. также* Критический путь
- Задержка реакции, отклика, 131–135. *См. также* Короткий тракт
- Задержки, ЛЭ. *См.* Задержки распространения в HDL (только модели), 239–240
- Заказные ИС (ASIC), 676
- Закон Амдала, 557
- Закрытое накопление времени, 611
- Замещение страниц, 583
- Запись после записи, (WAW) конфликт, 522
- Запись после чтения, (WAR) конфликт, 522–523. *См. также* Конфликты
- Зарезервированный сегмент, 406
- Защита памяти, 582–583
- Земля, Общий (GND), 57
- Символ, УГО, 69
- Знаковое расширение (знака) 51, 370
- HDL для, 505
- И**
- Идиомы, 226
- Иерархичность, 36
- Иерархия памяти, 553–554
- ИЛИ-НЕ вентиль, 55–56, 151, 172, 669
- ИС (7402), 669
- КМОП, 70
- псевдо-*n*-МОП логика, 72
- таблица истинности, 56
- Импульсные помехи, гонки, 136–139
- Инвертированный ИЛИ-И (OAI) вентиль, 85
- Инверторы, 53, 162, 227. *См. также* Вентиль НЕ скрещенные, 150, 151
- в HDL, 227, 252
- И-НЕ (7400), 669
- И-НЕ вентиль, 55, 55, 69
- КМОП, 69–70, 69–70
- Инициализация массивы в С, 738–739
- переменные в С, 724
- Инструкции (команды) для чисел с плавающей точкой, MIPS, 415–416
- Инструкции сдвига, 375–376, 375
- Инструкции типа R, 368–369
- Инструкции, команды MIPS, 357–416, 709–712
- арифметические, 360–361, 366, 378, 413–414, 711–712
- ветвление, 378–380
- с плавающей точкой, 415–416, 711
- для доступа в память. *См.* Загрузки, Сохранение
- F-типа, 416
- I-типа, 369, 369–371
- J-типа, 371, 371
- логические, 374–377
- умножение и деление, 378, 414
- R-типа, 368–369, 710–711
- установить, если меньше, 384–385, 414
- со знаком и без знака, 413–414
- Интегральные схемы (ИС), 685–686
- Интерфейс памяти, 549–550
- ИС малой степени интеграции (SSI), 668
- ИС микропроцессора 80386, 533, 534
- ИС микропроцессора 80486, 534, 535
- ИС микропроцессора Core Duo, 538
- ИС микропроцессора Core i7, 538
- ИС средней степени интеграции (MSI), 670
- ИС. *См.* Интегральные схемы

- Искл. ИЛИ, элемент, 55
 Искл. ИЛИ-НЕ, элемент, 56
 Исключения, 411–413, 511–515
 исключения. *См.* регистр причины исключения
 Коды причины исключения, 412
 ЕРС. *См.* регистр специального назначения
 ЕРС,
 обработчик, 412
 поддержка в процессоре, 511–515
 схемы, 512–513
 контроллер, 514–515
 Исполняемый файл, 409
 Исследование законов мышления (Буль), 40
- К**
- КА. *См.* Конечные автоматы
 Карно, Морис, 117
 Карта памяти
 MIPS, 404–405, 410, 587
 PIC32, 590–591
 Карты Карно (K-maps)
 Карты Карно (K-maps), 117–126, 137–139, 170
 минимизация логики с, 119–125
 простая импликанта, 105, 117–121, 136–137
 дешифратор семисегментного индикатора,
 121–123
 с безразличными переменными, 124–125
 без импульсных помех, 138
 Катод, 64
 Килобайт (кБ), 46
 Килобит (кб/кбит), 46
 Кластер мультипроцессорный, 528
 КМОП. *См.* Логика на комплементарных
 транзисторах металл-окисел-полупроводник
 Код операции, (Opcode), 368, 708–709
 Код операции. *См.* Opcode, Операторы
 в C, 726–729
 в HDL, 227–235
 битовые, 227–231
 приоритет, 235
 сокращения, 230–231
 таблица, 235
 тернарный, 231–232
 Кодирование One-cold, 176
 Кодирование состояний, КА, 173–175, 178. *См.*
также Двоичное кодирование, кодирование
 one-cold, Прямое (one-hot) кодирование
 Кодировка команд, x86, 417–423, 418
 Коды Грэя, 118
 Коды условий. *См.* Флаги состояния
 Кольцевой генератор, 162, 162
 Команд на такт (IPC), 445
 Команды для чисел со знаком и без знака,
 413–414
 Команды переменного сдвига, 377
 Команды, x86, 417–425
 Комбинационная композиция, 94
 Комбинационная логика, 223
 проектирование, 91–147
 Булева алгебра, 99–106
 Булевы уравнения, 96–99
 «строительный блок», функциональный
 узел, 125–130, 294–312
 задержки, 130–135
 безразличные, 124–125
 Карты Карно (K-maps), 117–125
 многоуровневая, 106–115
 приоритет, 96
 временные характеристики, 130–137
 двухуровневая, 110
 X (конфликт). *См.* Конфликт
 X (безразлично). *См.* Безразлично (X)
 Z's (плавающая). *См.* Плавающая (Z)
 HDL и. *См.* Языки описания аппаратуры
 таблицы истинности с безразличиями, 110,
 124–125, 259
- Комментарии
 в C, 719
 в ассемблере MIPS, 358
 в SystemVerilog, 227
 в VHDL, 227
 Компаратор величины, 303
 Компаратор равенства, 303
 Компараторы, 303–305
 Компилятор и опции командной строки, 760–762
 Компилятор, 406–407
 для C, 718–719, 760–761
 Компиляция, в C, 718–719, 760–761
 Компоновщик, 408–409
 Компьютер со сложным набором команд (CISC),
 362, 415
 Компьютеры со сложным набором команд
 (RISC), 359
 Конвейеризация, 205–207. *См. также*
 конвейерный процессор MIPS
 Конвейерный процессор MIPS, 479–498
 абстрактное представление, 481
 управление, 484–485
 тракт данных, 482–483
 описание, 479–482
 конфликты, 484–495. *См. также* Конфликты
 производительность, 496–499
 пропускная способность, 480
 Конденсаторы, 65
 Конечные автоматы (КА, FSM), 166–184, 259–263
 восстановление по электрической схеме, 182–185
 делитель на 3 на КА, 174–176, 265–266
 декомпозиция, 180–183, 184
 в HDL, 259–263
 конфигурация ЛЭ для, 338–339
 КА Мили, 176–178
 КА Мура, 176–178
 многотактное управление, 467–479, 476, 478
 КА распознавателя – битового шаблона
 улитки, 177–179, 266–267
 кодирование состояний, 173–175. *См. также*
 Двоичное кодирование, кодирование one-

- cold, Прямое (one-hot) кодирование диаграммы переходов, 168, 169
КА светофора, 166–173
- Константы
в С, 719–720
в ассемблере MIPS, 366, 377. *См. также* Непосредственные данные
- Контекста переключения, 527
- Конфигурируемые логические блоки (CLB), 336, 673. *См. также* Логические элементы, вентили
- Конфликт (X), 114–115
- Конфликт ветвления. *См.* Конфликт управления
- Конфликт чтения после записи (RAW), 485, 522. *См. также* Конфликты
- Конфликты данных, 484–491
- Конфликты управления, 486, 491–494
- Конфликты. *См. также* Блок обнаружения конфликтов
- конфликты управления, 486, 491–494
- конфликты данных, 486–491
- чтения после записи (RAW), 485, 523
- Разрешение
- конфликты управления, 491–494
- пересылка через байпас, 486–488
- приостановка конвейера, 488–491
- Запись после чтения (WAR), 524
- Запись после записи (WAW), 524–525
- Конъюнктивная форма (POS), 98
- Корпуса с двумя рядами выводов (DIP), 65, 668, 685–686
- Корпуса, кристаллы, 685–687
- Кремния диоксид (SiO₂), 65
- Кристалл, чип, 65
- мультипроцессоры, 528
- Критический путь, 132–135, 458
- Кружочек, Пузырёк, 51, 104, 489,9
- давить, 104–105, 111–113
- Куча, 405
- Кэш последовательностей микроопераций, 537
- Кэш прямого отображения, 558–562, 560
- Кэши, 557–573
- адресные поля,
- смещение блока, 565–566
- байтовое смещение, 560
- номер набора, 560
- тег, 560
- улучшенный, 569–573
- эволюция, в MIPS, 573
- многоуровневый, 569
- неблокирующий, 654
- организация, 568
- прямого отображения, 559–567
- полностью ассоциативный, 564–565
- многосекционный наборно-ассоциативный, 563–564
- параметры,
- блока, 557
- размер блока, 557, 565–566
- емкость (C), 556–557
- степень ассоциативности (N), 556, 563
- число наборов (S), 557
- производительность по попаданиям, 554–557
- процент попаданий, 554–557
- промах, 554–557, 571
- емкость, 571
- неизбежный, 571
- конфликт, 563, 571
- штраф, 565
- процент промахов 554–557
- уменьшение, 571–572
- процент промахов и параметры кэша, 571–572
- стратегия замены 567–568
- биты состояния
- бит изменения (D), 572
- бит использования (U), 568
- бит достоверности (V), 560
- стратегия записи, 571–573
- отложенная запись, 571–573
- сквозная запись, 571–573

Л

- Латентность в два такта, lw, 488
- Латентность, 205–208, 479–481, 488
- Линии передачи, 690–703
- характеристический импеданс (Z₀), 700–701
- вывод, 700–701
- согласованная нагрузка, 691–693
- рассогласованная нагрузка, 695–698
- нагрузка холостого хода, 693–694
- коэффициент отражения (kr), 701–702
- вывод, 701–702
- параллельная и последовательная нагрузка, 698–700
- нагрузка короткого замыкания, 694
- когда использовать, 700
- Линия битов, 322
- Линия слов, 322
- Листовая (Терминальная) функция, 397
- Литерал, 95, 139
- Ловушки, 412
- Логика на комплементарных транзисторах металл-окисел-полупроводник (КМОП), 62–70
- Логика
- “Катание шарика”, “Пихание кружочка”,
- Перенос инверсии, 112–114
- комбинационная. *См.* Комбинационная логика семейства, серии, 682–684
- элементы. *См.* Логические элементы
- Минимизация аппаратуры. *См.* Упрощение уравнений и Минимизация аппаратуры многоуровневая. *См.* Многоуровневая комбинационная логика
- программируемая, 671–679
- последовательная. *См.* Последовательная логика

- на уровне транзисторов. *См.* Транзисторы двухуровневая, 110
- Логика, семейства, 58, 682–685
совместимость, 61
логические уровни, 58
характеристики, 682, 684
- Логические вентили, 53–56, 228, 668
- И. *См.* И
элемент И-ИЛИ, 85
многоходовые элементы, 56–57
- И-НЕ. *См.* И-НЕ вентиль
ИЛИ-НЕ. *См.* ИЛИ-НЕ вентиль
ИЛИ. *См.* ИЛИ вентиль
ИЛИ-И-НЕ (ОАИ) вентиль, 85
с задержкой в HDL, 239–240
Искл. ИЛИ-НЕ. *См.* Искл. ИЛИ-НЕ, вентиль
Искл. ИЛИ. *См.* Искл. ИЛИ, вентиль
- Логические инструкции, 374–375
- Логические уровни, 57–58
- Логические элементы (ЛЭ), 335–339
в Cyclone IV, 337
функции, реализуемые с, 338–339
- Логический вентиль ИЛИ (OR gate), 54
- Логический синтез, 225–226, 226
- Логическое моделирование, 224–225
- Локальность, 551
- Локальные переменные, 399–400
- ЛЭ. *См.* Логический элемент
- М**
- Макетные платы, 686–687
- Макстермы, 95
- Манипуляции битами, 239
- Мантисса, 315–316
- Массивы, 386–390, 738–745
доступ к, 386–388, 738
как входной аргумент, 739–741
байты и символы, 386–388, 743–745
сравнение или присвоение, 744
объявление, 738
индексирование, 385–387, 738–742
инициализация, 738–739
многомерный, 741–742
- Масуока, Фудзэ, 329
- Матрицы логических элементов, 332–340. *См. также* Programmable logic arrays и Field programmable gate arrays
на уровне транзисторов, 340–341
- Матрицы памяти, 321–330. *См. также* Память запоминающий элемент, 322–327
HDL для, 331–333
логика на, 330–331
организация, 321–323
- Машинный код, ассемблер и, 506
- Машинный язык, 367–372
форматы, 367–370
F-тип, 416
I-тип, 369, 369–371
J-тип, 371, 371
- R-тип, 368–369
рашифровка, 371–372
хранимые программы, 372–373, 373
трансляция с языка ассемблера в машинный, 369
- Метастабильность, 197–203
метастабильное состояние, 151, 197
время разрешения, 198–199, 201–204
синхронизаторы, 199–201
- Микроархитектура, 421–541. *См. также*
Архитектура
улучшенная. *См.* Улучшенная микроархитектура,
архитектурное состояние. *См.* Архитектурное состояние
описание, 440–443
процесс проектирования, 441–443
представление в HDL, 498–510
многотактный процессор. *См.* MIPS
многотактный процессор
анализ производительности, 444–446. *См. также* Анализ производительности
конвейерный процессор. *См.* MIPS
конвейерный процессор
однотактный процессор. *См.* MIPS
однотактный процессор
x86, 532–540
развитие, 533
- Микроконтроллер, 588
PIC32 (PIC32MX675F512H), 589–593, 590
в корпусе TQFP с 64 выводами, 591
схема минимальной рабочей конфигурации, 592
с ПК, последовательный интерфейс, 608
цоколевка, 591
карта виртуальной памяти, 591
- Микроконтроллерное устройство (MCU), 588
- Микрооперации (micro-ops), 535
- Микропроцессоры, 31, 45, 355
архитектурное состояние, 373
разработчики, 514
высокопроизводительные, 513
- Миллионы операций в секунду, 479
- Минимизация аппаратуры, 111–112. *См. также*
Минимизация уравнений
- Минимизация уравнений
с Булевой алгеброй, 105–106
с картами Карно. *См.* карты Карно
- Минтермы, 95
- Многокристальный модуль (МКМ), 653
- Многопоточность, 526
- Многопоточный процессор, 526–527
- Многотактный процессор MIPS, 460–479
устройство управления, 468–476
тракт данных, 460–467
производительность, 478–481
- Многоуровневая комбинационная логика, 110–114.
См. также Логика
- Многоуровневые таблицы страниц, 584–586

- множимое, 308
 Модули, на HDL
 поведенческий и структурный, 222–223
 параметризованные модули, 272–275
 Модуль памяти с двумя рядами выводов (DIMM), 648
 Модульность, 36
 Монитор VGA (Video Graphics Array), 625–631
 цоколевка разъема, 627
 драйвер для, 629–632
 МОП-транзисторы. *См.* Полевые транзисторы
 металл-оксид-полупроводник
 Мощность в статическом режиме, 74
 МСИ ИС. *См.* Малой степени интеграции
 Мультиплексоры, 125–128
 определение, 125–126
 HDL для
 поведенческая модель, 231–233
 параметризованные N-битные, 272–273
 структурная модель, 241–244
 логика на, 127–129
 символ и таблица истинности, 126
 Мультипроцессоры
 кристалл, микросхема, 528
 гетерогенные, 529–531
 гомогенные, 529
 Мур, Гордон, 68
 Мура автоматы 167, 176
 таблицы переходов и выходов, 178
 диаграммы переходов, 177
 временные диаграммы 179
 Мура закон, 62
- Н**
 Набор инструкций, команд 357, 439–440. *См. также* Архитектура
 Набор регистров, 360–361. *См. также*
 Регистровый файл
 Наиболее (самый) значимый байт (MSB), 45, 46, 365
 Наиболее (самый) значимый бит (msb), 45, 46
 Наименее значимый байт (LSB), 45, 365
 Наименее значимый бит (lsb), 45
 Накопитель с жестким диском, 554–555, 574.
 См. также Жесткий диск, Твердотельный
 диск и Виртуальная память
 Напряжение питания, 57. *См. также* VDD
 Не число (Not a number, NaN), 314
 НЕ, вентиль, 53
 ИС (7404), 669
 КМОП, 69
 Неархитектурное состояние, 440
 Недействительный логический уровень, 238
 Неизбежные промахи, 570
 Нелистовая (нетерминальная) функция, 397
 Неопределенная команда, исключение, 412–413,
 510–513
 Непосредственная адресация, 401
 Непосредственные операнды, 366, 377. *См. также*
 Константы
 32-бит, 378
 immediate поле, 366–367
 логические операции, 374
 Непрерывного присваивания оператор, 228, 243,
 253, 260
 Неустойчивые схемы, 162
 Неявная старшая единица, 313
 Низкий, 58. *См. также* 0, OFF
 Низковольтная КМОП-логика (LVCMOS), 62
 Низковольтная ТТЛ-логика (LVTTTL), 62
 Нойс, Роберт, 62
 Номер набора, биты, 560
 Номер страницы, 577
 Номер физической страницы (PPN), 578
- О**
 Оберегаемые регистры, 396, 397
 Оберегаемые регистры, 396–397, 397
 Обозначение шин на схемах, 93
 Обороты в минуту (RPM), 634
 Обратный порядок, Big-endian, 227
 Обратный порядок, память, 364–367
 Общего назначения В/В (GPIO), 594
 порты (выводы) PIC32, 596
 Светодиоды и кнопки, пример, 594–595
 Объем информации, 39
 Ограничение времени предустановки, 187, 188–191
 с расфазировкой тактовых сигналов, 195–197
 Ограничение времени удержания, 186–194
 с расфазировкой тактов, 195–197
 Ограничение времени удержания, 190, 191,
 192–195
 Ограничение максимальной задержки.
 См. Ограничение времени предустановки
 Ограничение минимальной задержки.
 См. Ограничение времени удержания
 Однобитный динамический предсказатель
 переходов, 517–518
 Однократно программируемая логика (OTP), 672
 Однотактный процессор MIPS, 446–459
 управление, 452–455
 тракт данных, 446–452
 пример функционирования, 455
 HDL для, 498–510
 производительность, 459
 Оксид, окисел, 66
 Операнды
 MIPS, 360–366
 Непосредственные операнды (константы),
 366, 377
 память, 362–365
 регистры, 360–362
 x86, 418–420, 419
 Оперативное запоминающее устройство (RAM),
 324–326, 330
 Оператор Case, в HDL, 254–256. *См. также*
 оператор Switch/case
 Оператор switch/case
 в C, 732–733

- в HDL. *См.* Оператор case в ассемблере MIPS, 382
- Оператор одновременного присваивания сигнала, 228, 233–234, 244, 252–258
- Операторы сокращения, 230–231
- Операторы управления последовательностью действий
 - условные операторы. *См.* Условные операторы циклы. *См.* Циклы
- Отображение, 558
- П**
- Пакованная арифметика, 525
- Память данных, 442
 - HDL для, 509
- Память команд (инструкций) (IM), 442, 481
 - MIPS, 510
- Память с байтовой адресацией, 364–366
 - обратный порядок, 364–365
 - прямой порядок, 364–365
- Память с словной адресацией, 364, 365
- Память с прямым порядком, 364–365, 365
- Память с удвоенной скоростью обмена (DDR), 326, 648
- Память. *См. также* Массивы памяти
 - режимы адресации, 419
 - площадь и задержки, 326–327
 - массивы. *См.* Memory arrays
 - среднее время доступа к памяти, 555
 - обратный порядок (big-endian), 227, 364–365
 - байтовая адресация, 364–366
 - HDL для, 331–333
 - иерархия, 554
 - прямой порядок (little-endian), 227, 364–365
 - логика на основе, 330–332
 - главная, 554
 - операнды в, 362–365
 - физическая, 574
 - порты, 323
 - защита, 582. *См. также* Виртуальная память
 - типы, 324–329
 - DDR, 324–325
 - DRAM, ДОЗУ, 324
 - флэш, 329–330
 - регистровый файл, 327
 - ROM, ПЗУ, 327–330
 - SRAM, СОЗУ, 325–326
 - виртуальная, 553. *См. также* Виртуальная память
- Параллелизм на уровне команд (ILP), 523, 526
- Параллелизм на уровне потоков (TLP), 527
- Параллелизм, 205–208
- Параллельный В/В, 596
- Передаточная характеристика, 59, 60
- передаточная характеристика, 59–61. *См. также* Допускаемые уровни шумов
- Передаточный ЛЭ, 72
- Передача по значению, 737
- Передача по ссылке, 737
- Переднее крыльцо, 626
- Переименование регистров, 524–526
- Перекрестно включенные инверторы, 151, 152
 - бистабильный режим, 152
- Переменные в С, 721–726
 - глобальные и локальные, 724–725
 - инициализация, 725
 - базовые типы данных, 722–723
- Переменные состояния, 150
- Переполнение
 - обработка исключения при, 411–413, 511–514
 - при сложении, 47
- Период тактового сигнала, 188, 445
- Периодические прерывания, 613–614
- Периферийная тактовая шина (PBCLK), 593
- Периферийные устройства. *См.* Интерфейсы ввода-вывода
- Персональный компьютер (PC). *См.* x86
 - Персональный компьютер (PC) системы В/В, 644–650
 - системы сбора данных, 650–651
 - DDR3 память, 648
 - сеть, 648–649
 - PCI, 647
 - SATA, 649
 - USB, 646–647, 650–651
- Петля фазовой синхронизации (PLL), 628
- Печатная плата (PCB), 688–689
- Плавающий (Z), 115–116
 - в HDL, 237–239
- Плавающий потенциал на выходе, 159
- Пластина, подложка, вафля, вэйфер, 65
- Пластины, 574
- Поведенческое моделирование, 222–223
- Подложка, 65–66
- Показатели памяти. *См.* Среднее время доступа к памяти
- Полевые транзисторы металл-оксид-полупроводник (MOSFET), 62
 - модели переключения, 68
- Полностью ассоциативный кэш, 564–565
- Полный сумматор, 93, 233, 234, 253, 297
 - always/process, работа с оператором, 252
- Положительный перепад, фронт (передний), 131
- Полубайты, nibблы, 45–46
- Полупроводники, 63
 - промышленность, продажи, 31
- Полусумматор, 294, 294
- Попадание, 555
- Пороговое значение напряжения, 67
- Последним пришёл – первым ушёл (LIFO), 393. *См. также* Стэк
- Последовательная логика, 149–211, 317–321
 - счетчики, 317
 - конечные автоматы. *См.* Конечные автоматы
 - триггеры, 155–159. *См. также* Регистры
 - зашелки, 151–154
 - D, 154
 - RS, 151–153

- регистры. *См.* Регистры
 регистры сдвига, 318–320
 временные характеристики. *См.* Временные
 характеристики, анализ
- Последовательность функциональные узлы.
См. Последовательная логика
- Последовательный ввод-вывод, 596–609
 SPI. *См.* Serial peripheral interface
 UART. *См.* Universal Asynchronous Receiver
 Transmitter
- Последовательный обмен данными с
 компьютером, 607–609
- Постоянное запоминающее устройство (ROM),
 324, 327–328, 327–329
 реализация на транзисторах, 340–341
- Потери на упорядочение, 189–190, 196, 208, 498
- Потребляемая мощность, 73–75
- Предсказание переходов, 517–518
- Преобразование форматов (atoi, atol, atof), 759
- Преобразование чисел
 двоичное в десятичное, 41–42
 двоичное в шестнадцатичное, 44
 десятичное в двоичное, 42, 44
 шестнадцатичное в двоичное, 43
 шестнадцатичное в двоичное и десятичное,
 43, 44
 взятие дополнения до двух, 46, 6
- Прерывания, 406, 612–614
 PIC32, 612–614
- Префиксное дерево, 300
- Префиксный сумматор, 298–300, 299
- Привод прямого напряжения, 640
- Привод с ограничителем по постоянному току, 642
- Примеси, атомы, 63
- Приоритет
 схема, 109–110
 приоритетный шифратор, 145–146, 147
- Приостановки, 488–492. *См. также* Конфликты
- Причины исключения, регистр, 411–412, 511
- Проводник, 107
- Программирование
 массивы. *См.* Массивы
 переходы. *См.* Переходы
 условные операторы, 381–382
 константы. *См.* Константы, Непосредственные
 операнды
 вызовы функций. *См.* Функции в С. *См.* С,
 программирование на
 в С. *См.* С, программирование на
 в MIPS, 370–395,3
 инструкции, 707–710
 логические инструкции, 374–375
 циклы. *См.* Циклы
 умножение и деление, 378
 инструкции сдвига, 375–376, 375
- Программируемое плавкими перемычками ПЗУ, 328
- Программируемые ПЗУ (PROM), 327, 329,
 672–676
- Программы обработки прерываний (ISR), 612.
См. также Исключения
- Проектирование синхронных логических схем,
 161–165
- Прозрачность защелки. *См.* D-триггер
- Производительности процессоров сравнение
 многотактный процессор MIPS, 478–479
 конвейерный процессор MIPS, 496
 одноктактный процессор MIPS, 458–459
- Промахи, 553–555, 570
 из-за недостаточной емкости, 570
 неизбежный, 570
 из-за конфликтов, 562, 570
- Промахи из-за конфликтов, 570
- Промахи из-за недостаточной емкости, 570
- Промахи
 в кэш, 553
 из-за недостаточной емкости, 570
 неизбежные, 570
 из-за конфликтов, 570
 страничная ошибка, 575
- Пропускная способность, 205–208, 443–444,
 479–482, 526
- Простая импликанта, 105, 117
- Простой при неверно предсказанном переходе,
 491–492
- Пространственная локальность, 551, 565–568
- Пространственный параллелизм, 205–206
- Пространство подкачки, 583
- Простые программируемые логические
 устройства (ППЛУ, SPLD), 335
- Протокол TCP и интернет-протокол (TCP/IP), 648
- Прокладной вентиль. *См.* передаточный элемент
- Процессор с внеочередным выполнением команд,
 521–524
- Процессор цифровой обработки сигналов (DSP),
 531
- процессоры Pentium, 535, 536
 Pentium, 445, 536, 537–538
 Pentium II, 535
 Pentium III, 445, 535, 536
 Pentium M, 537
 Pentium Pro, 535
- Прямое (one-hot) кодирование, 173–176
- Прямой порядок шины в HDL, 227
- Псевдо-n-МОП логика, 72–73, 72
 вентиль ИЛИ-НЕ, 71
 ПЗУ и ПЛМ, 340–341
- Псевдокоманды, 410–411
- Псевдопрямая адресация, 402–403
- P**
- Работа с файлами, в С, 755–757
- Раздавливание, 523
- Разрыв процессор-память, 552
- Распространения сигнал, 296
- Расфазировка тактовых сигналов, 194–197
- Расфазировка. *См.* Расфазировка тактовых
 сигналов

- Реализация цифровых систем, 667–706
 74xx серии, логика. *См.* 74xx серии, логика
 программируемая логика (ASICs), 676
 монтаж, 690–693
 макетные платы, 686–687
 работа с документацией, 677–683
 экономика, 704–707
 семейства логических элементов, 685–687
 корпусирование, 685–688
 печатные платы, 688–689
 программируемая логика, 671–678
 Регистр инструкций, команд (IR), 461, 468
 Регистр результата (rd or rt), 448–449, 455, 464
 Регистр скорости передачи данных (BRG), 600
 Регистр специального назначения (EPC), 411–413
 Регистровая адресация, 401
 Регистровый файл (RF)
 HDL для, 504
 в конвейерном процессоре MIPS (запись по
 отрицательному фронту), 482
 описания регистров MIPS, 360–361
 схема, 327
 применение в процессорах MIPS, 442
 Регистры с сигналом разрешения, 248–249. *См.*
также Триггеры
 Регистры сдвига, 318–320
 Регистры со сбросом, 245–247
 Регистры Сопроцессора 0, 512. *См. также*
 Причины и Регистр специального назначения
 Регистры специального назначения (SFR), 590
 Регистры. *См.* Триггеры, регистры MIPS, и
 регистры x86
 Регулярность, 36
 Редко используемые данные (LRU) замена, 567–568
 двухсекционный наборно-ассоциативный кэш,
 567–568, 568
 Режимы адресации
 MIPS, 400–403
 базовая, 401
 непосредственная, 401
 РС-относительная, 401–402
 псевдопрямая, 402–403
 регистровая, 401
 x86, 419
 Рекурсивные вызовы функций, 397–399
 Решетка кремния, 64
 Решетка, кремний, 63
 Роберт Деннард, 324
- С**
 САПР. *См.* Системы автоматизированного
 проектирования
 Свободное пространство, 227
 Связный список, 749–750
 Сегмент глобальных данных, 404–405
 Сегмент данных, 404
 Сегмент динамических данных, 405
 Сегмент кода, текста, 405, 409
 Сегмента дескриптор, 424
 Сегментация, 425
 Серводвигатель, 633, 638–640
 Сигналы генерации, 296, 298
 Сигналы управления, 134, 305
 Символы (char), 388–390, 722, 743
 массивы, 388–390. *См. также* Строки
 типы С, 743
 Символьные ЖК-дисплеи, 622–625
 Симметричная многопроцессорность (SMP).
См. гомогенная многопроцессорность
 Синергетический процессор (SPU) ISA, 531
 Синергетических процессорных ядер (SPEs), 530
 Синхронизаторы, 199–201, 199–200
 Синхронно сбрасываемые триггеры, 158
 Синхронное динамическое ОЗУ (SDRAM), 326
 Синхронные последовательные схемы, 161–165,
 163. *См. также* Конечные автоматы
 временные характеристики. *См.* Временные
 характеристики, анализ
 Синхронные схемы, 163–164
 Системы автоматизированного проектирования
 (САПР), 112, 173
 Системы ввода/вывода (В/В), 587–651
 драйверы устройств, 588–589, 608
 встроенные системы В/В. *См.* Встроенные
 системы В/В
 регистры В/В, 586–587
 с отображением на память В/В, 586–587
 ПК системы В/В. *См.* ПК системы В/В
 Системы памяти, 549–550
 MIPS, 573
 анализ производительности, 554–556
 x86, 652–656
 Системы сбора данных (DAQs), 650–651
 тmuDAQ, 650
 Системы счисления, 40–50
 двоичная, 41–42, 43–44
 сравнение, 51–52, 52
 десятичная, 40
 оценка степеней двойки, 46
 с фиксированной точкой, 311, 311–312
 с плавающей точкой, 312–315
 сложение, 315–316, 316
 особые случаи, 314
 шестнадцатичная, 43–45, 44
 отрицательное и положительное, 47
 со знаком, 47
 без знака, 41–43
 Скалярный процессор, 518
 Сканируемый триггер, 320–321
 Сканирующие цепочки, 320–322
 Склеивания теорема, 102
 Слабый подтягивающий транзистор, 73
 Сложение, 43–44, 46–49, 287, 292–300, 356.
См. также сумматоры
 двоичное, 43–44
 плавающая точка, 312–314
 команды MIPS, 407–408, 707

- двоичное со знаком, 47–51
 - См. также* Сумматоры, Сложение
 - Смещение страницы, 577
 - Смещение, 462, 463
 - Смещенный порядок, 313–314
 - Совершенная индукция, доказательство теорем, 104–105
 - Совершенная форма. *См.* Дизъюнктивная форма, Конъюнктивная форма
 - Соединение АТА, 649
 - Сохранения
 - сохранить байт (sb or sbu), 364–366, 389–390
 - сохранить полуслово (sh or shu), 415
 - сохранить слово (sw), 364–366
 - Сохраняемые вызываемой функцией регистры, 397
 - Способы округления, 315
 - Сравнение
 - аппаратное. *См.* Компараторы, АЛУ на ассемблере MIPS, 384–385, 414–415 в ALU, 304
 - Среда тестирования, HDLs, 275–279
 - для процессора MIPS, 506–507
 - простая, 276–277
 - с самопроверкой, 277–278
 - с тестовыми векторами, 278–279
 - Среднее время доступа к памяти (AMAT), 555, 569
 - Среднее время наработки на отказ (MTBF), 200–201
 - Стандартные библиотеки, 752–760
 - math, 759–760
 - stdio, 753–755
 - чтение и запись файлов, 755–757
 - printf, 753–755
 - scanf, 754–755
 - stdlib, 758–760
 - exit, 758
 - преобразование форматов (atoi, atol, atof), 759
 - rand, srand, 757–758
 - string, 760
 - Статическая дисциплина, 60–62
 - Статическое ОЗУ (SRAM), 324, 325
 - Статическое предсказание переходов, 516
 - Стек, 393–399. *См. также* Вызовы функций при рекурсивном вызове функции, 398
 - оберегаемые регистры, 396–398
 - кадр стека, 396, 400
 - регистр указателя стека (\$sp), 394
 - хранение дополнительных аргументов в, 399–400
 - хранение локальных переменных в, 399–400
 - стираемые программируемые ПЗУ (EPROM), 329, 672
 - Страницы, 576
 - Страничная ошибка, 575
 - Странное число, 51
 - Стратегии замещения страниц, 583
 - Стратегии записи, 571–572
 - отложенная запись, 571–571
 - сквозная запись, 571–571
 - Строки, 390, 744–745. *См. также* Символы (char)
 - Структурное моделирование, 222–224, 241–244
 - Структуры(struct), 745–747
 - Сумматор с последовательным переносом, 295, 295–296, 298
 - Сумматор с распространяющимся переносом (CPA). *См.* Сумматор с последовательным переносом, Сумматор с ускоренным переносом, и Префиксный сумматор
 - Сумматор с ускоренным переносом (CLA), 294–296, 295
 - Сумматоры, 294–300
 - с ускоренным переносом, 295
 - с распространяющимся переносом, 293
 - полный, 90, 293
 - полу-, 293
 - HDL для, 234, 253
 - префиксный, 298
 - с последовательным переносом, 293
 - Суперскалярный процессор, 518–520
 - Схема конечного автомата. *См.* Конечные автоматы
 - Схемы с несколькими выходами, 108–109
 - Схемы сдвига и циклического сдвига, 306
 - Схемы сдвига, 306–308
 - Схемы циклического сдвига, 306–308
 - Схемы
 - серия 74xx. *См.* логика серии 74xx
 - Заказные интегральные схемы (ASICs), 676
 - неустойчивые, 161
 - асинхронные, 163, 165–166
 - комбинационные. *См.* Комбинационная логика
 - определение, 91
 - задержка, 130–135
 - с несколькими выходами, 108
 - приоритет, 108
 - последовательные. *См.* Последовательная логика
 - синхронные, 165–166
 - синхронные последовательные, 163–166, 165
 - синтез, 337, 340, 343
 - временные характеристики, 130–137
 - с двухступенчатым конвейером, 207–208
 - без импульсных помех, 138
 - Схемы, правила рисования, 69, 107
 - Счётчик команд (PC), 373, 401, 442, 446
 - Счетчик команд в архитектуре x86 (EIP), 418
 - Счетчики, 315
 - делитель-на-3, 286
- Т**
- Таблица символов, 408
 - Таблица страниц, 576, 578–579
 - номер, 584
 - смещение, 584

- Таблицы истинности, 51
 дешифратор ALU, 454, 455
 синтез мультиплексора по, 125
 дешифратор семисегментного индикатора, 121
 RS-триггер, 151, 152
 с безразличностью (незначимыми битами), 110, 124–125, 259
 с неопределенными и плавающими входами, 238–239
- Таблицы преобразования (LUT), 331, 336
- Таймеры, 610–612
 генерация задержки, 611–612
- Такты ожидания, 652
- Твердотельные диски (SSD), 554–555. *См. также* Флеш-память и Накопитель на ЖД
- Теорема идемпотентности, 101
 Теорема идентичности, 101
 Теорема инволюции, 101
 Теорема о дистрибутивности, 102
 Теорема о нулевом элементе, 101
 Теорема поглощения, 102
 Теорема согласованности, 102, 104
- Тернарные операторы, 231, 272
- тестируемое устройство (DUT), 275
- Тестируемый модуль (UUT), 271
- Тестовый набор, 444
- Техническая документация, 677–682
- Типичные ошибки в С, 762–767
- Типы данных, 736–751
 массивы. *См.* Массивы
 символы. *См.* Символы (char)
 динамическое распределение памяти. *См.* Динамическое распределение памяти (malloc and free)
 связный список. *См.* Связный список
 указатели. *См.* Указатели
 строки. *См.* Строки (str)
 структуры. *См.* Структуры (struct)
 typedef, 747–748
- Ток покоя, 74
- Тракт данных
 Многотактного процессора MIPS, 460–466
 Конвейерного процессора MIPS, 482–483
 Однотактного процессора MIPS, 446–452
- Транзистор с плавающим затвором, 329. *См. также* Флеш-память
- Транзисторно-транзисторная логика (TTL), 61–62, 682–683
- Транзисторы, 62–70
 биполярные, 62
 КМОП, 62–69
 ЛЭ на, 69–72
 триггеры и защелки, 159–160
 MOSFET, 62
 п-МОП, 65–71, 66–70
 р-МОП, 65–71, 66–70
 псевдо-п-МОП, 71–72
 ПЗУ и ПЛМ, 340–343
 Передаточный логический вентиль, 72
 Трансляция и запуск программы, 406–412, 407
 Триггер ведущий-ведомый, 155
 Триггер с функцией разрешения, 156–157
 Триггер с функцией сброса, 158
 Триггер, синхронизируемый уровнем. *См.* D-триггер
 Триггер, синхронизируемый фронтом, 156
 Триггер, синхронизируемый фронтом. *См.* триггер
 Триггеры, 155–159, 245–250. *См. также* Регистры
 непосредственное последовательное соединение, 190, 200–205, 249–250. *См. также* Синхронизатор
 сравнение с защелками, 160
 с функцией разрешения, 157–158
 HDL для, 506. *См. также* Регистры
 метастабильное состояние. *См.* Метастабильность
 регистр, 155–156
 со сбросом, 158
 сканируемый, 320–321
 регистр сдвига, 318–320
 число транзисторов, 156, 160
 на уровне транзисторов, 159–160
 Триггеры-защелки, 152–154
 сравнение с триггерами, 149, 158
 D, 155, 160
 SR, 151–153, 152
 на транзисторном уровне, 159–160
- ТТЛ. *См.* Транзисторно-транзисторная логика
- Тэг, 560
- ## У
- Указатели, 736–738, 740, 744, 746, 748
- Улучшенная микроархитектура, 513–531
 предсказание переходов. *См.* Предсказание переходов
 длинные конвейеры. *См.* Длинные конвейеры
 гетерогенные мультипроцессоры. *См.* Гетерогенные мультипроцессоры
 Гетерогенные мультипроцессоры
 гомогенные мультипроцессоры. *См.* Гомогенные мультипроцессоры
 Гомогенные мультипроцессоры
 многопоточность. *См.* Многопоточность
 процессор с внеочередным выполнением. *См.* Процессор с внеочередным выполнением
 переименование регистров. *См.* Переименование регистров
 Переименование регистров
 одиночный поток команд, множественный поток данных. *См.* Одиночный поток команд, множественный поток данных
 суперскалярный процессор. *См.* Суперскалярный процессор
 Суперскалярный процессор
 Умножение, 378, 414. *См. также* Умножитель
 инструкция MIPS, 378
 Команды для чисел со знаком и без знака, 413
 Умножитель со знаком, 272
 Умножитель, 308–309

- схема, 309
- HDL для, 309
- Универсальный асинхронный приемопередатчик (UART), 603–609
 - аппаратное подтверждение связи, 605
 - регистр STA, 606
- Уолл, Ларри, 55
- Управление двигателем, 641
- Управление сложностью, 33–36
 - цифровая абстракция, 33–34
 - дисциплина, 34–36
 - иерархия, 36–37
 - модульность, 36–37
 - регулярность, 36–37
- Условное присваивание сигнала, 230–231
- Условное присваивание, 230–231
- Условные операторы
 - в С, 731–733
 - if, 731–733
 - if/else, 731
 - switch/case, 731–733
 - в HDL, 246, 254–258
 - case, 254–256
 - casez, case?, 259
 - if, if/else, 256–259
 - в ассемблере MIPS, 381–382
 - if, 381–382
 - if/else, 382
 - switch/case, 382
- Условные переходы, 378–379
- Условный оператор, 230–231
- Устройство управления. *См. также*
 - Дешифратор АЛУ, Главный дешифратор Многотактного процессора MIPS, 466–478
 - Конвейерного процессора MIPS, 484–485
 - Однотактного процессора MIPS, 452–457
- Утечки ток, 74
- Ф**
- Физическая память, 574
- Физические страницы, 575
- Физического адреса расширение, 655
- Фиксация, 185
- Флаги состояния, 420
- Флаги, 306
- Флеш-память, 329. *См. также* Твердотельные диски
- Формат двойной точности, 314–315
- Форматы команд, MIPS
 - F-типа, 416
 - I-типа, 369, 369–371
 - J-типа, 371
 - R-типа, 368–369
- Форматы команд, x86, 417–423
- Форматы одинарной точности, 314–315. *См. также* Числа с плавающей точкой (запятой)
- Х**
- Хоппер, Грейс, 405

Хранимая программа, 372–373

Ц

- Ц/А преобразование, 615–619
- ЦАП. *См.* Цифро-аналоговое преобразование
- Целевой адрес ветвления (BTA), 402–403, 451
- Целевой адрес перехода (JTA), 402–403, 457
- Целочисленный функциональный блок (IEU), 536
- Цена промаха, 565
- Цикл While, 383–384, 734
- Циклические пути, 161
- Циклы, 383–385, 733–735
 - в С,
 - do/while, 734–735
 - for, 735
 - while, 734
 - в ассемблере MIPS,
 - for, 384–385
 - while, 383–384
- Цифро-аналоговые преобразователи (DAC), 614
- Цифровая абстракция, 34–35, 38–40, 57–61
- Цифровые схемы. *См.* Логика

Ч

- Чарльз Беббидж, 38
- Частичные произведения, 308
- Частота дискретизации (скорость выборки), 614
- Частотная манипуляция (FSK), 632
 - и GFSK сигналы, 632
- Числа без знака, 50
- Числа в прямом коде со знаком, 47–48, 312
- Числа с плавающей точкой, 314–315
 - сложение, 315–316
 - форматы одинарной и двойной точности, 314–316
 - в программировании. *См.* Float и Double
 - округление, 315
 - особые случаи
 - бесконечность, 315
 - NaN, 315
- Числа с фиксированной запятой, 311–312
- Число попаданий, 555–557
- Число тактов на команду (CPI), 445
- Число тактов на команду (CPI), 514, 516

Ш

- Шаговые двигатели, 634, 639–644
 - биполярный шаговый двигатель, 639–641
 - привод с полшагом, 640–641
 - привод с двумя одновременно включаемыми фазами, 640–641
 - волновой привод, 640–641
- Шестнадцатеричного в двоичную или десятичную преобразование, 44, 45
- Шестнадцатеричные числа, 44–46
- ШИМ (PWM). *См.* Широтно-импульсная модуляция
- Шина, 93
 - с тремя состояниями, 116

- Широтно-импульсная модуляция (PWM, ШИМ), 620–621
аналоговый выход с, 621
скважность, 620
сигнал, 620
- Э**
Эзернет, Ethernet, 648
Экономика, 704
Электрически стираемое программируемое ПЗУ (EEPROM), 329
Электронно-лучевая трубка (CRT), 625–626. *См. также* монитор VGA
интервал горизонтального гашения, 626
интервал вертикального гашения, 626
Элемент И, 54–55, 228
ИС (7408, 7411, 7421), 669
таблица истинности, 53, 56
на КМОП транзисторах, 71–72
Элемент И-ИЛИ, 85
Элементы ввода/вывода (IOE), 335
ЭЛТ. *См.* Электронно-лучевая трубка
- Ю**
Юникод (Unicode), 388
- Я**
Ядра, 528
Язык ассемблера, MIPS, 357–420, 709–713. *См. также* Инструкции MIPS
инструкции, команды, 356–364, 709–713
логические инструкции, 374–375
операнды, 358–366
трансляция кода высокого уровня в, 361
трансляция машинного языка в, 372
трансляция в машинный язык, 369–370
- Язык ассемблера, x86. *См.* команды x86
Язык. *См. также* Инструкции, команды ассемблера, 357–365
машинный, 367–372
мнемоники, 358
трансляция с ассемблера в машинный, 369
Языки описания схем (HDLs). *См. также* SystemVerilog, VHDL
емкость, 571
комбинационная логика, 223, 252
битовые операторы, 227–229
блокирующие и неблокирующие присваивания, 260–264
case операторы, 254–255
условные присваивания, 230–231
задержки, 239–240
типы данных, 268–272
история, 222–223
if операторы, 256–259
внутренние переменные, 233–235
числа, 235
операторы и приоритет, 235
операторы сокращения, 230
модули, 222–223
параметризованные модули, 272–275
универсальные строительные блоки, 504–507
последовательная логика, 245–251, 262–266
моделирование и синтез, 224–226
однотактный процессор MIPS, 499–511
структурное моделирование, 238–241, 3
среда тестирования, тестовое окружение, 275–279, 506–508