
Оглавление

ПРЕДИСЛОВИЕ	11
Предмет (темы) книги	14
Курсы	15
Ресурсы	17
Структура	18
Проекты	19
Второе издание	20
Благодарности	21
I. АППАРАТНОЕ ОБЕСПЕЧЕНИЕ	23
Привет, нижний мир	23
От Nand до «Тетриса»	25
Абстракция и реализация	29
Методология	31
Путь, который вас ожидает	33
1. Булева логика	35
1.1. Булева алгебра	36
1.2. Логические вентили	40
1.3. Аппаратное конструирование	44
1.4. Спецификация	50
1.5. Реализация	58
1.6. Проект	62
1.7. Перспектива	64
2. Булева арифметика	67
2.1. Арифметические операции	68
2.2. Двоичные числа	68
2.3. Двоичное сложение	71
2.4. Двоичные числа со знаком	72
2.5. Спецификация	74
2.6. Реализация	82
2.7. Проект	84
2.8. Перспектива	85

3. Память	87
3.1. Устройства памяти	88
3.2. Секвенциальная логика	90
3.3. Спецификация	97
3.4. Реализация	102
3.5. Проект	107
3.6. Перспектива	108
4. Машинный язык	111
4.1. Машинный язык: обзор	113
4.2. Машинный язык Nask	118
4.3. Программирование на языке Nask	135
4.4. Проект	138
4.5. Перспектива	141
5. Компьютерная архитектура	143
5.1. Основы компьютерной архитектуры	144
5.2. Аппаратная платформа Nask: спецификация	152
5.3. Реализация	160
5.4. Проект	165
5.5. Перспектива	168
6. Ассемблер	173
6.1. Общие принципы	174
6.2. Спецификация машинного языка Nask	177
6.3. Перевод с языка ассемблера в двоичный код	180
6.4. Реализация	182
6.5. Проект	188
6.6. Перспектива	191
II. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ	193
II.1. Примеры программирования на языке Jack	195
II.2. Компиляция программы	201
7. Виртуальная машина I: обработка	205
7.1. Парадигма виртуальной машины	207
7.2. Стековая машина	210
7.3. Спецификация VM, часть I	217
7.4. Реализация	218
7.5. Проект	230
7.6. Перспектива	235
8. Виртуальная машина II: управление	239
8.1. Высокоуровневая магия	240
8.2. Ветвление	243
8.3. Функции	246
8.4. Спецификация VM, часть II	255
8.5. Реализация	257
8.6. Проект	266
8.7. Перспектива	271

9. Высокоуровневый язык	275
9.1. Примеры	277
9.2. Спецификация языка Jack	283
9.3. Написание приложений на языке Jack	296
9.4. Проект	299
9.5. Перспектива	301
10. Компилятор I: синтаксический анализ	303
10.1. Основы	305
10.2. Спецификация	316
10.3. Реализация	320
10.4. Проект	325
10.5. Перспектива	331
11. Компилятор II: генерация кода	333
11.1. Генерация кода	335
11.2. Спецификация	361
11.3. Реализация	362
11.4. Проект	374
11.5. Перспектива	379
12. Операционная система	381
12.1. Основы	383
Эффективность прежде всего	385
Умножение	386
Деление	388
Квадратный корень	390
12.2. Спецификация ОС Jack	403
12.3. Реализация	404
12.4. Проект	413
План тестирования	414
Полный тест	418
12.5. Перспектива	419
13. Веселье продолжается	421
Аппаратные реализации	422
Улучшения аппаратной части	423
Высокоуровневые языки	423
Оптимизация	424
Обмен данными	424
ПРИЛОЖЕНИЕ 1. ПОСТРОЕНИЕ БУЛЕВЫХ ФУНКЦИЙ	425
П1.1. Булева алгебра	425
П1.2. Построение булевых функций	427
П1.3. Выразительная сила Nand	429
ПРИЛОЖЕНИЕ 2. ЯЗЫК ОПИСАНИЯ АППАРАТУРЫ	433
П2.1. Основы HDL	433
П2.2. Многобитные шины	438
П2.3. Встроенные микросхемы	440

П2.4. Последовательностные микросхемы	442
П2.5. Визуализация микросхем	446
П2.6. Практический справочник по HDL	449
ПРИЛОЖЕНИЕ 3. ЯЗЫК ОПИСАНИЯ ТЕСТОВ	457
П3.1. Общие рекомендации	458
П3.2. Тестирование микросхем в симуляторе аппаратуры	461
П3.3. Тестирование программ на машинном языке в симуляторе ЦПУ	471
П3.4. Тестирование программ ВМ в эмуляторе ВМ	473
ПРИЛОЖЕНИЕ 4. НАБОР МИКРОСХЕМ НАСК	477
ПРИЛОЖЕНИЕ 5. НАБОР СИМВОЛОВ НАСК	479
ПРИЛОЖЕНИЕ 6. API ОС JACK	481
Math	481
String	482
Array	483
Output	483
Screen	484
Keyboard	485
Memory	486
Sys	486
АЛФАВИТНЫЙ УКАЗАТЕЛЬ	487

Предисловие

Услышанное я забываю; увиденное запоминаю; сделанное понимаю.

— Конфуций (551–479 до н. э.)

Обычно утверждается, что просвещенные люди XXI века должны иметь какое-то представление о ключевых концепциях, лежащих в основе так называемой четверки BANG: «биты», «атомы», «нейроны» и «гены» (*Bits, Atoms, Neurons, and Genes*). И хотя наука добилась значительных успехов в исследовании основ во всех этих сферах, вряд ли мы когда-нибудь поймем до конца, как на самом деле устроены атомы, нейроны и гены.

Чего, впрочем, нельзя сказать о «битах» и о вычислительных системах в целом: несмотря на фантастическую сложность современных компьютеров, понять, как они работают и как устроены, можно полностью. Поэтому, когда мы с благоговением смотрим на окружающий нас физический мир, приятно осознавать, что хотя бы одна сфера из четверки BANG полностью доступна для человеческого понимания.

И действительно, в первые годы существования компьютеров общее представление о работе вычислительных машин мог получить любой любопытный человек. Взаимодействие между аппаратным и программным обеспечением было достаточно простым и прозрачным, позволяющим окинуть взглядом всю картину работы компьютера. Увы, по мере усложнения цифровых технологий эта ясность была практически утрачена: самые фундаментальные идеи и технические

приемы информатики — сама суть этой сферы — теперь скрыты под множеством слоев не до конца понятных интерфейсов и проприетарных реализаций. Неизбежным следствием такого усложнения стала специализация: изучение прикладной информатики превратилось в погоню за множеством нишевых курсов, каждый из которых охватывает лишь отдельный аспект этой области.

Мы написали эту книгу, поскольку у нас сложилось впечатление, что многие студенты, изучающие информатику, не видят леса за деревьями. Типичный студент пробегает через серию курсов по программированию, теории и инженерии, не останавливаясь, чтобы оценить красоту общей картины. А картина в целом такова, что аппаратные, программные и прикладные системы тесно связаны между собой, что не очевидно через скрытую паутину абстракций, интерфейсов и контрактных реализаций.

Невозможность рассмотреть общую картину порождает у многих учащихся и профессионалов тревожное ощущение, будто они не до конца понимают, что происходит внутри компьютеров. И это прискорбно, поскольку компьютеры — самые важные машины XXI века.

Согласно нашему мнению, лучший способ понять, как работают компьютеры, — это собрать один из них с нуля. Исходя из этого, мы пришли к следующей идее: опишем простую, но достаточно мощную компьютерную систему, предложив учащимся создать ее аппаратную платформу и разработать иерархическую структуру программной системы с нуля. В процессе работы будем делать это правильно. Мы утверждаем, что создание компьютера общего назначения (или универсального компьютера) на основе базовых принципов — это очень серьезная и важная задача.

Как следствие, мы решили воспользоваться уникальной образовательной возможностью не только создать некий механизм, но и проиллюстрировать на практике основные принципы планирования и управления крупными проектами по разработке аппаратного и программного обеспечения. Кроме того, мы стремились показать захватывающий процесс создания фантастически сложных и полезных систем на основе фундаментальных принципов посредством тщательных рассуждений и модульного планирования.

Результатом этих усилий стал практический курс, известный теперь под общим названием «*От Nand до “Тетриса”*»: практическое путешествие, начинающееся с самого элементарного логического элемента Nand, двадцать проектов спустя заканчивающееся созданием универсальной компьютерной системы, способной запустить «Тетрис», а также любую программу, которая только придет в вашу голову. Несколько раз спроектировав, построив, перепроектировав и восстановив компьютерную систему, мы написали эту книгу в таком формате, чтобы это мог сделать любой учащийся. Мы также открыли веб-сайт, где выложили все материалы и программные инструменты нашего проекта, доступные для всех, кто хочет изучать или преподавать курс «От Nand до “Тетриса”».

Реакция была ошеломляющей. Сегодня курсы «От Nand до “Тетриса”» преподаются во многих университетах, средних учебных заведениях, учебных лагерях программирования, на онлайн-платформах и в хакерских клубах по всему миру. Наша книга и наши онлайн-курсы приобрели огромную популярность, и тысячи учащихся — от старшеклассников до инженеров Google — регулярно публикуют отзывы, в которых описывают курс «От Nand до “Тетриса”» как лучший образовательный опыт в своей жизни. Как однажды сказал Ричард Фейнман: «То, что я не могу создать, я не понимаю». Практическая программа «От Nand до “Тетриса”» как раз посвящена пониманию через создание. Очевидно, что в людях очень силен менталитет создателя и они страстно привязываются к этой идее.

После публикации первого издания книги мы получили множество вопросов, комментариев и предложений. По мере того как мы решали эти вопросы и вносили изменения в онлайн-материалы, между нашей веб-версией курса «От Nand до “Тетриса”» и книгой появились расхождения. Кроме того, мы подумали, что многие разделы книги могли бы выиграть от большей ясности изложения и от лучшей организации. Поэтому, после того как мы и так уже откладывали насколько могли данный процесс, мы решили засучить рукава и написать новый вариант данной книги, в результате чего в свет вышло это издание. В оставшейся части предисловия описывается именно оно, а в конце сравнивается с предыдущим.

Предмет (темы) книги

Книга предлагает учащимся получить значительный объем знаний в ходе выполнения ряда заданий по конструированию аппаратного и программного обеспечения. В частности, в процессе выполнения практических проектов затрагиваются следующие темы.

- *Аппаратное обеспечение*: булева арифметика, комбинационная логика, последовательностная (секвенциальная) логика, проектирование и реализация логических элементов (вентилей), мультиплексоры, триггеры, регистры, блоки оперативной памяти (ОЗУ), счетчики, язык описания аппаратуры (HDL), моделирование микросхем, верификация и тестирование микросхем.
- *Архитектура*: проектирование и реализация АЛУ/ЦПУ, такты и циклы, режимы адресации, логика выборки и выполнения, набор команд, ввод/вывод с привязкой к памяти.
- *Языки низкого уровня*: разработка и реализация простого машинного языка (двоичного и символического), наборы команд, программирование на языке ассемблера, ассемблеры.
- *Виртуальные машины*: автоматы на основе стека, арифметика стека, вызов и возврат функций, работа с рекурсией, разработка и реализация простого языка виртуальных машин.
- *Языки высокого уровня*: разработка и реализация простого объектно-ориентированного Java-подобного языка: абстрактные типы данных, классы, конструкторы, методы, правила разметки, синтаксис и семантика, ссылки.
- *Компиляторы*: лексический анализ, синтаксический разбор, таблицы символов, генерация кода, реализация массивов и объектов, двухуровневая компиляция.
- *Программирование*: реализация ассемблера, виртуальной машины и компилятора, следуя предоставленным API. Может быть выполнено на любом языке программирования.
- *Операционные системы*: разработка и реализация управления памятью, математические библиотеки, драйверы ввода/вывода,

- обработка строк, текстовый вывод, графический вывод, поддержка языков высокого уровня.
- *Структуры данных и алгоритмы*: стеки, хеш-таблицы, списки, деревья, арифметические алгоритмы, геометрические алгоритмы, соображения по поводу времени исполнения.
 - *Программная инженерия*: модульное проектирование, парадигма интерфейса/реализации, проектирование и документирование API, модульное (поблочное) тестирование, проактивное планирование тестирования, обеспечение качества, программирование на большом уровне (программирование крупных систем).

Уникальная особенность практической программы «От Nand до “Тетриса”» заключается в том, что все эти темы связно представлены и четко ориентированы на главную цель: создание современной компьютерной системы с нуля. Собственно, это и было нашим критерием отбора тем: минимальный набор знаний, необходимый для конструирования универсальной компьютерной системы, способной выполнять программы, написанные на объектно-ориентированном языке высокого уровня. Как оказалось, этот критический набор включает в себя большинство фундаментальных концепций и методов, а также некоторые из самых красивых идей прикладной информатики.

Курсы

Курсы «От Nand до “Тетриса”» обычно предлагаются как общие курсы для студентов и аспирантов, а также пользуются большой популярностью среди тех, кто учится самостоятельно. Курсы на основе этой книги, можно сказать, «перпендикулярны» типичному учебному плану по информатике и компьютерным наукам (*Computer Science*), их можно проходить на любой стадии обучения. Два естественных места в учебном плане — это CS-2, то есть вводный курс, но уже после основ программирования, и CS-99 — обобщающий курс в конце программы. Первый предполагает системно-ориентированное введение в прикладную информатику с заделом на будущее, а второй представляет

собой обобщающий блок, заполняющий пробелы, оставленные предыдущими курсами.

Еще один набирающий популярность вариант — это курс, объединяющий в одной структуре ключевые темы из традиционных курсов по архитектуре компьютера и курсов по компиляции. В любом случае, какова бы ни была обозначенная цель «От Nand до “Тетриса”», они могут иметь самые разные названия, в том числе «Элементы вычислительных систем», «Конструирование цифровых систем», «Организация компьютера», «Создаем компьютер» и, конечно же, «От Nand до “Тетриса”».

Книга и проекты имеют модульную структуру, начиная с самого основного разделения на часть I «Аппаратное обеспечение» и часть II «Программное обеспечение», каждая из которых состоит из шести глав и шести проектов. Хотя мы рекомендуем пройти весь курс, вполне возможно изучать каждую из двух частей отдельно. Книга и проекты могут поддерживать два независимых курса, каждый продолжительностью шесть-семь недель, то есть типичный семестровый курс или два семестровых курса — в зависимости от выбора тем и темпа обучения. Книга полностью самодостаточна: все необходимые знания для построения описанных в ней аппаратных и программных систем предоставляются в ее главах и проектах. Часть I «Аппаратное обеспечение» не требует предварительных знаний, что делает проекты 1–6 доступными для любого студента и самоучки. Часть II «Программное обеспечение» и проекты 7–12 требуют предварительного изучения программирования (на любом языке высокого уровня).

Курсы «От Nand до “Тетриса”» предназначены не только для тех, кто специализируется на информатике или «компьютерных науках». Они подходят для студентов любой специальности, желающих получить практические знания в областях аппаратных архитектур, операционных систем, компиляции и разработки программного обеспечения — и все это в рамках одного курса. И опять-таки единственным предварительным условием (для части II) является знание основ программирования. Действительно, многие студенты курса «От Nand до “Тетриса”» — это те, кто не выбирал в качестве своей основной специальности информатику, но прошел курс введения в информатику и пожелал узнать больше, не перегружая себя программой

из нескольких курсов. Довольно много среди учащихся и разработчиков программного обеспечения, пожелавших «спуститься пониже» и понять, как работают собственно технологии, чтобы повысить свое мастерство высокоуровневого программирования.

В связи с острой нехваткой разработчиков в индустрии аппаратного и программного обеспечения растет спрос на компактные и целенаправленные образовательные программы по прикладной информатике. Они часто принимают форму учебных лагерей программирования и онлайн-курсов, разрабатываемых для подготовки учащихся к рынку труда без прохождения полного курса и получения академической степени. Любая сколько-нибудь серьезная образовательная программа должна предлагать как минимум рабочие знания в области программирования, алгоритмов и систем. «От Nand до “Тетриса”» — это уникальная возможность охватить системные элементы таких программ в рамках одного курса. Кроме того, проекты «От Nand до “Тетриса”» — это привлекательное средство синтеза и практического применения знаний в области алгоритмов и программирования, полученных на других курсах.

Ресурсы

Все необходимые инструменты для создания описанных в книге аппаратных и программных систем свободно предоставляются в комплекте программного обеспечения «От Nand до “Тетриса”». В их число входят симулятор аппаратного обеспечения, эмулятор ЦПУ, виртуальная машина-эмулятор (все из открытых источников), учебные материалы и описанные в книге исполняемые версии ассемблера, виртуальной машины, компилятора и операционной системы. Кроме того, на сайте www.nand2tetris.org размещены все материалы проекта — около двухсот тестовых программ и скриптов, что позволяет постепенно разрабатывать и тестировать каждый из двенадцати проектов. Программные инструменты и материалы проекта можно использовать в исходном виде на любом компьютере под управлением Windows, Linux или macOS.

Структура

Часть I «Аппаратное обеспечение» охватывает главы 1–6. В главе 1 после введения в булеву алгебру описывается логический элемент (вентиль) Nand (И-НЕ), на основе которого строится набор других элементарных логических вентилях. В главе 2 описывается комбинационная логика и создание набора сумматоров, что подводит к конструированию АЛУ. В главе 3 описывается последовательная (секвенциальная) логика и создание регистров и устройств памяти, что подводит к конструированию ОЗУ. В главе 4 обсуждается низкоуровневое программирование и разбирается машинный язык в его символической и двоичной формах. Глава 5 объединяет микросхемы, описанные в главах 1–3, в аппаратную архитектуру, способную выполнять программы, написанные на машинном языке, представленном в главе 4. В главе 6 обсуждается низкоуровневая трансляция программ, что завершается созданием ассемблера.

Часть II «Программное обеспечение» состоит из глав 7–12 и требует подготовки в области программирования (на любом языке) на уровне вводного курса информатики. В главах 7–8 разбираются стековые автоматы и описано построение JVM-подобной виртуальной машины.

В главе 9 описывается объектно-ориентированный Java-подобный язык высокого уровня. В главах 10–11 обсуждаются алгоритмы синтаксического анализа и генерации кода, а также описывается построение двухуровневого компилятора. В главе 12 представлены различные алгоритмы управления памятью, алгебраические и геометрические алгоритмы и описана реализация операционной системы, которая применяет эти алгоритмы на практике. Операционная система предназначена для устранения разрывов между языком высокого уровня, реализованным в части II, и аппаратной платформой, построенной в части I.

Книга основана на парадигме *«от абстракции к реализации»*. Каждая глава начинается с «Введения», в котором описываются соответствующие концепции и типовая аппаратная или программная система. Следующим разделом всегда следует «Спецификация», описывающая системную абстракцию, то есть различные операции, которые она должна тем или иным образом осуществлять. После описания