

ОТ ТРАНЗИСТОРА ДО РОБОТА



Сари Лада

От транзистора до робота

<https://litres.ru/74062664>

SelfPub; 2026

Аннотация

Вы нажимаете кнопку питания — и за долю секунды оживают миллиарды крошечных переключателей. Как из горстки кремния и двух состояний — ноль и единица — вырастает цифровой мир с его фотографиями, музыкой, нейросетями и роботами?

Эта книга — путешествие сквозь все слои современных технологий. Вы узнаете, как процессор управляет миллиардами операций в секунду, почему интернет продолжает работать, когда рвётся один кабель, как робот «видит» мир и принимает решения, и почему Япония делает роботов-компаньонов, а Китай — самые дешёвые гуманоиды на планете.

Содержание

Часть 1 — Из чего сделан компьютер	4
Конец ознакомительного фрагмента.	48

Сари Лада

От транзистора до робота

Часть 1 — Из чего сделан компьютер

Глава 1. Маленький переключатель, который изменил мир

Ты открываешь ноутбук. Может быть, утром, ещё немного сонная. Может быть, когда все уже спят, и только экран светится в темноте. Ты нажимаешь кнопку питания, и через несколько секунд перед тобой — рабочий стол, уведомления, мессенджеры, весь твой цифровой мир.

Возможно, ты не думаешь о том, что произошло внутри. Хорошая технология именно так и работает: незаметно.

Но сегодня мы заглянем внутрь. Это одна из самых удивительных историй, которые человечество когда-либо придумало. История о том, как из простого «да» и «нет» вырос целый цифровой мир.

Начнём с вопроса, который может казаться глупым
Что умеет компьютер?

Если ты спросишь случайного человека на улице, он, скорее всего, скажет что-то вроде: «ну, он считает», «хранит файлы», «запускает программы». И всё это правда. Но если копнуть глубже — до самого дна, до последнего слоя — окажется, что компьютер умеет делать ровно одну вещь.

Он умеет различать два состояния: ток есть или тока нет. Единица или ноль. Да или нет.

Всё. Больше ничего.

Никакой магии. Никакого «понимания». Никакого разума — по крайней мере, на этом уровне. Просто миллиарды крошечных переключателей, каждый из которых в каждый момент времени находится в одном из двух состояний.

И из этой основы выросло всё: фотографии, музыка, видеозвонки, онлайн-банки, карты, игры, социальные сети, искусственный интеллект. Всё это — комбинации единиц и нулей, организованные с невероятной точностью и скоростью.

Что такое транзистор — и почему он главный герой этой истории

Переключатель, о котором мы говорим, называется транзистор. Это маленький электронный компонент, который умеет пропускать или не пропускать через себя электрический ток — в зависимости от того, подаётся ли на него управляющий сигнал.



Рис. 1.1. Транзистор как управляемый илюз: положение затвора определяет силу тока.

Представь обычный выключатель света на стене. Ты нажимаешь — свет горит. Отпускаешь — гаснет. Транзистор работает по тому же принципу, только вместо пальца — электрический сигнал, вместо лампочки — следующий элемент схемы, и вместо долей секунды — миллиардные доли секунды.

Транзистор был изобретён в 1947 году тремя учёными из Bell Laboratories — Уильямом Шокли, Джоном Бардином и Уолтером Браттейном. За это открытие они получили Нобелевскую премию по физике в 1956 году. В тот момент мало кто понимал, что именно произошло. Маленькая деталька

из германия, которая работала лучше, чем предыдущие вакуумные лампы.

До транзистора компьютеры были монстрами. ENIAC, один из первых электронных компьютеров, созданный в 1945 году, занимал целую комнату площадью около 170 квадратных метров, весил почти 30 тонн и содержал около 18 000 вакуумных ламп. Он потреблял столько электричества, что, по легенде, когда его включали, огни в соседних кварталах Филадельфии слегка тускнели. Он постоянно ломался: лампы перегорали, их нужно было менять чуть ли не каждый день.

Транзистор был меньше, надёжнее, потреблял несравнимо меньше энергии. И главное — его можно было делать очень маленьким. Очень-очень маленьким.

Интегральная схема: когда миллион уместается на ногте

Следующий шаг случился в конце 1950-х годов. Инженеры задались вопросом: а что, если разместить сразу несколько транзисторов на одной пластине из полупроводникового материала — кремния? И соединить их прямо там, без проводов?

Так родилась интегральная схема — или, как её часто называют, чип. Джек Килби из Texas Instruments и Роберт Нойс из Fairchild Semiconductor независимо друг от друга пришли к этой идее примерно в одно время. Килби получил Нобелевскую премию в 2000 году — уже в наши дни, спустя де-

сятилетия после открытия.

Первые интегральные схемы содержали буквально несколько транзисторов. Несколько — это два, пять, десять. Уже это казалось революцией.

Но потом началась гонка миниатюризации, которая не останавливается до сих пор.

В 1965 году Гордон Мур, один из основателей Intel, сформулировал наблюдение, которое вошло в историю как закон Мура: количество транзисторов на чипе удваивается примерно каждые два года, при этом стоимость одного транзистора падает, что позволяет производить более мощные чипы за ту же цену. Это не физический закон — скорее, инженерное обещание, которое индустрия давала себе и выполняла. Десятилетиями.

Давай просто посмотрим на цифры.

1971 год, первый процессор Intel 4004: 2300 транзисторов. Размер каждого — около 10 микрон. Для сравнения: толщина человеческого волоса — около 70 микрон. То есть транзистор был в семь раз тоньше волоса.

2020 год, процессоры Apple M1: около 16 миллиардов транзисторов. Технологический процесс — 5 нанометров (шаг литографии). Нанометр — это миллионная часть миллиметра. Для сравнения: атом кремния имеет диаметр около 0,2 нанометра. То есть современный транзистор — это примерно 25 атомов в поперечнике.

Двадцать пять атомов.

Это уже не инженерия в привычном смысле. Это работа с материей на почти атомарном уровне. Это физика, химия, квантовая механика — всё вместе, в производственных цехах, где воздух чище, чем в операционной.

Язык единиц и нулей

Хорошо, транзисторы маленькие и их много. Но как из миллиардов переключателей получается, например, фотография?

Давай разберёмся с языком.

Компьютер работает в двоичной системе счисления. В привычной нам десятичной системе мы используем десять цифр: от 0 до 9. В двоичной — только две: 0 и 1. Каждая такая цифра называется битом (от английского binary digit — двоичная цифра).

Один бит — это один транзистор в одном из двух состояний. Ноль или единица. Нет тока или есть ток.

Из восьми битов складывается байт. Байт — это уже маленький кирпичик, из которого строится всё остальное. Один байт может хранить, например, один символ текста — букву «а» или цифру «7».

Тысяча байт — килобайт. Миллион байт — мегабайт. Миллиард байт — гигабайт. Именно в этих единицах мы измеряем размер файлов, объём памяти, скорость интернета.

Теперь про фотографию. Цифровое изображение — это сетка из маленьких точек, пикселей. Каждый пиксель — это определённый цвет. Цвет задаётся тремя числами: интенсив-

ность красного, зелёного и синего (это модель RGB). Каждое число — от 0 до 255, то есть умещается в один байт. Значит, на один пиксель нужно три байта.

Фотография с телефона — скажем, 12 мегапикселей — это 12 миллионов пикселей, то есть 36 миллионов байт в несжатом виде. Около 36 мегабайт. Поэтому существуют форматы вроде JPEG: они сжимают изображение, убирая детали, которые человеческий глаз всё равно не замечает, и уменьшают размер в десятки раз.

Точно так же работает и звук. Звуковая волна — это непрерывное колебание. Чтобы записать её в цифре, её «измеряют» тысячи раз в секунду — этот процесс называется оцифровкой. Каждое измерение — число, число — биты, биты — транзисторы. Музыкальный трек на три минуты — это десятки миллионов измерений, упакованных в файл.

Видео — это просто много фотографий в секунду плюс звук. 24 кадра в секунду для кино, 60 для плавного геймплея. Если умножить размер одного кадра на количество кадров, получим, почему видео занимает столько места и почему алгоритмы сжатия — это целая наука.

Полупроводники: почему кремний правит миром

Силиконовая долина — Silicon Valley — называется так именно в честь кремния. Но почему именно кремний? Почему не медь, не железо, не стекло?

Всё дело в свойствах материала. Есть проводники — материалы, которые хорошо проводят электрический ток (медь,

золото, серебро). Есть изоляторы — материалы, которые ток не проводят почти совсем (пластик, стекло, резина). А есть полупроводники — материалы, которые находятся посередине и чьи свойства можно менять.

Кремний — именно такой. Сам по себе он проводит ток плохо. Но если добавить в него крошечное количество другого вещества — например, фосфора или бора, — его свойства меняются. Этот процесс называется легированием. Легированный кремний можно заставить проводить ток в одном направлении, блокировать в другом, усиливать сигнал — то есть делать всё то, что нужно для транзистора.

Кроме того, кремний — один из самых распространённых элементов на Земле. Обычный песок — это в основном диоксид кремния. Из него и делают чипы, хотя процесс очистки и преобразования невероятно сложен: нужен кремний чистотой 99,9999999% — девять девяток после запятой.

Сегодня исследователи активно работают над новыми материалами — графеном, нитридом галлия, карбидом кремния, — которые могут однажды заменить или дополнить кремний. Физические пределы миниатюризации уже почти достигнуты: транзисторы размером в несколько атомов начинают вести себя по законам квантовой механики, и классическая электроника там уже не работает. Но это — история следующих десятилетий.

Как транзисторы складываются в логику

Мы разобрались, что транзистор — это переключатель.

Но как из переключателей получается, например, сложение двух чисел?

Ответ — в логических вентилях. Это базовые «строительные блоки» вычислений, каждый из которых сделан из нескольких транзисторов.

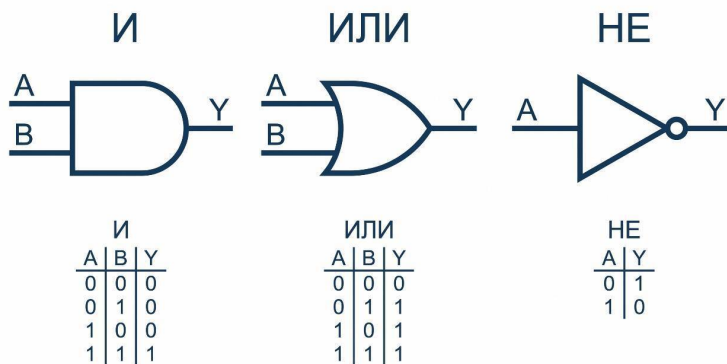


Рис. 1.2. Логические вентили И, ИЛИ, НЕ с таблицами истинности.

Самые простые из них называются И (AND), ИЛИ (OR) и НЕ (NOT).

Вентиль И: выдаёт сигнал «1» только тогда, когда оба входных сигнала равны «1». То есть: «и то, и другое должно быть включено». Как розетка с двумя выключателями — свет горит, только если оба нажаты.

Вентиль ИЛИ: выдаёт «1», если хотя бы один из входов равен «1». Как дверь, которую можно открыть с любой из двух сторон.

Вентиль НЕ: просто инвертирует сигнал. Было «1» — стало «0», и наоборот.

Из этих трёх типов вентилях можно построить любую логическую схему. Буквально любую. Сложение, вычитание, сравнение, выбор — всё это комбинации И, ИЛИ, НЕ.

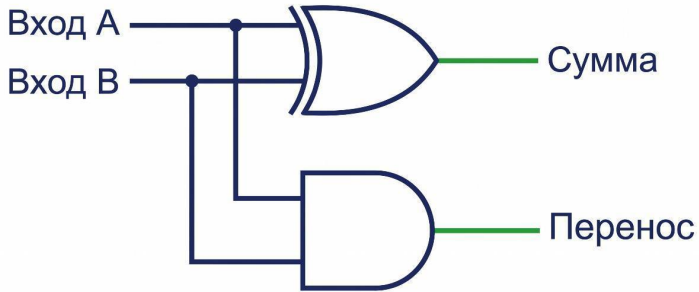


Рис. 1.3. Полусумматор: вентили XOR и AND складывают два бита.

Например, сумматор — схема, складывающая два однобитных числа, — требует нескольких вентилях (бит суммы, бит переноса). Сумматор для многозначных чисел — боль-

ше. Арифметико-логическое устройство (АЛУ) — сердце процессора, которое выполняет математические операции, — это уже тысячи вентиляей, то есть десятки тысяч транзисторов. И так далее, слой за слоем.

Компьютер — это иерархия абстракций. На самом нижнем уровне — физика: электроны, кремний, квантовые эффекты. Чуть выше — транзисторы. Ещё выше — логические вентили. Потом — схемы (сумматоры, регистры, счётчики). Потом — процессор. Потом — машинный код. Потом — операционная система. Потом — приложения. Потом — то, что видишь ты на экране.

Каждый слой скрывает от тебя детали нижнего и предоставляет удобный интерфейс для верхнего. Это принцип абстракции, и он — один из главных секретов того, как люди вообще справляются со сложностью. Никто в мире не держит в голове все слои одновременно. Инженер, разрабатывающий чип, не думает о том, как будет выглядеть пользовательский интерфейс приложения. Разработчик приложения не думает о том, как именно транзисторы хранят биты. Каждый работает на своём уровне.

Закон Мура умирает — и что будет дальше

Мы упоминали закон Мура: каждые два года количество транзисторов на чипе удваивается. Это правило работало с поразительной точностью почти полвека. Именно оно обеспечило тот экспоненциальный рост вычислительной мощи, который привёл нас из эпохи комнатных монстров вроде

ENIAC в эпоху смартфонов мощнее суперкомпьютеров прошлого.

Но в последние годы закон Мура начинает давать сбои. Транзисторы стали настолько маленькими, что дальнейшая миниатюризация натывается на фундаментальные физические ограничения. Когда размер транзистора — несколько нанометров, электроны начинают «просачиваться» сквозь барьеры (тонкие слои материала, через который электрон «не должен» проходить) за счёт квантового туннелирования. На квантовом уровне частица не находится в одной точке, она описывается волновой функцией — размытым облаком вероятностей. Это уже не инженерная проблема — это квантовая физика.

Индустрия отвечает на это несколькими способами. Первый — усложнение структуры чипа: вместо того чтобы делать транзисторы ещё меньше, их стали складывать в трёхмерные структуры. Так появились трёхмерные NAND-флеш-память и сложные многоуровневые чипы.

Второй способ — специализация. Вместо одного универсального процессора — несколько специализированных. GPU (графический процессор) считает задачи параллельно, что идеально для графики и нейросетей. NPU (нейронный процессор) оптимизирован специально для операций машинного обучения. TPU (тензорный процессор) разработан Google специально для своих нейросетей. Каждый делает своё дело лучше, чем универсальный.

Третий путь — квантовые вычисления. Это совсем другой принцип: вместо битов, которые равны строго 0 или 1, используются кубиты, которые могут находиться в суперпозиции — одновременно и 0, и 1. Квантовый компьютер может решать определённые задачи экспоненциально быстрее классического. Но технология пока сырая, хранить кубиты без ошибок крайне сложно, и до массового применения ещё далеко.

Так что закон Мура, может быть, и умирает в своей классической форме. Но инженерная изобретательность — нет. Просто теперь рост идёт не по одной оси (уменьшение транзистора), а сразу по нескольким.

Почему это всё важно для нас

Я думаю, что понимать основы технологий, которые пронизывают буквально каждый момент нашей жизни, — это что-то вроде умения читать. Никто не требует, чтобы ты знала лингвистику или историю письменности, чтобы читать книгу. Но понимание того, как язык работает, делает тебя более точным читателем, более внимательным к словам, более свободным в обращении с текстом.

То же самое с технологиями. Когда ты понимаешь, что компьютер — это просто очень быстрые переключатели, организованные очень умным образом, — он перестаёт быть магическим чёрным ящиком. Он становится инструментом.

Ещё одна причина — разговор о будущем. В следующих главах мы дойдём до роботов, до искусственного интеллек-

та, до вопросов о том, что произойдёт с нашей работой и нашим бытом в ближайшие десятилетия. Эти разговоры намного интереснее, когда ты понимаешь, из чего сделаны эти системы. Из кремния, из единиц и нулей, из логики И/ИЛИ/НЕ.

Итог: от переключателя к миру

Компьютер — это устройство, которое умеет работать с двумя состояниями: ноль и единица. Физически это реализуется через транзисторы — крошечные электронные переключатели на основе кремния. Современный смартфон содержит их больше десяти миллиардов, каждый размером в несколько нанометров.

Транзисторы объединяются в логические вентили (И, ИЛИ, НЕ), те — в схемы, те — в процессоры. Информация кодируется как последовательности битов: тексты, изображения, звук, видео — всё это просто числа, записанные в двоичной системе.

Закон Мура обеспечивал удвоение мощности чипов каждые два года на протяжении полувека. Сейчас классическая миниатюризация замедляется, и индустрия ищет новые пути: трёхмерные структуры, специализированные процессоры, квантовые вычисления.

Глава 2. Процессор: сердце, которое считает

Ты когда-нибудь замечала, как ноутбук начинает греть-

ся, когда открываешь слишком много вкладок? Или как телефон чуть тормозит в момент, когда одновременно снимает видео, загружает файл и проигрывает музыку? Это процессор работает на пределе своих возможностей.

В предыдущей главе мы разобрались, из чего сделан компьютер на самом нижнем уровне: транзисторы, биты, логические вентили. Теперь поднимаемся на один уровень выше и познакомимся с тем, кто дирижирует этим миллиардным оркестром.

Процессор. Центральный процессор. CPU — Central Processing Unit. Это сердце любого вычислительного устройства: компьютера, телефона, умной колонки, банкомата, навигатора в машине. Там, где есть вычисления — есть процессор.

Что делает процессор — в одном предложении

Процессор выполняет инструкции. Одну за другой, наверняка быстро.

Это звучит скромно. На самом деле это — всё. Буквально всё, что происходит в компьютере, сводится к тому, что процессор получает инструкцию, выполняет её и переходит к следующей. Показать пиксель на экране — инструкция. Сложить два числа — инструкция. Открыть файл — тысячи инструкций, следующих одна за другой.

Программа — это просто очень длинный список инструкций. Когда ты запускаешь браузер, операционная система загружает этот список в память и говорит процессору: «начи-

най отсюда». Дальше процессор читает и выполняет, читает и выполняет — пока не дойдёт до конца или пока ты не закроешь программу.

Инструкции бывают очень простыми. «Возьми число из ячейки памяти номер 1042». «Прибавь к нему число из ячейки 1043». «Запиши результат в ячейку 1044». «Если результат больше нуля, перейди к инструкции номер 278». Всё. Из таких крошечных шагов складываются операционные системы, игры, нейросети.

Это немного похоже на рецепт. Или на нотный лист. Ноты сами по себе — просто символы на бумаге. Но когда музыкант читает и играет их одну за другой — рождается симфония.

Внутри процессора: анатомия

Процессор — это не однородная пластина кремния. Внутри него несколько чётко разграниченных зон, каждая со своей задачей. Давай пройдемся по главным.

Арифметико-логическое устройство (АЛУ) — это «рабочие руки» процессора. Именно оно выполняет математические операции: сложение, вычитание, умножение, деление. И логические: сравнение чисел, побитовые операции. Всё, что связано с реальными вычислениями, проходит через АЛУ.

Регистры — это крошечная сверхбыстрая память прямо внутри процессора. Их немного — десятки, иногда сотни штук, — но они работают на той же скорости, что и сам

процессор, без какой-либо задержки. Это «руки», в которых процессор держит числа прямо во время работы: то, с чем он сейчас считает, и то, что только что посчитал.

Устройство управления — это «дирижёр» внутри дирижёра. Оно читает инструкции из памяти, декодирует их (то есть понимает, что именно нужно сделать) и отдаёт команды АЛУ и другим блокам. Оно не считает само — оно организует счёт.

Кэш-память — об этом подробнее чуть позже, но коротко: это несколько уровней быстрой памяти прямо на кристалле процессора. Промежуточное звено между молниеносными регистрами и относительно медленной оперативной памятью.

Блок предсказания переходов — один из самых элегантных механизмов. Программы полны условий: «если X, делай A, иначе делай B». Пока процессор выполняет проверку X, блок предсказания угадывает, какая ветка скорее всего понадобится, и заблаговременно начинает её подготавливать. Если угадал — выигрыш в скорости. Если нет — откатывается и начинает правильную ветку. Современные предсказатели угадывают верно в 95–99% случаев.

Это похоже на опытного официанта, который уже несёт тебе меню и воду, ещё пока ты только садишься за столик. Он не знает точно, что ты закажешь — но знает, что меню и вода нужны почти всегда.

Тактовая частота: метроном внутри

Ты наверняка видела в характеристиках компьютера что-то вроде «процессор 3,2 ГГц». Что это значит?

Внутри процессора есть генератор тактового сигнала — он похож на метроном, только вместо ударов в минуту — миллиарды ударов в секунду. Каждый «удар» — это один такт, один момент времени, когда процессор может выполнить элементарное действие.

Герц — это одно колебание в секунду. Гигагерц — миллиард. Процессор с частотой 3,2 ГГц делает 3,2 миллиарда тактов каждую секунду.

Раньше каждая инструкция требовала несколько тактов, но сегодня, благодаря конвейерной обработке, процессоры стремятся завершать в среднем больше одной инструкции за такт. Этот показатель называется IPC (instructions per cycle), и именно он, наряду с частотой, определяет реальную скорость.

Конвейер: почему процессор не ждёт



Рис. 2.1. Конвейерная обработка: четыре инструкции обрабатываются одновременно на разных стадиях.

Представь автомойку. Машина въезжает, проходит несколько зон: сначала намыливание, потом щётки, потом ополаскивание, потом сушка. Если делать это последовательно для каждой машины — то есть полностью вымыть одну, потом взять следующую — большинство зон большую часть времени простаивают.

Умная автомойка работает иначе: пока первая машина на ополаскивании, вторая уже на щётках, третья намыливается, четвёртая только въезжает. Все зоны заняты одновременно. Пропускная способность резко растёт.

Процессор работает точно так же. Это называется конвейер-

ерная обработка (pipeline). Выполнение каждой инструкции разбито на несколько стадий: выборка (взять инструкцию из памяти), декодирование (понять, что делать), исполнение (сделать), запись результата. Пока одна инструкция исполняется, следующая уже декодируется, а ещё следующая — выбирается из памяти.

Современные процессоры имеют конвейеры глубиной 10–20 стадий. Это значит, что одновременно «в пути» находятся 10–20 инструкций на разных стадиях готовности. И именно поэтому предсказание переходов так важно: если программа вдруг меняет направление (условный переход), весь конвейер нужно «промыть» и начать заново — а это потеря тех самых тактов, которые уже были потрачены на неверные инструкции.

Ядра: когда одного дирижёра недостаточно

До середины 2000-х годов процессоры были одноядерными. Чтобы сделать их быстрее, инженеры повышали тактовую частоту. Это работало — до определённого предела.

Проблема в том, что чем выше частота, тем чаще переключаются транзисторы, а каждое переключение требует затрат энергии на перезарядку крошечных конденсаторов внутри них — поэтому с ростом частоты резко растёт выделение тепла. К 2004 году некоторые процессоры Intel разогревались до температур, при которых охлаждение становилось инженерным кошмаром. Гонка за гигагерцами упёрлась в тепловой предел.

Решение оказалось элегантным: вместо одного очень быстрого ядра — два умеренно быстрых. Потом четыре. Потом восемь, шестнадцать, тридцать два.

Ядро — это, грубо говоря, отдельный полноценный процессор внутри одного чипа. Каждое ядро читает свои инструкции, выполняет свои задачи, имеет свои регистры и кэш. Несколько ядер могут работать параллельно — одновременно, независимо.

Это меняет всё — но с одной оговоркой. Параллельная работа нескольких ядер даёт выигрыш только тогда, когда задачу можно разбить на независимые части. Это называется параллелизм задачи.

Хороший пример: видеоредактор применяет фильтр к миллиону пикселей. Каждый пиксель обрабатывается независимо — можно раздать работу по ядрам, и она выполнится в восемь раз быстрее на восьмиядерном процессоре.

Плохой пример: ты читаешь книгу. Каждая следующая страница зависит от предыдущей. Такая задача называется последовательной, и здесь дополнительные ядра почти не помогают.

Вот почему для разных задач нужны разные процессоры. Видеомонтаж, рендеринг, машинное обучение — любят много ядер. Однопользовательская игра, работа с базой данных в реальном времени, сложные расчёты с длинными цепочками зависимостей — предпочитают меньше ядер, но быстрее.

Кэш: память, о которой ты не думаешь

Представь, что ты работаешь за столом. На столе — несколько листов бумаги с числами, которые тебе сейчас нужны. В ящике стола — толстая папка с данными, к которым обращаешься реже. В шкафу за спиной — архив со всем остальным. На складе через улицу — старые дела, которые почти никогда не нужны.

Именно так организована память в компьютере. И именно так работает кэш.

Оперативная память (RAM) — это, грубо говоря, склад через улицу. Большая, вмещает много, но добраться до неё занимает время. В масштабах процессора — целые десятки и сотни тактов ожидания. Для процессора, который делает миллиарды операций в секунду, это катастрофически долго.

Кэш — это стол и ящики. Маленький (мегабайты против гигабайт RAM), но находится прямо на кристалле процессора и работает на его скорости.

Современные процессоры имеют несколько уровней кэша — обычно три.

L1 — самый маленький (32–64 килобайта на ядро) и самый быстрый. Доступ — 4–5 тактов. Это буквально «стол»: то, с чем процессор работает прямо сейчас.

L2 — побольше (256 килобайт — несколько мегабайт), чуть медленнее. 10–20 тактов. Ящик стола.

L3 — самый большой (8–64 мегабайта, иногда больше), общий для всех ядер. 30–50 тактов. Шкаф.

И только если данных нет ни в одном кэше — процессор идёт в RAM. Это называется промах кэша, и это дорого: 100–300 тактов ожидания, в течение которых процессор фактически стоит на месте.

Вся архитектура кэша построена на принципе локальности: если процессор обратился к какому-то куску памяти, он, скорее всего, скоро обратится к соседнему куску. Поэтому кэш загружает не один байт, а целую «строку» — 64 байта подряд. Это работает, потому что программы действительно обрабатывают данные последовательно — массивы, строки текста, кадры видео.

Кстати, именно поэтому хорошие программисты думают о кэше. Код, который обращается к памяти последовательно, работает в разы быстрее кода, который «прыгает» по случайным адресам — даже если количество операций одинаково. Разница в производительности может быть десятикратной просто за счёт того, попадают ли данные в кэш или нет.

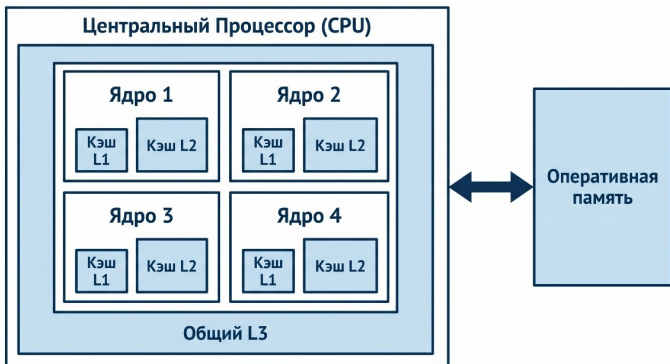


Рис. 2.2. Иерархия ядер и кэша: L1 и L2 у каждого ядра, общий L3 для всех.

Процессор и тепло: почему ноутбук греется

Вернёмся к тому, с чего начали: горячий ноутбук.

Каждый переключающийся транзистор потребляет крошечное количество энергии. Умножь крошечное на миллиарды — получишь заметное. Умножь заметное на миллиарды переключений в секунду — получишь тепло.

Это фундаментальная физика: электрический ток в проводнике всегда порождает тепло. Избежать этого нельзя — можно только управлять им.

Именно поэтому в компьютерах есть системы охлаждения: радиаторы, вентиляторы, тепловые трубки, а в дорогих

игровых системах — жидкостное охлаждение. Тепло от процессора нужно отвести, иначе он перегреется и начнёт замедляться или вовсе выключится — это встроенная защита.

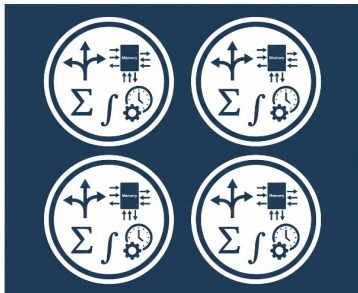
Современные процессоры умеют динамически менять свою частоту и напряжение в зависимости от нагрузки. Когда ты читаешь статью — частота низкая, тепла мало, батарея расходует медленнее. Открыла тяжёлое приложение — частота растёт, тепло растёт, вентилятор начинает шуметь. Это называется динамическое масштабирование частоты (DVFS — Dynamic Voltage and Frequency Scaling).

В смартфонах та же история, только охлаждение пассивное — радиатор без вентилятора. Поэтому телефон при долгой нагрузке может заметно нагреваться и замедляться: это не поломка, а тепловой троттлинг — осознанное снижение частоты, чтобы не перегреться.

Apple несколько лет назад произвела впечатление на всю индустрию своими чипами серии M. Секрет не только в высокой производительности — а в том, что такую производительность удалось получить при очень низком энергопотреблении. MacBook на M-чипе работает тихо, почти не греется и держит заряд весь день. Это результат нескольких архитектурных решений сразу: эффективные ядра для простых задач, производительные — для сложных, единая память для процессора и графики. Об этом — чуть ниже.

CPU против GPU: два разных подхода к скорости

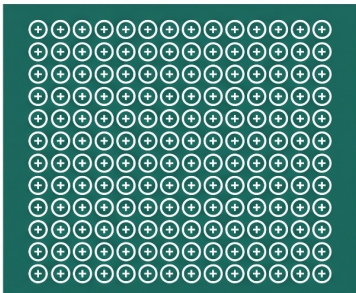
Процессор (CPU)



МОЩНЫЕ ЯДРА

Каждое ядро умеет всё: логика, ветвления, сложные вычисления

Видеокарта (GPU)



ПРОСТЫЕ ЯДРА

Каждое ядро умеет одно: простая математика, но все одновременно

Рис. 2.3. CPU против GPU: несколько мощных ядер против тысяч простых.

Мы много говорили о центральном процессоре — CPU. Но в современных устройствах почти всегда есть ещё один процессор — графический, GPU (Graphics Processing Unit). И разница между ними — отличная иллюстрация того, что «быстрее» значит разное для разных задач.

CPU — это несколько очень умных и очень быстрых ядер. Современный топовый настольный процессор имеет 8–24 производительных ядер. Каждое ядро может быстро выполнять сложные, разветвлённые последовательности инструкций, справляться с непредсказуемыми задачами, обрабатывать ошибки, управлять другими компонентами. CPU — это

универсальный солдат.

GPU — это тысячи маленьких, простых ядер. Видеокарта NVIDIA RTX 4090 содержит 16 384 шейдерных ядер. Каждое из них значительно проще и медленнее одного ядра CPU. Но когда им дают задачу, которую можно разбить на тысячи независимых маленьких подзадач — GPU сокрушает CPU.

Классическая задача для GPU — отрисовка 3D-сцены. На экране миллионы пикселей. Цвет каждого пикселя нужно вычислить на основе геометрии, источников света, текстур, теней. Это независимые вычисления — каждый пиксель не зависит от соседнего. GPU запускает тысячи таких вычислений параллельно и за долю секунды рисует кадр.

А потом оказалось, что та же математика — тысячи параллельных умножений матриц — лежит в основе обучения нейронных сетей. И GPU, созданные для игровой графики, вдруг стали главным инструментом машинного обучения. Именно поэтому сегодня NVIDIA — одна из самых дорогих компаний в мире: её GPU используются не для игр, а для обучения больших языковых моделей.

Смартфоны и современные ноутбуки всё чаще объединяют CPU и GPU в один чип — систему на кристалле (SoC, System on Chip). Кроме того, туда добавляют специализированные блоки: нейронный процессор (NPU) для задач ИИ, видеodeкодер, сигнальный процессор для камеры. Это называется гетерогенная архитектура: не один универсальный

блок, а оркестр специалистов.

Как процессор «общается» с остальными частями

Процессор — главный, но не единственный. Ему нужно постоянно взаимодействовать с памятью, хранилищем, экраном, клавиатурой, сетью. Как это происходит?

Всё общение идёт через шины — наборы электрических проводников, по которым передаются данные и команды. Главная шина соединяет процессор с оперативной памятью и называется шиной памяти или системной шиной. Есть шины для периферии: PCIe (по ней подключается видеокарта и быстрые SSD), USB, SATA.

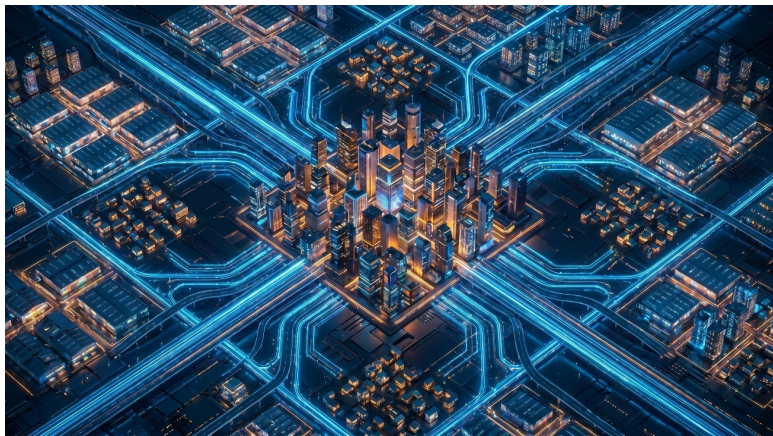


Рис. 2.4. Процессор как город: специализированные кварталы соединены шинами данных.

Но современные чипы уходят от классических шин в сторону более сложных межсоединений прямо на кристалле. В упомянутых чипах Apple M вся система — процессор, графика, память — находится на одном кристалле, и данные между ними передаются по внутренним высокоскоростным шинам. Это называется *unified memory architecture*, единая архитектура памяти: CPU и GPU видят одну и ту же оперативную память без необходимости копировать данные туда-обратно. Отсюда и эффективность.

Когда процессору нужны данные с диска или из сети, он не делает это сам. Он отдаёт команду контроллеру соответствующего устройства: «принеси мне данные оттуда». Пока контроллер работает, процессор может делать что-то другое. Когда данные готовы, контроллер посылает процессору прерывание — сигнал «я готов». Процессор прерывает текущую работу, забирает данные и продолжает. Прерывания происходят сотни тысяч раз в секунду — ты нажала клавишу, пришёл сетевой пакет, таймер сработал, звуковая карта требует следующую порцию звука.

Немного истории: как процессоры стали такими

Intel 4004, 1971 год. Первый коммерческий микропроцессор. 2300 транзисторов, тактовая частота 740 кГц. Разработан для японского калькулятора Busicom. Производительность — примерно 60 000 операций в секунду.

Intel 8086, 1978 год. Предок всей архитектуры x86, которая до сих пор используется в большинстве персональных

компьютеров. 29 000 транзисторов, 5–10 МГц.

Intel Pentium, 1993 год. 3,1 миллиона транзисторов, 60–66 МГц. Первый процессор, который появился в домах по-настоящему массово.

Intel Core 2 Duo, 2006 год. Первый массовый двухъядерный процессор для настольных ПК. Переломный момент: индустрия признала, что путь «быстрее и горячее» исчерпан, и перешла к пути «больше ядер».

Apple M1, 2020 год. 16 миллиардов транзисторов, 5-нанометровый техпроцесс, 8 ядер CPU плюс 8 ядер GPU на одном кристалле. Переломный момент другого рода: в реализации Apple — с унифицированной памятью, тесной интеграцией CPU и GPU на одном кристалле и широким конвейером — ARM-архитектура оказалась эффективнее традиционных x86-ноутбуков.

Apple M4 Max, 2025 год. Около 92 миллиардов транзисторов, 3-нанометровый техпроцесс, до 16 ядер CPU и 40 ядер GPU на одном кристалле. Производительность, которая ещё десять лет назад требовала стойки серверного оборудования.

За 54 года — от 2300 транзисторов до 200 миллиардов. Рост в 87 миллионов раз. Ни одна другая технология в истории человечества не демонстрировала ничего подобного.

Что происходит за долю секунды

Давай пройдем весь путь один раз — медленно, чтобы почувствовать масштаб.

Ты пишешь в мессенджере букву «п» и нажимаешь кла-

вишу.

Контроллер клавиатуры фиксирует нажатие и посылает процессору прерывание. Процессор откладывает текущую задачу, переключается на обработчик прерывания клавиатуры. Обработчик определяет, какая клавиша нажата, — смотрит в таблицу кодов. Код «п» — это число 1087 в кодировке Unicode.

Процессор передаёт это число операционной системе. Та — в фокусное окно (мессенджер). Программа мессенджера принимает символ, добавляет его в строку вводимого текста, просит операционную систему перерисовать текстовое поле. Операционная система передаёт задачу системе отрисовки, которая вычисляет, какие пиксели должны измениться. GPU получает команду отрисовки — и за следующий кадр (менее 16 миллисекунд при 60 кадрах в секунду) обновляет экран.

Ты видишь букву «п» на экране.

Всё это — от нажатия до отображения — занимает единицы миллисекунд. За это время процессор выполнил миллионы инструкций.

И это — простой случай. Нажатие «Отправить» запустит цепочку на порядки длиннее: шифрование, формирование сетевого пакета, передача через Wi-Fi или мобильную сеть, маршрутизация через несколько серверов, доставка собеседнику, его устройство расшифровывает, отображает, посылает уведомление о прочтении обратно.

Почему это важно понимать

Мы живём в мире, где вычислительная мощь стала дешёвой и повсеместной. Смартфон в твоём кармане мощнее суперкомпьютеров 1990-х годов. За этой обыденностью стоит невероятная инженерная работа. Каждое поколение процессоров — это годы исследований, миллиарды долларов инвестиций, решение задач, которые ещё недавно казались нерешаемыми.

Понимать, как работает процессор, полезно не только само по себе. Это меняет то, как ты думаешь о производительности, о том, почему одни программы работают быстро, а другие медленно, о том, почему «больше ядер» не всегда значит «быстрее» для твоей конкретной задачи.

И это даёт контекст для разговора о том, что будет дальше. Потому что архитектурные решения в процессорах — параллелизм, специализация, эффективность — это ровно те же принципы, которые лежат в основе робототехники и искусственного интеллекта.

Итог: дирижёр миллиардного оркестра

Процессор — это устройство, которое выполняет инструкции. Одну за другой, в строгом порядке, с невероятной скоростью.

Внутри него — несколько специализированных блоков: АЛУ для вычислений, регистры для мгновенного хранения, устройство управления для координации, кэш для быстрого доступа к памяти, предсказатель переходов для опережающей работы.

Конвейер позволяет обрабатывать несколько инструкций одновременно на разных стадиях. Несколько ядер позволяют выполнять несколько потоков инструкций параллельно. Тактовая частота задаёт ритм, но не является единственным показателем скорости.

CPU и GPU — два разных подхода: несколько умных и быстрых ядер против тысяч простых и параллельных. Разные задачи требуют разного.

Каждое нажатие клавиши запускает миллионы операций. Каждый кадр на экране — результат слаженной работы процессора, памяти, графики, операционной системы.

Глава 3. Память: краткосрочная и долгая

Ты работаешь над презентацией. Открыто несколько вкладок с референсами, текстовый редактор, таблица с данными, мессенджер. Потом внезапно отключается свет. Компьютер гаснет.

Ты включаешь его снова. Презентация — там, где сохранила в последний раз. Вкладки браузера — часть восстановилась, часть нет. Всё, что не было сохранено за последние полчаса, — исчезло.

Это принципиальное устройство компьютерной памяти. И чтобы его понять, нужно разобраться в одном ключевом различии — между памятью, которая живёт, пока есть питание, и памятью, которая помнит, даже когда ты всё

выключила.

Два типа памяти: рабочий стол и архив

В компьютере есть два принципиально разных вида хранения информации — оперативная память (RAM) и постоянное хранилище (жёсткий диск или SSD). Их часто путают, особенно когда говорят просто «память». Но это совершенно разные вещи, с разными задачами, разными технологиями и разными свойствами.

Самая простая аналогия — рабочий стол и архив.

RAM — это твой рабочий стол. Большой, удобный, всё под рукой. Сюда ты раскладываешь всё, с чем работаешь прямо сейчас: открытые документы, запущенные программы, данные, которые нужны немедленно. Стол быстрый — протянула руку, взяла. Но когда ты уходишь домой — убираешь всё со стола. Завтра он снова пустой.

SSD (или жёсткий диск) — это архив. Шкаф с папками, где хранится всё, что ты когда-либо сохранила: файлы, фотографии, программы, операционная система. Архив медленнее стола — нужно встать, найти нужную папку, достать документ. Зато он никуда не исчезает. Ты можешь уйти на год, вернуться — а все файлы на месте.

Когда ты включаешь компьютер, он берёт часть содержимого архива (операционную систему, необходимые программы) и раскладывает на стол. Когда открываешь документ — он тоже переключается на стол. Когда сохраняешь — копия уходит обратно в архив. Когда выключаешь — стол очища-

ется.

Именно поэтому несохранённые изменения пропадают при отключении питания: они жили только на столе и не добрались до архива.

Оперативная память: как она устроена

RAM — Random Access Memory, память с произвольным доступом. «Произвольный доступ» означает, что можно мгновенно обратиться к любой ячейке памяти по её адресу — не нужно перематывать ленту с начала, как в старых магнитофонах, не нужно последовательно перебирать содержимое. Указал адрес — получил данные.

Физически современная оперативная память — это тип, называемый DRAM (Dynamic RAM, динамическая оперативная память). Каждый бит хранится в крошечном конденсаторе — элементе, способном удерживать электрический заряд. Заряд есть — единица. Заряда нет — ноль.

Но у конденсаторов есть неприятная особенность: они постепенно разряжаются. Сами по себе, просто со временем. Если ничего не делать, все биты превратятся в нули — данные исчезнут.

Поэтому оперативная память постоянно «обновляется» — специальная схема регулярно считывает каждую ячейку и восстанавливает заряд, если он упал ниже порога. Это происходит тысячи раз в секунду, для каждой ячейки из миллиардов. Именно поэтому она называется «динамической» — данные нужно постоянно поддерживать.

И именно поэтому RAM энергозависима: как только пропадает питание — некому обновлять конденсаторы, заряд утекает, данные исчезают. Секунды — и стол пуст.

Скорость у оперативной памяти впечатляющая: современные планки DDR5 передают данные со скоростью десятков гигабайт в секунду. Это в сотни раз быстрее, чем даже самый быстрый SSD. Именно поэтому процессор работает с данными в RAM, а не напрямую с диска.

Сколько RAM нужно и почему

Объём оперативной памяти определяет, сколько задач компьютер может держать «на столе» одновременно.

Когда RAM заканчивается, операционная система начинает хитрить: она берёт те части памяти, к которым давно не обращались, и временно выгружает их на диск — в так называемый файл подкачки (swap). Когда эти данные снова понадобятся, она загружает их обратно в RAM, попутно выгружая что-то другое.

Технически это работает. Практически — это катастрофа для скорости. Диск в сотни раз медленнее RAM. Когда система начинает активно использовать подкачку, компьютер «тормозит»: всё начинает работать медленно, вентиляторы гудят, диск шуршит. Если ты когда-нибудь сидела и смотрела, как курсор превращается в крутящееся колёсико на несколько секунд — скорее всего, это было оно.

Сколько RAM достаточно? Это быстро меняющийся ответ, но сегодня ориентиры примерно такие. 8 гигабайт — ми-

нимум для комфортной работы с офисными задачами. 16 гигабайт — комфорт для большинства пользователей, включая несколько браузерных вкладок, редактирование фото, лёгкий монтаж. 32 гигабайта и больше — профессиональный видеомонтаж, работа с большими таблицами данных, разработка, виртуальные машины.

Один из мифов про компьютеры: «RAM нужно освободить». Некоторые приложения даже продаются как «очистители памяти». Операционная система сама управляет памятью — и незанятая RAM это не «чистая», а «бесполезная» RAM. Хорошая операционная система всегда держит память максимально заполненной полезными данными, чтобы не ходить лишний раз на диск. На macOS и iOS система особенно агрессивно сжимает данные в памяти, поэтому 90% заполненности — действительно норма и не повод для беспокойства. На Windows высокое потребление памяти тоже в целом нормально: система использует свободную RAM под кэш файлов. Заметные торможения появляются лишь тогда, когда начинается активный сброс данных в файл подкачки на диске — а при 16+ ГБ RAM это случается редко.

Твёрдотельный накопитель: флеш-память и её секреты

Переходим к постоянному хранилищу. Сегодня в большинстве ноутбуков и смартфонов стоит SSD — Solid State Drive, твёрдотельный накопитель. До него царили жёсткие диски — HDD (Hard Disk Drive). Давай разберёмся с обои-

ми.

Жёсткий диск — это механическое устройство. Внутри него вращаются магнитные пластины — «блины» — с огромной скоростью: 5400 или 7200 оборотов в минуту. Над ними скользит магнитная головка, которая считывает и записывает данные, намагничивая крошечные участки поверхности. Намагничен в одну сторону — единица, в другую — ноль.

Это работает. И работало десятилетиями. Но у механики есть неустранимые ограничения: нужно физически переместить головку к нужному месту на диске — а это занимает время, измеряемое в миллисекундах. По меркам процессора — вечность.

SSD работает принципиально иначе. Никаких движущихся частей. Данные хранятся во флеш-памяти — особом типе памяти, которая, в отличие от RAM, не требует питания для хранения данных.

Физически флеш-память построена на транзисторах с плавающим затвором. Обычный транзистор имеет один управляющий электрод — затвор. У флеш-транзистора их два: один управляющий и один «плавающий» — изолированный со всех сторон слоем диэлектрика. На этот плавающий затвор можно загнать электроны и запереть их там. Они никуда не денутся без специального воздействия — изоляция держит.

Есть заряд на плавающем затворе — транзистор ведёт се-

бя иначе, чем без заряда. Это и есть бит. Ноль или единица. Без питания, без обновления, без ухода — просто сидят запертые электроны и хранят информацию.

Теоретически флеш-ячейка может хранить заряд десятилетиями. На практике — столько, сколько позволяет изоляция, которая со временем деградирует.

SLC, MLC, TLC, QLC: сколько бит в одной ячейке

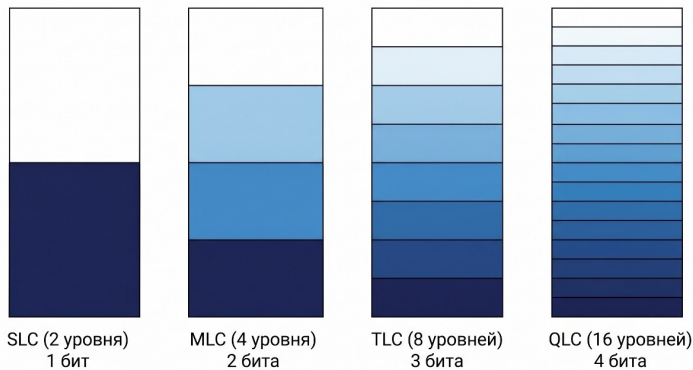


Рис. 3.1. Типы ячеек флеш-памяти: чем больше бит в ячейке, тем ниже ресурс записи.

Самое простое устройство флеш-ячейки — SLC, Single Level Cell. Одна ячейка хранит один бит: заряд есть или нет. Просто, надёжно, быстро — и дорого, потому что на единицу площади помещается мало данных.

Инженеры решили: а что, если различать не два уровня заряда, а четыре? Тогда одна ячейка хранит два бита: 00, 01, 10, 11 — четыре состояния вместо двух. Это MLC, Multi Level Cell. Плотность удваивается, цена снижается.

Потом пошли дальше: восемь уровней заряда в одной ячейке — три бита. TLC, Triple Level Cell. Именно этот тип стоит в большинстве потребительских SSD и смартфонов сегодня.

И наконец QLC — Quad Level Cell: шестнадцать уровней, четыре бита. Максимальная плотность, минимальная цена. Используется в недорогих накопителях и флеш-накопителях большого объёма.

Но за плотность приходится платить. Чем больше уровней нужно различать, тем точнее должен быть контроллер при чтении и записи. Разница между соседними уровнями заряда становится всё меньше — и тем сложнее их надёжно различить. Результат: QLC медленнее SLC, быстрее деградирует и менее надёжна.

Для обычного пользователя это практически незаметно — современные контроллеры SSD делают огромную работу по исправлению ошибок и управлению ресурсом. Но профессионалы, которым нужна максимальная надёжность и скорость, — выбирают SLC или MLC, платя за это значительно больше.

Почему SSD изнашивается — и почему это не страшно

Флеш-память имеет ограниченный ресурс записи. Каждый раз, когда ты записываешь данные в ячейку, слой изоляции вокруг плавающего затвора чуть-чуть деградирует. После определённого количества циклов записи ячейка перестаёт надёжно удерживать заряд — становится «битой».

Для современных 3D TLC-ячеек ресурс составляет порядка 3000–5000 циклов перезаписи на одну ячейку. Для QLC — около 1000–1500 циклов: именно поэтому QLC-накопители дешёвы, но менее долговечны при интенсивной записи. Звучит мало. Но контроллер SSD применяет технологию, называемую выравниванием износа (wear leveling): он следит за тем, чтобы данные равномерно распределялись по всем ячейкам, а не писались постоянно в одни и те же. Так нагрузка размазывается по всему накопителю, и реальный срок службы оказывается многолетним даже при интенсивном использовании. TLC — золотая середина для большинства пользователей; QLC стоит выбирать только если накопитель нужен для хранения, а не для постоянной перезаписи.

Производители указывают ресурс SSD в TBW — Terabytes Written, терабайт записанных данных. Типичный потребительский SSD на 1 терабайт имеет ресурс 300–600 TBW. Если ты записываешь 50 гигабайт в день (что очень много для обычного пользователя), это 18 терабайт в год. До исчерпания ресурса — 15–30 лет.

Есть ещё одна особенность флеш-памяти, которая влияет на скорость: перед записью нового значения ячейку нужно

сначала стереть. И стереть нельзя по одной ячейке — только блоками, которые содержат тысячи ячеек сразу. Это создаёт интересную ситуацию: запись маленького файла может требовать считывания целого блока, его изменения и обратной записи — даже если менялась одна ячейка. Это называется *write amplification*, усиление записи, и хороший контроллер минимизирует это через хитрые алгоритмы управления данными.

NVMe, SATA, eMMC: буквы, которые определяют скорость

SSD подключаются к компьютеру по-разному, и это сильно влияет на скорость.

SATA — старый стандарт, разработанный ещё для жёстких дисков. Максимальная скорость передачи данных — около 550 мегабайт в секунду. Для первых SSD это было революцией по сравнению с HDD (100–150 МБ/с). Но интерфейс стал узким местом: флеш-память способна на большее.

NVMe — новый стандарт, разработанный специально для флеш-памяти, подключается через шину PCIe напрямую к процессору. Скорость — 3500–7000 мегабайт в секунду для современных накопителей. В 10 раз быстрее SATA. Именно NVMe стоит в большинстве современных ноутбуков и настольных ПК.

eMMC — тип памяти, используемый в бюджетных устройствах и смартфонах. Медленнее NVMe, но компактнее и дешевле. Смартфоны высокого класса используют UFS

(Universal Flash Storage) — быстрее eMMC, медленнее лучших NVMe, зато очень компактный и энергоэффективный.

Скорость хранилища ощущается в конкретных вещах: время загрузки операционной системы, скорость открытия программ и файлов, скорость копирования данных. Переход с HDD на NVMe SSD — это, пожалуй, самое ощутимое ускорение, которое можно сделать со старым компьютером.

Жёсткий диск: почему он ещё не умер

Раз SSD быстрее, надёжнее (нет движущихся частей) и потребляет меньше энергии — почему жёсткие диски вообще ещё существуют?

Цена за гигабайт. Жёсткий диск объёмом 4 терабайта стоит примерно столько же, сколько SSD на 500 гигабайт. Разница в восемь раз. Для задач, где нужно хранить огромные объёмы данных и скорость не критична — резервные копии, архивы видео, серверное хранилище, — HDD остаётся оптимальным выбором.

Дата-центры Google, Amazon, Microsoft хранят колоссальные объёмы данных именно на жёстких дисках — петабайты и эксабайты архивов, резервных копий, редко запрашиваемых данных. Для этих задач цена важнее скорости.

Плотность записи на жёстких дисках тоже продолжает расти. Современные HDD используют технологию HAMR (Heat-Assisted Magnetic Recording) — нагрев лазером позволяет записывать биты на ещё более крошечных участках поверхности. Накопители на 20–30 терабайт уже существуют и

постепенно дешевеют.

Но для персональных компьютеров и ноутбуков жёсткие диски уходят в прошлое. Разница в ощущениях между ноутбуком с HDD и с NVMe SSD настолько велика, что экономить на этом не имеет смысла.

Как компьютер организует данные: файловая система

Мы разобрались с физическим уровнем памяти. Но как компьютер знает, где что лежит? Как он находит твой файл среди терабайтов данных?

Ответ — файловая система. Это программный слой, который организует данные на диске и предоставляет привычный нам интерфейс: файлы, папки, имена.

На самом низком уровне диск — это просто огромная последовательность пронумерованных блоков, по 4 или 8 килобайт каждый. Файловая система ведёт таблицу: «файл vacation.jpg занимает блоки 1042, 1043, 1050, 1051, 2307». Когда ты открываешь файл — система смотрит в таблицу, собирает нужные блоки и передаёт их как единое целое.

Файл не обязан лежать в памяти подряд. Он может быть разбросан по всему диску — это называется фрагментация. На жёстких дисках это было серьёзной проблемой: головка должна была физически перемещаться между разными участками пластины, что замедляло чтение. Отсюда — «дефрагментация», которую советовали делать раз в месяц.

Конец ознакомительного фрагмента.

Текст предоставлен ООО «Литрес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на Литрес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.