

ИСТОРИЯ ЭЛЕКТРОННЫХ КОМПЬЮТЕРОВ

От ламповых машин к микропроцессорам

Леонид Черняк



Леонид Черняк

**История электронных
компьютеров**

«Автор»

2026

Черняк Л.

История электронных компьютеров / Л. Черняк — «Автор»,
2026

Книга состоит из двух частей – в первой последовательно изложен процесс эволюции компьютеров от первых – ламповых до современных, во второй – эволюция программного обеспечения от языка ассемблера до наших дней.

© Черняк Л., 2026
© Автор, 2026

Содержание

Часть I. История электронных компьютеров	7
Введение	7
Когда люди были компьютерами	7
Люди-компьютеры в России и СССР	9
Компьютинг как основа цифровой цивилизации	9
Компьютинг и пирамида DIKW: эволюция от данных к здравому смыслу и принятию решений	11
Компьютинг в повседневной жизни	12
Глава 1. Аппаратная и программная основы компьютеринга	13
Об электронике	13
Эпоха ламп	14
Лампы как основа компьютеров первого поколения	15
История полупроводников и зарождение транзисторной эры	16
Ранние открытия: от Вольта до Брауна	16
Лосев и кристадин	16
Другие пионеры	17
Bell Labs и рождение транзистора	17
Конфликты, приведшие к созданию будущего	17
Влияние на Силиконовую долину	18
Гордон Тил и Texas Instruments	19
Память	19
Магнитная память на ферритовых сердечниках	19
Пионеры ферритовой памяти	20
Полупроводниковая память	21
Интегральные микросхемы	22
Микропроцессоры	23
Почему мы все еще говорим «микропроцессор»?	23
CISC и RISC	23
CISC–Intel, восхождение, доминирование и уступка трона	24
Apple и ARM: уход в другое измерение	25
История RISC	25
Три школы RISC: три взгляда на простоту	25
Первые RISC-процессоры	26
Период 2000–2025: Конец безальтернативности	26
Apple и ARM	27
RISC–V: открытая архитектура, которая меняет правила игры	27
Исторические корни RISC–V	28
Графический процессор: от рендеринга к ИИ	28
Рождение GPU: NVIDIA и ATI	29
GPU выходит за рамки	29
GPU и искусственный интеллект	29
Глава 2. Нулевое поколений компьютеров	30
Первопроходцы	30
Великолепная семерка	31

ABC – машина, воскресшая по суду	31
Компьютеры на телефонных реле	32
Калькулятор с дистанционным доступом	33
Harvard Mark I	34
Наследник Аналитической машины Бэббиджа	35
ENIAC – в одном шаге от настоящего компьютера	36
Глава 3. Школа Мура и создание первого электронного компьютера	38
Феномен Школы Мура: инженерия и наука	38
От стереотипов к реальности	38
Цикл науки, инженерии и менеджмента	39
Роль Школы Мура: случайность или закономерность?	39
Первый электронный компьютер	43
Глава 4. EDVAC	46
От ENIAC к EDVAC	46
Хранимая программа – гениальная простота	47
Система команд	47
О Джоне фон Неймане	47
Тайм-лайн EDVAC	49
Когда теория пытается догнать практику	51
Глава 5. Компьютеры первого поколения	53
IAS	53
Основные черты IAS	54
UNIVAC – первый коммерческий компьютер	54
BINAC: первая попытка создания коммерческой машины	54
UNIVAC I	55
UNIVAC I на выборах 1952 года	55
UNIVAC II и последующие модели	56
Whirlwind I: первый компьютер реального времени	56
Архитектура и принципы работы	56
Влияние на последующее поколение компьютеров	57
IBM: от калькулятора до империи	57
SSEC – «калькулятор для витрины»	57
IBM 701 – рождение настоящего компьютера	58
IBM100 – SAGE	58
Британский путь в компьютеринге	59
От Colossus к «национальному компьютеру»	60
Манчестерская инженерная школа	60
Кембридж и рождение бизнес-компьютеринга	61
Наследие британского пути	61
Барабанные машины	62
Глава 6. Второе поколение ЭВМ	64
Революция в элементной базе и памяти	65
Транзисторы	65
Ферритовая революция	65
Внешняя память: магнитная эра	65
Компьютеры новых типов	66
Мэйнфреймы второго поколения	68

Глава 7. Мэйнфреймы – от третьего поколения до наших дней	72
Мэйнфреймы третьего поколения	72
Рождение IBM System/360	73
От идеи к железу: реализация System/360	74
Основные трудности при создании IBM System/360	74
Принципиальные новшества System/360	75
Железо и софт	76
Модели IBM/360	77
IBM System/370	79
IBM System/390	80
IBM zSeries (начало 2000-х)	80
IBM System z	81
Конец ознакомительного фрагмента.	83

Леонид Черняк

История электронных компьютеров

Часть I. История электронных компьютеров

Введение

Когда люди были компьютерами

В 1613 году впервые в английском языке было зафиксировано употребление слова *computer*, тогда оно значило «человек считающий». Так много лет называли человека, выполнявшего вычисления, эта профессия имеет богатую историю. Передо мной лежит объемистая книга с неожиданным для современников названием «When computers were human», в ней более 400 страниц. Ее автор – профессор Университета Джорджа Вашингтона Дэвид Гриер. Он раскрывает эту историю с глубокой древности до середины прошлого века.

С появлением электронных компьютеров в 1950-х профессия человека-компьютера естественным образом исчезла, но ее наследие никуда не делось. Именно эти люди выработали культуру точности и ответственности, документирования и проверки – основы будущего программирования. Первые программисты во многом унаследовали мышление своих предшественников: они тоже управляли вычислительными процессами, но на другом уровне.

Сегодня мы видим в людях-компьютерах не только предшественников машин, но и участников глубокой культурной трансформации. Они превратили вычисление из индивидуального умения в технологический процесс и тем самым подготовили появление того вычислительного мира, в котором мы живем.

Еще в Древнем Египте и Месопотамии существовала особая категория писцов, чьей обязанностью был счет: они учитывали урожай, следили за налогами, контролировали производство каменных блоков для пирамид. Их труд был основой государственного управления. В античной науке мы встречаем ту же практику. Птолемей и другие астрономы не работали в одиночку: рядом с ними находились ученики и помощники, которые сводили наблюдения в таблицы, складывали и вычитали углы, проводили рутинные операции. Тогда-то и возникла форма коллективного счета, сохранившаяся надолго – ученый задавал формулу, а его «компьютеры» выполняли ее механически.

Развитие науки и мореплавания в эпоху Возрождения многократно увеличило потребности в вычислениях. В ответ возникли такие «предприятия» как, например, обсерватория Тихо Браге в Ураниборге на острове Вен, она работала как настоящая вычислительная мастерская. Несколько десятков ассистентов, студентов и практикантов занимались наблюдениями и сводили результаты в таблицы. У них было свое разделение труда – один переводил углы в градусы, другой складывал и вычитал величины, третий заносил итог в журнал. Обстановка напоминала ремесленную мануфактуру, где каждый выполнял свою часть работы, и только вместе они могли получить готовый результат.

К концу XVIII века организация вычислительного труда стала еще более похожей на фабрику. Знаменитый пример – «мануфактура логарифмов» барона де Прони. Она стала ответом на запрос администрации императора Наполеона I, которой требовались новые таблицы прежде всего в связи с переходом на метрическую систему. Прони организовал труд расчетчиков наподобие пирамиды, на ее вершине находилась небольшая группа математиков, которые

выводили формулы. Средний слой «мастеров» превращал формулы в пошаговые инструкции. А в основании пирамиды трудились десятки женщин, отличавшихся в прошлом низкой социальной ответственностью, им доставались только операции сложения и вычитания. Их тетради были заполнены колонками чисел с простыми указаниями: «возьмите значение из столбца А, прибавьте к столбцу В, результат занесите в С». Никто из бывших жриц любви не понимал всей картины в целом, но их усилиями рождались самые разнообразные математические таблицы. Это был первый в истории опыт индустриализации математического труда.

В XIX веке такие практики стали нормой. Государства и университеты создавали вычислительные бюро, где десятки и сотни «людей-компьютеров» трудились над морскими эфемеридами, астрономическими и навигационными таблицами. В рабочих залах британской Королевской обсерватории многочисленные сотрудники сидели за длинными столами, выполняя однотипные операции. Каждый результат проверялся дважды или даже трижды: если числа не совпадали, вычисление делали снова. В США данные о переписи населения с середины XIX века обрабатывались целыми армиями счетчиков. До появления табуляторов и перфокарт Холлерита тысячи людей с бумагой и карандашами складывали и сводили данные воедино.

Особую страницу в этой истории открыли женщины-компьютеры. В конце XIX века директор Гарвардской обсерватории Эдвард Пикеринг нанял группу женщин, вошедших в историю как «гарвардские компьютеры». Работа была рутинной – они сидели среди ящиков с фотопластинами, пользуясь лупами и линейками, измеряли расстояния, переводили их в числа, складывали колонки, записывали результаты в журналы. Ошибки отмечались красным карандашом, каждый шаг проверялся. За скромную зарплату они выполняли механическую работу – и при этом именно их труд привел к открытиям, изменившим астрономию.

Первая мировая война потребовала создания баллистических таблиц для артиллерии и для инженерных расчетов. Были организованы специальные бюро, где студенты и клерки неделями решали одни и те же уравнения сотни раз. Во Вторую мировую войну таких людей стали привлекать тысячами. В США при армии и флоте создавались женские вычислительные группы. Большие залы напоминали экзаменационные аудитории: десятки женщин сидели за рядами столов с карандашами, бумагой и арифмометрами. Каждая выполняла один шаг численного решения дифференциального уравнения, затем лист передавала дальше. Работа была не только медленной, но и чреватой ошибками. Чтобы их исключить, одна и та же задача решалась параллельно и в другой группе, а полученные результаты сравнивались. Многие вспоминали этот труд как монотонный и изматывающий, но сознание важности – «мы делаем это для фронта» – придавало силы.

Даже в Лос-Аламосе расчеты, связанные с первыми атомными бомбами, выполнялись вручную. Математики, в основном женщины, сутками считали уравнения диффузии и цепные реакции. Единственная ошибка могла перечеркнуть недели работы. В воспоминаниях участников встречаются истории, как результатов многодневных расчетов шли насмарку из-за одной пропущенной цифры.

К середине XX века пределы ручных вычислений стали очевидны. Армии людей-компьютеров, сидевшие в залах с карандашами и тетрадями, не успевали за требованиями со стороны войны и науки. При этом вероятность ошибки росла вместе с объемом данных, а скорость оставалась ограниченной человеческими руками, что стало импульсом к созданию электронных машин. Первый ENIAC в США и первая МЭСМ в СССР задумывались именно как «замена тысячам человеческих компьютеров», и в этом не было метафоры – устройства освобождали рабочие залы от людей с их бумагой и калькуляторами. Их труд остался недооцененным, но именно он стал фундаментом науки и техники XIX–XX веков. И когда вычисления переселились в машины, они унаследовали не только методы, но и организацию, созданную невидимыми миру тружеников от египетских писцов и учеников Птолемея до гарвардских компьютеров и участников Манхэттенского проекта.

Люди-компьютеры в России и СССР

Россия и СССР имеют собственную богатую историю людей-компьютеров. В XVIII–XIX веках в Петербургской Академии наук и Пулковской обсерватории трудились группы вычислителей, которые составляли астрономические каталоги и таблицы. Пулково, открытое в 1839 году, быстро стало «астрономической столицей мира», и его слава опиралась не только на телескопы, но и на десятки людей, которые часами сводили наблюдения в аккуратные ряды чисел. В XIX веке вычислительный труд требовался и в инженерии: выпускники Института корпуса инженеров путей сообщения выполняли рутинные расчеты для строительства мостов и железных дорог.

При статистических комитетах в губерниях и в Министерстве финансов работали так называемые «числовики», которые занимались сводом и обработкой данных для бюрократической машины. В Военно-морском ведомстве и Артиллерийском комитете существовали группы, которые рассчитывали таблицы для стрельбы и навигации.

С 1920-х годов в СССР с внедрением централизованного управления экономикой потребность в вычислениях резко возросла. Госплан, Центральное статистическое управление и другие органы содержали целые отделы счетчиков, которые вручную сводили экономические показатели пятилеток. В университетах и научных институтах создавались вычислительные лаборатории: там студенты и сотрудники решали уравнения на бумаге или с помощью арифмометров. В СССР механический арифмометр «Феликс» выпускался серийно с 1929 по 1978 год, порядка 3 000 000 штук).

Во время Великой Отечественной войны труд вычислителей стал критически важным. В артиллерийских академиях и оборонных НИИ работали группы, составлявшие баллистические таблицы и решавшие задачи аэродинамики для авиации. В авиационных КБ коллективы «компьютеров» просчитывали профили крыльев и нагрузки конструкций. В этих залах царил та же атмосфера, что и в американских или британских центрах: длинные ряды столов, листы бумаги, калькуляторы, бесконечные проверки и перекрестные сверки.

В послевоенные годы, во время атомного проекта, потребность в ручных вычислениях достигли пика. В закрытых городах Арзамас-16 и Челябинск-70 трудились сотни молодых женщин-математиков. Их задачей было решать уравнения диффузии, нейтронного переноса, теплопередачи. Использовались арифмометры, логарифмические линейки и разного рода электромеханические калькуляторы. Ошибка одного человека могла перечеркнуть недели работы, и потому каждый шаг дублировался несколькими группами. В воспоминаниях участников сохранилась картина огромных залов, где люди сутками исполняли роль живых алгоритмов.

К середине XX века пределы ручного труда стали очевидны и в СССР. Именно это стало толчком к созданию электронных машин. МЭСМ Сергея Лебедева (1951) задумывалась как замена «армии вычислителей», трудившихся на оборонные и научные задачи.

Таким образом, история человеческих компьютеров в России и СССР – это развитие от астрономических бюро XVIII–XIX веков, через статистические и инженерные коллективы начала XX века, до оборонных и атомных проектов 1940–50-х годов. Их труд, рутинный и тяжелый, но организованный фабрично и коллективно, был фундаментом науки, техники и обороны, и подготовил почву для перехода вычислений в электронные машины.

Компьютинг как основа цифровой цивилизации

До 1970-х годов компьютер оставался редким, уникальным и дорогостоящим объектом. Он воспринимался прежде всего как машина, как инженерный артефакт, созданный для выполнения сложных расчетов. На этом раннем этапе комплексное явление, которое сегодня

мы называем computing, еще не существовало. Хотя в статьях создатели первых электронных компьютеров Джон Мокли и Преспер Эккерт использовали термины *electronic computing* или *digital computing*, но они понимали под этим именно саму машину как физический объект, вокруг которого формировалась соответствующая сфера деятельности.

С компьютеров третьего поколения и особенно с появлением персональных компьютеров второй половины 1970-х годов понятие *computing* стало пониматься шире – как деятельность, осуществляемая в самых разных контекстах. Но в массовом сознании того времени главным оставался сам компьютер – настольная машина, символ технологической независимости и нового образа жизни.

Следующий перелом связан с сетями. В 1990-е годы персональный компьютер стал лишь узлом в глобальной системе, а вычисления и работа с данными вышли за пределы локального устройства. *Computing* стал рассматриваться как распределенный процесс и инфраструктура, соединяющая людей и организации. В этот период появляются такие направления, как *distributed computing*, где множество машин объединяются для совместного решения задач (примером служит проект *SETI@home*, использовавший вычислительные мощности добровольцев для анализа радиосигналов из космоса), и *parallel computing*, позволяющий выполнять сложные операции одновременно на нескольких процессорах, что использовалось в суперкомпьютерах IBM серии *Blue Gene* для биоинформатики и климатического моделирования.

С началом XXI века *computing* окончательно освобождается от привязки к отдельному устройству. С распространением облачных сервисов (*cloud computing*, например, *Amazon Web Services*, запущенные в 2006 году) и мобильных технологий (*mobile computing*, смартфоны и планшеты) компьютер перестает быть центром. Он становится терминалом, а вычисления превращаются в услугу. Можно хранить данные и обрабатывать их без собственного сервера, используя глобальную инфраструктуру. *Computing* становится ресурсом – невидимым, но необходимым, как электричество, которое пронизывает все аспекты современного общества.

Сегодня *computing* существует во множестве форм и сфер применения. Среди них *Scientific computing*, применяемый для моделирования и анализа данных в науке и инженерии; *Business computing*, обеспечивающее управление бизнес-процессами и экономическую аналитику; *Cognitive computing*, имитирующее процессы человеческого мышления и принятия решений; а также *Social computing*, формирующее новые способы взаимодействия и коммуникации через платформы и социальные сети. В последние годы активно развиваются *edge computing*, когда обработка данных происходит ближе к источнику, снижая задержки и нагрузку на сеть, и *ubiquitous computing*, когда встроенные процессоры и сенсоры пронизывают повседневную жизнь – от «умных» домов до носимых устройств.

Особое значение имеет вопрос взаимодействия человека и машины. В ранние эпохи вычисления осуществлялись в пакетном режиме (*batch computing*), когда задачи подавались на вход, а результаты выдавались спустя определенное время. Персональные компьютеры открыли эру интерактивных вычислений (*interactive computing*), позволяя пользователю управлять процессом в реальном времени. Реальные системы управления, от авиации до медицины, потребовали *real-time computing*, где скорость реакции критична. Сегодня на повестке дня автономные вычисления (*autonomous computing*): автомобили с автопилотом, интеллектуальные ассистенты и роботы принимают решения без постоянного контроля человека. *Computing* постепенно становится партнером человека в деятельности, а не просто инструментом.

Таким образом, за несколько десятилетий произошел переход от «компьютера» как отдельного устройства к *computing* как универсальному явлению общечеловеческой значимости. В нем собственно компьютер утратил центральную роль; главным стало не устройство, а процесс – способность преобразовывать данные в информацию, знания и решения в различных контекстах.

Поэтому важно различать computing и информационные технологии. Эти понятия принадлежат к разным порядкам явлений. Computing – фундаментальное, почти философское понятие, связанное со способностью общества обрабатывать информацию, моделировать реальность и создавать новые знания. Оно охватывает все формы вычислений, от первых мейн-фреймов и распределенных систем до облаков, квантовых машин и pervasive computing (все-проникающий, встраиваемый в повседневные устройства). Computing существует как явление, формирующее культуру, инфраструктуру и способы взаимодействия человека с окружающим миром.

Информационные технологии (ИТ), напротив, относятся к конкретным прикладным решениям. Это организации, сервисы, программные продукты и сети, которые используют вычислительные возможности для поддержки деятельности компаний, учреждений и пользователей. ИТ – это индустрия, инфраструктура, операционные процессы; оно существует внутри явления computing, как конкретная реализация его возможностей. Можно провести аналогию: если computing – это электрификация как природно-технический феномен, то ИТ – это лампы, электросети и бытовые приборы, которые позволяют людям пользоваться электричеством. Computing задает потенциал и направление развития, ИТ – прикладное воплощение, используемое для решения конкретных задач.

Фокус на ИТ и его видимых проявлениях иногда искажает восприятие эволюции вычислительной сферы. Люди чаще замечают новые программы, приложения и сервисы, воспринимая их как «движущую силу прогресса», в то время как само явление computing развивается на более фундаментальном уровне – через новые архитектуры, методы обработки информации, распределенные системы, внедрение pervasive (всепроникающий) и cognitive (способный к мышлению) computing. ИТ отражает лишь прикладную, наиболее заметную часть этого процесса, и без осознания разницы между уровнем прикладного инструмента и фундаментальной среды легко потерять понимание истинной эволюции от компьютера как отдельного устройства к computing как универсальной инфраструктуре и культурной практике.

Таким образом, computing сегодня – это универсальный язык цифровой эпохи. Он определяет то, как мы думаем, общаемся и создаем технологические и культурные системы. ИТ остается важным прикладным воплощением этого языка, но явление computing гораздо шире: оно задает фундаментальные возможности и направления, в которых отдельные компьютеры и информационные технологии – лишь инструменты. Понимание этой разницы важно для корректной оценки современной цифровой эволюции: computing формирует фундамент, на котором строятся видимые информационные технологии.

Компьютинг и пирамида DIKW: эволюция от данных к здравому смыслу и принятию решений

Для глубокого понимания эволюции компьютеринга стоит обратиться к концепции пирамиды DIKW, описывающей четыре уровня обработки данных: Data (Данные), Information (Информация), Knowledge (Знания) и Wisdom (Здравый смысл и принятие решений). Эта модель показывающая, как сырые факты превращаются в информацию, знания и в способность принимать обоснованные решения сложилась в конце восьмидесятых годов XX века. В оригинале он состоит из четырех компонентов:

- Данные, получаемые из внешнего мира в результате человеческой деятельности или от различных датчиков и других устройств.
- Информация, создаваемая посредством анализа отношений и взаимосвязей между фрагментами данных в результате ответа на вопросы: Кто? Что? Где? Сколько? Когда? Почему?
- Знания получаются в результате синтеза полученной информации и человеческого разума, а в последние годы и ИИ.

- Глубокое понимание или здравый смысл служит основой для принятия решений.

Ретроспективно глядя, идеи, лежащие в основе DIKW, можно приложить к эволюции компьютеринга.

- 1940–1950-е годы – зарождение понятий «данные» и «информация». Клод Шеннон формализовал теорию информации как меру неопределенности. В это время Джон Моукли и Преспер Эккерт создают машины, работающие исключительно с данными.

- 1960–1970-е годы – переход к уровню информации. Компьютеры третьего поколения и ПК позволили структурировать данные, создавать отчеты, базы данных и визуализировать результаты. Computing превращается из работы с данными в деятельность, позволяющую извлекать полезную информацию.

- 1990-е годы – переход на уровень знаний. Распространение сетей и появление распределенных и параллельных систем (distributed computing, parallel computing) дают возможность объединять ресурсы множества машин, выявлять закономерности и строить модели. Computing на этом уровне превращает информацию в знания, используемые для прогнозирования и поддержки решений.

- 2000-е – 2010-е годы – уровень здравого смысла и принятия решений. Зарождение ИИ.

Компьютинг в повседневной жизни

Современный компьютеринг наличествует не только в лабораториях, исследовательских центрах или офисах – он уже прочно вошел в нашу повседневную жизнь. Смартфон, по сути, является миниатюрным компьютером, который всегда с нами. Он собирает данные о наших действиях, анализирует их, предоставляет информацию и помогает принимать решения. В определенном смысле смартфон уже частично выступает в роли интеллектуального партнера, способного предлагать рекомендации и оптимальные варианты действий.

Компьютинг стал обязательным атрибутом жизни, а смартфон – его осязаемое и доступное воплощение. Каждый день мы взаимодействуем с данными, информацией, знаниями и элементами цифровой мудрости, часто почти не замечая этого. Наши устройства аккумулируют и интерпретируют информацию, помогают нам принимать решения быстрее и точнее, а порой даже подсказывают, что делать в ситуациях, когда у нас нет времени или ресурсов на полноценный анализ.

В итоге можно сказать, что современный компьютеринг стал частью человеческого опыта: мы не только пользуемся технологиями, но и встраиваем их в собственное мышление, делая процесс взаимодействия с данными и знаниями естественной и почти непрерывной частью повседневной жизни.

Глава 1. Аппаратная и программная основы компьютеринга

Компьютер не имеет аналогов в истории техники. Он сочетает в себе два начала: аппаратную платформу и программное обеспечение (ПО). Железо обеспечивает счет и работу с данными, а ПО превращает эти ресурсы в инструменты для анализа, управления и даже творчества. Без программного обеспечения самый совершенный процессор остается лишь набором транзисторов и схем, а без железа самые изощренные алгоритмы не смогут воплотиться в действия.

На первый взгляд все кажется простым и очевидным. Но если отвлечься от привычного взгляда, перед нами открывается парадокс: компьютер соединяет материальное и нематериальное, вещь и идею, тело и программу. Железо само по себе мертво, софт сам по себе бесплоден. Только вместе они образуют целостность – особую машину, которой в истории техники не было. У книги есть текст, но он не управляет бумагой; у станка может быть программа, по ней он выполняет заданные операции. В культуре и природе можно найти аналогии – мозг и сознание, генотип и фенотип, вещество и энергия, – но в артефактах два начала – техника и мысль никогда еще не соединялись столь органично.

Эта взаимодополняемость формирует особый союз: аппаратное обеспечивает «мышцы», программное – «разум», и вместе они создают универсальный инструмент, превращающий данные в знания, вычисления – в понимание, а инструкции – в действия. Двойственность делает компьютер не просто машиной, а феноменом культуры.

В этой главе мы проследим эволюцию обоих начал, чтобы понять, как их развитие шло рука об руку и почему современный компьютер стал тем, чем мы его знаем. А вся книга в целом будет сосредоточена прежде всего на аппаратной основе компьютеринга – на его «теле», без которого никакая программа не может обрести жизнь.

Об электронике

У многих из нас еще со школьной скамьи сохранились обрывочные воспоминания об истории электричества. Греческое слово «электрон» – «янтарь», который при трении начинал притягивать легкие предметы. Таинственная электрофорная машина со стеклянными вращающимися дисками, стоявшая в школьном кабинете. Опыт с воздушным змеем, устремленным к грозovým облакам. Лейденская банка, первые гальванические элементы. И, конечно, целая плеяда великих физиков – Алессандро Вольты, Луиджи Гальвани, Андре-Мари Ампер, Джеймс Джоуль, Георг Ом, Майкл Фарадей и многие другие. Их имена навсегда закрепились не только в истории науки, но и в языке самой физики: они стали названиями единиц измерения и законов, которые мы по привычке используем в повседневной жизни.

На этом фоне история электроники более молодой отрасли физики и инженерии, изучающей поведение электронов и создающей устройства на их основе, известна куда хуже. И это при том, что в современном мире мы сталкиваемся с электронными приборами, пожалуй, даже чаще, чем с чисто электрическими. Поэтому, чтобы говорить об истории электронных компьютеров, нужно начать с истории самой электроники. Ее развитие можно условно разделить на несколько этапов:

- первые шаги, когда электроника еще только отделялась от «большой электротехники»;
- эпоха вакуумных ламп;
- изобретение полупроводниковых приборов;
- появление интегральных схем и микропроцессоров;
- развитие магнитной и полупроводниковой памяти.

Именно эта цепочка открытий и инженерных решений есть путь, который привел к современному «железу» компьютера.

Эпоха ламп

От первых шагов до вакуумных приборов – диода и триода

В электронике, как и в любой отрасли науки и технологий, к открытиям и изобретениям причастно гораздо больше людей, чем можно включить в исторический экскурс. Родоначальниками работ, приведших к появлению электроники, были немецкие физики-экспериментаторы Генрих Гейслер и Юлиус Плюккер. В 1857 году они совместно изобрели первый электронный прибор – «трубку Гейслера». Пробраз современных газоразрядных приборов представлял собой частично вакуумированную стеклянную трубку с двумя металлическими электродами – катодом и анодом. При подаче высокого напряжения поток ионов вызывал свечение, названное флуоресценцией. Дальнейшие исследования флуоресценции привели к появлению неоновому освещению в начале XX века.

Следующий шаг сделал Иоганн Гитторф, изучавший физику вакуумной трубки. Он обнаружил обратный поток заряженных частиц от катода к аноду, который назвал катодным лучом, и описал его свойства. Подлинным основоположником электроники как науки стал Джозеф Томсон, лауреат Нобелевской премии по физике 1906 года. Он показал, что анодные лучи, проходящие через разреженный газ, состоят из ионизированных атомов и молекул, а катодные лучи через вакуум – из отрицательно заряженных элементарных частиц, впоследствии названных электронами. Честь открытия электрона и введение этого термина в науку принадлежит ирландскому физика Джорджу Стони (1897 год).

Совокупность этих открытий проложила путь к созданию разнообразных электронных приборов: вакуумных ламп, рентгеновских трубок, люминесцентных ламп. Карл Браун, лауреат Нобелевской премии 1909 года, изобрел кинескоп – катодно-лучевую трубку, которая некоторое время использовалась в компьютерах как устройство памяти, а многие годы – как экран для телевизоров.

Появлению электронных ламп, ставших аппаратной основой радио, телевидения и первых компьютеров, предшествовало открытие термоэлектронной эмиссии в 1873 году английским физиком Фредериком Гатри. Он создал прообраз электронной лампы – термоэмиссионный диод, который его ученик Джон Флеминг в 1904 году превратил в полноценный электровакуумный прибор.

Венцом этих разработок стал ламповый триод Джей Ли де Фореста, в котором появился третий управляющий электрод – сетка. Эта конструкция позволяла варьировать напряжение на сетке и управлять током анод-катод, усиливая сигнал. Триод стал основой огромного числа приборов и породил новую индустрию.

Де Форест был не только талантливым ученым и инженером, но и успешным предпринимателем. В 1909 году в Пало-Альто при его участии была основана первая лаборатория Федеральной телеграфной компании. Его деятельность и сподвижников превратила побережье от Сан-Хосе до Пало-Альто из бескрайних капустных полей в крупнейший научно-производственный кластер – будущую Силиконовую долину.

На этих землях вскоре появились новые легендарные проекты. В нескольких шагах от лаборатории де Фореста, на улице Эль Каминьо, в 1938 году выпускники Стэнфорда Дэвид Паккард и Билл Хьюлетт собрали свой первый осциллограф. Рядом возник клуб «Домашняя пивоварня», где позже собирались первые ПК, включая Apple I. По мере развития автомагистралей и кампусов здесь формировался современный ландшафт технологического производства.

Если рассматривать Калифорнию как отдельную экономику, ее валовой внутренний продукт сопоставим с ведущими странами мира, уступая лишь США, Китаю и Японии. Именно здесь, на некогда сельскохозяйственных угодьях, выросли научные и производственные кластеры, формирующие современный облик электроники, программирования и высоких технологий.

Первый вариант триода де Фореста, лампа Audion, использовалась только для усиления звуковых сигналов. Немецкие инженеры усовершенствовали конструкцию, сделав катод цилиндрическим с нагреваемой нитью, сетку – перфорированной, а анод – нитью. Компактная компоновка позволила создавать двойные триоды и лампы с большим числом электродов, варьируя размеры и мощность.

Ламповая электроника нашла применение в радио, телевидении, радиолокации и вычислительных машинах. В первых компьютерах существенным было облечение надежности работы, а не усиление, использование недогруженных ламп позволяло решать эту задачу.

Лампы как основа компьютеров первого поколения

Ламповая электроника позволила автоматизировать выполнение двоичных логических и арифметических операций – двух столпов всех существующих компьютеров. Отмечено всего три попытки создания троичных компьютеров. В середине XIX века ирландец Джон Фоулер построил деревянный макет, в 1960-х годах Н.П. Брусенцов на феррит-транзисторных троичных компонентах создал уникальную «Сетунь», а примерно в то же время в Канаде разрабатывался компьютер QTC-1, о котором почти ничего не известно. В последнее время появились сообщения о том, что IBM и Samsung ведут исследования в этом направлении.

Одной из причин доминирования двоичной системы является простота ее реализации. Атомами любой вычислительной инфраструктуры – от простейших компьютеров до суперкомпьютеров – служат двоичные триггеры, собранные в многоразрядные регистры для хранения данных и выполнения операций. Триггер – это компонент, способный занимать одно устойчивых состояний. Он может быть двоичным, троичным и так далее. Для наиболее используемых двоичных триггеров в английском языке используется образный термин flip-flop, по аналогии с пляжными шлепками.

Простейший двоичный триггер, будь то ламповый или полупроводниковый, собирается из двух триодов. Управляющий электрод одного триода (в лампе это сетка) подключается к аноду другого, и пара триодов сохраняет одно из двух устойчивых состояний. Если на сетку подать управляющий импульс, триггер переходит в другое состояние: одно состояние отождествляется с 0, другое – с 1. Триггер легко упаковывается в лампу типа двойной триод, которая хранит один разряд двоичного числа. Из двоичных триггеров формируются регистры для записи и хранения данных, с ними выполняются логические, арифметические и коммутационные операции. Из регистров собирается арифметико-логическое устройство (АЛУ) – главный компонент процессора.

Простота триггера оказалась гениальной: это устройство стало одним из важнейших кирпичей истории компьютеров. Обычно изобретение двоичного триггера приписывают британским физикам Уильяму Эклсу и Фрэнку Джордану. В 1918 году они подали патент на ионное реле (Improvements in Ionic Relays), подтвержденный в 1929 году. Устройство позволяло усиливать ток для телефонии и телеграфии, заменяя механические реле – ненадежные и медленные. Кстати, Эклс предложил термин diode – «трубка с двумя электродами» (от латинского di – «дважды» и hodos – «путь»), которое стало первым электронным устройством.

Меньше известно, что почти одновременно к идее электронного реле пришел российский инженер Михаил Александрович Бонч-Бруевич. В 1918 году он доложил о своем изобретении на заседании Российского общества радиоинженеров в сообщении «Комбинированные харак-

теристики катодных реле». Он занимался этой темой с 1914 года, используя триггер для усиления радиосигнала. Более широко триггеры оказались востребованы только через двадцать лет, когда они стали основой компьютеров первого поколения (1945–1959).

Новая жизнь триггеров началась в 1940-х, точнее это произошло в 1942 году при создании экспериментального компьютера Атанасова-Берри (ABC), далее – в 1943 году в британском электронно-механическом компьютере Colossus в Блетчли-Парке, и, наконец, в 1945 году при создании полноценного электронного компьютера ENIAC. Теоретическое обоснование для дальнейшего развития двоичных логических схем появилось в работе Клода Шеннона «Математическая теория коммуникаций» (1948).

Ламповая электроника стала базой для компьютеров первого поколения (1940–1956). В них лампы использовались в центральном процессоре, а электроннолучевые и акустические трубки – в ранних конструкциях памяти. Использование недогруженных ламп обеспечивало высокую надежность и стабильность работы, что было критично для выполнения программ. Таким образом, ламповая электроника не только позволила автоматизировать вычисления, но и создала фундамент для всей последующей истории вычислительной техники.

История полупроводников и зарождение транзисторной эры

Ранние открытия: от Вольта до Брауна

Трудно поверить, но термин «полупроводник» старше «вакуумной лампы» на целых 60 лет! Первым его использовал Алессандро Вольта в докладе Лондонскому Королевскому обществу в 1782 году. Он наблюдал, как разные материалы ведут себя при замыкании ими контактов заряженного конденсатора: металлы вызывают мгновенный разряд, дерево и другие изоляционные материалы – нет. Материалы, при замыкании которых конденсатор разряжался не мгновенно, Вольта назвал полупроводниками.

В течение следующих 50 лет исследования ограничивались регистрацией удельного сопротивления материалов, находящихся между металлами и изоляторами. К современному пониманию полупроводников эти наблюдения не относятся.

Прогресс начался в 30-е годы XIX века, когда Майкл Фарадей обнаружил, что сульфид серебра обладает отрицательным температурным коэффициентом. В 1839 году Эдмонд Беккерель создал первый в мире фотоэлектрический элемент, прибор, который преобразует энергию фотонов в электрическую. В 1873 году Уиллоуби Смит впервые описал эффект фотопроводимости на кристаллическом селене – прообраз современных фоторезисторов.

В 1874 году Фердинанд Браун заметил, что сопротивление в кристаллах галенита меняется в зависимости от направления протекания тока. Так возник точечный электрический выпрямитель, собранный из металлической иглы и галенита. В начале XX века Гринлиф Пикард применил полупроводники для обнаружения радиосигналов, что дало толчок к созданию детекторных радиоприемников.

Лосев и кристадин

В начале 1920-х годов Олег Владимирович Лосев заметил эффект отрицательного дифференциального сопротивления на кристаллах оксида цинка. Одновременно он обнаружил свечение, ставшее много позже основой светодиодов. Лосев пытался использовать свое изобретение для усиления сигналов в радиоприемниках и передатчиках.

История Лосева – пример драматичной судьбы ученого, родившегося в неподходящее время и в неподходящем месте. Хотя ему и присвоили звание кандидата наук, большую часть жизни он провел на лаборантских должностях. Он остался в городе во время блокады в Ленин-

граде и умер от голода. На Западе его признали намного раньше – поклонником стал Хьюго Гернсбэк, который построил радиоприемник по спецификациям Лосева и написал статью в Radio News в 1924 году.

Механизм кристадинного эффекта до сих пор не полностью изучен. Многие специалисты связывают его с туннельным эффектом, но его природа отличается от туннельного диода.

Другие пионеры

Юлий Лилиенфельд, родившийся во Львове, предложил в 1926 году трехэлектродную полупроводниковую структуру, прообраз полевых транзисторов, и еще разработал электролитический конденсатор, который используется до сих пор. В 1935 году Оскар Хайль получил британский патент на свой вариант полевого транзистора. В 20–30-е годы другие исследователи отмечали проявления поверхностного полупроводникового эффекта, но это не приводило к практическим результатам. Крупный прорыв сделал британский физик-теоретик Алан Уилсон, опубликовавший в 1939 году книгу «Полупроводники и металлы».

Bell Labs и рождение транзистора

Планомерные исследования по созданию полупроводникового триода начались в 1936 году под руководством Мервина Келли в одном из крупнейших исследовательских центров своего времени Bell Labs. Он создал отдел по физике твердого тела и привлек к работе в нем Уильяма Шокли, Рассела Ола и Джека Скаффа. Ол случайно обнаружил первый полупроводниковый P-N переход, разрезав кремниевый слиток.

После Второй мировой войны работа продолжилась. Шокли и Стэнли Морган возглавляли отдел физики, привлекли Уолтера Браттейна, Джона Бардина и других. Они сосредоточились на кремнии и германии. В лаборатории Ола они увидели его «дезистеры» – устройства, напоминавшие кристадин Лосева.

Формально руководителем был Шокли, который разрабатывал проект полупроводникового усилителя на полевом эффекте. Его модель оказалась ошибочной – электроны локализовывались на поверхности полупроводника. Бардин и Браттейн исправили проблему и в декабре 1947 года создали полупроводниковый диод и усилитель, который лег в основу точечного транзистора. Шокли затем разработал инжекционный транзистор на P-N переходах – более практичный и простой в изготовлении.

26 февраля 1948 года Bell Labs подала две патентные заявки – одну Бардина и Браттейна, другую Шокли. Термин «transistor» возник как сочетание transresistance и созвучия с варисторами и термисторами. 30 июня 1948 года состоялась пресс-конференция, на которой объявили о транзисторе. Первый коммерческий точечный транзистор использовала Western Electric в 1951 году, но успех был ограничен.

Конфликты, приведшие к созданию будущего

Bell Labs раздирали острые личные и профессиональные конфликты. Уильям Шокли, совмещавший научный дар с высокой амбициозностью и неоднозначной социальной позицией, часто сталкивался с Джоном Бардином и Уолтером Браттейном. Эти разногласия носили как научный, так и личный характер: Шокли стремился к полному контролю над исследованиями, навязывал свои идеи и методы, в то время как Бардин и Браттейн обладали самостоятельным мышлением и практическими способностями.

Конфликт достиг апогея после создания точечного транзистора. Несмотря на коллективный успех, Шокли не мог смириться с тем, что значительная часть прорыва произошла без его непосредственного участия. В результате Бардин и Браттейн покинули Bell Labs, посвятив

себя другим направлениям – в частности, изучению сверхпроводимости. Бардин в 1957 году опубликовал фундаментальную статью «Теория сверхпроводимости», которая стала ключевой в физике твердого тела, и позже получил вторую Нобелевскую премию.

Шокли же остался одиозной фигурой. Он основал Shockley Semiconductor, надеясь продолжить исследования и коммерциализировать полупроводниковые технологии. Однако его авторитарный стиль управления и неспособность работать в команде привели к масштабной утечке талантов: лучшие инженеры и исследователи, не согласные с руководством, покинули компанию и образовали группу, получившую название «Вероломная восьмерка» (The Traitorous Eight). В ее состав вошли:

- Гордон Мур – позднее сооснователь Intel, разработчик закона Мура, один из ведущих теоретиков микроэлектроники.
- Роберт Нойс – сооснователь Fairchild Semiconductor и Intel, изобретатель интегральной схемы.
- Жан Хофф – инженер и руководитель проектных групп, внес значительный вклад в разработку производственных процессов.
- Роберт Крой – специалист по химии полупроводников и производству кремниевых кристаллов.
- Виктор Грин – эксперт по технике нанесения слоев и изготовлению P-N переходов.
- Джек Бил – занимался проектированием схем и тестированием прототипов транзисторов.
- Фред Вайс – инженер-электронщик, участвовал в создании первых коммерческих транзисторов Fairchild.
- Джон Ред – занимался исследованиями по увеличению производительности кремниевых приборов.

Эти восемь ученых создали Fairchild Semiconductor, компанию, ставшую символом инновационной культуры Силиконовой долины. На базе Fairchild позднее возникли Intel, AMD, National Semiconductor и другие технологические гиганты.

Влияние на Силиконовую долину

Уход «Вероломной восьмерки» из Shockley Semiconductor стал катализатором формирования уникальной инновационной экосистемы:

- Опора на социальные сети и коллективный опыт: компании создавались людьми, уже имеющими профессиональные связи, что усиливало эффект передачи знаний и ноу-хау.
- Движение по собственному пути: новые предприятия использовали инновационные решения, независимые от крупных корпораций, что позволяло быстро внедрять передовые технологии.
- Фокус на коммерческом успехе: в отличие от корпоративной модели Bell Labs, где приоритет отдавался фундаментальным исследованиям, Fairchild и ее последователи ориентировались на быстрое внедрение инноваций на рынок.
- Таким образом, личные конфликты и противоречия внутри Bell Labs, несмотря на драматические последствия для отдельных людей, стали неожиданным толчком к развитию современной высокотехнологичной индустрии, а также заложили основу принципов, характерных для будущей Силиконовой долины.

Гордон Тил и Texas Instruments

Гордон Тил, химик-физик из Техаса, работавший в Bell Labs с 1930 года, предложил выращивать монокристаллы с P-N переходом, но получил отказ. В 1952 году Texas Instruments приобрела лицензию на германиевые транзисторы и пригласила Тила на работу. Он создал отдел по образцу Bell Labs и совместно с Морганом Спарксом усовершенствовал технологию, вырастив p-p-н германиевый транзистор прямо из расплава.

В апреле 1954 года команда Тила создала первый коммерческий кремниевый транзистор, а 10 мая Тил представил его на конференции IRE. В течение пары лет транзисторы стали компонентной базой второго поколения компьютеров, сочетаясь с диодами, резисторами, конденсаторами и магнитными сердечниками.

Память

На заре компьютерной эры проблема адекватной памяти была, пожалуй, острее, чем сложности, возникающие при создании новых процессоров. Процессоры строились из стандартных радиокомпонентов, имевшихся в большом объеме на рынке – ламп, резисторов и конденсаторов. Для памяти же ничего полностью подходящего из готового не существовало, приходилось адаптировать под нужды вычислительной техники то, что хотя как-то подходило.

Для компьютера EDVAC Преспер Эккерт использовал забытый ныне прибор – ртутную акустическую линию задержки. Такая трубка, заполненная парами ртути, могла хранить данные благодаря пьезокристаллическим преобразователям на концах. Они сохраняли акустический импульс, посланный в трубку, который продолжал циркулировать между преобразователями. Первоначально такие трубки создавались для разделения подвижных и неподвижных целей в радиолокации. Они были крайне дорогими, громоздкими и сложными в эксплуатации. Эккерт применил тот же принцип динамического хранения данных, но на магнитоотрицательных материалах, что оказалось гораздо практичнее ртути. Однако емкость таких линий задержки не превышала сотен битов.

Альтернативой линиям задержки стала электронно-лучевая трубка (ЭЛТ) Уильямса—Килберна, названная в честь британцев Фредди Уильямса и Тома Килберна. Она стала первым запоминающим устройством с произвольным доступом (RAM). В этой трубке электронный луч буквально записывал данные на экран в виде заряженных и незаряженных точек. Считывание обеспечивал металлический диск перед экраном, отражавший луч. Возвращаемый сигнал зависел от того, была ли точка заряжена или нет. На этом принципе строились ЗУ емкостью до 2048 битов – существенно быстрее, чем акустические линии задержки.

Существовала и более амбициозная конструкция. Команда RCA под руководством Владимира Зворыкина создала лампу Selectron, которая сохраняла статический заряд, подобно ЭЛТ. Однако производство оказалось чрезвычайно сложным, и коммерческая версия так и не была создана. RCA планировала выпустить 200 трубок емкостью 4096 битов к концу 1946 года, но к середине 1948 года эти планы не были реализованы.

Таким образом, выходом из положения стала ферритовая память, которая со временем стала основой для большинства первых компьютеров следующего поколения.

Магнитная память на ферритовых сердечниках

Память на ферритовых сердечниках продержалась около четверти века, с 1955 по конец 1970-х она монополюбно господствовала во всех компьютеринге. Чаще всего в форме колечек миллиметрового размера, реже – на биксах, сердечниках с двумя взаимно перпендикулярными

отверстиями. Сердечки собирались в квадратные матрицы, а матрицы – в кубы, достигая удельной емкости до 32 килобит на литр физического объема.

Внешне такая память выглядела крайне скромно, если не сказать примитивно. В Компьютерном музее в Маунтин-Вью на фоне сложных систем смотрятся кустарными поделками кубические конструкции из деревянных реек с тысячами сердечников внутри. Кропотливую работу по соединению проводами крошечных сердечников могли выполнять только тонкие девичьи пальчики, поэтому процесс изготовления называли «вышивкой памяти». В одной статье этот период даже назвали «Временем, когда компьютерную память вышивали женщины» (That Time When Computer Memory Was Handwoven by Women). Массовое производство ферритовой памяти перенесли в страны Юго-Восточной Азии, где миниатюрные местные девушки плели память в цехах, которые напоминали ковроткацкие мастерские.

Физической основой была способность сердечников сохранять направление намагниченности (эффектом гистерезиса). Сердечник являлся простейшим магнитным флип-флопом, хранящим 1 бит. Ферритовая память обладала двумя важными преимуществами: она сохраняла данные при отключении питания (не волатильная, non-volatile), а также была устойчива к ионизирующему излучению, что делало ее востребованной в военной технике вплоть до 1990-х годов.

Идея использования магнитных материалов для памяти возникла еще в 1945 году у Преспера Эккерта при создании ENIAC, а в 1946 году – у Джорджа Девола, известного созданием промышленных роботов UNIMATE. Однако тогда практическая реализация оказалась крайне сложной, последующая история ферритовой памяти – это не столько история научных открытий, сколько череда инженерных решений и патентных споров.

Пионеры ферритовой памяти

Одним из первых патентов на этот тип памяти получил Фредерика Вехэ, финн по происхождению, муниципальный служащий в Лос-Анджелесе. Он занимался экспериментами с магнитными материалами дома и получил несколько патентов. Его карьера трагически оборвалась: вместе с женой он погиб в автомобильной аварии.

Другим ключевым изобретателем был Эн Ванг. Он эмигрировал в США из Китая до прихода Мао, получил Ph.D. по физике в Гарварде в 1948 году, работал с Говардом Эйкенем над проектом Mark IV и тогда в очередной раз изобрел память на магнитных сердечниках. Однако он не получил финансирования и с 1951 года продолжил работу в созданной им компании Wang Laboratories. В 1955 году после патентования памяти Ванг продал права IBM за 500 000 долларов. Позже компания Wang выпускала мини-компьютеры, включая Wang VS, совместимый с IBM System/360, которые использовались в СССР, например, при разработке «Искры-226».

Джей Форрестер успешно сочетал академическую деятельность с практической инженерной работой. Он разрабатывал свою версию памяти для исторического компьютера Whirlwind, в него куб 32×32×16 бит была установлена летом 1953 года. Форрестер отмечал: «Изобретение Ванга не оказало влияния на мою разработку. Память Ванга была более дорогой и сложной». Он также вспоминал в 1975 году: «Нам потребовались годы, чтобы убедить индустрию в том, что память на магнитных сердечниках решает проблему недостающего звена в компьютерных технологиях. Затем мы семь лет отстаивали наш приоритет в патентных судах».

Магнитные сердечники применялись не только в оперативной памяти (ОЗУ), но и в постоянной памяти (ПЗУ). Программы и данные записывались методом физической прошивки проводами, и с тех пор сохранился термин «прошивка ПЗУ», хотя в современных полупроводниковых ПЗУ процедура записи полностью изменилась.

Полупроводниковая память

Для начала уточним терминологию, сложившуюся более полувека назад. Память с произвольным доступом (Random Access Memory, RAM) – это оперативная память компьютера, к которой имеет прямой доступ центральный процессор (CPU). Главное отличие RAM на полупроводниках от ферритовой памяти заключается в энергозависимости (volatile): данные сохраняются только при подаче питания и теряются при его отключении.

Исторически выделились два типа RAM, это деление сохраняется до сих пор:

- Статическая RAM (SRAM) – использует триггеры на транзисторах (один триггер на один бит). Триггер устойчив и сохраняет состояние при наличии питания. SRAM называется «статической», потому что для удержания данных дополнительных действий не требуется.

- Динамическая RAM (DRAM) – хранит бит данных в конденсаторе. Изоляторы не идеальны, поэтому конденсаторы постепенно разряжаются; для поддержания данных требуется периодическая подзарядка. По этой причине RAM такого типа называется «динамической».

SRAM работает быстрее и потребляет меньше энергии, но сложнее и дороже. DRAM медленнее, требует подзарядки, но дешевле и позволяет создавать большие массивы памяти. В компьютере используется компромисс: SRAM применяют в кэше, где размер ограничен стоимостью, а основная память строится на DRAM и измеряется гигабайтами.

Первые попытки создания полупроводниковой памяти

Первые образцы электронной памяти на дискретных полупроводниках начались еще в 1950-х, задолго до появления интегральных схем. Они создавались для сверхоперативной регистровой памяти, ускоряющей обмен данными. Однако на технологическом уровне того времени такие решения были слишком дорогими и не могли стать массовыми.

Более чем через десять лет компания Signetics, основанная выходцами из Fairchild, смогла создать микросхему типа SRAM для работы в качестве сверхоперативной памяти. Эту память называли scratchpad memory – «блокнот для черновых записей» между процессором и основной магнитной памятью. По сути, это был аналог современного кэша L1.

В 1963 году Роберт Норман изобрел более совершенную биполярную SRAM, а в 1964 году Джон Шмидт создал 64-битную MOS-р-канальную SRAM. В апреле 1969 года Intel выпустила Intel 3101, первую промышленно произведенную SRAM. С тех пор SRAM постоянно совершенствовалась и к 1995 году стала стандартом для кэширования процессоров в серверах.

DRAM: идея и реализация

В отличие от SRAM, история DRAM насыщена событиями и повлияла на все устройства: от серверов до смартфонов. Ей человечество обязано сотруднику IBM Роберту Деннард (1932).

Деннард родился в Техасе, получил бакалавра и магистра в Далласе (1954 и 1956), а Ph.D. в 1958 году в Технологическом институте Карнеги в Питтсбурге. В IBM он занимался исследованиями памяти. Идея создания DRAM пришла к нему как инсайт, напоминающий притчу о Ньюtone и яблоке: после долгой встречи коллег по улучшению ферритовой памяти он заметил очевидные недостатки – низкую скорость и трудоемкость ручной сборки.

При этом его проект MOS-SRAM был перспективным, но требовал шесть транзисторов на один бит. Тогда Деннард пришла мысль: а нельзя ли обойтись одним транзистором, используя заряд конденсатора для хранения 0 или 1. Идея показалась слишком смелой, и некоторое время IBM не воспринимала ее всерьез.

В 1967 году Деннард удалось довести проект до патентной заявки на одностранзисторную DRAM, выданной в 1968 году. В 1970 году Intel создала успешный 1-килобитный DRAM-чип, а к середине 1970-х появились 4-килобитные чипы с одностранзисторной ячейкой. Рост емкости шел по закону Мура.

Деннард сам признавал: «Я знал, что это будет большая вещь, но не знал, что она будет иметь такое широкое влияние, какое имеет сегодня».

DRAM доминирует и используется в виде Double Data Rate Synchronous DRAM (DDR SDRAM) – синхронной динамической памяти с высокой скоростью передачи данных.

Интегральные микросхемы

Создание интегральных микросхем (Integrated Circuit, IC) открыло путь к компьютерам третьего поколения. Часто термин чип используют как синоним IC в корпусе с контактами, но точнее называть чипом сам полупроводниковый кристалл с электронными компонентами, а корпус – лишь его конструктивной оболочкой.

Историю IC часто упрощают, сводя ее к двум именам – Джека Килби и Роберта Нойса, которых называют «отцами интегральной схемы». На самом деле в развитии технологии участвовали десятки инженеров и ученых, а заслуги Килби и Нойса до сих пор вызывают споры. Присуждение Нобелевской премии Килби также считается некоторыми исследователями спорным, ведь речь шла не об одиночном изобретении, а о становлении целой отрасли.

Идея объединить несколько транзисторов на одной подложке возникла еще в 1949 году у немецкого физика Вернера Якоби, который собрал пять транзисторов в усилитель – по сути, первую интегральную схему. Однако его работа осталась экспериментом.

В 1952 году английский инженер Джеффри Даммер, позже названный «пророком IC», писал: «С появлением транзистора станет возможным создавать схемы в виде единого модуля без соединительных проводов». Он даже сделал модель твердотельного триггера как монолитный блок, но без необходимых технологий практического применения добиться не удалось.

- В 1950-е годы появлялись прототипы интегральных схем:
- Харвик Джонсон и Торкель Уоллмарк из RCA подали заявку на однокристалльный генератор.
- Артур Д'Азаро и Иэн Росс из Bell Labs создали счетчик для телефонии.
- Японец Ясуо Таруи предложил собственную концепцию интеграции.

Даже неудачный «четырёхслойный диод» Шокли можно рассматривать как попытку объединить функции нескольких компонентов в одном кристалле. Все эти разработки оставались узкоспециализированными.

Прорыв обеспечил Джек Килб из Texas Instruments. В 1958 году он доказал возможность размещения транзисторов, резисторов и конденсаторов на одном кристалле и соединения их между собой. В 1959 году TI представила первую коммерческую интегральную схему – двоичный триггер.

Однако соединения Килби выполнялись вручную золотыми «летающими» проводками, что делало процесс дорогим и ненадежным (одна схема стоила около 500 долларов – дороже лампового аналога). Формально Килби создал интегрированную схему, но не микросхему: его элементы были размером 0,1–1 мм («мезоуровень»). Настоящие IC стали появляться только с переходом к микроуровню (1–100 мкм), а затем к наноуровню.

Килби был изобретателем, а не исследователем-физиком, и не решил ключевых задач, тем не менее его эксперимент продемонстрировал принцип универсальной интеграции. Главный практический прорыв сделали инженеры Fairchild Semiconductor.

- Жан Эрни предложил покрывать транзисторы диоксидом кремния, что привело к планарной технологии.
- Роберт Нойс применил ее для создания первой кремниевой монолитной схемы.
- Курт Леговец разработал изоляцию p—n-переходами, сделавшую массовое производство возможным.

Fairchild в 1960-е годы стала центром интегральной революции. Брюс Дил, Эндрю Гроув, Эд Сноу и их коллеги провели фундаментальные исследования взаимодействия кремния и диоксида кремния, которые легли в основу классического учебника Гроува *Physics and Technology of Semiconductor Devices*.

Патентные споры между TI и Fairchild длились много лет. Верховный суд США признал приоритет Fairchild. В 2000 году Килби получил половину Нобелевской премии «за вклад в изобретение интегральной схемы». Это решение носило компромиссный характер: к тому времени Нойс уже умер, а патентные войны остались в прошлом. Историк Бо Лоек отметил: «Идея Килби была настолько непрактична, что от нее отказались даже в TI. Его патент имел ценность лишь как объект торга».

Сегодня история IC рассматривается как результат коллективной работы множества изобретателей. Главное, что интегральная схема стала фундаментом современной микроэлектроники, а не продуктом одиночного гения.

Микропроцессоры

Почему мы все еще говорим «микропроцессор»?

Когда в 1970-х годах появились первые интегрированные процессоры, слово «микро» указывало на то, что весь управляющий функционал, который на тот момент занимал несколько компьютерных плат, а еще раньше одну или даже несколько стоек, теперь оказался собран на одной кристаллической подложке, которая, собственно говоря, называется чип. Этот чип заключают в корпус с внешними контактами, на первых в 70-х их было от 100 до 200, в наиболее мощных современных до 6000. Вот этот пирожок с начинкой-чипом и называется микропроцессором. Подчеркнем, чип – это только начинка. Но не только размер, еще префикс «микро» связывают с тем, что многие современные CPU работают под управлением специального уровня управления, его называют микроархитектурой. Этот уровень неперемнная часть одной из двух разновидностей архитектуры, именуемой CISC (complex instruction set computing или complex instruction set computer). Альтернативная версия – архитектура с жестко запрограммированными командами RISC (reduced instruction set computer), здесь команды обрабатываются напрямую аппаратном уровне, без промежуточного микрокода. В последние годы граница между CISC и RISC постепенно размывается.

CISC и RISC

На протяжении более чем полувека микропроцессоры развивались по двум независимым траекториям. Первоначально доминировала архитектура, которую позже назвали CISC (Complex Instruction Set Computer). Она появилась в индустрии в 1970-х годах и характеризовалась сложными командами, способными выполнять несколько действий за одну инструкцию. Программисту было проще писать код, а сам код получался компактным. За это процессору приходилось тратить больше времени на декодирование команд, сложнее было распараллеливать выполнение и оптимизировать конвейер.

Примерно через десять лет появилась академическая парадигма RISC (Reduced Instruction Set Computer). В отличие от CISC, RISC использовала простые, компактные

инструкции, которые чаще всего выполняются за один такт процессора. Сложные операции разбивались на несколько простых команд. Преимущества RISC проявлялись в высокой скорости работы, простоте конвейеризации и хорошей масштабируемости. Недостатками были увеличенный объем кода и более сложная компиляция.

В 1980-х обе архитектуры сосуществовали как отдельные парадигмы: CISC росла в недрах индустрии, обеспечивая удобство программирования и компактность кода, а RISC развивалась в академической среде, делая упор на скорость и эффективность исполнения инструкций. Параллельное существование этих подходов объясняется тем, что они решают разные задачи и обладают уникальными преимуществами в определенных условиях.

Современные процессоры часто комбинируют элементы обеих архитектур. Это позволяет использовать преимущества CISC в компактности и удобстве программирования, а RISC – в скорости выполнения и эффективной конвейеризации. Таким образом, история архитектур процессоров – это история постепенного сочетания двух подходов, каждый из которых развивался почти независимо, но в итоге дополнил другой.

CISC–Intel, восхождение, доминирование и уступка трона

Когда в 1971 году компания Intel представила миру первый коммерческий микропроцессор – Intel 4004, никто еще не знал, что начинается новая технологическая эра. Микросхемы размером с ноготь стали сердцем цифровой революции, а сама Intel – ее главным мотором.

На протяжении четырех десятилетий Intel удерживала лидерство в индустрии микропроцессоров. Ее продукты стояли внутри подавляющего большинства персональных компьютеров, от офисных машин до домашних игровых станций. Слоган «Intel Inside» стал символом надежности, производительности и статуса.

Хронология доминирования:

- 1980-е: архитектура x86 закрепляется как индустриальный стандарт.
- 1990-е: Pentium становится нарицательным именем; компьютеры ассоциируются с Intel.
- 2000-е: линейка Intel Core возвращает компании лидерство после кратковременного успеха AMD.

В это время AMD и другие конкуренты существовали – иногда даже догоняли, но почти никогда не опережали. Intel задавала темп, определяла архитектурные стандарты и даже диктовала, как будет выглядеть будущее вычислений.

Точка перегиба: замедление гиганта

Но у каждой империи есть предел господства. У Intel он начался, казалось бы, с незначительного, но рокового сбоя: компания не справилась с переходом на более тонкий техпроцесс – с 14-нм на 10-нм. То, что должно было случиться в 2016 году, откладывалось до 2019 и далее. В то время как TSMC, крупнейший контрактный производитель чипов, шагнул вперед и начал осваивать 7-нм техпроцессы, на которых работали уже и Apple, и AMD.

В 2017 году AMD представила архитектуру Zen – и внезапно стала снова опасным соперником. Серия процессоров Ryzen показала отличную многопоточную производительность, а серверные процессоры EPYC начали теснить Intel в ее, казалось бы, неприступной крепости – дата-центрах.

AMD предлагала больше ядер, меньше энергопотребления и лучшую цену. Это изменило правила игры. Если раньше Intel могла позволить себе медленные итерации, теперь это стало роскошью.

Apple и ARM: уход в другое измерение

В 2020 году еще один удар пришел из неожиданного направления. Apple отказалась от процессоров Intel, на которых работали ее ноутбуки и десктопы более 15 лет, и перешла на собственные чипы, построенные на архитектуре ARM. Так родился Apple M1 – энергоэффективный и невероятно мощный процессор, который потряс рынок.

ARM-чипы давно доминировали в смартфонах и планшетах, но с M1 стало ясно: они могут тягаться и с «большими» x86-процессорами. Intel пропустила революцию мобильных устройств – и теперь теряет позиции и в мире настольных решений.

2020-е: трон шатается

К началу 2020-х годов Intel столкнулась с конкуренцией сразу с нескольких фронтов:

- AMD – в классических x86-компьютерах и серверах.
- Apple и ARM – в мобильных и десктопных устройствах.
- TSMC – в области производства полупроводников, где Intel отстала по техпроцессу.

Финансово это стало ощутимо: в 2021 году TSMC обогнала Intel по рыночной капитализации, а в технологическом плане – ее лидирующая позиция была поставлена под сомнение впервые за десятилетия.

В 2021 году у руля Intel встал ветеран индустрии Пэт Гелсингер. Под его руководством компания объявила масштабную стратегию по модернизации производств, разработке новых архитектур и вхождению в рынок контрактного производства – прямой конкуренции с TSMC.

Цель – вернуть Intel в статус не просто одного из игроков, а лидера. Будет ли это возможно? Вопрос пока открыт.

Intel была символом цифровой эпохи. Она фактически изобрела микропроцессор, сделала его массовым продуктом и десятилетиями диктовала правила игры. Но даже самые могучие технологические империи сталкиваются с вызовами. Сегодня она больше не единоличный владыка микропроцессорного трона. Но если история чему-то учит – в том числе и история самой Intel – так это тому, что даже гиганты способны на второе дыхание.

История RISC

RISC (Reduced Instruction Set Computer) – это архитектурный подход, при котором используется малое количество простых и быстро исполняемых инструкций. Предпочтение отдается командам одинаковой длины и формата, основной упор делается на работу с регистрами, а не с памятью. Главная идея RISC проста: чем проще инструкция, тем быстрее и эффективнее ее выполнение.

Три школы RISC: три взгляда на простоту

В начале 1980-х годов в разных точках США почти одновременно родились три проекта, которые заложили фундамент концепции RISC. Каждый из них отражал свой подход к мышлению, проектированию и контролю над сложностью. Это проекты IBM 801, Berkeley RISC и Stanford MIPS. Они развивались независимо, но объединяет их общий дух – дух нового времени, когда громоздкие, многозначные инструкции уступили место простым, быстрым и управляемым конструкциям.

Хронологически первый прообраз RISC появился не в университете, а в корпорации IBM в середине 1970-х. Проект 801, названный по номеру здания, разрабатывался для телекоммуникационного оборудования. Автор, инженер Джон Кок, увидел, что большинство сложных CISC-инструкций не ускоряют выполнение программ, а наоборот, мешают. В результате родилась архитектура, где каждая инструкция короткая, однозначная и выполняется за один такт.

Почти одновременно профессор Дэвид Паттерсон в Калифорнийском университете в Беркли запустил проект, ориентированный изначально на обучение студентов архитектуре компьютеров. Паттерсон подошел к RISC как ученый: анализировал статистику использования инструкций и выявлял, какие реально ускоряют работу программ. Проект Berkeley RISC стал первой архитектурой, напрямую названной RISC, и оказал огромное влияние на последующие ARM и SPARC.

Третий путь открыл Джон Хеннеси в Стэнфорде. Его проект MIPS (Microprocessor without Interlocked Pipeline Stages – микропроцессор без взаимоблокирующих стадий конвейера) был академическим, но сразу ориентированным на реальные продукты. Хеннеси тесно работал с компиляторами, считая, что они должны брать на себя большую часть ответственности за оптимизацию кода.

Первые RISC-процессоры

Процессор IBM 801 никогда не продавался как коммерческий продукт; он использовался внутри IBM для телекоммуникационных систем. На его основе появились IBM ROMP – первый RISC в коммерческом продукте (рабочая станция IBM RT, 1986) и архитектура POWER/PowerPC, применяемая в суперкомпьютерах, игровых консолях и серверах.

Berkeley RISC (RISC I, 1981; RISC II, 1983) был реализован в университетской лаборатории на кремнии с частотой до 3 МГц. Эти чипы служили исследовательской целью, но оказали большое влияние на архитектуру SPARC (Sun Microsystems, 1987), которая широко использовалась в рабочих станциях.

Stanford MIPS появился в 1984 году. В 1985 Хеннеси основал MIPS Computer Systems, и вскоре были выпущены первые коммерческие процессоры: R2000 (1986) и R3000 (1988), применявшиеся в UNIX-системах и игровых приставках. В 1990-х MIPS использовался в продуктах SGI, Cisco и Sony (PlayStation 1).

Что осталось из первых RISC и кто выжил?

Большая часть классических MIPS-процессоров уже не выпускается для массового рынка, но архитектура продолжает жить в специализированных и встроенных системах. SPARC сохранялся в серверном сегменте, однако компании постепенно переходят на x86 или ARM. IBM POWER – самый живой представитель классического RISC, активно развивающийся и применяемый в суперкомпьютерах, мощных серверах и мэйнфреймах IBM Z.

Современные процессоры сочетают идеи CISC и RISC. Например, процессоры Intel и AMD содержат RISC-ядра, на которых выполняются сложные CISC-инструкции в виде микроопераций. Это позволяет сочетать богатство x86-инструкций и эффективность RISC-исполнения. ARM же остается классическим RISC с огромной экосистемой, доминируя в смартфонах, встроенных системах и серверных решениях, постоянно улучшая поддержку 64-битных расширений (ARMv8 и далее).

Период 2000–2025: Конец безальтернативности

В период с 2000 по 2025 годы архитектура x86 постепенно теряет статус безальтернативного стандарта. В мобильной и встраиваемой технике ее начинают теснить RISC-процессоры, такие как ARM и MIPS. На десктопах и серверах x86 по-прежнему удерживает позиции за счет традиций, огромной программной базы и совместимости.

В начале 2000-х развернулась «битва частот»: процессоры Athlon и Pentium 4 демонстрировали все более высокие тактовые частоты. Intel столкнулась с проблемой перегрева, а AMD выигрывала в эффективности. В 2003 году AMD представила архитектуру x86-64 (AMD64),

добавив 64-битность к x86 без отказа от старых программ. Этот шаг позволил архитектуре работать с большими объемами оперативной памяти и сохранил совместимость со старым ПО.

В 2006 году Intel вернула лидерство с выходом процессоров серии Core. Переход к многоядерным решениям и энергосберегающим технологиям сделал Intel Core 2 Duo новым стандартом для настольных и серверных систем.

С 2010-х годов ARM стремительно набирает популярность в мобильных устройствах: iPhone и Android выводят x86 из мобильного рынка, а попытка Intel Atom конкурировать с ARM оказалась неудачной. В 2017 году мировое сообщество столкнулось с уязвимостями Spectre и Meltdown, которые затронули процессоры обеих архитектур.

В 2020 году Apple полностью отказалась от x86, выпустив собственные процессоры Apple Silicon (M1 и последующие), показывая, что ARM способен превосходить x86 по энергоэффективности даже на десктопе. В 2023–2025 годах x86 теряет эксклюзивность: Microsoft выпускает Windows на ARM, а специализированные ИИ-чипы (NPU, GPU, TPU) забирают часть вычислений у CPU. Сегодня x86 остается важной архитектурой для корпоративного и ПК-сегмента, но больше не является «центром мира» вычислительной техники.

Apple и ARM

Когда в 2007 году Apple выпустила первый iPhone, внутри него находился ARM11 – простой и энергоэффективный процессор. Выбор ARM был очевиден: эта архитектура идеально подходила для мобильных устройств, где важны компактность, энергосбережение и низкое тепловыделение. Уже тогда Apple поняла: архитектура важнее, чем частоты и мегабайты.

С 2010 года, начиная с A4, Apple самостоятельно разрабатывает процессоры на базе ARM. С каждым поколением (A5, A6, A7...) они становились все мощнее, опережая конкурентов по производительности на ватт. В A7 Apple впервые представила 64-битную мобильную архитектуру, чем опередила индустрию минимум на два года. К концу 2010-х чипы Apple A-серии уже были ближе к десктопному классу, чем к традиционным мобильным решениям.

С появлением iPad Pro стало очевидно: ARM-чипы Apple выходят за пределы «мобильного». A12X, A14X – это полноценные вычислительные платформы с высокой графической и вычислительной производительностью. Именно в iPad Pro архитектура Apple Silicon впервые начала напоминать то, что позже окажется в Mac.M. Переход на

В 2020 году Apple объявила о переходе с x86 (Intel) на собственные ARM-чипы – Apple Silicon. Первый из них, M1, стал настоящим технологическим прорывом:

- Собственные высокопроизводительные и энергоэффективные ядра.
- Интеграция CPU, GPU, RAM и Neural Engine в одном чипе.
- Unified Memory Architecture: единая шина между компонентами.
- Поддержка Rosetta 2 для запуска x86-приложений без значительной потери скорости.

RISC-V: открытая архитектура, которая меняет правила игры

Еще недавно мир процессоров был жестко поделен. На персональных компьютерах – x86 от Intel и AMD, в смартфонах и встраиваемой электронике – ARM. Любой, кто хотел сделать собственный чип, был обязан покупать лицензию и играть по правилам владельцев архитектуры. Это ограничивало свободу и тормозило эксперименты.

В 2010 году в Калифорнийском университете в Беркли группа ученых решила попробовать другой путь. Они разработали RISC-V – простую и свободную архитектуру команд, которая изначально задумывалась как удобный инструмент для обучения и исследований. Но идея оказалась гораздо сильнее учебного проекта.

Главное новшество RISC-V в том, что ее спецификация открыта для всех. Это не готовый процессор, а набор правил, описывающих, какие команды понимает машина. Любая компания, университет или энтузиаст могут на ее основе создать собственное ядро: крошечный микроконтроллер для датчика, процессор для смартфона или даже чип для суперкомпьютера.

Архитектура устроена модульно. Есть минимальный набор инструкций, необходимый для запуска системы, а дополнительные возможности подключаются как расширения: арифметика с плавающей точкой, работа с векторами, криптография. Такой подход делает RISC-V гибкой: она подходит и для простейших устройств, и для высокопроизводительных решений.

Исторические корни RISC-V

RISC-V вырос не на пустом месте. Его предшественниками были ранние проекты по созданию RISC (Reduced Instruction Set Computer) – «процессоров с сокращенным набором команд».

В конце 1970-х годов Дэвид Паттерсон и его команда в Беркли создали RISC-I и RISC-II. Их идея была в том, чтобы вместо сложных и медленных инструкций сделать небольшой, но быстрый набор команд. Это оказалось эффективнее: упрощенный процессор мог работать быстрее и использовать меньше транзисторов.

Эти разработки породили целое поколение промышленных архитектур: MIPS, SPARC, PowerPC, ARM. Каждая из них добилась успеха в своем сегменте – от серверов до мобильных устройств.

Однако со временем большинство таких архитектур стали закрытыми и дорогими для лицензирования.

RISC-V, появившийся в 2010-м, стал возвращением к истокам: это пятая исследовательская архитектура из Беркли (отсюда «V» в названии). Она сохранила дух RISC – простоту и эффективность – и при этом добавила то, чего раньше не было: открытую спецификацию, доступную каждому.

Хронология развития

2010 – старт проекта в Беркли.

2014 – публикация первой версии спецификации.

2015 – основание некоммерческой организации RISC-V Foundation (позже RISC-V International).

2018–2019 – к проекту подключаются крупные компании: NVIDIA, Western Digital, Google.

2020-е – выход готовых процессоров на RISC-V, использование в IoT, контроллерах памяти, суперкомпьютерах.

Можно сказать, что RISC-V – это аналог Linux в мире железа: открытая основа, на которой каждый строит свое. И, как и с Linux, сначала это казалось академической забавой, а теперь все больше похоже на новую технологическую реальность.

Графический процессор: от рендеринга к ИИ

В начале было изображение

В 1980-х компьютеры умели показывать графику, но делали это неохотно. Центральный процессор (CPU) рисовал все – от иконки до трехмерного объекта, – а его архитектура, заточенная под последовательную обработку, страдала от перегрузки. Когда на горизонте появились видеоигры, стало очевидно: нужен новый подход.

Так появились первые графические ускорители. Они брали на себя простейшие графические операции – наложение текстур, рендеринг спрайтов – и быстро стали неотъемлемой частью игровых ПК и рабочих станций.

Рождение GPU: NVIDIA и ATI

В 1999 году NVIDIA представила GeForce 256, устройство, в котором маркетологи впервые употребили термин "Graphics Processing Unit" (GPU). Это был качественный скачок: GPU стал самостоятельным вычислительным блоком с возможностью аппаратной трансформации, освещения и обработки треугольников. CPU, наконец, перестал быть обязанным делать все сам.

Но NVIDIA не была единственным игроком. Канадская компания ATI Technologies, основанная в 1985 году, развивала собственную линию графических решений. В 2000-х ATI выпустила серию Radeon, которая составила серьезную конкуренцию линейке GeForce. Борьба двух гигантов, NVIDIA и ATI, во многом определяла темп развития всего графического рынка.

В 2006 году компанию ATI приобрела AMD, и с тех пор видеокарты под брендом AMD Radeon продолжают противостоять GeForce в ожесточенной технологической гонке.

GPU выходит за рамки

К середине 2000-х годов стало ясно: GPU умеют нечто большее, чем просто рендеринг графики. Их архитектура с десятками, а затем и сотнями ядер позволяла выполнять тысячи параллельных операций одновременно. Это было находкой для научных вычислений и инженерных задач, особенно там, где важна скорость массовых вычислений. Так появилось направление GPGPU – General-Purpose computing on Graphics Processing Units. Ключевым моментом стало появление CUDA (Compute Unified Device Architecture) от NVIDIA в 2006 году. CUDA дала программистам инструменты для написания кода, запускаемого на GPU, как на полноценном вычислительном устройстве.

С тех пор GPU применяются в: молекулярной динамике и физике высоких энергий, финансовом моделировании, криптографии и хешировании, видеорендеринге и симуляции жидкостей, биоинформатике и секвенировании генома.

GPU и искусственный интеллект

Настоящий взрыв произошел в 2010-х годах, когда нейросети стали обрабатывать миллионы параметров. Стандартный CPU не справлялся с объемом и характером этих вычислений. А вот GPU оказался идеальным кандидатом: массовые матричные операции – его стихия.

Сегодня обучение больших языковых моделей (LLM), генеративных ИИ, компьютерного зрения и распознавания речи происходит именно на GPU. NVIDIA с ее архитектурой

Глава 2. Нулевое поколение компьютеров

Первопроходцы

В начале 1940-х годов мир оказался на пороге невидимой революции. Идея автоматической вычислительной машины возникала почти одновременно у нескольких ученых и инженеров, но каждый видел ее по-своему – через призму своей профессии, характера и культурной среды. Каждый проект отражал личность создателя и дух времени, а вместе они закладывали фундамент новой эры вычислительной техники.

В Германии Конрад Цузе работал почти в одиночку в Берлине. Его Z3, созданная в 1941 году, стала первой в мире программируемой цифровой вычислительной машиной, но не электронной. Машина могла работать с числами с плавающей точкой, память и вычислительные блоки были четко разделены, а архитектура отличалась строгой логикой. В подходе Цузе ощущалась немецкая педантичность: точность, порядок и надежность были для него неотъемлемой частью инженерного творчества.

В это же время в США физик Джон Атанасов в Университете Айовы разрабатывал машину для решения систем линейных уравнений. Его Atanasoff—Berry Computer (ABC) впервые использовал двоичную арифметику, вакуумные лампы и электрическую память на конденсаторах. Машина была рациональной и прагматичной, как ее создатель, сосредоточенный на эффективности и функциональности, не универсальной.

В Bell Labs, Джордж Стибиц создавал свои прототипы на кухне собственного дома. Его калькулятор Model K и последующий Complex Number Calculator открыли новую идею – управление машиной на расстоянии через телеграф. Стибиц видел компьютер не только как инструмент расчетов, но и как средство связи, предвосхитив сетевое взаимодействие задолго до появления интернета.

В Гарварде Джордж Айкен создавал машины как инструмент мышления. Harvard Mark I (ASCC), завершенная совместно с IBM в 1944 году, работала медленно, но стабильно, управляясь программой с перфоленки и обрабатывая числа с плавающей точкой. Для Айкена компьютер был продолжением интеллекта математика – инструментом, с помощью которого абстрактные формулы превращались в реальные результаты.

По заказу армии США Джон Эккерт и Джон Моукли построили ENIAC – машину, которая занимала целый зал. Она была универсальной и мощной, хотя не программировалась «внутри». ENIAC отражал возможности индустрии и государства, демонстрируя масштаб коллективного инженерного труда и практическую ориентацию вычислительной техники.

Ванневар Буш создавал свой гигантский Differential Analyzer, длиной десять метров, предназначенный для численного решения дифференциальных уравнений. Машина сочетала инженерную изобретательность с практическими потребностями физиков, инженеров и баллистиков. Для Буша вычислитель был не просто механизмом, а инструментом расширения человеческого разума.

Каждая из этих машин и каждый из этих инженеров были по-своему первопроходцами. Их подходы отличались, их задачи были разными, но объединяла одно – стремление расширить возможности человеческого интеллекта и превратить абстрактную мысль в рабочую, осязаемую машину. Именно их усилия создали фундамент, на котором выросла современная вычислительная техника.

Великолепная семерка

ABC – машина, воскресшая по суду

О брошенной своим создателем по причине ухода в армию машине ABC никто бы, возможно, и не вспомнил, если бы в 1971 году она неожиданно не оказалась в центре скандального судебного процесса. Компания Sperry Rand, в ту пору производитель мэйнфреймов UNIVAC, обвинила своего конкурента Honeywell в якобы имевшем место нарушении патента на электронный компьютер. Этот патент прежде принадлежал Джону Моукли и Пресперу Эккерту, заявку они подали еще в 1947 году, но официально документ выдали только в 1964-м, но еще в 1950 году права вместе с фирмой Eckert—Mauchly Computer Corporation перешли к Sperry Rand.

Honeywell стала ключевым ответчиком, потому что она публично отказалась платить лицензионные отчисления, в то время как остальные компании также игнорировали патент, но делали это тихо, не бросая прямого вызова Sperry Rand.

Процесс длился почти два с половиной года. В нем адвокаты Honeywell выбрали необычную стратегию, не оспаривая факта нарушения, они сосредоточились на доказательстве недействительности собственно самого патента. В качестве главного аргумента они извлекли из забвения Atanasoff—Berry Computer и представили его как первый в мире электронный цифровой вычислитель. Если это признать, то приоритет изобретения принадлежит Джону Атанасову, а Моукли, как утверждала защита, недобросовестно воспользовался его идеями.

Аргументация выглядела спорной, но судья Эрл Ларсон неожиданно и, прямо говоря, вопреки логике принял ее. В 1973 году он вынес историческое решение: патент Моукли и Эккерта был признан недействительным, но он и не перешел к Атанасову, а был передан в общественное достояние (public domain).

Особую роль в формировании образа незаслуженно обиженного Атанасова сыграли супруги Артур и Элис Буркс. Артур участвовал в разработке ENIAC, а позже на пару с Элис они посвятили себя историческим исследованиям. Их книга «Кто изобрел компьютер. Процесс, изменивший компьютерную историю» (1996) активно утверждает первенство Атанасова. Со временем многие историки посчитали такой взгляд чрезмерно ангажированным. Ассоциация вычислительной техники (АСМ) и ряд независимых исследователей отмечали, что решение суда могло быть несправедливым по отношению к Моукли и Эккерту.

Но самое громкое эхо прозвучало не в США, а в Болгарии. Там Атанасова провозгласили национальным героем, была учреждена президентская премия John Atanasoff Award, десятки школ и улиц получили его имя, даже астероид с номером 3546 получил имя Atanasoff. Ирония заключалась в том, что сам Атанасов не считал себя связанным с исторической родиной, он родился и прожил всю жизнь в США, не говорил по-болгарски и не имел болгарского гражданства. Это классический пример национальной героизации представителей диаспоры, можно привести множество примеров, когда страна с охотой присваивает себе заслуги человека, добившегося мирового признания, практически не имея к исторической родине прямого отношения.

Джон Атанасов родился в 1903 году. Его отец, болгарский эмигрант, сменил фамилию Atanasov на Atanasoff, а мать преподавала математику. В 1921 году Джон поступил в Университет штата Флорида, получил образование по электротехнике и затем защитил докторскую диссертацию по теоретической физике. В своей научной работе он постоянно сталкивался с механическими счетными машинами – арифмометрами Монгое и табуляторами IBM, и был недоволен их функциональной ограниченностью.

Поэтому для начала 1936 году ученый построил небольшой аналоговый вычислитель *Laplaceometer* для анализа поверхностей, но искал более универсальное решение. По воспоминаниям, вдохновение к созданию ABC пришло к нему после ночной автомобильной поездки и беседы за бокалом виски. Именно тогда он сформулировал четыре принципа будущей машины:

- она должна быть электронной;
- работать в двоичной системе;
- использовать конденсаторную память с регенерацией;
- основываться на логических арифметических операциях.

При поддержке Университета штата Айова Атанасов получил грант в сумме 650 долларов и привлек к сотрудничеству аспиранта Клиффорда Берри. К концу 1939 года они построили прототип ABC.

С началом Второй мировой войны Атанасов и Берри оказались заняты оборонными проектами, а их машина заброшенной и частично разобранной. Атанасов больше к ней не возвращался, лишь в ходе судебного процесса против Honeywell ABC вновь была извлечена из забвения, и создатель рассказал о своей работе.

Технически ABC была необычной. В ней использовалась регенеративная память на конденсаторах, рассчитанная примерно на 300 чисел или команд. Арифметическое устройство оперировало векторами данных, что сближало ее с более поздними векторными машинами. Программы и данные хранились отдельно – по сути, это был один из ранних примеров реализации гарвардской архитектуры. Машина была сравнительно компактной (размером с письменный стол) и потребляла около 1 кВт энергии.

До нашего времени от оригинала сохранились описания и фрагменты памяти. По ним в 1990-е годы в Айовском университете была построена рабочая реплика.

По итогам процесса Атанасов не получил коммерческой выгоды от своего изобретения. Он продолжал работать в университетах, выступал на конференциях и позже охотно ездил в Болгарию. Его вклад был признан лишь исторически.

История ABC – это не рассказ о краже идей, а пример того, как новые замыслы циркулируют в научном сообществе и влияют друг на друга. Машина Атанасова и Берри стала одной из вех на пути к современным компьютерам.

Компьютеры на телефонных реле

В условиях фашистской Германии Конрад Цузе создавал свои вычислительные машины в полной изоляции от остального мира. Пока в США и Великобритании активно разрабатывали электронные компьютеры, он работал без доступа к зарубежным публикациям и научным контактам. Эта «информационная блокада» сделала его достижения по-настоящему уникальными.

Родившийся в 1910 году в Берлине, Цузе поначалу увлекался живописью и архитектурой. В 1935 году стал инженером-проектировщиком в авиационной фирме Heinkel, где столкнулся с громоздкими ручными расчетами и задумался о создании автоматической вычислительной машины. В 1936 году он сформулировал главный принцип: вычислитель должен преобразовывать одну спецификацию в другую по заданным правилам. В 1943 году он даже дал собственное определение компьютеринга – «получение новой спецификации на основании старой и предписаний».

Первая машина Цузе, Z1, была построена в 1938 году. Она представляла собой программируемый механический двоичный компьютер, где арифметические и логические операции выполнялись приспособлениями, изготовленными из металлических пластин. Память вмещала 64 слова по 22 бита, программы считывались с перфоленты. Машина работала нестабильно, но уже здесь проявилось инженерное новаторство Цузе.

В 1940 году Цузе построил Z2, электромеханический компьютер на телефонных реле. В 1941 году появилась Z3 – полноценный программируемый цифровой компьютер на 2600 реле. Позднее было доказано, что Z3 обладала всеми свойствами тьюринг-полной машины. Во время войны Цузе работал над Z4, которая была завершена в 1945 году и позже вывезена в Швейцарию. В 1950 году Z4 заработала в Цюрихском политехникуме (ETH Zürich) и стала первой коммерчески используемой вычислительной машиной в Европе, работая до середины 1950-х годов.

Параллельно, в 1942–1946 годах, Цузе разработал проект языка Plankalkül («план вычислений») – первого в истории формального языка программирования для записи алгоритмов. Его идеи опередили свое время на десятилетия, но публикация состоялась лишь в 1972 году, а практическая реализация появилась в 1970–1980-х.

В 1940 году Цузе основал фирму Zuse Apparatebau, но в 1945-м ее здания были уничтожены бомбежкой. В 1949 году он создал компанию Zuse KG, первую компьютерную фирму в Германии. Z4 стала основой для дальнейших разработок: появились релейные Z5 и Z11 (последняя оказалась успешной в университетах и геодезии). К середине 1950-х стало ясно, что механические и релейные системы исчерпали себя, и Цузе перешел на электронику, выпустив ламповую Z22, а затем транзисторные Z23 и Z31.

В 1967 году Siemens приобрел Zuse KG, а ее основатель отошел от бизнеса. Оставив индустрию, Цузе занимался различными проектами: интересовался ветряными двигателями, теорией сетей Петри и вновь вернулся к живописи. В 1980-е годы он даже выставлялся под псевдонимом Kuno See, сочетая инженерию и искусство.

Сегодня заслуги Конрада Цузе признаются как один из основополагающих вкладов в развитии вычислительной техники. В условиях политической изоляции ему удалось создать целое семейство машин, заметно превосхитивших мировое развитие, а Plankalkül стал предвестником современных языков программирования. Цузе остался в истории как изобретатель, который почти в одиночку и в тени войны открыл свою дорогу к цифровой эпохе.

Калькулятор с дистанционным доступом

Имя Джорджа Стибица сегодня известно гораздо меньше, чем других первопроходцев компьютерной эры. Он открыл два направления, без которых трудно представить современную информатику: релейную вычислительную технику и концепцию удаленного доступа к компьютеру.

В 1930-е годы Стибиц работал в Bell Labs, где параллельно с Клодом Шенноном занимался применением булевой алгебры. Их пути разошлись: Шеннон ушел в теорию информации, а Стибиц – в практическую автоматизацию вычислений. Оба заложили фундамент «цифровой революции», но Стибиц пошел по пути инженерной реализации и доказательства работоспособности идей на практике.

Первый прототип релейного калькулятора Стибиц собрал в 1937 го на собственной кухне. Для конструкции использовались подручные материалы: лампочка от сигнального фонаря, звонок и выключатель от дверного звонка. Машина получила шуточное название Model K (от kitchen – кухня). Лампочка моргала при работе, а звонок сигнализировал о завершении вычисления. Этот скромный эксперимент стал началом целого направления. Сегодня Model K хранится в Смитсоновском институте.

В 1940 году в Bell Labs Стибиц создал первый настоящий релейный калькулятор – Complex Number Calculator (Model I). Машина насчитывала около 450 телефонных реле и имела несколько принципиально новых особенностей:

- ввод и вывод осуществлялся через телетайпы;
- данные представлялись в двоично-десятичной форме;

- к машине можно было подключать несколько удаленных терминалов;
- отдельные арифметические блоки обрабатывали одновременно реальную и мнимую часть комплексного числа.

11 сентября 1940 года Стибиц организовал первую в истории демонстрацию удаленного доступа: терминалы, установленные в Дартмутском колледже (Нью-Гэмпшир), связывались по телефонной линии с калькулятором в Bell Labs в Нью-Йорке. По сути, это был прообраз будущих компьютерных сетей и удаленных вычислений.

Model I предназначался для расчетов с комплексными числами. Несмотря на то, что одна операция занимала до 30 секунд, машина исправно работала почти девять лет – с 1940 по 1949 год. Это демонстрировало не только надежность релейной техники, но и эффективность инженерной мысли Стибица.

Джордж Стибиц вошел в историю как человек, впервые связавший вычислитель с удаленным терминалом и продемонстрировавший возможности релейной автоматики. Он доказал, что надежные цифровые вычисления возможны даже без электронных компонентов, а его демонстрация 1940 года стала первым шагом к будущим компьютерным сетям, послужив вдохновением для последующих проектов в ARPANET и системах удаленного доступа.

Harvard Mark I

Машина Harvard Mark I выпадает из ряда остальных машин нулевого поколения – она осталась в истории как пример утопического гигантизма. В истории технологий есть особые моменты, когда грандиозные замыслы выходят за пределы чисто утилитарного и становятся символами эпохи, почти мифами – монументами человеческой дерзости и мечты. Они демонстрируют дух времени, когда технологии не просто решали задачи, а становились олицетворением веры в безграничные возможности. Среди таких проектов:

- Titanic – символ индустриального оптимизма начала XX века, попытка создать непотопляемое чудо инженерии;
- Hindenburg – воздушный гигант эпохи дирижаблей, метафора мечты и опасности утопического гигантизма;
- Spruce Goose (H-4 Hercules) – деревянный гигант, символ технической дерзости и почти утопической амбиции.

Эпоха гигантских проектов завершилась. Сегодня величие измеряется точностью, скоростью и универсальностью, а не размером и сложностью. Но память о тех легендарных машинах живет – это памятники дерзости и мечты, уроки о границах возможностей и цене амбиций.

Два Говарда: Эйкен и Хьюз

В 1940-е годы жили два человека по имени Говард, создавшие машины-гиганты. Говард Хьюз построил деревянный летающий монстр Spruce Goose, а Говард Эйкен – гигантский вычислитель Harvard Mark I. Машины были совершенно разными: одна из дерева и воли, другая из металла и логики. Но обе стали символами эпохи, созданными не столько для использования, сколько как доказательство человеческой дерзости. Обе пережили своих создателей: Mark I теперь экспонируется в музее Гарвардского университета, Hercules – в Авиационном музее в Мак-Минвилле.

Идея Harvard Mark I напрямую восходит к Чарльзу Бэббиджу и его Analytical Engine. Хотя между ними лежит почти столетие, между ними прослеживается удивительная идейная преемственность. Эйкен наткнулся на труды Бэббиджа и почувствовал себя завершающим то, что тот начал. В предисловии к отчету о Mark I (1944) Эйкен называет Бэббиджа «первооткрывателем современной вычислительной машины» и посвящает ему свою работу.

Конструкция и технические характеристики

Mark I строилась на электромеханических реле, двигателях и роторах. Она могла выполнять арифметические операции, читать команды с бумажной ленты и следовать программе без вмешательства человека. В этом смысле функциональная модель Mark I повторяет замысел Бэббиджа: автоматическое исполнение сложных вычислений.

Практическое конструирование началось в 1939 году, а в январе 1943 ASCC (Automatic Sequence Controlled Calculator) выполнила первую тестовую программу. Машина имела:

- длину 15,5 м, высоту 2,4 м, глубину 0,6 м;
- вес 35 тонн, потребляемую мощность 25 кВт;
- 765 тыс. деталей и около 800 км проводов.

Числа хранились в регистрах из 10-зубчатых колес (24 колеса на регистр) с механизмом сквозного переноса десятков. Сложение занимало 0,3 секунды, умножение – 6 секунд, деление – 15 секунд, извлечение квадратного корня – около минуты. Главное достоинство – программирование последовательности действий, что стало революцией.

Mark I использовалась для баллистических таблиц, навигации и артиллерийских расчетов. После Mark I появились Mark II и Mark III, но вскоре на сцену вышли электронные компьютеры: ENIAC, EDVAC, UNIVAC, IBM 701, вытеснив электромеханику.

Первые «леди программирования»

Mark I стала первой программируемой машиной, а первую программу для нее написала Грейс Хоппер. В 1940 году Хоппер поступила на службу в Вооруженные силы США и стала третьим программистом Mark I в Бюро артиллерийских вычислительных проектов при Гарварде.

Однажды машина вышла из строя, и причина оказалась забавной: между контактами реле застрял мотылек. Его приклеили в журнал с подписью «First actual case of bug being found». Так родился термин bug и практический debugging.

После войны Хоппер создала первый компилятор, стояла у истоков COBOL, получила звание контр-адмирала и прозвище «бабушка COBOL».

Если Бэббидж был пророком, видевшим возможности вычислений в пространстве идей, Хоппер была строителем: она воплощала идеи в реальные программы, делая машины понятными человеку. Лавлейс предвидела, Хоппер создавала. Вместе они символизируют становление программирования как науки и искусства.

Наследник Аналитической машины Бэббиджа

Среди создателей машин нулевого поколения одной из ярких фигур был Ванневар Буш. Его знаменитый дифференциальный анализатор – лишь часть его деятельности. Буш сочетал научный талант, инженерное мышление и организаторские способности, что позволило ему создать модели управления крупными научными проектами и в итоге подготовить основу для будущего развития вычислительной техники и Интернета.

- В Буше сочетались три ключевых качества:
- Крупный академический ученый и преподаватель;
- Универсальный инженер с масштабным видением;
- Искусный организатор и администратор.

Даже после выхода на пенсию он продолжал работать с инженерными проектами, например, изготавливая уникальные хирургические инструменты.

До Второй мировой войны Буш преподавал в МИТ, был деканом и занимался научной работой. В 1940 году президент Рузвельт пригласил его в администрацию в качестве научного советника. Буш организовывал работу ученых и инженеров, включая многих европейских эмигрантов, в рамках американского военно-промышленного комплекса. Его действия способствовали продвижению США на передовые позиции в науке и технике.

Буш осознал необходимость использования высоких технологий в военной индустрии и сумел преодолеть сопротивление как со стороны консервативных военных, так и со стороны европейских ученых-эмигрантов, направив их усилия на практическое применение науки в обороне. Его вклад в организацию научной работы был настолько значительным, что его прозвали «царем американской науки» – за участие в Манхэттенском проекте и руководство рядом оборонных исследований.

Первая версия дифференциального анализатора была создана в 1931 году в МИТ. Машина решала дифференциальные уравнения, что являлось ключевым инструментом для баллистики, электротехники и других прикладных дисциплин. Анализатор Буша стал связующим звеном между разностной машиной Бэббиджа и интегратором Кельвина, объединив автоматическое решение задач и физическое непрерывное вычисление. Машина стала универсальной платформой для моделирования реального мира.

Конструкция включала:

- шесть механических интеграторов с вращающимися дисками и колесами;
- координатные передатчики для задания входных функций и считывания результатов;
- цепи шкивов, валов и дифференциальных механизмов для передачи вращения между блоками;
- электромоторы и системы обратной связи для стабилизации работы;
- панель настройки для задания структуры уравнения.

Машина была аналоговой: вместо чисел использовались физические величины – углы, скорость вращения, линейные смещения. Решение уравнения воспроизводилось как динамический процесс, а результаты записывались на самописце или считывались оператором.

Дифференциальный анализатор породил небольшое семейство аналоговых машин. Оригинал, построенный Бушем совместно с Гарольдом Хейзенем, стал прототипом модификаций в Пенсильванском университете (1935), которые повлияли на будущих создателей ENIAC – Моукли и Эккерта. Военные версии использовались для расчета баллистических таблиц в Абердинской Баллистической лаборатории. Копии и модификации машин появились в университетах Манчестера, Оксфорда, Кембриджа, Торонто, а в Великобритании один анализатор применял Алан Тьюринг при ранних экспериментах с вычислительными моделями.

Дифференциальный анализатор Буша стал не только инженерным проектом, но и организационным и концептуальным мостом к цифровой эпохе, демонстрируя, как научное видение может быть преобразовано в практический инструмент инженерии и обороны.

ENIAC – в одном шаге от настоящего компьютера

До настоящего компьютера оставался один шаг – и он был сделан с появлением ENIAC (Electronic Numerical Integrator and Computer). Название машины отражало ее переходный характер и особенности понимания идеи компьютера в 1940-х:

- Electronic – впервые вычисления осуществлялись не механически, а с помощью электронных ламп. Это означало радикальный технологический сдвиг.
- Numerical Integrator – изначально ENIAC проектировался для расчета баллистических таблиц; интегрирование дифференциальных уравнений было ключевой задачей.

- Computer – слово, которое тогда еще чаще обозначало человека, выполняющего вычисления. Название символизировало замену человеческого труда машиной: «мы создаем электронный вычислитель, который выполняет работу человека».

Инженерное открытие сделали Джон Моукли и Преспер Эккерт, но выдающуюся роль в реализации проекта сыграл Герман Гольдштейн. Работая в Баллистическом исследовательском центре армии США, он стал посредником между разработчиками и военными, убедил руководство в стратегической важности ENIAC и обеспечил финансирование. Голдстейн продемонстрировал, что даже гениальные инженеры нуждаются в инфраструктуре и визионерах для воплощения своих идей.

- ENIAC был массивной конструкцией:
- около 18 000 вакуумных ламп;
- площадь около 170 м²;
- потребление энергии до 150 кВт.

При этом машина могла выполнять около 5000 операций сложения в секунду, что в сотни раз превышало скорость электромеханических предшественников. Работа с ENIAC требовала тщательного охлаждения и постоянного обслуживания; температура в зале могла достигать невыносимых уровней.

Особую роль сыграли женщины-программистки – шесть сотрудниц, включая Джин Бартик и Кэй МакНалти. Они разработали методы ручного программирования машины, оперируя сложной системой кабелей и переключателей. Их вклад долго оставался в тени, но без них проект не достиг бы успеха.

ENIAC создавался как инструмент Второй мировой войны для ускорения артиллерийских расчетов. Хотя машина была завершена в 1945 году и продемонстрирована в 1946-м, она оказала колоссальное влияние на дальнейшие разработки:

Технологический прорыв – переход к полностью электронным вычислениям радикально ускорил обработку информации.

- Переходная архитектура – ENIAC не имел хранимой программы, но заложил идею универсальности: машину можно было перепрограммировать для разных задач.
- Ограничения – сложность перепрограммирования и отсутствие хранимой программы стимулировали разработку EDVAC и архитектуры фон Неймана.
- Институциональный эффект – ENIAC показал возможность электронных вычислений, изменив отношение военных, инженеров и ученых к вычислительной технике.
- Социальное значение – женщины-программистки ENIAC положили начало истории женского участия в программной инженерии.

ENIAC не был компьютером в полном смысле, но стал мостом к ним. Он доказал практическую возможность электронных вычислений и запустил эпоху дальнейших разработок.

Глава 3. Школа Мура и создание первого электронного компьютера

Феномен Школы Мура: инженерия и наука

Создание первого электронного компьютера с хранимой в памяти программой обычно представляют слишком упрощенно. Устоявшаяся легенда звучит так: «Великий математик Джон фон Нейман предложил схему, которая в дальнейшем получила название архитектура фон Неймана, а инженеры, следуя его гениальному предвидению, построили EDVAC, который стал прототипом всех последующих машин». Этот придуманный сюжет повторяется от книги к книге, хотя реальная история оказалась гораздо сложнее.

Для полноценного понимания происходившего полезно обратиться к системному подходу, разработанному австрийским биологом и философом Людвигом фон Берталанфи. У него «подмоченная репутация» – вступил в нацистскую партию в 1938 году и использовал свои связи с режимом для продвижения академической карьеры. Это не умаляет его научного вклада. Говоря о связи теории и практики он подчеркивал, что инженерная практика и фундаментальная наука должны рассматриваться как части одной большой системы. Он писал: «Практика рождает новые идеи, а теория помогает создавать еще более совершенные устройства». Такой взгляд позволяет увидеть историю первого компьютера во всей ее полноте.

От стереотипов к реальности

Системному пониманию истории технологий мешают укоренившиеся стереотипы. Мы привыкли считать, что теория всегда предшествует практике: сначала ученые открывают законы природы, затем инженеры воплощают их в устройствах. Часто в качестве образца приводят ядерную физику: исследователи изучили процессы деления атома – и лишь после этого появились реакторы и атомное оружие.

Такая схема прочно закрепилась и в представлениях о вычислительной технике. Архитектура фон Неймана нередко подается как исходная теоретическая модель, из которой выросли все последующие компьютеры. Однако реальная история развития вычислений была куда менее линейной и значительно более взаимной: практика нередко опережала теорию, а инженерные решения появлялись раньше их строгого концептуального осмысления.

Не только в компьютеринге во множестве случаев технические решения появлялись раньше науки, способной их объяснить. Паровая машина была создана задолго до того, как возникла термодинамика. Двухколесный велосипед, изобретенный Луи и Пьером Мишо в 1860-е годы, получил объяснение своей устойчивости лишь через столетие. Аналогично с первым компьютером: инженеры Джон Моукли и Преспер Эккерт разработали схему EDVAC, исходя из опыта создания экспериментального ENIAC. Только затем Джон фон Нейман увидел их проект, переосмыслил его и предложил свое описание принципов работы машины.

Здесь проявляется принципиальное отличие компьютерных наук от естественных. Естественные науки исследуют то, что существует независимо от человека. Компьютерные науки создают искусственные объекты. Сначала появляется артефакт, а теория объясняет и обобщает его. Иногда практика порождает собственные науки – теорию алгоритмов, теорию вычислимости и, позднее, целые направления в искусственном интеллекте, включая машинное обучение, компьютерное зрение и обработку естественного языка. Эти дисциплины не могли возникнуть до появления самих вычислительных машин.

Цикл науки, инженерии и менеджмента

На протяжении десятилетий приоритет в развитии вычислительной техники то переходил к инженерам, то возвращался к ученым. Архитектура x86 является примером чисто инженерного продукта. Решения в ней диктовались практическими соображениями, а теория лишь постфактум анализировала и объясняла эти решения. Новая архитектура RISC, наоборот, возникла в университетских лабораториях: сначала это была научная идея, а позже она воплотилась в инженерные решения. Иногда наука подключалась к существующей технологии и «оборачивала» ее в формализм. Так произошло с системами управления базами данных: сначала создавались утилитарные решения, а затем вокруг них сформировалась теория реляционной алгебры с доказательствами корректности.

Нельзя забывать и третью силу – менеджеров, которые создают условия для реализации технических и научных идей. В истории ENIAC и EDVAC такую роль сыграл Герман Гольдштейн, который обеспечил проект ресурсами, связями и защитой от бюрократии. Позднее аналогичную функцию выполняли лидеры вроде Стива Джобса. Он не создавал технических новшеств сам, но превращал инженерные находки и дизайнерские идеи в продукты, определявшие направление всей отрасли.

Школа Мура воплощала гармоничный союз инженеров, ученых и менеджеров. Здесь роли распределялись гибко, по историческому ритму: сначала ведет техника, затем ее осмысляет наука, потом снова наступает черед инженерии, а менеджеры все это время создают среду для циклического развития и реализации идей.

Роль Школы Мура: случайность или закономерность?

Эта смена ритмов хорошо согласуется с концепцией Томаса Куна. Американский философ науки в книге «Структура научных революций» ввел ключевое понятие – парадигма. По его мысли, развитие науки и техники идет не плавно и не линейно, а скачками: старая парадигма исчерпывает себя, и ее место занимает новая. Новые идеи рождаются не как внезапное озарение отдельного гения, а как ответ среды, где уже накопились все предпосылки.

История техники подтверждает эту логику. Изучая путь телеграфа, телефона, радио, автомобиля или самолета, мы видим, что близкие идеи появляются параллельно в разных странах и у разных людей. Иногда «точка кипения» наступает чуть раньше в одном месте – и тогда приоритет очевиден. Если же несколько претендентов приходят к решению почти одновременно, это оборачивается многолетними спорами и патентными тяжбами.

Тем ярче выглядят такие редкие исключения, когда вперед вырывается одна единственная локальная группа энтузиастов и у нее нет конкурентов. Так случилось в начале 1940-х годов в Школе Мура: здесь был создан первый электронный цифровой компьютер ENIAC, а затем сформулированы архитектурные принципы EDVAC – прямого предшественника всех современных универсальных машин.

Почему же именно Школа Мура оказалась лидером, а не такие авторитетные центры, как MIT, Кембридж или промышленный гигант IBM? Однозначного ответа нет, но можно выделить несколько факторов. Во-первых, относительная свобода от академических догм. Пенсильванский университет не имел такой громоздкой иерархии и административного давления, как более престижные университеты. Это открывало дорогу дерзким замыслам, которые в MIT могли показаться слишком рискованными, а в IBM – подрывающими устоявшийся бизнес.

Во-вторых, военный заказ. Армии США срочно требовались быстрые баллистические расчеты, и это обеспечило проект ресурсами, которых не было у университетских энтузиастов и которые были немыслимы в частной индустрии без военного давления.

В-третьих, важен был предыдущий опыт. Работа с дифференциальным анализатором показала инженерам пределы аналоговых методов. На фоне этого осознания цифровой подход выглядел не просто экспериментом, а реальным выходом из тупика.

В-четвертых, уникальная команда. Моукли с инженерным воображением, Эккерт с практическими решениями, Гольдштейн с организацией и связями, а затем фон Нейман, который придал происходящему интеллектуальную рамку. Подобное сочетание редко встречается в одном месте и в одно время.

Для сравнения: МІТ в это время оставался в русле аналоговых систем, ІВМ продолжала совершенствовать электромеханику и не решалась на радикальный переход, а в Великобритании Colossus, хотя и стал электронным прорывом, был строго секретным и узким по применению. Универсальной машины там не возникло именно из-за секретности и сосредоточенности на узкой военной задаче.

Школа Мура стала примером того, как свобода, инициатива и военное финансирование сошлись в одной точке. Но такие центры не вечны. Как только первоначальная цель достигнута, начинается нормализация: структура бюрократизируется, творческая энергия уходит вместе с людьми, и место пионеров занимают администраторы. Это не исключение, а правило. Так случилось и со Школой Мура после завершения проектов ENIAC и EDVAC. Похожие процессы можно увидеть и в других примерах – например, в Хегох PARC, где в 1970-х появились графический интерфейс, мышь, Ethernet и объектно-ориентированное программирование, но уже в 1980-е годы центр утратил свою уникальность.

Основанная в 1923 году по завещанию Альфреда Фитлера Школа Мура, школа быстро стала центром передовых исследований. Заслуга ее директора Гарольд Пендера в создании благоприятной среды, в которой удалось успешно реализовать идею электронного компьютера. А проведенные по его инициативе летние лекции 1946 года (Moore School Lectures) способствовали передаче знания о проектировании компьютеров и стали катализатором последующих проектов по всему миру. Случившееся в Школе Мура показало, что технологическая революция результат редкого в истории резонанса, когда на стыке науки, практики и личностей рождается новое.

Путь к ENIVAC

В годы Второй мировой войны Школа Мура стала уникальным местом: здесь впервые в истории сошлись военная необходимость и амбиции пионеров цифрового века. Присущая школе ориентация на прикладные задачи оказалась особенно востребованной, когда потребовалось создавать новые вычислительные средства в условиях войны.

В 1942 гоу. Армия США столкнулась с лавинообразным ростом потребности в расчетах баллистических таблиц для новых типов артиллерийских орудий, по этим таблицам артиллеристы подбирали необходимые параметры выстрела. Прежний метод расчета – ручной труд «людей-компьютеров», главным образом женщин, среди которых были будущие программистки ENIAC, – оказался бессилем перед требуемыми сроками и объемами.

Военные обратились за помощью именно в Школу Мура, поскольку здесь еще в 1935 году был построен дифференциальный анализатор – крупнейший механический вычислитель своего времени, применявшийся для расчетов в самых разных областях науки и техники. Созданный по проекту Ванневара Буша, он был настоящим техническим шедевром, но для баллистических нужд оказался слишком медленным и, как любое сложное механическое устройство, капризным.

К этому времени в Школе уже работал Джон Моукли. Он давно задумывался об электронике как об альтернативе механике и в нужный момент представил доклад 1942 года «A Note on the Use of Vacuum Tubes for Computation». В нем он предложил не что-то радикально новое, а электронную версию дифференциального анализатора. Воплотить эту идею не удалось, и от

прямого воспроизведения механики вскоре отказались, но этот замысел стал спусковым механизмом будущих событий.

Эта неудача свела Моукли с 23-летним Преспером Эккертом – одержимым инженером, известным своей маниакальной тягой к точности и надежности. Так возник тандем, которому предстояло войти в историю: Эккерт – рационалист, не выпускающий из рук паяльника, и Моукли – мечтатель, мыслящий образами. Вместе они смогли создать то, что было бы не под силу каждому из них поодиночке. Вскоре они сосредоточили свои усилия на принципиально новом решении, альтернативном дифференциальному анализатору.

В этой истории появился и еще один важный персонаж, сыгравший неоднозначную, но судьбоносную роль. Лейтенант Герман Гольдштейн, представитель армейской лаборатории в Абердине, на первых порах стал верным союзником Моукли и Эккерта. Как математик, он быстро понял масштаб их замысла и взял на себя функции посредника между Школой Мура и армейским командованием. Сделал он это блестяще: именно благодаря его усилиям в 1943 году было подписано соглашение на сумму шестьдесят одна тысяча семьсот долларов. Эта сумма, скромная по меркам военных заказов, дала проекту старт. Машина получила кодовое имя ENIAC (Electronic Numerical Integrator and Computer). Контракт обязывал к строгим срокам, отчетности и армейскому контролю, но при этом управление проектом оставалось за Школой Мура и ее деканом Гарольдом Пендером.

Таймлайн ENIAC – первого в мире универсального электронного компьютера общего назначения – от замысла до снятия с эксплуатации:

- 1942 – физик Джон Моукли предлагает идею электронного дифференциального интегратора.
- Начало 1943 – у Моукли совместно с Преспером Эккертом складывается первое представление о будущей машине.
- Май 1943 – официальное одобрение проекта армией США, решение о финансировании через Бюро баллистики для расчетов артиллерийских таблиц.
- Июнь 1943 – начало работ в Университете Пенсильвании, Moore School of Electrical Engineering, под руководством Джона Моукли и Преспера Эккерта.
- 1944 – изготовление и тестирование компонентов.
- Осень 1945 – завершение физической сборки.
- 10 декабря 1945 – ENIAC полностью закончен.
- 14 февраля 1946 – официальная презентация ENIAC в Филадельфии. Проект вызвал огромный резонанс в научной и военной среде.
- 1946 – ENIAC передан Армии США и установлен в Абердинском испытательном полигоне, штат Мэриленд.
- Использовался для баллистических расчетов, ядерных симуляций в рамках проекта по водородной бомбе, криптоанализа и научных задач.
- 1948 – модифицирован механизм хранения программ, приближение к схеме фон Неймана.
- 2 октября 1955 – ENIAC официально выведен из эксплуатации.

Создание ENIAC

ENIAC представлял собой сложнейший инженерный комплекс. Построить его в 1940-х годах, когда электронная техника только зарождалась, было настоящим подвигом. Поразительно, что собственными силами сотрудников Школы Мура в подвале университета удалось собрать «монстр», состоявший из сорока отдельных шкафов с электроникой, каждый весом примерно шестьсот килограммов. Шкафы были модульными и выполняли разные функции: арифметические операции, управление программой, ввод и вывод данных и другие задачи.

Каждый шкаф имел собственную систему охлаждения – встроенные вентиляторы для отвода тепла от ламп.

Наибольшую техническую проблему составлял низкая надежность электронных ламп. А их было около восемнадцати тысяч, на них строилась вся логика и управление машиной. Они потребляли около сто пятидесяти киловатт, что создавало серьезные проблемы с охлаждением и питанием. Преспер Эккерт предложил простое, но эффективное решение: поскольку лампы не выполняли предусмотренное для них усиление сигналов, а использовались исключительно для логических функций, оказалось возможным понизить напряжение на нитях накаливания. Кроме того, в команде работали специалисты, которые проводили диагностику и меняли лампы «на лету». Благодаря этому, несмотря на сложность машины, удалось довести средний срок наработки всей машины на отказ до десяти – пятнадцати часов.

К моменту завершения работы над машиной война уже подходила к концу, и возникли новые приоритеты. Первым практическим применением ENIAC стало моделирование ядерного взрыва. Прежде такие расчеты занимали бы месяцы ручной работы, а машина справилась за сорок восемь часов непрерывной работы.

Несовершенство «первого блина» было очевидно как с точки зрения эксплуатации, так и еще более с точки зрения способа задания условий решаемой задачи. Сегодня мы называем это действием программированием, но тогда такого понятия не существовало. Текста программы как такового не было: программирование ENIAC являлось физическим процессом, состоявшим из ручной коммутации кабелей и установки в нужные позиции переключателей. До идеи программы, состоящей из последовательности команд, Джон Моукли и Преспер Эккерт еще просто не дошли.

Нам это может показаться странным, однако, размышляя об ENIAC, следует избегать снисходительного отношения к прошлому – того, что по-английски называют *hindsight bias*, или эффектом поздней очевидности – когнитивное искажение, при котором после того, как событие произошло или информация стала известна, прошлое кажется очевидным и предсказуемым, хотя до этого момента результат было трудно или невозможно предугадать.

Об этом эффекте важно помнить, говоря о Джоне Моукли и Преспере Эккерт. Пока другие сомневались, они действовали. Пока остальные обсуждали теорию и спорили о машине Тьюринга, они «на коленке» собрали машину, которая изменила ход истории. Без них история компьютеринга пошла бы другим путем – каким, не нам гадать. И, что самое главное, они сами обнаружили врожденные слабости своей машины и компенсировали их в следующем проекте – EDVAC.

Великий дуэт

Джон Моукли родился в семье ученого-физика Себастьяна Моукли. Его предки эмигрировали из Швейцарии, а необычная фамилия является результатом языковой адаптации немецкоязычной формы к американской среде. Имя читается и как Мокли, и Мочли.

С детства Джон проявлял тягу к знаниям и увлекался изобретениями. Первым из них был контактный датчик на ступенях лестницы, ведущей в детскую. Он включал сигнальную лампочку, когда мать поднималась проверить, спит ли ребенок, и позволял Джону успеть выключить свет и спрятать книгу.

Преподавание и наука в те годы не были доходным делом, поэтому отец порекомендовал более практичное инженерное образование, особенно актуальное после окончания Первой мировой войны. Джон согласился и без труда получил стипендию в Университете Джона Хопкинса в Балтиморе. Однако после двух лет учебы он разочаровался в инженерии. В письмах отцу он сравнивал работу инженера с работой повара: оба действуют по книгам, содержащим рецепты (cookbook). Позже Джон осознал ошибку – он стал великим инженером-творцом.

Продолжив обучение, он переключился на физику и добился заметных успехов, а в двадцать пять лет получил степень Ph.D.

Вскоре Джон обзавелся семьей и двумя детьми. Сочетание эрудиции, глубоких знаний и незаурядной внешности позволило ему стать одним из лучших молодых преподавателей в Ursinus College в Пенсильвании, известном своими либеральными взглядами на образование. Его «коронной» лекцией была научно-популярная рождественская демонстрация, на которой он показывал действие законов физики на эффектных опытах, придуманных им самим. Однажды он вызвал фурор, выехав на сцену на роликовых коньках в русском костюме. Затем он сделал волчок и продемонстрировал, как угловая скорость увеличивается, если исполнитель сжимается.

Научным увлечением Моукли стало применение расчетных методов в метеорологии. Это возникло в связи с появлением подходов к численному моделированию атмосферных процессов в конце XIX века. Не видя перспектив в существующих средствах, Джон задумался о создании альтернативных. В письмах он писал о своем непреодолимом упрямстве в стремлении к цели. Окружающие называли его *underdog*. Этот термин, часто переводимый как «аутсайдер» или «неудачник», точнее обозначает «темную лошадку», способную на неожиданные поступки. Жизнь показала, что Моукли оказался *underdog* именно в этом смысле – как Давид, победивший Голиафа.

Преспер Эккерт, известный как Прес, родился в Филадельфии в состоятельной семье застройщика и владельца недвижимости. Ему было предопределено получить образование, связанное с бизнесом, и войти в высшее общество. Однако природное дарование изменило этот путь. С пяти лет он увлекался радио. В двенадцать лет он победил на городском научном конкурсе, а затем строил самые разнообразные электронные приборы, включая музыкальную систему для местного крематория. В детстве он любил забавные поделки. Венцом его творчества стал шуточный прибор для измерения страстности поцелуев с индикаторной панелью из десяти лампочек.

Сначала Эккерт учился в местной привилегированной бизнес-школе. Не выдержав ее атмосферы, он перешел в Электротехническую школу Мура (Moore School of Electrical Engineering), позже слившуюся с Пенсильванским университетом. В то время школа считалась вторым по уровню подготовки техническим учебным заведением после Массачусетского технологического института. Выбор не в пользу МИТ, вероятно, объясняется нежеланием переезжать в Бостон и жить в кампусе. Прес плохо ориентировался в формальностях повседневной жизни. Однокурсница вспоминала, что на вопрос, почему он всегда ходит в белой рубашке и черном галстуке, он ответил: «Я не знаю, это мне дает мама каждое утро».

После окончания школы в 1940 году Эккерт получает свой первый патент. Дальнейшая его деятельность связана с выполнением военных заказов в области радиолокации, где ему пришлось использовать один из немногих дифференциальных анализаторов конструкции Ванневара Буша. Летом 1941 года он участвовал в американской программе «Инженерное дело, наука и военное управление», организованной Министерством обороны. Именно там он познакомился с Джоном Моукли, который тоже перешел на работу в Школу Мура к аналоговым устройствам.

Так возник великий дуэт, которому предстояло изменить ход истории вычислительной техники. Моукли и Эккерт объединили свои таланты: рационализм и педантичность одного – с мечтательностью и изобретательностью другого, создав команду, способную решать задачи, которые поодиночке были бы непосильны.

Первый электронный компьютер

Если быть точным, то ABC следует признать первым электронным счетным устройством, однако компьютером в полном смысле слова оно не было. Тот факт, что в 1941 году Джон Моукли посетил Джона Атанасова в Университете штата Айова, где Атанасов продемонстрировал ему ABC и использование в ней электронных ламп, дал повод для разговоров о возможном заимствовании. Даже если удастся обнаружить элементы «подсмотренного», их нельзя считать нарушением авторского права. Между двумя машинами дистанция колоссальна: ENIAC создавался как универсальный программируемый компьютер, тогда как ABC был лишь специализированной машиной для решения систем линейных уравнений. Сходство ограничивалось использованием ламповых регистров и последовательных схем. Вне всяких сомнений ENIAC по праву можно считать первым электронным компьютером.

Работа над ENIAC началась в августе 1942 года с внутренней служебной записки Джона Моукли «Использование высокоскоростных устройств на вакуумных лампах для вычислений». В ней он утверждал, что электронные схемы могут выполнять вычисления значительно быстрее, чем любые механические или электромеханические устройства. Моукли предлагал использовать электронные узлы для выполнения последовательности операций – сложения, вычитания, умножения и деления. Однако с архитектурной точки зрения проект оставался близок к механическим вычислителям. Поэтому записка стала концептуальным мостом между ними и электронными цифровыми машинами.

Следуя этому замыслу, сочетая старую архитектуру с электроникой, в начале 1943 года Моукли и Эккерт направили свои усилия на попытку создания цифровой версии дифференциального анализатора Буша. Иначе говоря, цель заключалась в создании «электронной версии» машины для решения дифференциальных уравнений.

Вскоре стала очевидной ограниченность такого подхода: заказчику был нужен универсальный аппарат для любых баллистических расчетов. Осознав это, Моукли и Эккерт пришли к исторически значимому выводу: не следует совершенствовать старое – требуется создавать принципиально новое. И этим новым стала возможность перепрограммирования.

ENIAC еще не был программируемым в современном смысле: в нем не было разделения на аппаратную часть и программу, а программирование сводилось к ручной перенастройке на уровне соединений и переключателей. Однако сама идея универсальности открыла новую эпоху в вычислительной технике. Согласование позиций сторон прошло быстро, и 1 июля проект получил финансирование на разработку Электронного числового интегратора (Electronic Numerical Integrator). Позже к названию добавилось «и компьютера» – так возникло сокращение ENIAC. В целях секретности проект сначала назывался «Белый слон», затем – Project PX.

Менеджером проекта стал Герман Гольдштейн. Он сыграл роль своеобразной «повивальной бабки» проекта. Впоследствии он привлек к работе над ENIAC свою жену Адель, которая сначала была одной из малозаметных программисток, а позже заняла более заметное и активное положение.

При проектировании ENIAC Моукли и Эккерт не знали ни об Аналитической машине Бэббиджа, ни машинах Айкена. Удивительно то, что ее преемник EDVAC, оказалась близок к концепции Аналитической машины и особенно к тому, интерпретировала эту машину Леди Августа Лавлейс. Об этом подробно в книге «История неэлектронных компьютеров».

ENIAC состоял из трех основных подсистем.

- Первая – вычислительная – включала автономные электронные модули сложения, вычитания, умножения и деления.
- Вторая – память для команд и данных – была частично электронной (на лампах), программы и константы задавались с помощью коммутаторных панелей и переключателей.
- Третья – система управления координировала работу вычислительной системы, она выполняла роль программатора.

Существовала также вспомогательная система ввода-вывода на перфокартах и принтеры IBM. ENIAC имел тактовый генератор с частотой около 100 кГц. Его роль отличалась от современных ЦПУ: сигналы генератора координировали микрошаги внутри операции, а переход от одной команды к другой осуществлялся через заранее сконфигурированные сигнальные цепи – так называемое «программирование потока управления через сигнальные соединения».

В июле 1943 года была собрана первая команда разработчиков из двенадцати человек.

Архитектура ENIAC

Архитектура ENIAC уникальна и соответственно программирование тоже. В отличие от позднейших фоннеймановских компьютеров, где программа – это линейная последовательность инструкций в памяти, ENIAC работал как набор специализированных модулей, которые настраивались и соединялись вручную. Программирование представляло собой трехуровневый процесс: проектирование маршрутов данных, определение временной организации вычислений и физическую конфигурацию коммутационных панелей.

ENIAC состоял из отдельных вычислительных блоков, каждый из которых выполнял строго определенные функции. Центральную роль играли двадцать аккумуляторов – десятиразрядных регистров-счетчиков, способных хранить число и выполнять сложение и вычитание. Дополняли их один умножитель, блок деления и извлечения квадратного корня, генераторы констант и устройства ввода-вывода для работы с перфокартами, индикаторами и печатью.

На этом уровне алгоритм представлялся как граф: вершины – вычислительные модули, ребра – каналы передачи данных. Программирование заключалось в «прокладывании маршрутов» чисел между модулями: какое значение хранится в аккумуляторе, куда передается и какой блок его обрабатывает.

ENIAC не исполнял программы линейно. Работа модулей синхронизировалась управляющими импульсами (control pulses), задающими моменты начала и окончания операций. Это позволяло параллельное выполнение: несколько аккумуляторов могли работать одновременно, пока другой блок ожидал результат.

Программа воспринималась как сеть зависимостей и синхронизированных действий, распределенных во времени. Такой подход концептуально близок к современным параллельным вычислениям, конвейерной обработке и распределенным системам.

Программист определял зависимости: «результат аккумулятора 1 отправить в аккумулятор 3, пока умножитель ожидает данные от аккумулятора 4». Алгоритм таким образом представлялся не как список инструкций, а как граф действий с временными ограничениями.

Физически программа фиксировалась в виде конфигурации кабелей и переключателей на панелях. Каждый модуль имел входы и выходы, соединяемые проводами, а режим работы задавался тумблерами: сложение, вычитание, направление передачи данных и т. д.

Состояние машины можно описать матрицей: строки – модули, столбцы – возможные соединения и режимы, ячейки – конкретные настройки. Алгоритм сводился к полной физической схеме, собранной вручную. Эта модель напоминает современные FPGA или телефонные станции середины XX века.

Глава 4. EDVAC

Когда Нил Армстронг ступил на поверхность Луны, он произнес: «Это всего лишь маленький шаг одного человека, но гигантский скачок всего человечества». Да, в 1969 году этот шаг рассматривался не иначе как начало новой эры. Но если оглянуться назад, то окажется, что другой, куда менее заметный «шаг», но более важный был сделан за четверть века до этого. Его не транслировали по телевидению, его не сопровождали многообещающие речи, но его последствия оказались несравнимо масштабнее. Имя этому шагу – EDVAC.

Эта аббревиатура расшифровывается как Electronic Discrete Variable Automatic Computer. Слово «компьютер» к тому времени уже существовало, но именно здесь оно впервые приобрело тот смысл, который мы вкладываем в него сегодня. До этого «компьютерами» называли либо людей-счетчиков, либо громоздкие машины, выполнявшие одну-две специфические функции.

У предшественника EDVAC – ENIAC это слово тоже значилось в названии. Но сам он, при всей своей мощи и грандиозности, не был компьютером в современном понимании. ENIAC не умел хранить программы. Чтобы настроить на новую задачу, инженерам и программистам приходилось буквально перекраивать его устройство: переключать сотни тумблеров, перекладывать кабели, менять панели. Это походило скорее на ремонт, чем на программирование.

Тем не менее именно ENIAC впервые доказал: цифровая электроника годится для вычислений и способна значительно превзойти механические и аналоговые устройства. Но сам он стал скорее доказательством концепции, чем универсальной моделью.

А вот EDVAC – это уже не демонстрация, а настоящий прототип будущего. Он вобрал в себя все, что мы сегодня связываем со словом «компьютер». Его создателями были инженеры Преспер Эккерт и Джон Моукли. На завершающем этапе к ним в качестве внешнего обобщающего их изобретение гуру присоединился Джон фон Нейман, он сумел превратить инженерный проект в строгую теоретическую модель. Позже именно из этой модели вырастет целая новая область знаний – современная информатика.

От ENIAC к EDVAC

ENIAC был удивительной машиной – работа с ним напоминала настройку огромного музыкального органа, где для исполнения каждой новой мелодии нужно было бы вручную переставлять трубы.

Программа в ENIAC – это сплетение кабелей, комбинация переключателей и панелей. Чтобы настроить новую программу, инженеры прокладывали маршруты электрических сигналов: вот вход в умножитель, вот выход в сумматор, а вот дальше результат идет в следующий блок. Получался не набор строк, а целая «электрическая маршрутная карта».

Неудивительно: ENIAC вырос из опыта работы с дифференциальным анализатором, который решал уравнения при помощи шестеренок, валов и дисков. Эккерт и Моукли словно заменили механические шестерни на электронные лампы – но принцип остался тем же: чтобы поменять задачу, нужно было переключить «внутренности».

Очень быстро стало ясно: это тупик. Машина может быть сколько угодно быстрой, но если на ее перепрограммирование уходит неделя, то практической пользы мало. Инженерам нужна была новая идея – способ разделить «железо» и «логику».

Хранимая программа – гениальная простота

Решение оказалось удивительно изящным. Зачем каждый раз перенастраивать машину, если можно просто хранить последовательность операций в ее памяти, машина сама будет читать инструкции одну за другой, словно кулинарную книгу рецептов.

Сама идея памяти была не нова. Еще Жаккар управлял ткацким станком с использованием перфокарт с записанными на них плетениями, а Ада Лавлейс в XIX веке описала для аналитической машины Бэббиджа алгоритмы, где были даже ветвления и циклы. Позже эта идея внешней программы была реализована в гарвардской архитектуре. Во всех этих устройствах программа отделена от процессора.

Эккерт и Моукли сделали радикальный шаг. Они предложили хранить инструкции там же, где и данные. Команда машинного языка стала такой же записью в памяти, как число. Машина могла прочитать ее, изменить или даже переписать. По сути, программа превратилась в хранилище внутреннего состояния машины. Эта простая на первый взгляд идея изменила все. Компьютер впервые стал по-настоящему универсальным.

Система команд

Вместе с памятью для программ появилась и другая новинка – система команд.

Теперь машина понимала формализованный набор команд-инструкций. Каждая команда делилась на две части:

- код операции (что нужно сделать: сложить, вычесть, перейти),
- адреса операндов (с какими числами работать и куда положить результат).

Особым прорывом стали команды условных переходов. С их помощью машина могла менять ход программы в зависимости от результата предыдущей операции: например, «если число ноль – иди туда, если положительное – сюда». В сочетании с безусловными переходами это дало возможность создавать циклы, ветвления и, в конечном счете, сложные алгоритмы.

Важно, что команды и данные находились в одном адресном пространстве. Это позволяло программе «думать» о самой себе – читать, изменять и перезаписывать собственные инструкции. Для будущего программирования это оказалось решающим.

Набор команд EDVAC был ограниченным, но принципиально новым. Впервые появился аппаратный декодер инструкций – устройство, которое по коду операции понимало, что именно нужно сделать. С этого момента общение человека с машиной перестало быть физическим (кабели, тумблеры) и стало логическим.

До EDVAC понятие «язык общения с машиной» в современном смысле отсутствовало. Программирование на ENIAC было больше похоже на инженерный монтаж. Эккерт и Моукли первыми предложили формализованный набор команд, открыв дорогу универсальным компьютерам. Этот шаг и стал тем самым фундаментом, на котором стоит вся современная вычислительная техника.

О Джоне фон Неймане

Двое непосредственных создателей EDVAC были представлены в Главе 3. Джона фон Неймана нельзя назвать третьим – он не создавал машину. Возможно, хотя и не задокументировано, он как консультант оказал некоторое влияние на процесс разработки. Но его историческая роль скорее в другом: он стал популяризатором EDVAC, написав и опубликовав за своим именем так называемый меморандум, или, точнее, обзор «First Draft of a Report on the EDVAC» – собрание умозаключений по поводу проекта будущей машины. В этом документе он раскрыл замысел создания компьютера с хранимой программой, что дало основа-

ние в последующем растиражировать эту схему под названием «Архитектура фон Неймана». На самом деле он не был автором описанной схемы, и, похоже, не претендовал на авторство, хотя по каким-то причинам истинные авторы он не упомянул. Важно отметить: сама «Архитектура фон Неймана», существенно усовершенствованная по сравнению с черновиком First Draft, была реализована в компьютере IAS, о котором будет подробно рассказано в Главе 5.

Урожденный Янош Лайош Нейман использовал дворянскую частицу «von», хотя его принадлежность к дворянству была условной – она было куплено его отцом Микшой Нейманом, венгерским евреем, банкиром и промышленником. Заметим, что Антон Павлович Чехов отказался от дворянства, которое ему было присвоено императором Николаем II. Состоятельная семья фон Неймана сильно пострадала за 133 дня существования Венгерской советской республики, и у юного Яноша тогда сформировались стойкие антикоммунистические взгляды. Позже, уже в США, он стал убежденным противником СССР и одним из самых жестких сторонников холодной войны, всерьез рассматривая возможность превентивных ядерных ударов по Советскому Союзу.

Существует легенда, рассказанная Германом Гольдштейном, о знаменательной встрече фон Неймана с ним, в ту пору молодым лейтенантом, на железнодорожной платформе. Лейтенант якобы раскрыл ему секрет создания компьютеров в инженерной Школе Мура, чем вызвал интерес великого ученого к деятельности небольшой группы инженеров. Однако истинные причины были серьезнее: работа над атомной бомбой требовала огромного объема расчетов, и люди-компьютеры с арифмометрами с ней не справлялись. Нейман изучал все доступное: релейные машины Стибица, машины Эйкена, анализатор Буша, табуляторы IBM и ENIAC. Возможно, Гольдштейн лишь сообщил о новом проекте, что вызвало цепочку событий, получивших в последующем неоднозначную оценку.

Специфические черты личности фон Неймана, позволяли ему мгновенно схватывать суть услышанного, перехватывать инициативу и распространять чужие идеи, не обременяя себя упоминанием авторства первоначальных разработчиков. Он не одинок, среди великих таких немало. Давление авторитета со стороны большого ученого известно как эффект Матфея, описанного в «Притче о талантах» в Евангелии от Матфея. В науке эффект встречается повсеместно, особенно остро ощущался в советской науке.

Тьюринг и фон Нейман

Не упомянуть Тьюринга в повествовании об EDVAC невозможно, потому что почти во всех учебниках Нейману приписывают роль посредника между Универсальной машиной Тьюринга (UMT) и EDVAC, что создает впечатление, будто UMT был непосредственным прототипом EDVAC. Такая трактовка истории дает возможность представить создание программируемого компьютера как академический плод исключительно математической мысли, что выглядит красиво, но лишено какой-либо исторической достоверности. Тьюринг имеет косвенное отношение созданию программируемого компьютера.

Первое знакомство Тьюринга с фон Нейманом было заочным. В 1933 году Тьюринга, тогда еще 21-летнего студента, за успехи в учебе по принятой в Кембридже традиции наградили серьезным научным фолиантом. Совершенно случайно им оказалась книга фон Неймана «Математические основы квантовой механики», на момент публикации которой автору исполнилось всего 30 лет, а у него на счету уже было свыше 50 научных статей. Такое совпадение выглядит не иначе как знак судьбы.

Через пару лет она прозорливо свела их очно во время визита Неймана в Кембридж. После обстоятельной беседы он порекомендовал Тьюрингу пройти стажировку у него в Принстонском университете, и Тьюринг принял приглашение. Учитывая репутацию университета и Института передовых исследований (IAS), ставшего усилиями властей Германии центром интеллектуальной жизни. Здесь работали Алонсо Черч, его ученики Стивен Клейни и Джон

Россер, а в IAS неоднократно бывал Курт Гедель; позже здесь оказались Эйнштейн и Oppenгеймер.

Мог ли выпускник Кембриджа мечтать о большем, чем о таком сообществе коллег? В октябре 1936 года он оказался в кампусе университета, где, помимо основной деятельности, активно занимался спортом – был членом софтбоольной команды, занимался бегом и спуском по реке на каноэ, но с коллегами близко не сходилась и держался отстраненно. Вскоре после приезда из Лондона ему прислали гранки статьи «On Computable Numbers», которую он после правки отправил обратно, и в январе 1937 года статья вышла в свет. В 2008 году ученый совет Принстона признал Тьюринга вторым по значимости выпускником, хотя статья была написана все же в Кембридже. Первым по значимости выпускником Принстона ученый совет признал Джона Нэша, выдающегося математика, лауреата Нобелевской премии по экономике (1994) за вклад в теорию игр. Попутно отметим, что о Нэше «Игры разума» (A Beautiful Mind, 2001), менее известный чем «Игра в имитацию» (The Imitation Game, 2014) о Тьюринге, содержательнее.

Единственным человеком, с которым а Принстоне у Тьюринга возник личный контакт, оказался фон Нейман. Внешне они были полной противоположностью друг другу: фон Нейман – бонвиван, обновлявший гардероб ежегодно, стремившийся стать настоящим американцем, активно демонстрировавший антикоммунистические взгляды; Тьюринг – мрачный по натуре, скромный англичанин, с трудом научившийся водить старый Ford и одетый в поношенную одежду из твида. Контраст был заметен, что не мешало содержательному взаимодействию.

Для фон Неймана странности английского стажера не имели значения. Он предложил Тьюрингу остаться в качестве ассистента с неплохой зарплатой по окончании полутора лет стажировки, но на этот получил отказ. Тьюринг был патриотом и видел угрозу нацистской Германии, писал другу: «Надеюсь, Гитлер не вторгнется в Англию до моего возвращения». Он тогда еще не осознавал своей роли в будущей криптовойне.

На этом научное продуктивное сотрудничество двух гениев завершилось, но идея программируемой машины Тьюринга оставила неизгладимый след в памяти фон Неймана. В описании EDVAC эта теория прямо не использовалась, однако фон Нейман рекомендовал участникам проекта ознакомиться со статьей «Computable Numbers» для расширения кругозора. По окончании войны фон Нейман и Тьюринг до 1949 года еще поддерживали переписку, в которой они расходились во мнении об интеллектуальном потенциале компьютеров: Тьюринг размышлял о том, может ли машина мыслить, а фон Нейман видел в компьютере исключительно средство обработки данных. И, черт возьми, он оказался прав!

Тайм-лайн EDVAC

Работа по проекту EDVAC делится на несколько этапов:

- Хронология развития идеи (август 1944 – март 1945)
- Август 1944. ENIAC близок к завершению, но становятся очевидны его ограничения.

Моукли и Эккерт обсуждают идею хранения программы и данных в памяти, а также возможность автоматического исполнения команд, выбираемых из памяти.

- Сентябрь 1944. Эккерт предлагает использовать ртутные линии задержки в качестве памяти, способной хранить данные и инструкции. Моукли работает над логической моделью и идеей системы команд. Появляется мысль о двоичной арифметике: проще реализовать и требуется меньше компонентов.

- Ноябрь 1944. Вырабатываются основы архитектуры: память, арифметико-логическое устройство, устройство управления, устройство ввода/вывода. Оформляется формат команд (код операции и адреса операндов). Принято решение об унифицированной памяти для дан-

ных и кода и внутреннем дешифраторе команд вместо внешнего коммутатора. Начинается ранний архитектурный дизайн.

- Январь 1945. Проект получает название EDVAC, включающее слова «discrete variable». Ведутся обсуждения формата команд, синхронизации памяти и ALU, реализации дешифратора и идей fetch-decode-execute. Голдштейн подготавливает проектную документацию для армейского заказчика.

- Март 1945. В качестве консультанта подключается Джон фон Нейман. Он не иницирует новую архитектуру, но дает теоретическую интерпретацию, вводит термины, сравнимые с машиной Тьюринга. Голдштейн предлагает фон Нейману оформить идеи в виде локального отчета для армейского заказчика.

- При участии фон Неймана (апрель 1945 – июнь 1946)

- Апрель 1945. Фон Нейман активно работает над теоретическим оформлением архитектуры. Вводится унифицированная терминология: память, арифметическое устройство, устройство управления. Начинает формироваться идея программируемой машины как универсального автомата, устанавливается логическая связь с машиной Тьюринга.

- Май 1945. Появляется черновик текста, позже известный как «First Draft of a Report on the EDVAC». Фон Нейман пишет текст самостоятельно, акцентируя внимание на логической структуре и принципах работы машины, а не на схемотехнике. Сопровождающие с Моукли, Эккертом и Голдштейном корректируют черновики.

- Июнь 1945. Завершается «First Draft». Отчет размножен на mimeографе и распространяется в научной среде, влияя на параллельные проекты в США и Великобритании (например, ACE Тьюринга). Хотя автором указан только фон Нейман, идеи являются коллективной работой.

- Июль—август 1945. Публикация отчета вызывает конфликт: авторство указано только за фон Нейманом, что лишает инженеров возможности патентования архитектуры.

- Сентябрь 1945. Моукли и Эккерт подготавливают свой отчет «Automatic High Speed Computing: A Progress Report on the EDVAC» для армейского заказчика. Он остается секретным, подробно описывает текущий статус проектирования, технические детали и уточненную архитектуру. Отчет подтверждает приоритет Эккерта и Моукли в разработке хранимой программы на инженерном уровне, в то время как фон Нейман остается теоретиком и популяризатором.

- Судьба двух документов складывается по-разному. Отчет фон Неймана быстро распространяется и закрепляет за ним репутацию «отца современной архитектуры компьютера». Отчет Эккерта и Моукли остается секретным и начинает проявлять влияние только позднее – через машины вроде UNIVAC.

- Финал совместной работы (осень 1945 – июнь 1946)

- Осень 1945. Эккерт и Моукли начинают искать возможности создания собственной компании и постепенно отходят от EDVAC. Фон Нейман уходит в Принстонский университет и начинает планировать IAS Machine, логического наследника EDVAC.

- Декабрь 1945 – март 1946. Продолжается техническая реализация EDVAC под руководством Клерка Везерберна: разработка арифметического блока, линий задержки, регистров, управляющей логики и внутренней документации.

- Июнь 1946. ENIAC введен в эксплуатацию, EDVAC все еще опытно-экспериментальный. В Moore School проходит Летняя школа, посвященная архитектуре и проектированию электронных машин. Используются идеи EDVAC, включая стек вызовов и подпрограммы.

- Завершение проекта (осень 1946 – август 1949)

- Осень 1946. Развитие EDVAC затруднено из-за отсутствия финансирования и ухода части команды.

- Август 1948. Начата подготовка технической документации: протоколы загрузки, описание команд, инструкции по замене ламп и тестированию. Формируется группа пользователей-программистов. На ранних этапах программирование как дисциплина почти не поднималось; основное внимание уделялось архитектуре памяти, команд, схем управления и физической реализации.

- Август 1949. EDVAC формально готов к передаче в BRL. Машина реализует архитектуру с хранимой программой и развитым набором инструкций.

Когда теория пытается догнать практику

Современные курсы по информатике и теории алгоритмов практически всегда начинают с машины разбора Тьюринга. Но с исторической точки зрения это в корне неверно. Первые компьютеры создавались вовсе не как реализация теоретических моделей. Никто из инженеров, причастных к ним, не читал и не знал о существовании статьи Тьюринга 1936 года. Они решали прагматические задачи, их целью было ускорение вычислений и не более того. Никто не говорил: «давайте построим физическую реализацию универсального автомата». Это один из тех случаев, когда теория догнала практику, объяснение возникло после явления, а не наоборот и стало ретроспективным основанием, а не исходной мотивацией.

ENIAC, EDVAC, UNIVAC, Manchester Baby, как и ранние компьютеры были реакцией на конкретные потребности – расчет баллистических таблиц, моделирование физических процессов, криптоанализ, атомный проект. Никто из создателей этих машин не задумывался об универсальности. Они действительно могли эмулировать любую вычислимую функцию, но никто этого не планировал, универсальность возникла, как побочный эффект инженерного прагматизма.

Теория вычислений как самостоятельная область сложилась позже. В 1950–60-х годах логики и математики начали систематизировать понятие алгоритма. Появились альтернативы машине Тьюринга: нормальные алгоритмы Поста, рекурсивные функции Клини, λ -исчисление Черча. На этом фоне стало ясно, что компьютер, построенный по принципам, изложенным фон Нейманом, – это, по сути, физическая реализация универсальной машины Тьюринга. Но это был вывод, а не предпосылка. Так же, как Луна не летает вокруг Земли потому, что ее кто-то туда поставил.

Установление параллели между машиной Тьюринга и архитектурой фон Неймана стало делом историков, логиков и специалистов по теоретической информатике. Хронология формирования этой дисциплины такова:

- 1936. Алан Тьюринг и Алонзо Черч независимо предложили модели вычислений – машину Тьюринга и λ -исчисление.

- 1940. Эмиль Пост и Стивен Клини развивают альтернативные формализмы вычислимых функций.

- 1945. Джон фон Нейман пишет доклад о EDVAC, не упоминая Тьюринга. Компьютеры создаются по инженерной логике.

- 1949. Первая рабочая машина с хранимой программой – EDSAC. С нее начинается практическое программирование.

Так, шаг за шагом, формируется представление о компьютере как о материальной реализации машины – якобы это был не инженерный замысел, а интеллектуальный вывод по фактам.

Со временем в образовании прочно укоренился подход «сначала теория, потом практика». И вот уже школьников просят писать программы для машин Тьюринга. В вузах начинают курсы с формальных моделей вычислений, еще до знакомства с реальными языками про-

граммирования. Однако следует иметь в виду, что машина Тьюринга – великая абстракция, следовательно она не предназначена для обучения программированию. Она предназначена для анализа пределов вычислений. Ее место – в финале, как апофеоз теоретического мышления, а не в начале, где

Что же касается универсальности, то она оказалась побочным эффектом. Жизнь показала, что любая достаточно выразительная система становится тьюринг-полной сама собой. (Достаточно выразительная система умеет хранить данные, выполнять условия и повторять действия, благодаря чему способна вычислять любые алгоритмы, как машина Тьюринга). Это значит, что универсальность – это не достижение, а побочный эффект. Если в системе есть возможность хранения переменных, условные переходы, циклы или рекурсия, то тьюринг-полнота появляется почти автоматически. Компьютер стал универсальным не потому, что это было задумано, а потому что иначе он был бы слишком ограничен.

Глава 5. Компьютеры первого поколения

Компьютеры первого поколения производились с 1940-х до середины 1950-х годов, их объединяет общая элементная база – электронные лампы в качестве логических элементов. Машины были громоздкими, потребляли огромное количество электроэнергии, переводя ее в тепло, отличались низкой надежностью. Нарботка на отказ компьютеров первого поколения составляла порядка часов и вероятность отказа была практически неизбежной, поэтому они требовали постоянного технического обслуживания.

Размеры этих машин были колоссальны – они занимали целые машинные залы весили десятки тонн. Программирование осуществлялось только на машинном языке: программы вводились вручную при помощи переключателей или перфокарт, а результаты вычислений выводились на печать.

Оперативная память была на линиях задержки ЭЛТ очень небольшой.

Основное назначение компьютеров первого поколения заключалось в выполнении сложных научных и военных расчетов – например, баллистических, статистических и инженерных. Среди наиболее известных машин этого периода – EDVAC, UNIVAC I, EDSAC, МЭСМ и БЭСМ-1, по непонятной причине забывают об IAS – машине, в полном смысле реализованной по архитектуре фон Неймана и при его участии, хотя по ее образу и подобию было создано порядка 10 машин.

IAS

Обычно в учебниках и популярных изданиях встречается утверждение: «архитектура фон Неймана – это EDVAC». На самом деле собственно фон-неймановская изложена в другом более фундаментальном документе – «Preliminary Discussion of the Logical Design of an Electronic Computing Instrument». По нему была построена машина, названная в IAS в честь Института перспективных исследований (IAS), расположенного в Принстоне, под непосредственным руководством самого фон Неймана, а ее главным разработчиком был Джулиан Бигелу. Он не просто воплотил теоретические идеи фон Неймана, а фактически создал прототип того, что позже стало стандартом архитектуры всех компьютеров: с хранимой программой, общей памятью для данных и инструкций и последовательным исполнением команд.

Термин «архитектура фон Неймана» появился в конце 1950-х – начале 1960-х годов уже задним числом, как ретроним. Он использовался для того, чтобы отличать модель компьютеров с хранимой программой, реализованную в проектах EDVAC/IAS, от альтернативных архитектур, которые начали появляться в тот период.

Изначально идея машины с хранимой программой не носила имени фон Неймана – это был результат совместной работы нескольких исследователей. Но с появлением других подходов к организации компьютеров потребовалось специальное обозначение классической схемы с единым хранилищем для данных и инструкций, последовательным выполнением команд и центральным процессором. Таким образом, термин «архитектура фон Неймана» выполняет ту же функцию, что и другие ретронимы, например «черно-белое телевидение» или «акустическая гитара» – он уточняет старый объект в контексте новых разработок.

Принципиально важным шагом стало то, что IAS разрабатывалась в академическом институте и ее технический проект был опубликован полностью и без каких-либо изъятий. Не было ни патентов, ни попыток коммерческого контроля. По этим причинам архитектура, как бы сегодня сказали, оказалась в public domain, что позволило десяткам исследовательских центров в США, Европе, СССР, Израиле, Австралии и Японии легально создавать ее клоны. В

отличие от EDVAC, который несколько лет оставался скорее прототипом «на бумаге», именно IAS стал образцом для повторения.

Иногда можно встретить странное утверждение, будто IAS была асинхронным или даже параллельным. Это неверно, но том уровне технологий такое было невозможно, работа машины была строго синхронизирована тактовым генератором, до параллелизма его было еще не одно десятилетие.

Основные черты IAS

- Произвольный доступ к памяти. В отличие от последовательных устройств хранения (например, ртутных линий задержки в EDVAC), IAS использовала электронно-лучевые трубки Уильямса, которые позволяли обращаться к любым ячейкам напрямую по адресу.
- Фиксированный формат команд. Каждая инструкция состояла из двух частей: код операции и адрес, что упрощало декодирование и построение управляющей логики.
- Разделение функциональных блоков. Машина имела четко выделенные компоненты: арифметико-логическое устройство (ALU), управляющий блок, память и регистры. Каждый блок имел строго определенные функции.
- Синхронная модульная организация. Все узлы работали под управлением тактового генератора, что обеспечивало надежность и воспроизводимость работы схемы.
- Ясность и простота архитектуры. Модульная структура позволяла легко масштабировать и адаптировать машину, что сделало IAS эталоном для последующих компьютеров 1950-х годов.

Эта архитектура оказалась настолько продуманной, что стала основой для целого поколения машин, включая ILLIAC, MANIAC, ORACLE, WEIZAC, БЭСМ и другие. Хотя эти компьютеры различались по реализации и масштабу, их архитектурный облик был легко узнаваем и восходил к IAS.

UNIVAC – первый коммерческий компьютер

Проект ENIAC в 1945 году застрял на несколько лет и пути его создателей разошлись, Джон Моукли и Преспер Эккерт хотели превратить компьютер в коммерческий продукт, но эта позиция противоречила мнению руководства Университета Пенсильвании: университет требовал, чтобы все разработки и патенты, связанные с ENIAC и будущими машинами, принадлежали ему.

В результате в феврале 1946 года, почти сразу после официальной презентации ENIAC в Принстоне, Моукли и Эккерт подали в отставку и через несколько месяцев зарегистрировали собственную фирму – Eckert-Mauchly Computer Corporation (EMCC). Это было первое предприятие в мире, полностью сосредоточенное на разработке и производстве электронных компьютеров.

BINAC: первая попытка создания коммерческой машины

Первой разработкой EMCC стал BINAC (Binary Automatic Computer, 1949) – машина была задумана как двухпроцессорная система с резервированием: две одинаковые машины работали параллельно и сравнивали результаты, чтобы исключить ошибки. Концепция была смелой, но опередила время – подобные системы высокой надежности реально появились только через три десятилетия.

На практике BINAC часто выходил из строя. Он был собран в единственном экземпляре для Northrop Aircraft, и достоверно неизвестно, работал ли он после установки – скорее всего, нет. Убытки составили почти 200 тыс. долларов. Одной из причин неудачи было то, что созда-

телей из соображений секретности не допустили к машине после поставки, а инженеры заказчика не смогли с ней справиться. Этот опыт стал уроком: сопровождение компьютеров должно было входить в обязательные обязанности – сначала в ЕМСС, а затем и во всей отрасли.

UNIVAC I

После краха BINAC команда ЕМСС, насчитывавшая всего 12 человек, приступила к работе над UNIVAC I. Однако и этот проект оказался под угрозой: в авиакатастрофе погиб главный инвестор, Гарри Страус, глава American Totalisator Company, инженер, создававший электромеханические системы для тотализаторов с конца 1930-х годов. Он был готов инвестировать более 1 млн долларов. После его гибели преемники отказались поддерживать проект, и 1 февраля 1950 года Моукли и Эккерт продали ЕМСС корпорации Remington Rand, что позволило продолжить разработку UNIVAC I.

UNIVAC I стал прямым наследником EDVAC, который все еще оставался исследовательской машиной. В EDVAC впервые реализовали идею хранимой программы, но она имела маленький объем памяти и оставалась медленной. Ввод данных происходил через перфокарты или вручную через переключатели, других устройств ввода/вывода не было – EDVAC был прототипом, проверкой возможности работы компьютера.

UNIVAC I задумывался как практическая, коммерческая машина. Он унаследовал основные черты EDVAC, включая память на ртутных трубках, но был адаптирован для реального мира:

- увеличена память;
- добавлены магнитные ленты для хранения больших объемов данных;
- появилось удобное программирование через перфокарты и ассемблер;
- расширены возможности ввода/вывода – консоль, телетайп, позже принтер.
- Технические характеристики UNIVAC I
- 5 200 электронных ламп,
- вес 13 тонн,
- потребление 125 кВт,
- частота 2,25 МГц,
- сложение – 120 мкс, умножение – 1800 мкс, деление – 3600 мкс,
- память – 1 000 слов по 72 бита на ртутных линиях задержки,
- команда – 2 слова по 36 бит,
- ввод/вывод – восемь магнитофонов, консоль, телетайп, позже – принтер.

Первоначально в UNIVAC I не было перфокарт из-за патентов IBM, но позже Remington Rand разработала 90-колоночную альтернативу. Числа кодировались в двоично-десятичной системе Excess-3, что облегчало работу с отрицательными значениями. Всего было произведено 46 машин, стоимость каждой – около 1,5 млн долларов.

UNIVAC I на выборах 1952 года

UNIVAC I прославился на президентских выборах в США 1952 года, где соперничали Дуайт Эйзенхауэр и Эдлай Стивенсон. В порядке эксперимента телевизионная компания CBS привлекла машину для прогноза результатов. Программу для обработки данных написала Грейс Хоппер, будущий контр-адмирал ВМС США и один из пионеров программирования.

Машина смогла рассчитать распределение выборщиков на основе всего лишь 1 % поступивших данных: примерно 438 за Эйзенхауэра против 93 за Стивенсона. Прогноз выглядел неожиданно – редакторы CBS сначала не решились объявить его в эфире. Итог оказался почти точным: 442 к 89 голосам выборщиков. Эта демонстрация превратила UNIVAC в символ

надежности и современности, открыв путь к использованию компьютеров в бизнесе и статистике.

UNIVAC II и последующие модели

В 1958 году вышел UNIVAC II, промежуточная версия между UNIVAC I и будущими моделями. Идея оставалась прежней, но машина стала быстрее и надежнее, она сохранила совместимость с программами UNIVAC I, а архитектура во многом соответствовала IAS фон Неймана. Основные новшества: повышение производительности, надежности и улучшение устройств ввода/вывода.

UNIVAC II по-прежнему использовал ртутные линии задержки, переход на память на магнитных сердечниках произошел только с UNIVAC III, что обеспечило увеличение скорости и надежности. Было выпущено около 50–60 машин. Машины UNIVAC II конкурировали с IBM 700-й серии, которая закрепила лидерство на рынке научных и бизнес-вычислений.

Whirlwind I: первый компьютер реального времени

Особое место среди компьютеров первого поколения занимает Whirlwind I, разработанный в Лаборатории цифровых компьютеров MIT. Проект продолжал работы с тем же названием, начатые еще во время Второй мировой войны, когда возникла необходимость создать имитатор полетов, а также решать задачи стабилизации поведения самолетов и повышения точности бомбометания. Директором проекта был Джей Форрестер, его заместителем – Роберт Эверетт.

Изначально создавался аналоговый компьютер, который успешно справлялся с поставленными задачами, но он оказался крайне сложным при перепрограммировании для разных типов самолетов. В 1947 году, ознакомившись с ENIAC, Форрестер и Эверетт пришли к выводу, что решением проблемы может быть программируемый цифровой компьютер.

Работой над цифровой версией занималась команда из 175 человек. Первый экземпляр был собран в декабре 1950 года, а в апреле 1951 года машина была введена в эксплуатацию. Whirlwind создавался как лабораторный стенд и с самого начала был адаптирован для работы в реальном времени, что было уникально для начала 1950-х.

В качестве устройства ввода/вывода впервые использовался экран на электронно-лучевой трубке для визуализации процессов управления. Машина могла выполнять до 20 000 операций в секунду, что для начала 1950-х было значительным показателем.

Первоначально использовалась память на электронно-лучевых трубках Уильямса, обеспечивавшая быстрый доступ к данным с малой емкостью. В 1953 году ее заменили на магнитно-ядерную память на ферритовых сердечниках, вдвое увеличив производительность системы.

Архитектура и принципы работы

- Программа и данные находились в единой памяти;
- Инструкции выполнялись последовательно (fetch-decode-execute);
- Модульная конструкция позволяла расширять возможности машины и адаптировать ее под новые задачи.

В сочетании с памятью на сердечниках и работой в реальном времени Whirlwind стал прототипом всех управляющих компьютеров.

Влияние на последующее поколение компьютеров

Проект Whirlwind воспитал новое поколение инженеров. Среди них был великий Кен Олсен, который участвовал в разработке цифровых систем МИТ и позже создал TX-0 (Transistorized Experimental Computer Zero) – первую транзисторную машину МИТ, архитектурно наследующую идеи Whirlwind, но более компактную и удобную. Опыт, полученный на Whirlwind позволил Олсену в 1957 году основать Digital Equipment Corporation (DEC), корпорацию, занимавшую долгое время второе место в мире.

Все последующие модели DEC – от PDP-1 до легендарного VAX – сохраняли черты Whirlwind:

- модульность;
- реактивность;
- ориентацию на управление и взаимодействие с пользователем.
- Основные достижения и значение Whirlwind I
- Первый компьютер, работающий в реальном времени;
- Первая машина с графическим выводом на экран;

IBM: от калькулятора до империи

В 1930–1960-е годы корпорация IBM была столпом информационной инфраструктуры США. Ее технологии обеспечивали средствами обработки данных всех – от бизнеса до государственных структур. Томас Уотсон-старший, бессменный лидер компании на протяжении полувека, сформулировал кредо IBM так: «IBM – это не просто бизнес. Это глобальная инсти-туция, установленная навечно».

Уотсон был человеком эпохи электромеханических табуляторов и обладал монополией на перфокарты формата 80 колонок. Его отношение к вычислительной технике формировалось на этом опыте и не смог сразу поверить в ее потенциал. До своей частичной отставки в 1952 году Уотсон проводил независимую техническую политику, кульминацией которой стало решение 1946 года о создании Selective Sequence Electronic Calculator (SSEC). Руководил проектом Фрэнк Гамильтон, ранее совместно с Говардом Эйкенем работавший над Harvard Mark I.

SSEC – «калькулятор для витрины»

Проект SSEC находился под личным контролем Уотсона, который настойчиво называл устройство «калькулятором», привычно оставляя слово computer за «людьми-вычислителями». Машина сочетала электромеханическое наследие IBM и начальные наработки в электронике. SSEC был установлен в зале на первом этаже штаб-квартиры IBM на Мэдисон-авеню, в Нью-Йорке, три стеклянные стены машина открывали вид с улицы и SSEC мгновенно стала городской достопримечательностью. Она представляла собой стойки, расставленные буквой «П», общей длиной 55 метров, содержащие 21 400 реле и 12 500 электронных ламп. В машинном зале SSEC впервые применили фальшполы, скрывавшие кабельную разводку, а футуристический дизайн вдохновил кинематографистов.

Несмотря на визуальную эффектность, практическая отдача от машины была скромной. Главным приложением стали расчеты эфемерид Луны, которые пригодились в 1970-х в рамках программы «Аполлон». Были и внешние заказчики, но низкая надежность системы помешала широкому применению. Программист Джон Бэкус, будущий создатель Fortran, вспоминал: «Вы должны были присутствовать при выполнении вашей программы, потому что машина останавливалась каждые три минуты, и никто, кроме вас, не знал, с чего продолжить». Преспер Эккерт лаконично резюмировал: «SSEC превзошел пределы гигантизма – и не мог работать».

SSEC был установлен в январе 1948 года, а демонтирован – в июле 1952-го. На его месте IBM представила IBM 701 – первый серийный компьютер компании, выпущенный в количестве 19 экземпляров.

IBM 701 – рождение настоящего компьютера

1952 год стал поворотным для IBM. Корпорацию возглавил Томас Уотсон-младший, сумевший осознать угрозу со стороны UNIVAC и других электронных машин, способных обрабатывать данные с перфокарт значительно быстрее. IBM подошла к вопросу системно: входить в электронику, но с гарантированной выгодой. Начали с военного заказа – первым компьютером с хранимой программой стал Defense Calculator, позже переименованный в IBM 701.

Проект возглавил Натан Рочестер, ранее работавший над арифметическим устройством Whirlwind, его заместителем был Джерри Хаддад, впоследствии создавший массовую IBM 650. В роли научного консультанта выступал Джон фон Нейман, и неудивительно, что архитектура IBM 701 оказалась IAS-подобной.

В отличие от универсального UNIVAC, IBM 701 проектировалась под конкретные задачи – прежде всего решение дифференциальных уравнений в частных производных, необходимых для авиации и обороны. Для оперативной памяти использовались трубки Уильямса.

Работа над машиной началась в январе 1951 года и завершилась весной 1952-го. Презентация в стеклянном зале была столь же торжественной, как и SSEC, но теперь речь шла о машине для обработки данных (Electronic Data Processing Machines), подчеркивая ее практическую значимость. Гражданская версия под индексом IBM 702, начиная с 14-го экземпляра, оснащалась памятью на ферритовых сердечниках – революция в надежности и производительности.

Развитие серии IBM 700/7000 продолжалось более десяти лет. В этих машинах появлялись новые архитектуры, поддержка плавающей точки, переход с ламп на транзисторы, но программная несовместимость между моделями сохранялась. Только с появлением в 1964 году серии System/360 IBM создала единое совместимое семейство, объединившее промышленность, науку и бизнес под одним архитектурным стандартом.

IBM100 – SAGE

Стимулом к созданию системы управления ПВО SAGE (Semi-Automatic Ground Environment) стал испытательный взрыв первой советской атомной бомбы 29 августа 1949 года, являвшейся копией американского «Толстяка».

Появлению SAGE предшествовала созданная силами ВВС США сеть наблюдательных постов за воздушными объектами – Ground Observer Corps (GOC). В 1953 году в ее составе действовало более 8 тысяч постов, где участвовали свыше 305 тысяч добровольцев. Однако GOC не имела связи с системами раннего обнаружения самолетов противника, и ее решили заменить «Полуавтоматической наземной вычислительной средой» – SAGE.

На выбор стратегии создания SAGE повлияла система взглядов Ванневару Буша, формулировавшего правила взаимодействия государства, науки и бизнеса при решении крупных государственных задач. Буш разработал их, будучи руководителем National Defense Research Committee во время войны и в послевоенные годы. Позже он стал советником по науке президента Дуайта Эйзенхауэра.

Принципы Буша частично применялись в Манхэттенском проекте, но наиболее полно – в SAGE, а также при создании национальных лабораторий и координирующих ведомств вроде NASA и DARPA. Главная особенность гибкой научно-производственной модели Буша – ориентация на доверие к специалистам и на технологии двойного назначения, в отличие от жестко административной «оружейной» модели СССР.

Головной организацией стал MIT, а не министерство. Массачусетский технологический институт привлек частные компании в качестве субподрядчиков.

SAGE состояла из 23 центров, каждый оснащался компьютером A/N FSQ-7 (Army-Navy / Fixed Special eQuipment). Он был построен на базе незавершенного проекта Whirlwind II MIT и доработан IBM.

Компьютер A/N FSQ-7:

- вес – 250 тонн;
- потребление – 3 МВт;
- число вакуумных ламп – 60 тысяч;
- обслуживался около 100 операторами;
- к одному компьютеру подключалось до 50 мониторов, что позволяло одновременно отслеживать около 400 самолетов;
- данные поступали от сотен радаров и других источников, машина интегрировала их в единую картину воздушного пространства.

Согласно экспертным оценкам, стоимость системы SAGE составляла 8–12 млрд долларов, что сопоставимо с Манхэттенским проектом.

Субподрядчики распределились следующим образом:

- IBM – аппаратное обеспечение;
- Burroughs – коммуникационное оборудование;
- Линкольновская лаборатория MIT – системная интеграция;
- RAND Corporation – программное обеспечение.

Первый управляющий центр SAGE вступил в действие в ноябре 1956 г., последний – в 1962 г. Для системы было написано 500 тысяч строк кода. Надежность впечатляет: SAGE выключалась всего на 10 часов в год.

SAGE была первой системой с полноценным человеко-машинным интерфейсом. Каждая станция управления имела экран с электронно-лучевой трубкой, на котором отображалась текущая воздушная обстановка.

Оператор мог взаимодействовать с изображением с помощью светового пера – устройства, позволяющего указать на объект на экране и:

- запросить параметры;
- передать координаты;
- инициировать перехват.

Эта технология стала предтечей графического интерфейса пользователя (GUI): курсор, клики, визуальная обратная связь и обработка событий в реальном времени. Появление GUI задолго до Xerox PARC и Macintosh показывает, что потребность в интуитивном управлении сложными системами возникла намного раньше, чем массовая персонализация компьютеров.

SAGE стал технологическим и культурным прецедентом: впервые показано, что человек может взаимодействовать с компьютером напрямую – в диалоге, а не через бумагу или перфокарты.

SAGE стал первым технически реализованным прообразом государственной системы как информационной машины, где наблюдение, моделирование, анализ и реакция объединены в одном технологическом контуре. Он не просто анализировал данные – система принимала решения и управляла действиями (сил ПВО) в реальном времени.

SAGE доказал работоспособность кибернетической утопии управления: если государство – система, его поведение можно анализировать, прогнозировать и направлять при достаточном уровне технологий.

Британский путь в компьютеринге

Великобритания начала свою компьютерную историю блестяще. Она не пошла по американскому пути массового производства машин, а выбрала иной вектор развития: умные архитектуры, оригинальные алгоритмы и смелые концептуальные решения. Английские компьютеры часто рождались не ради рынка, а ради идей. В этом кроется их особое очарование.

И хотя США быстро вышли в лидеры индустрии, британская школа сохранила самобытность. Acorn Computers с BBC Micro и предшественниками ARM, Sinclair Research с компактными ZX Spectrum, Autonomy с байесовской аналитикой данных, ARM Holdings и ее архитектуры для миллиардов чипов, а также Darktrace с алгоритмами киберзащиты – все это продолжение традиции, в которой ценность измеряется не серийностью, а интеллектом.

От Colossus к «национальному компьютеру»

Опыт Блетчли-парка, полученный при создании криптоаналитической машины Colossus стал катализатором британского старта. В отличие от США, где в дело сразу включились университеты и промышленность, в Великобритании инициатором выступило Министерство снабжения армии (Ministry of Supply). В 1945 году, еще до публикации знаменитого американского меморандума о EDVAC, оно предложило создать British national computer – национальный вычислитель.

Исполнителем выбрали Национальную физическую лабораторию (NPL), относительно небольшое учреждение, которым руководил сэр Чарльз Дарвин, родственник великого естествоиспытателя. В команду вошли лучшие умы Блетчли-парка – Макс Ньюман и Алан Тьюринг. Ньюман, игравший ключевую роль в организации работы над Colossus и в математической подготовке специалистов Блетчли, предложил идею вычислительного центра как общенационального ресурса. Тьюринг разработал проект Automatic Computing Engine (ACE), универсальной общенациональной машины. В нем ощущалось математическое вдохновение, но и некоторая инженерная мегаломания.

Проект выглядел внушительно: полная ACE должна была содержать около 6 000 ламп, работать с памятью на 1 000 слов по 32 бита, а скорость выполнения операций оценивалась в 1 миллион операций в секунду – показатели, поражающие для середины 1940-х. Однако быстро выяснилось, что идея слишком амбициозна. Ньюман вскоре ушел в Манчестер, где развернул собственную линию работ, Тьюринг взял годовой отпуск и работал в Кембридже и Манчестере. К 1947 году от идеи «British national computer» отказались. Главной причиной стала экономика: страна, разрушенная войной, не могла позволить себе столь масштабный проект.

Упрощенный вариант, Pilot ACE, содержал около 800 ламп, использовал ртутные линии задержки общей емкостью 352 слова по 32 бита и достигал скорости до 1 миллиона инструкций в секунду – фактически он стал одним из самых быстрых компьютеров своего времени. Но программировать его было крайне сложно. Тем не менее, машина работала до 1957 года, а ее промышленный вариант – DEUCE (Digital Electronic Universal Computing Engine) – выпускался English Electric (35 экземпляров) и содержал уже около 1 450 ламп.

Манчестерская инженерная школа

В Манчестере в дело включился именно Макс Ньюман, перешедший сюда из NPL. Он привел с собой опыт Блетчли и понимание организационной стороны крупных проектов. Вместе с инженерами Фредериком Уильямсом и Томом Килбурном он основал школу, где математика и инженерия соединились.

Уильямс, специалист по радарам, предложил использовать электронно-лучевые трубки не для вывода изображения, а для хранения данных: люминофор сохранял след, а луч мог его обновлять и считывать. Вместе с Килбурном и Джеффом Тутиллом он в 1946 году начал эксперименты, а к 1948 году построил SSEM (Small Scale Experimental Machine) – «Baby».

Машина содержала 550 ламп и могла хранить 32 слова по 32 бита. Ее первая программа из 17 команд проработала 52 минуты.

Следующим шагом стал Manchester Mark I, в полной версии имевший около 4 200 ламп, память на 128 слов и скорость порядка 1 миллисекунды на операцию. В нем использовались «длинные слова» (40 бит), а также впервые применялись индексные регистры. Эта машина стала основой для коммерческого компьютера Ferranti Mark I (1949), содержавшего более 4 000 ламп и память на 2 048 слов. На нем была реализована ранняя система программирования, а Тони Брукер создал язык Autocode.

Дальнейшее развитие линии: Mercury (1957, 2 000 слов памяти на магнитных барабанах, скорость до 60 000 операций в секунду), а затем Atlas (1962) – самый быстрый компьютер своего времени. Atlas содержал около 90 000 транзисторов, оперативную память на 16 000 слов по 48 бит и барабанное хранилище на 96 000 слов. Именно здесь впервые была реализована концепция виртуальной памяти. В 1970-е появились MU5 и MU6, ставшие прототипами мэйн-фреймов ICL.

Кембридж и рождение бизнес-компьютинга

В Кембридже работу возглавил Морис Уилкс. Получив доступ к меморандуму фон Неймана, он создал проект EDSAC (Electronic Delay Storage Automatic Calculator), заверченный в 1949 году. Машина содержала около 3 000 ламп, память на 512 слов по 17 бит на ртутных линиях задержки и выполняла порядка 650 инструкций в секунду. Это был первый по-настоящему полезный компьютер с хранимой программой.

Неожиданным союзником оказалась чайная империя J. Lyons & Co. Компания имела собственные подразделения инженеров и математиков и стремилась автоматизировать бизнес-процессы. Руководители Джон Симмонс и Реймонд Томпсон сразу оценили потенциал компьютеров. Они отправились в США, встречались с Германом Гольдштейном и вернулись с твердым решением внедрять вычислительные машины для управления производством.

Сначала Lyons финансировала проект EDSAC, но затем пригласила бывшего аспиранта Уилкса Джона Пинкертон для создания собственной машины. Так появился LEO I (Lyons Electronic Office), запущенный в 1951 году. Он состоял из 21 шкафа, содержал около 6 000 ламп и 64 ртутные линии задержки на 2 048 слов. Тактовая частота достигала 500 кГц, а скорость – около 600 инструкций в секунду. Его эксплуатация произвела сенсацию: это был первый в мире бизнес-компьютер.

Компания LEO Computers выпустила обновленные версии: LEO II (1957, до 10 000 ламп, 4 096 слов памяти) и LEO III (1962, транзисторная, память на 8 192 слова). Всего было произведено около 150 машин, установленных по всему миру, включая Австралию, Южную Африку и даже Чехословакию.

Наследие британского пути

История британских компьютеров первого поколения – это чередование амбициозных проектов и прагматичных решений.

- ACE и его наследники показали силу идей Тьюринга, но и ограниченность чисто математического подхода.
- Манчестерская школа Ньюмана, Уильямса и Килбурна создала основу инженерной традиции и вывела Великобританию на уровень коммерческого производства.
- Кембриджская линия подарила миру бизнес-компьютинг через LEO.

Хотя позже Британия уступила США в гонке объемов и скоростей, ее влияние оказалось интеллектуальным. От ранних идей до современных ARM и Darktrace проходит прямая линия: британский путь – это умные архитектуры, новые алгоритмы и смелые концепции.

Сегодня уже невозможно представить мир без ARM-процессоров, работающих в миллиардах устройств. Но их корни уходят в 1945 год, когда в NPL обсуждали создание «British national computer». Тогда, в послевоенной Британии, родилась традиция, в которой ценность измеряется не числом выпущенных машин, а качеством идей.

Это и есть подлинное наследие Алана Тьюринга, Макса Ньюмана и их соратников: компьютеринг как интеллектуальное приключение, а не только как индустрия.

Барабанные машины

В 1950–1970-х годах штат Миннесота был «тайной Силиконовой Долиной», не случайно его называют «Землей 10 000 самых засекреченных компьютерных проектов» (The Land of 10,000 Top-Secret Computer Projects). Третий компьютерный кластер после территории между Сан-Франциско и Сан-Хосе и окрестностей Бостона, растянувшихся вдоль Шоссе 128, он стал сначала резиденцией для компаний Univac, Control Data, IBM, Honeywell, а позже и для других, причастных к тому, что делалось в целях обеспечения национальной безопасности в период холодной войны. Флер таинственности сыграл не лучшую роль в развитии региона; сейчас делается многое для восстановления его статуса, но на этот раз упор делается на развитие медицинских технологий.

Здесь в 1946 году Уильям Норрис, позже возглавивший Control Data Corporation, и Говард Ингстрем (1902–1962), математик и криптограф, завершивший карьеру на посту заместителя директора Агентства национальной безопасности (АНБ), создали компанию Engineering Research Associates (ERA). Они собрали в ней несколько десятков специалистов для создания машин, способствующих раскрытию шифров, прежде всего советских. Им предстояло построить две такие машины – Goldberg и Demon, аналогичные более известному Colossus. В этом деле они не были одни: существовала целая индустрия, производившая подобные машины. Goldberg, Demon и другие описаны в книге Стивена Будянски «Борцы с кодами: Взломщики шифров АНБ и секретная война разведки против Советского Союза» (Code Warriors: NSA's Codebreakers and the Secret Intelligence War against the Soviet Union).

Третьей машиной стал Atlas, построенный в 1950 году; в нем впервые был использован магнитный барабан для программы и данных, который выполнял и функции оперативной памяти. На основе Atlas были созданы коммерческая версия ERA 1101 и секретная Atlas II. Вскоре Sperry Rand приобрела компании ERA и Eckert—Mauchly Computer Corporation с ее UNIVAC I, образовав компьютерное подразделение Sperry-UNIVAC. Недовольные этим, во главе с Норрисом, создали более успешную Control Data Corporation, среди них был великий Сймур Крей, основатель Cray Computers. ERA 1101 была переименована в UNIVAC 1101, а в обновленной версии 1954 года барабан был заменен ферритовой памятью.

По мотивам UNIVAC 1101 компанией Computer Research Corporation (CRC) была создана еще одна барабанная машина – CRC 102A. Она была небольшой и недорогой по тем меркам и удобной для математических расчетов. Однако проще и дешевле была машина LGP-30, созданная Стэнли Френкелем, чрезвычайно одаренным человеком: будучи аспирантом Роберта Оппенгеймера, он участвовал в Манхэттенском проекте и писал программы для расчета атомной бомбы на ENIAC. Он сумел придумать машину, состоящую всего из 119 двойных триодов и 1350 диодов. При весе 450 килограмм ее считали настольной, а стоимость по нынешнему курсу составляла чуть более 400 тысяч долларов. Не удивительно, что ее продали в количестве пятисот штук. В каком-то смысле ее можно считать предтечей мини-компьютеров.

Относительная простота барабанных машин открывала простор для творчества. Пример личного вклада показал Гарри Хаски, студент, которому повезло поработать в команде Моукли и Эккерта над проектами ENIAC и EDVAC, а затем еще год с Аланом Тьюрингом в NPL, когда тот создавал Pilot ACE. В 1953 году он работал в Университете Уэйна в Детройте и параллельно

в компании Bendix, где возглавил разработку компьютера G-15, ставшего более удачной реализацией замысла Pilot ACE, чем то, что сделал сам Тьюринг. Со сложностями программирования справились программы INTERCOM 1000, прообраз будущих операционных систем, и компилирующая программа Algo, значительно опередившая появление языка Algol; она позволяла вводить формулы, не зная устройства машины. G-15 комплектовался богатым набором периферийных устройств: от обычных для ввода и вывода перфокарт и телетайпа до специализированного сопроцессора – разностного анализатора и рулонного графопостроителя. Все это богатство поддерживалось процессором из 450 ламп и диском, способным хранить 2160 29-битовых слов. Иногда G-15 называют первым персональным компьютером.

В начале 1950-х IBM начала активно входить в мир электронной обработки данных. Пока в известной серии компьютеров IBM 700 разрабатывались на предприятии компании в Паук-кипси, Нью-Йорк, лаборатория в Эндикотте, штат Нью-Йорк, вносила свой вклад в историю информационных технологий – создавая передовую машину, которая в конце 1950-х называлась «рабочей лошадкой современной промышленности».

Самым массовым из барабанных компьютеров стал IBM 650, анонсированный в 1953 году. Название Magnetic Drum Data Processing Machine подчеркивало отличие от выпускавшихся тогда машин серии 700. Они служили для научных расчетов, а IBM 650 предназначался для обработки данных и может рассматриваться как первая попытка заменить табуляторы, занимавшие монопольное положение в государственных учреждениях и бизнесе. О нем говорили: «Это жизненно важный фактор продвижения машин с хранимой в памяти программой в индустрию и в бизнес». Успех превзошел все ожидания: вместо прогнозируемых 500–700 к 1962 году было произведено 2000 IBM 650. Ориентация на барабанную память оправдалась: машина оказалась достаточно надежной для операций с ограниченным объемом вычислений. Барабан диаметром четыре дюйма и длиной шестнадцать дюймов вращался со скоростью 12 500 оборотов в минуту и мог хранить 20 000 чисел, разделенных на 2 000 фрагментов. Для доступа к ним имелась специальная таблица. На протяжении всех лет выпуска IBM 650 машина комплектовалась новой периферией, в том числе магнитными лентами и дисками.

Глава 6. Второе поколение ЭВМ

К середине 1950-х годов вычислительная техника вошла в фазу переосмысления содержания. Было ясно, что ламповая природа и отсутствие технологий для памяти в машинах первого поколения лишали их будущего, при том, что сами архитектурные принципы – последовательная обработка данных, программное управление, структура из функциональных блоков (память, процессор, устройства ввода-вывода) – показали себя эффективными. Эти принципы могли быть перенесены в конструкции второго поколения почти без изменений, особых новаций не предполагалось. В этом смысле машины второго поколения казались скучнее, чем машины первого.

На этом этапе перед инженерами встала задача не переизобрести компьютер, а используя доступные новые технологии, сделать его быстрее, надежнее и практичнее. Архитектура оставалась знакомой: арифметико-логическое устройство, устройство управления, память, шины обмена, периферийные устройства. Однако уже в это время появились первые попытки стандартизации интерфейсов и модульного построения. ЭВМ начали проектировать не как уникальные изделия, а как модели в линейке со сменными или масштабируемыми блоками. Яркий пример – элементная база «Урал-10», позволявшая строить на ее основе универсальные машины «Урал-14» и «Урал-16», а также управляющие машины.

Кроме того, появились в зародыше операционные системы, хотя до полноценных ОС оставались годы. Например, использовались мониторы пакетной обработки (batch monitors), которые управляли запуском программ с перфокарт и контролировали порядок выполнения заданий без участия оператора. Одна из них Мониторная система «Дубна» была разработана в Объединенном институте ядерных исследований (ОИЯИ) в Дубне в начале 1960-х годов. Она предназначалась для автоматизации вычислений на советских ЭВМ второго поколения, в первую очередь на машинах типа БЭСМ-2, М-20 и БЭСМ-4.

Появление языков программирования Fortran, Algol и Cobol, ассемблеров и первых трансляторов ускорило написание программ и сделало вычислительные машины более доступными для прикладных специалистов. ЭВМ становилась не просто инструментом расчетов, а многофункциональной вычислительной системой.

Начало второго поколения вычислительной техники связывают с переходом от электронных ламп к транзисторам. Этот шаг оказался не просто инженерным улучшением: транзисторы сделали компьютеры надежнее, компактнее и доступнее. Первые образцы показали, как быстро может измениться представление о вычислительной машине.

- TRADIC (1954, Bell Labs, США) – один из первых экспериментальных компьютеров, в котором использовались почти исключительно транзисторы. Он показал, что новая элементная база может заменить лампы и при этом потреблять в сотни раз меньше энергии. Хотя машина была специализированной и не получила широкого распространения, именно с нее начался практический путь к транзисторным ЭВМ.

- TX-0 (1956, MIT, США) – первый полностью транзисторный универсальный компьютер. Построенный в Массачусетском технологическом институте, он был относительно мал по размерам и стал удобной исследовательской платформой. На TX-0 программисты впервые опробовали новые методы взаимодействия с машиной, включая использование дисплея и клавиатуры – черты, которые позже станут привычными.

- Philco Transac S-2000 (1957, США) – первая серийная коммерческая ЭВМ, выполненная на транзисторах. В отличие от лабораторных экспериментов, эта машина уже предназначалась для клиентов и стала символом перехода индустрии к массовому выпуску «второгопоколенных» компьютеров.

Революция в элементной базе и памяти

Транзисторы

Самым важным техническим рубежом, отделяющим второе поколение ЭВМ от первого, стала новая элементная база. На смену вакуумным лампам пришли транзисторы – компактные полупроводниковые приборы, обладавшие целым рядом преимуществ: они были надежнее, работали без нагрева, потребляли меньше энергии, не требовали сложного питания и занимали в десятки раз меньше места.

Ламповая ЭВМ – это, как правило, громоздкая конструкция, требующая систем вентиляции и регулярного технического обслуживания. Отказ даже одной лампы мог остановить всю машину. Транзисторы же обеспечили радикальное повышение отказоустойчивости и снижение эксплуатационных издержек.

Сравним по надежности ламповый UNIVAC I (1951) и транзисторный UNIVAC III (1962).

- UNIVAC I Средняя наработка на отказ (МТBF) составляла 100–200 часов. Причины отказов: лампы перегревались и выгорали, соединения и контакты реле изнашивались, возникали проблемы с питанием и охлаждением. Особенности эксплуатации: требовалась постоянная замена ламп, техническое обслуживание было почти непрерывным.

- UNIVAC III (1962). МТBF на порядок выше 1000–2000 часов и более. Причины отказов в основном связаны с периферией, питающими цепями и редко с транзисторами. Транзисторная конструкция позволила уменьшить тепловыделение, повысить скорость и стабильность работы, снизить потребность в постоянном обслуживании.

Ферритовая революция

Наряду с арифметикой и логикой, огромный шаг вперед был сделан в области оперативной памяти. Машины первого поколения использовали довольно экзотические и ненадежные методы хранения данных: линии задержки, ртутные трубки, электростатические накопители. Все эти решения были медленными, сложными и, как современная полупроводниковая память, требовали постоянного питания.

Ферритовая память – изобретение, основанное на использовании крошечных магнитных колец – стала технологическим прорывом. Каждый феррит хранил один бит, а состояние (0 или 1) определялось направлением намагниченности. Главное преимущество: ферритовая память была энергонезависимой – данные сохранялись даже при отключении питания. Кроме того, она отличалась высокой скоростью (до десятков микросекунд доступа) и долговечностью.

Например, в БЭСМ-6, одной из самых известных советских ЭВМ, использовалась ферритовая память с временем доступа около двух микросекунд. Такая скорость по тем временам считалась выдающейся и позволяла выполнять десятки тысяч операций в секунду.

Ферритовая память стала стандартом индустрии на два десятилетия. Даже с появлением полупроводников ее продолжали использовать – настолько она оказалась удачным компромиссом между стоимостью, скоростью и надежностью.

Внешняя память: магнитная эра

Второе поколение ЭВМ ознаменовалось также переходом от сугубо ручного ввода данных к автоматизированным средствам. Перфокарты и перфоленты, конечно, еще широко использовались много лет. К тому же появились и другие новые периферийные устрой-

ства: алфавитно-цифровые печатающие устройства, световые табло, управляющие консоли. Машины стали взаимодействовать с человеком не только через механические устройства, но и через специализированные программные интерфейсы.

Одновременно развивалась система внешней памяти. Магнитные ленты, барабаны и диски дали возможность хранить программы и данные вне оперативной памяти. Это открыло путь к хранению больших массивов информации, к созданию библиотек подпрограмм и первым попыткам реализации файловых систем.

Первый серийный дисковый накопитель IBM 305 RAMAC (1956) весил более тонны, содержал 50 дисков по 24 дюйма и вмещал всего пять мегабайт данных – объем, меньше обычной фотографии на современном смартфоне. Накопители второго поколения были медленными и громоздкими, но они сделали возможным то, что сегодня кажется само собой разумеющимся: работу с базами данных, сохранение результатов вычислений,

Компьютеры новых типов

Основную массу компьютеров второго поколения составили мейнфреймы, о них ниже, но облегченные по сравнению с ламповыми транзисторные технологии открыли возможность для создания экспериментальных компьютеров новых типов, они нашли свое продолжение позже с появлением микросхем. Упомянем ставшие наиболее известными и вошедшими в историю.

TRADIC, первый шаг в эпоху транзисторов

В 1954 году инженеры Bell Labs представили опытный компьютер TRADIC – одну из первых в мире машин, построенных почти полностью на транзисторах. Изначально проект создавался для Военно-воздушных сил США, где его задача заключалась не в массовых вычислениях, а в демонстрации возможностей новой элементной базы. TRADIC должен был показать, что транзисторы способны заменить громоздкие и ненадежные электронные лампы, лежавшие в основе всех предыдущих компьютеров.

Машина состояла примерно из 800 транзисторов и около 10 тысяч диодов, что по тем временам было настоящим техническим подвигом. По производительности TRADIC уступал ламповым мейнфреймам, но его главное преимущество заключалось в энергопотреблении и надежности. Вместо десятков киловатт, которые требовали ламповые компьютеры, TRADIC потреблял лишь несколько сотен ватт. Эта экономия энергии и уменьшение размеров позволили представить вычислительную технику в совсем новом свете: компактной и потенциально более доступной.

Инженеры столкнулись с целым рядом вызовов: транзисторы того времени были еще не полностью стандартизированы, их характеристики отличались от единицы к единице, а схемотехника приходилось проектировать практически «с нуля». Несмотря на эти сложности, TRADIC успешно выполнял свои расчетные задачи, что стало первым убедительным доказательством работоспособности транзисторной логики в реальных условиях.

Хотя TRADIC так и не превратился в массовый продукт, он сыграл ключевую роль в истории вычислительной техники. Он показал, что компьютеры могут быть меньше, надежнее и проще в обслуживании, закладывая основы второго поколения ЭВМ. Можно сказать, что TRADIC стал символом переходного этапа – от ламповых гигантов к машинам, которые вскоре перестанут быть экзотикой и станут частью лабораторий и исследовательских центров.

TX-0: лаборатория будущего

В 1956 году в Массачусетском технологическом институте (MIT) появился TX-0 – первый полностью транзисторный универсальный компьютер. В отличие от TRADIC, который

был демонстрационным образцом для военных, TX-0 создавался как рабочий инструмент для научных и учебных экспериментов.

Машина занимала несколько шкафов, но выглядела компактной по меркам того времени. Ее уникальной особенностью был дисплей на основе осциллографа и клавиатура, что позволило операторам взаимодействовать с компьютером напрямую. Это было настоящим прорывом: вместо пакетной обработки на перфокартах пользователи могли сразу видеть результат своих команд на экране.

TX-0 быстро стал «игровой площадкой» для инженеров и студентов. На нем пробовали создавать первые графические программы, текстовые редакторы и даже музыкальные синтезаторы. Именно здесь зародилась культура «игры с компьютером», которая позже получила название «хакерская». Машина показала, что транзисторный компьютер может быть не только средством расчетов, но и платформой для творчества и экспериментов.

Philco Transac: коммерческая реальность

В 1950-х компания Philco активно разрабатывала транзисторы и в 1957 году выпустила Transac S-2000 – первую серийную коммерческую ЭВМ на транзисторах, сделав значительный вклад в начало второго поколения компьютеров. В отличие от лабораторных образцов, эта машина была рассчитана на практическое использование: ее могли приобрести университеты, исследовательские центры и предприятия.

Transac использовал новые высокоскоростные диффузионные транзисторы, что делало его быстрее и надежнее большинства ранних машин. Машина была способна выполнять разнообразные научные и инженерные задачи, а также обрабатывать большие объемы данных. Этот компьютер стал первым ярким примером того, что транзисторная техника выходит из лабораторий в массовое применение.

Transac доказал, что транзисторные компьютеры – это не только экспериментальные устройства, но и надежная рабочая техника, способная заменить ламповые мэйнфреймы. Его выпуск ознаменовал начало эпохи второго поколения ЭВМ, когда новые технологии стали стандартом для промышленности и науки.

PDP-1: машина, с которой началась культура «хакеров»

В 1960 году в университетских лабораториях Массачусетского технологического института появился новый гость – PDP-1. В отличие от привычных мэйнфреймов, занимавших целые залы и требовавших армии операторов, этот компьютер выглядел почти «домашним»: несколько шкафов, пульт управления, а главное – подключенный к нему дисплей.

Инженеры компании Digital Equipment Corporation хотели доказать, что вычислительная техника может быть не только монстром для оборонных заказов, но и инструментом для исследователя. PDP-1 стоил около 120 тысяч долларов – по меркам ЭВМ это было дешево. Поэтому он стал доступен университетам, небольшим фирмам и лабораториям.

Но главное – PDP-1 оказался первым компьютером, с которым можно было работать вживую. В то время как на больших машинах задания обычно подавались пачками перфокарт и ответ приходилось ждать часами, здесь можно было сидеть за пультом и видеть реакцию машины мгновенно.

Так родилась новая культура – культура людей, которым хотелось не только «сдать задачу в вычислительный центр», но и самим играть с машиной, исследовать ее возможности, пробовать невозможное. Позднее таких людей назовут «хакерами» – не в сегодняшнем смысле, а как энтузиастов, влюбленных в компьютеры.

На PDP-1 они создали первые программы для компьютерной графики, музыкальные эксперименты и даже игру – знаменитую Spacewar! (1962). На экране, напоминающем осцилло-

граф, крутились космические корабли и стреляли друг в друга – для того времени это было чудом, почти фантастикой.

PDP-1 навсегда вошел в историю как компьютер, который показал: вычислительная машина может быть инструментом творчества, а не только строгим счетчиком чисел.

Мэйнфреймы второго поколения

На пару лет позже, в конце 1950-х появились и мэйнфреймы второго поколения – универсальных машин, предназначенных для решения широкого круга задач.

Само слово мэйнфрейм (от англ. main frame – «основная рама») изначально не обозначало тип компьютера. Так называли металлический шкаф, в котором размещались основные модули машины. Однако с течением времени этот термин стал нарицательным: так начали называть все большие универсальные компьютеры, обслуживавшие десятки и сотни пользователей. Они стали символом эпохи централизованных вычислений – от научных центров до банков и министерств.

Мэйнфрейм – это не просто большой компьютер. Это сложный вычислительный комплекс, рассчитанный на интенсивную эксплуатацию, зачастую круглосуточную. Он включал центральное устройство, модули памяти, блоки ввода-вывода, а также периферийные устройства – устройства чтения перфокарт, магнитные ленты, печатающие устройства. Управление происходило не интерактивно, а в пакетном режиме, пакеты, включавшие задания, программы и данные. подготавливали на перфокартах, а затем запускали одно за другим.

По сравнению с первыми машинами, мэйнфреймы второго поколения обладали заметно большей надежностью, высокой скоростью работы (сотни тысяч операций в секунду) и расширенными возможностями для хранения и обработки данных. Объем оперативной памяти составлял от нескольких до десятков тысяч слов (чаще всего 36-битных), в качестве постоянной памяти использовались магнитные ленты и барабаны, позже – диски.

Основные области применения мэйнфреймов второго поколения: в науке для численных расчетов, при проектировании авиационной и космической техники, в промышленности для планирования производства, расчета нагрузок, автоматизации управления технологическими процессами. И разумеется, в оборонной сфере – для моделирования разного рода атак, для расчета баллистических траекторий, для обработки данных с радаров и создания систем автоматического управления. На Западе, особенно в США, со второго поколения началось внедрение мэйнфреймов и в деловой сфере – банки, страховые компании, крупные торговые сети.

Мэйнфреймы как отдельная ветвь эволюции

Поначалу мэйнфреймы не выделялись как особый класс машин. Это были просто самые большие, мощные и дорогие компьютеры, предназначенные для сложных расчетов и обслуживания крупных организаций. Однако уже в конце 1960-х годов стало ясно: мэйнфреймы пошли по собственному пути развития, отличному от остальных ветвей компьютерной техники.

В отличие от будущих мини-компьютеров, ориентированных на локальные задачи, или персональных ЭВМ, рассчитанных на индивидуального пользователя, мэйнфреймы продолжали развиваться как машины коллективного, массового обслуживания. Их главными чертами стали:

- Централизация ресурсов: один мощный компьютер обслуживал десятки или сотни терминалов.
- Надежность и отказоустойчивость: мэйнфреймы проектировались для работы круглосуточно и без выходных, с резервированием критических узлов.
- Масштабируемость: систему можно было расширять и наращивать по мере роста потребностей.

- **Управляемость:** сложные диспетчерские механизмы, контроль доступа и планирование загрузки задач.

На базе этих требований возникла особая операционная среда – сначала OS/360, затем MVS, а позже – z/OS. Появились собственные подходы к программированию, администрированию, построению сетей и защите данных. Так, уже в 1970-е годы мэйнфреймы стали не просто техникой, а целой экосистемой с собственными языками, стандартами и профессиональными сообществами.

IBM и «семь гномов»: история о доминировании

Во времена второго поколения на рынке США сформировалась четкая иерархия производителей. Главной силой была IBM, державшая в своих руках до 70 % мирового рынка вычислительной техники. На рынке вычислительной техники США сформировался интересный феномен, получивший полушутливое название «семь гномов». Под этим прозвищем объединяли семь компаний, конкурировавших с IBM: Burroughs, Control Data Corporation (CDC), General Electric (GE), Honeywell, NCR, RCA и UNIVAC (часто вместе с Sperry Rand). Иногда в список включали Philco, Raytheon или Digital Equipment Corporation (DEC).

Название возникло из маркетингового фольклора и подчеркивало монополистическое превосходство IBM: компания контролировала диктовала стандарты и правила, а конкуренты, несмотря на свои технологические достижения, выглядели «мелкими» на ее фоне. Каждая из компаний-гномов занимала свою нишу: например, DEC была новатором в мини-компьютерах, CDC – в суперкомпьютерах, NCR – в системах автоматизации кассовых операций, Honeywell – во встроенных и авиационных системах. Ни одна из них не могла бросить серьезный вызов IBM на массовом рынке.

К началу 1980-х многие «гномы» либо были поглощены (например, Burroughs и Sperry Rand в Unisys), либо ушли в смежные области: GE сосредоточилась на промышленной автоматике, NCR – на кассовых системах, Honeywell – на авиации и безопасности.

Этот феномен имеет прямую историческую параллель в современной цифровой экономике. Акроним FAANG (Facebook/Meta, Apple, Amazon, Netflix, Google/Alphabet) описывает несколько гигантских технологических компаний, вокруг которых строится вся остальная экосистема стартапов и более мелких игроков. Например, стартапы вроде Snap (создатели Snapchat) или Spotify вынуждены интегрироваться в экосистему Google и Apple, используя их платформы для распространения и монетизации. Аналогично, в эпоху «семи гномов» стартапы и малые фирмы вроде DEC создавали технологические новшества, но их рыночная мощь была ограничена масштабами IBM.

Таким образом, «семь гномов» и FAANG – это разные эпохи одной и той же закономерности: технологические гиганты формируют рынок, концентрируют влияние и создают среду, в которой остальные участники обречены быть малыми или нишевыми игроками, пока не удастся создать альтернативную платформу или интегрироваться в экосистему.

Как IBM подавила конкурентов

Масштаб производства и агрессивная ценовая политика: IBM могла позволить себе продавать машины дешевле себестоимости (или с очень низкой наценкой), зная, что окупится на сопровождении, аренде и программном обеспечении. Малые фирмы просто не выдерживали этой конкуренции.

- **Доминирование в бизнесе и государственном секторе:** IBM первой активно продвигала мэйнфреймы в корпоративную среду – банки, страхование, производство. Эти отрасли «прилипли» к экосистеме IBM, и конкуренты уже не могли войти в эти ниши.

- **Вертикальная интеграция:** Компания контролировала весь цикл – от аппаратной части до программного обеспечения и сервисного обслуживания. Это обеспечивало надежность и совместимость, что особенно ценила бизнес-аудитория.

- **Маркетинг и обучение:** IBM вкладывалась в подготовку специалистов, создавала учебные курсы, методические материалы и даже финансировала кафедры в университетах. Это создавало лояльную базу кадров, которые с самого начала ориентировались на IBM.

- **Модель аренды, а не продажи:** Клиенты не покупали машины – они арендовали их, и тем самым попадали в зависимость от обновлений, обслуживания и программного обеспечения IBM. Это создавало эффект «замкнутого круга» и повышало стоимость перехода к конкуренту.

Модель доминирования IBM является примером того, как мощная корпоративная стратегия может задавить даже сильных технических конкурентов. К середине 1960-х годов IBM была не просто производителем ЭВМ – она стала экосистемой, в которую встроилось большинство клиентов, поставщиков и специалистов. Остальные могли существовать лишь как нишевые игроки или экспериментаторы.

Почему мэйнфреймы выжили?

Когда в 1980-е годы появилась волна персональных компьютеров, многие предрекали скорый конец мэйнфреймам. Но они не исчезли – напротив, нашли свою нишу, особенно в таких сферах, где требуются:

- высокая производительность;
- гарантированная надежность;
- долговременное хранение и обработка критических данных (например, банковские транзакции, государственные реестры, телекоммуникации).

Именно поэтому многие из крупнейших банков, авиакомпаний, страховых и государственных учреждений до сих пор используют мэйнфреймы, хотя снаружи это уже совсем не шкафы с лампочками, а современные стойки, порой неотличимые от серверов.

IBM 1401: компьютер, который пришел в офис

В 1959 году компания IBM представила компьютер, которому суждено было стать одной из самых массовых машин второго поколения – IBM 1401. Это был не мощный суперкомпьютер, не машина для ядерных расчетов и не мозг системы ПВО. IBM 1401 была рождена для гораздо более прозаичных задач – обработки платежных ведомостей, расчета налогов, инвентаризации, автоматизации канцелярии. И именно в этом заключалась ее революционность.

В отличие от предшествующих моделей, ориентированных на научные и военные применения, IBM 1401 проектировалась как «компьютер для бизнеса» – не слишком дорогой, достаточно мощный и в то же время понятный пользователю, далекому от программирования. Главным архитектором машины был инженер Фрэнсис Андервуд, работавший в исследовательском центре IBM в городе Покипси, штат Нью-Йорк. Он видел задачу не в том, чтобы создать очередной шедевр вычислительной техники, а в том, чтобы компьютер стал таким же привычным офисным инструментом, как калькулятор или печатная машинка.

Именно простота и доступность сделали IBM 1401 коммерческим успехом. В IBM даже не рассчитывали на большой спрос: изначально предполагалось продать около тысячи машин. Однако реальность превзошла ожидания – было поставлено более десяти тысяч экземпляров, а в некоторые периоды выпуск достигал одной машины в день.

Архитектурно 1401 во многом отличалась от своих «старших» собратьев. Во-первых, она использовала не двоичную, а десятичную логику (BCD – binary-coded decimal), что идеально подходило для обработки чисел в бухгалтерии, а также для хранения текстовых данных. Во-вторых, память имела переменную длину слова, что позволяло экономить ресурсы и эффек-

тивно работать с переменными по длине строками. Каждая запись завершалась специальным «маркером слова» (word mark), что упрощало организацию данных.

Машина была ориентирована на обработку символов, строк и таблиц, то есть на то, что сегодня назвали бы текстовой аналитикой. Она легко справлялась с печатью документов, считыванием и сортировкой перфокарт, формированием отчетов. Это был инструмент, который мог понять человек из бухгалтерии – и это было важнее, чем его вычислительная мощность.

Программирование велось на языке Autocoder – это был символический ассемблер, достаточно простой в освоении, с макрооператорами и мнемониками. Позже для IBM 1401 были адаптированы и более «человеческие» языки – COBOL и RPG, рассчитанные на бизнес-аналитиков и операторов, а не на инженеров.

Огромную роль сыграла и техническая политика IBM. Машину можно было арендовать – за сравнительно скромную по тем временам сумму (около 2500 долларов в месяц), с полным пакетом услуг: установкой, обучением, сопровождением. Более того, машина собиралась модульно: заказчик мог выбрать конфигурацию под свои задачи – объем памяти, тип накопителей, количество устройств ввода и вывода. Это делало IBM 1401 невероятно гибким решением, особенно для тех, кто переходил на нее с механических систем.

Большую роль сыграла и совместимость с существующей инфраструктурой: IBM обеспечила полную поддержку стандартных 80-колоночных перфокарт, старых принтеров и табуляторов. Это был не резкий технологический скачок, а аккуратная эволюция, рассчитанная на постепенную цифровизацию офиса.

На практике IBM 1401 использовалась в самых разных организациях – от банков и страховых компаний до железных дорог и госпочты. Она стояла в университетах, на складах, в органах статистики, в корпоративных архивах. Во многих учреждениях это была первая цифровая система, заменившая ряды механических счетных машин и картотек.

Историческое значение IBM 1401 трудно переоценить. Это был компьютер, который впервые массово попал в руки не ученых и не военных, а бухгалтеров, логистов, чиновников. Вокруг него сложилось настоящее сообщество пользователей: появились клубы, курсы, методички, типовые программы. Это была не просто машина – это была экосистема, предвосхищавшая то, что десятилетия спустя сделают Apple и Microsoft.

IBM 1401 можно справедливо назвать символом перехода от элитарных вычислений к массовым. Не мощность, а практичность, простота и масштабируемость сделали ее самым успешным коммерческим компьютером своего времени.

Глава 7. Мэйнфреймы – от третьего поколения до наших дней

Компьютеры первого поколения были построены на электронных лампах. Это были исключительно большие машины – мэйнфреймы, занимавшие целые залы. Они потребляли огромное количество электроэнергии, сильно грелись, часто выходили из строя и требовали штата инженеров для обслуживания.

Второе поколение сменило лампы на транзисторы. Благодаря этому компьютеры стали надежнее, быстрее и немного компактнее, но в целом оставались все теми же большими системами – мэйнфреймами. Их по-прежнему устанавливали в вычислительных центрах, научных институтах, банках, на предприятиях.

Третье поколение стало поворотным – с появлением интегральных схем вычислительная техника разделилась на два класса. С одной стороны, продолжали развиваться мощные мэйнфреймы, обслуживавшие крупные организации. С другой – возникли мини-компьютеры: заметно меньшие по размеру и цене, рассчитанные на лаборатории, отделы, университеты. Мини-компьютеры стали промежуточным звеном между большими системами и будущими персональными ЭВМ. Поэтому разделим рассказ о третьем поколении на две главы, одна будет посвящена хорошо известным мэйнфреймам, а другая – новому классу – мини-компьютерам.

Интересно то, что на третьем поколении развитие мэйнфреймов не прекратилось, хотя четвертого уже не было. В последующем примерно с десятилетним интервалом приходили на смену старым новые семейства, но радикально менялось не многое. И это несмотря на то, что с 1980-х годов мэйнфреймы много раз «хоронили» – сначала в пользу мини-компьютеров, затем персональных ПК, позже серверов и облаков, но они, точнее IBM, всякий раз находили способ выживания. Секрет прост: мэйнфреймы постоянно эволюционировали вместе с индустрией. Микроэлектроника, микропроцессоры, виртуализация, современные системы управления базами данных – все это было освоено и встроено в архитектуру мэйнфреймов. Они не остались в прошлом, а шагнули в цифровую эпоху.

В отличие от мэйнфреймов, для мини-компьютеров их поколение фактически стало первым и последним: до появления интегральных схем компактные вычислительные машины просто не существовали, а после этого они быстро эволюционировали в сторону персональных компьютеров нового Эпохи. Эти мини открыли новую нишу – небольшие, относительно дешевые и доступные системы для лабораторий, отделов и предприятий. Хотя сами машины были скромны по размеру и мощности, третье поколение для них было богато событиями: впервые появились интегральные схемы, поддержка операционных систем и стандартизированные архитектуры, что заложило основу будущей компактной вычислительной техники. Им будет посвящена отдельная глава.

Мэйнфреймы третьего поколения

Система IBM System/360 считается классическим представителем третьего поколения компьютеров, прежде всего потому, что она воплотила в себе характерные для этого этапа развития вычислительной техники принципы. Ключевым отличием третьего поколения стало использование интегральных схем вместо отдельных транзисторов, что позволило существенно уменьшить размеры процессоров и модулей памяти, повысить надежность и снизить энергопотребление. В System/360 применялись малые и средние интегральные схемы, благодаря чему инженеры смогли создавать более сложные вычислительные устройства, одновременно обеспечивая стабильность работы и меньшую склонность к отказам.

Помимо аппаратного прогресса, System/360 впервые предложила унифицированную архитектуру, при которой все машины линейки – от самых маленьких моделей для офисов до крупных вычислительных центров. IBM впервые предложила единое семейство совместимых машин – от младших до старших моделей, с одинаковой архитектурой команд (ISA, Instruction Set Architecture). Благодаря этому программы, написанные для одной модели System/360, можно было запускать на другой без переписывания – несомненное преимущество для того времени. Это дало возможность масштабировать мощности без переписывания программ, унифицировать операционные системы и драйверы устройств, а также внедрять многозадачность и виртуальную память. В отличие от машин второго поколения, где каждая система была уникальной, и перенос программ между ними был практически невозможен, System/360 создавала ощущение единой вычислительной экосистемы.

Эта линейка активно использовала высокоуровневые языки программирования, такие как COBOL и FORTRAN, а также первые крупные операционные системы вроде OS/360. Эти системы умели управлять несколькими одновременно выполняемыми задачами, изолировать процессы и координировать работу различных устройств ввода-вывода. Подобный уровень программно-аппаратной интеграции был характерен именно для третьего поколения, когда вычислительная система перестала быть просто набором транзисторов, а стала цельным организмом, состоящим из железа и софта, работающих согласованно.

Кроме того, в System/360 была впервые реализована стандартизация периферийных устройств. Диски, ленточные накопители, терминалы и принтеры могли подключаться ко всем моделям линейки одинаково, без переписывания драйверов и изменения программ. Ранее каждое устройство было уникально и работало только с конкретной машиной, что сильно ограничивало универсальность и масштабируемость.

Таким образом, System/360 воплотила все признаки третьего поколения: она использовала интегральные схемы, предлагала унифицированную архитектуру с совместимым набором команд, поддерживала операционные системы и высокоуровневые языки, а также стандартизировала периферию. Это позволило IBM впервые создать линейку универсальных, масштабируемых и совместимых машин, которые могли работать как отдельные устройства и одновременно формировать единое вычислительное пространство. Именно за эти свойства System/360 истории вычислительной техники относят к третьему поколению.

Рождение IBM System/360

В начале 1960-х IBM столкнулась с критически важной проблемой: компания выпускала множество отдельных моделей компьютеров с несовместимыми архитектурами и программным обеспечением. Клиентам приходилось переписывать программы при переходе на более мощные машины, а поддержка каждой модели обходилась дорого.

Для выхода из положения было решено объединить разнообразные вычислительные системы в единый стандарт, который позволял бы масштабировать мощности без переписывания программ. Стратегическая инициатива исходила от Томаса Уотсона-младшего, который видел необходимость рационализировать продуктовую линейку и обеспечить конкурентоспособность компании. Однако конкретная архитектурная концепция, набор инструкций, стандарты совместимости, операционные системы и проект периферийных устройств создавались и вырабатывались командой инженеров, ученых и архитекторов:

- Фред Брайант и его группа архитекторов отвечали за совместимость моделей;
- Джон Бэкус и другие специалисты разрабатывали стандарты программирования и набор инструкций;
- Десятки инженеров, математиков и экспертов по периферийным устройствам работали над масштабируемостью, надежностью и многопользовательской архитектурой.

Замысел состоял в том, чтобы построить архитектуру, способную покрыть весь рынок вычислительных машин. Это и нашло отражение в названии System/360 – намеке на полный охват вычислительного пространства. Линейка стала революцией: единая архитектура, совместимость программ между моделями, общие языки и операционные системы, возможность роста производительности без переписывания кода. Таким образом, System/360 – это продукт командной инженерной работы, в которой стратегическая рамка задавалась руководством, а все технологическое содержание и инновации исходили от инженеров и ученых IBM.

От идеи к железу: реализация System/360

После того как инженеры IBM определили цель – создать единый стандарт, охватывающий весь спектр вычислительных задач, – перед ними встал ряд конкретных проблем:

- Совместимость моделей разного уровня мощности. Команда должна была спроектировать архитектуру так, чтобы программы для «малых» моделей работали на «больших» без изменений. Это потребовало унификации набора инструкций (Instruction Set Architecture, ISA) и стандартов работы с памятью, вводом-выводом и периферией. Фред Брайант и его группа архитекторов разработали единый набор команд, который включал арифметические, логические и управляющие операции, а также инструкции для работы с потоками данных и устройствами.

- Масштабируемость и производительность. В линейке System/360 существовали модели от относительно скромных 360/30 и 360/40 до мощных 360/75 и 360/195. Инженеры использовали модульную конструкцию процессоров, позволяющую добавлять процессоры, расширять память и подключать новые устройства без перепроектирования всей системы. Для этого применялись блоки общей архитектуры, которые могли наращиваться как строительные блоки.

- Управление памятью и виртуализация. Одним из прорывов стало внедрение систем управления памятью и виртуальных адресов. Это позволило запускать несколько программ одновременно (многозадачность) и обеспечивать изоляцию процессов. Работу этих систем проектировали совместно инженеры по аппаратной архитектуре и специалисты по операционным системам.

- Периферийные устройства и стандартизация интерфейсов. Впервые IBM стандартизировала подключение дисковых накопителей, лент, принтеров и терминалов. Это позволяло клиенту без переделки ПО или аппаратуры подключать новое устройство. Команда инженеров по периферии разработала унифицированные интерфейсы и протоколы обмена данными.

- Программная поддержка. Джон Бэкус и группа разработчиков языков программирования создавали стандарты для COBOL, FORTRAN и ассемблера под System/360. Важной задачей было, чтобы программы на высокоуровневых языках корректно работали на всех моделях линейки. Это обеспечивало полную совместимость и минимизировало затраты клиентов на переписывание ПО.

В 1964 году IBM представила 360/30 и 360/40 – малые и средние модели, и 360/65 и 360/75 – более мощные варианты для крупных организаций. В 1965–1966 году линейка расширилась до самых производительных моделей, включая 360/195, способные выполнять миллионы операций в секунду. Каждая модель использовала один набор инструкций, единые стандарты памяти и ввода-вывода, но отличалась производительностью и количеством подключаемых модулей.

Основные трудности при создании IBM System/360

Создание принципиально новой архитектуры было необычайно сложной задачей. Требовалось объединить требования разных типов клиентов: малых и средних предприятий, крупных корпораций и научных центров. Каждая категория имела свои потребности – бизнес нуж-

дался в точных и надежных вычислениях, научные центры – в высокой производительности для сложных расчетов. Объединить все это в единой архитектуре, не ограничивая возможности мощных моделей и не перегружая малые, было инженерным вызовом.

Не меньшей проблемой оказалось определение набора команд. Команды компьютера задают его функциональность, и каждая прежняя модель имела собственный набор инструкций. При проектировании System/360 нужно было разработать универсальный компромиссный Instruction Set Architecture, который был бы достаточно богатым для сложных вычислений, но при этом не усложнял бы малые модели и не делал их медленными и дорогими. Найти баланс между универсальностью и эффективностью оказалось крайне непросто.

Одновременно инженеры сталкивались с задачей управления памятью и поддержки многозадачности. В крупных организациях уже тогда возникала потребность запускать одновременно несколько программ, обеспечивая при этом изоляцию процессов. Разработчикам пришлось создать систему виртуальной адресации и управления памятью, которая работала бы одинаково на всех моделях линейки и позволяла каждому процессу видеть «свою» память, независимо от физического объема ресурсов. Это было особенно сложно с учетом технических ограничений того времени: интегральные схемы только появлялись, дисковые и ленточные накопители были медленными, а интерфейсы ввода-вывода ограничены.

Не меньше усилий потребовала разработка масштабируемой аппаратной конструкции. Модели System/360 должны были различаться по мощности, но использовать одни и те же архитектурные принципы. Инженеры создали модульные процессоры и блоки памяти, которые можно было наращивать по мере необходимости, обеспечивая плавный переход от малых машин до крупных вычислительных центров.

Особое внимание уделялось периферийным устройствам. До System/360 каждая машина имела свои ленты, диски, принтеры и терминалы, что делало расширение и модернизацию сложными и дорогими. Для новой системы нужно было разработать унифицированные интерфейсы и протоколы обмена данными, которые позволяли подключать любую периферию ко всем моделям.

И, наконец, сама архитектура была бы бесполезна без программной поддержки. Программы должны были работать на всех моделях линейки без изменений. Для этого инженеры IBM синхронизировали разработку аппаратной части с созданием операционных систем и стандартизировали языки программирования COBOL, FORTRAN и ассемблер. Такое параллельное проектирование «железа» и софта было новым подходом для того времени и требовало тесного взаимодействия различных групп специалистов.

Только преодолев все эти трудности – разработав единую архитектуру, стандартизовав набор команд, спроектировав масштабируемые процессоры и модули памяти, унифицировав периферию и создав универсальные операционные системы – команда IBM смогла перейти к конкретной линейке моделей System/360, которая объединила малые, средние и крупные системы в единую экосистему, задав стандарты для всей индустрии.

Принципиальные новшества System/360

Создание System/360 требовало не просто улучшения существующих технологий, а разработки принципиально новых подходов как в аппаратной архитектуре, так и в программной поддержке.

Прежде всего, инженерам IBM пришлось впервые в истории объединить разноуровневые модели в единую архитектурную платформу. Прежде каждая машина проектировалась отдельно, и каждая имела собственный набор команд. System/360 ввела концепцию универсального набора инструкций, одинакового для всех моделей, что позволило запускать одну и ту же программу как на малой, так и на сверхмощной машине. Это была революционная

идея совместимости «по вертикали» – масштабируемости программного обеспечения вместе с ростом мощности аппаратуры.

Второе принципиальное новшество касалось управления памятью и многозадачностью. В System/360 впервые была реализована система виртуальной адресации и разделения памяти между процессами, что позволяло запускать несколько программ одновременно и изолировать их друг от друга. До этого большинство компьютеров работало с физической памятью напрямую, и любая ошибка в одной программе могла повредить данные другой. Новая концепция стала предвестником будущих систем виртуализации, которые сегодня лежат в основе серверов и облачных вычислений.

Третье – масштабируемая модульная архитектура процессоров и блоков памяти. Для того чтобы разные модели линейки отличались по мощности, но оставались совместимыми, инженеры придумали принцип «строительных блоков» – процессоры, память и устройства ввода-вывода проектировались так, чтобы их можно было наращивать. Это было совершенно новым подходом: до System/360 инженеры просто проектировали отдельную машину под каждую задачу.

Четвертое новшество – унификация периферийных устройств и интерфейсов. Впервые IBM стандартизовала работу дисков, лент, терминалов и принтеров, чтобы они одинаково работали со всеми моделями. Ранее каждая машина имела свои устройства, и смена оборудования часто означала дорогостоящую переделку программ. Унификация позволила создавать общую экосистему железа и ПО.

Наконец, принципиально новым был параллельный подход к разработке аппаратуры и программного обеспечения. Для System/360 инженеры и разработчики языков программирования работали вместе с архитекторами процессоров, синхронизируя набор инструкций, операционные системы и языковые стандарты. Это позволяло обеспечить полную совместимость между железом и софтом, что было беспрецедентно.

В совокупности эти новшества сделали System/360 не просто очередной линейкой компьютеров, а фундаментом для всей индустрии вычислительной техники. Универсальная архитектура, масштабируемость, виртуализация, стандартизированная периферия и интегрированная программная поддержка стали теми принципиальными решениями, которые определяли будущее компьютеров на десятилетия вперед.

Железо и софт

В отличие от прежних подходов, где сначала создавали железо, а потом пытались подстроить под него программное обеспечение, команда IBM подошла к System/360 как к единым комплексным системам. С самого начала инженеры процессоров, архитекторы памяти и контроллеров, разработчики операционных систем и языков программирования работали параллельно и синхронно.

Каждая группа имела свои задачи, но решения одной напрямую влияли на других. Например, архитекторы процессора проектировали набор команд так, чтобы он подходил под универсальные задачи – научные расчеты, обработку транзакций, управление вводом-выводом. При этом они консультировались с разработчиками операционной системы: как эти команды будут использоваться для управления памятью, планирования задач, обеспечения многопользовательского режима. Если программисты ОС видели, что какой-то набор команд усложнит поддержку виртуальной памяти или многозадачность, это сразу обсуждалось и корректировалось на уровне архитектуры.

Аналогично, проектирование виртуальной памяти и многозадачности было тесно связано с возможностями железа. Инженеры должны были заранее предусмотреть поддержку изоляции процессов, работу с виртуальными адресами и синхронизацию нескольких потоков.

Это напрямую влияло на конструкцию процессоров, регистров, контроллеров памяти. Каждый блок железа создавался с пониманием того, как его будут использовать программы, и каждая функция ОС проектировалась с учетом того, что реально может сделать процессор.

Периферийные устройства и интерфейсы – еще один пример интеграции. Разработчики ОС должны были управлять дисками, лентами, терминалами и принтерами одинаково на всех моделях. Поэтому архитекторы железа создавали унифицированные контроллеры, а программисты ОС одновременно разрабатывали драйверы и стандартизированные методы доступа. Это позволило достичь совместимости между малой и большой моделью без переписывания ПО.

Особенно сложным было согласование масштабируемости. Малые модели имели один процессор и небольшую память, крупные – несколько процессоров и огромные объемы памяти. Команды и операционная система должны были работать на всех моделях одинаково, а это означало проектирование интерфейсов и функций ОС с учетом возможного добавления процессоров, памяти и устройств. Такой уровень интеграции был тогда принципиально новым: инженеры не могли рассматривать железо и софт отдельно, они рожали систему как единое целое.

Таким образом, System/360 – это не просто машина с операционной системой. Это система, которая создавалась одновременно на двух уровнях: аппаратном и программном, где каждый элемент – процессор, память, контроллеры, ОС, языки программирования – был спроектирован с учетом работы всех остальных. Такой подход позволил IBM впервые создать линейку совместимых машин, где программы могли выполняться одинаково на малых и больших моделях, а железо и софт образовывали единый, согласованный организм.

Модели IBM/360

Когда команда IBM перешла от абстрактной идеи «универсальной системы» к созданию конкретных моделей, каждый элемент линейки проектировался одновременно с соответствующей поддержкой в операционной системе. Например, малые модели вроде 360/30 и 360/40 предназначались для небольших предприятий и отделов крупных организаций. Их процессоры были простыми по структуре, но полностью поддерживали общий набор команд, принятый для всей линейки. Это позволяло запускать те же программы, что и на больших моделях, хотя производительность была ограничена.

Для этих малых моделей инженеры создали минимальный блок процессора с базовыми регистрами и контроллерами памяти, одновременно разработав версию операционной системы, способную управлять ограниченными ресурсами. ОС должна была обеспечивать поддержку ввода-вывода, планирование задач и базовую многозадачность в условиях ограниченной памяти. Инженеры аппаратуры и разработчики софта тесно согласовывали, как каждая команда процессора будет интерпретироваться и выполняться в ОС, чтобы не нарушать совместимость с другими моделями.

Средние и крупные модели, например 360/65 и 360/75, требовали принципиально другого подхода к масштабируемости. В этих машинах использовались несколько процессоров, расширенные блоки памяти и высокопроизводительные контроллеры ввода-вывода. Архитекторы процессоров создавали модули, которые можно было добавлять или заменять без изменения основной логики команд, а программисты операционной системы проектировали механизмы управления ресурсами так, чтобы одинаково эффективно работать с малым и большим числом процессоров и объемом памяти. В результате одна и та же ОС могла запускаться как на малой, так и на большой модели, обеспечивая совместимость программного обеспечения и унификацию работы с пользователем.

Особое внимание уделялось периферийным устройствам. Диски, ленточные накопители, терминалы и принтеры разрабатывались так, чтобы одинаково работать на всех моделях. Для этого архитекторы железа создавали унифицированные контроллеры, а программисты ОС одновременно разрабатывали драйверы и стандартизированные методы доступа. Так, операционная система могла управлять устройствами независимо от мощности модели, обеспечивая клиентам возможность расширять систему без переписывания программ.

Одним из ключевых достижений было создание универсального набора команд, поддерживающего все модели. Инженеры процессоров и разработчики ОС тесно взаимодействовали, чтобы каждая команда могла выполняться на всех машинах одинаково. Это потребовало согласования работы арифметических операций, управления памятью, синхронизации процессов и ввода-вывода. Любое изменение на аппаратном уровне обсуждалось с программистами ОС, а новые функции софта учитывались при проектировании железа.

В результате совместного проектирования родились первые реальные системы System/360, где каждая модель представляла собой не отдельный компьютер, а часть единой архитектурной платформы. Малые машины обслуживали ограниченные задачи, большие – крупные вычислительные центры и банки, но все они работали одинаково, использовали один набор инструкций, унифицированные устройства и совместимое программное обеспечение. Именно эта интеграция железа и софта позволила IBM создать по-настоящему универсальную линейку компьютеров, ставшую фундаментом для всей индустрии на следующие десятилетия.

Для иллюстрации интеграции аппаратуры и программного обеспечения в линейке IBM System/360 мы выбрали две модели: 360/30 и 360/75. Причина в том, что они показывают крайние точки спектра возможностей системы. Малая модель 360/30 предназначалась для офисов и отделов средних предприятий, где важна была компактность, экономичность и работа с ограниченными ресурсами. На ее примере хорошо видно, как инженеры оптимизировали выполнение команд, управление памятью и ввод-вывод, сохранив совместимость с другими моделями.

Крупная модель 360/75 предназначалась для крупных вычислительных центров и банков, где требовалась высокая производительность, многопроцессорность и работа с большими объемами памяти и потоками данных. На ее примере можно проследить, как реализовывалась масштабируемость, управление ресурсами и поддержка сложной многозадачной среды.

Эти две модели позволяют увидеть полный диапазон решений, которые инженеры IBM внедрили, чтобы линейка System/360 стала единой и универсальной. Малая машина показывает оптимизацию ресурсов, большая – масштабирование и сложность управления, а единая архитектура объединяет их в одну систему.

Когда команда IBM приступила к созданию этих моделей, инженеры и разработчики программного обеспечения работали параллельно и синхронно. В 360/30 процессор имел минимальный набор регистров и простую структуру, поэтому пришлось тщательно продумывать, как выполнять все команды универсального набора эффективно на малой машине. Операционная система поддерживала многозадачность и управление памятью в условиях ограниченных ресурсов, а драйверы периферийных устройств позволяли подключать диски, ленточные накопители и терминалы точно так же, как на более мощных моделях.

В 360/75 задачи были иного масштаба. Множество процессоров, большие блоки памяти и высокопроизводительные контроллеры ввода-вывода требовали разработки модульной архитектуры, где процессоры и память можно было наращивать без изменения набора команд. Операционная система распределяла ресурсы между процессорами, синхронизировала параллельные процессы и обеспечивала безопасное многопользовательское использование. Контроллеры периферийных устройств и драйверы ОС создавались так, чтобы одинаково работать на малых и больших моделях, обеспечивая совместимость и унификацию.

Особое внимание уделялось согласованию арифметических и логических операций процессоров с механизмами ОС. На 360/30 каждая операция должна была выполняться быстро

на ограниченном железе, а на 360/75 – масштабироваться и работать в многопроцессорной среде. Инженеры разработали микропрограммы и аппаратные блоки так, чтобы одна и та же инструкция обеспечивала одинаковое поведение в разных условиях, а операционная система могла управлять процессами без зависимости от мощности машины.

В результате каждая модель System/360, от малой 360/30 до крупной 360/75, стала частью единой архитектурной системы. Программы, драйверы, операционная система и устройства ввода-вывода работали согласованно, обеспечивая совместимость и масштабируемость. Именно эта интеграция аппаратуры и софта позволила IBM превратить концепцию «универсальной системы» в реальные машины, способные удовлетворять потребности как малых офисов, так и крупных банков и научных центров.

Основные модели:

- Модели 20, 22, 25 – предназначены для малых предприятий и офисов.
- Модели 30, 40, 50, 65, 67 – средний диапазон, включая научные и корпоративные приложения.
- Модели 75, 85, 91, 95, 195 – высокопроизводительные системы для крупных вычислительных центров и научных исследований.
- Некоторые модели, такие как 64 и 66, были анонсированы, но не поступили в производство. Модели 60 и 62 также не были выпущены, замененные на 65. Википедия+1

• По данным, к концу 1970 года в США было установлено более 7 400 процессоров модели 20, что делает ее самой успешной по количеству установленных машин в линейке System/360. Однако, несмотря на популярность, количество работающих экземпляров модели 20 в 2020 году было ограничено

• Для других моделей точные данные о количестве выпущенных машин варьируются. Например, модель 91 была произведена в количестве от 10 до 20 экземпляров, из которых 4 использовались внутри IBM. Модель 195 также была выпущена ограниченным тиражом – около 20 систем.

• Решение о разработке System/370 было принято в 1968 году, а анонс состоялся в июне 1970 года. System/370 сохраняла совместимость с System/360, но включала новые архитектурные и технологические улучшения.

IBM System/370

Системы IBM System/370 появились в 1970 году как логическое продолжение System/360. Основная идея была в том, чтобы взять проверенную архитектуру, которая уже работала у клиентов, и сделать ее гораздо более гибкой и мощной. Системная точка зрения изменений сводилась к тому, что теперь машина могла работать с гораздо более крупными и сложными программами за счет виртуальной памяти. Раньше программы были «привязаны» к физической памяти – если памяти не хватало, программа не запускалась. В 370-й системе появились аппаратные механизмы, которые позволяли «подменять» части памяти и работать с программами, размер которых превышал объем физической памяти. Это дало возможность одновременно запускать больше задач и повышало надежность работы – программы теперь были защищены друг от друга и не могли «засорять» чужую память.

С аппаратной точки зрения основное изменение – использование интегральных схем вместо транзисторов. Это позволило сделать процессор компактнее, быстрее и надежнее. Появились новые каналы ввода-вывода и контроллеры для работы с дисками и ленточными накопителями, что увеличивало скорость обработки данных и расширяло возможности подключения периферии. Также стало возможным объединять несколько процессоров в одной системе – то, что мы сегодня называем многопроцессорной конфигурацией.

С точки зрения программного обеспечения изменения тоже были существенными. Операционные системы OS/VS1 и OS/VS2 научились работать с виртуальной памятью, появились новые механизмы многозадачности, управления памятью и защиты программ. Все это позволяло писать более сложные, устойчивые и безопасные приложения.

Количество выпущенных машин трудно назвать точно, но ясно, что они были очень популярны в банках, государственных учреждениях и научных лабораториях. S/370 стал настоящей рабочей лошадкой, обеспечив переход от транзисторных машин к интегральным схемам и подготовив почву для более мощных мейнфреймов следующего поколения.

IBM System/390

System/390 появилась примерно в начале 1990-х и стала логическим продолжением 370-й серии, но уже с серьезным обновлением архитектуры. Если 370-й фокусировался на виртуальной памяти и надежности, то 390-й ориентировался на масштабируемость, производительность и интеграцию с корпоративными системами.

С системной точки зрения здесь появилась архитектура ESA/390. Она расширила адресное пространство, улучшила поддержку многопроцессорных конфигураций и позволила системе обрабатывать одновременно еще больше задач. Машины 390-й серии умели работать с большим количеством устройств и каналов ввода-вывода, что делало их удобными для центров обработки данных и крупных предприятий.

Аппаратно машины стали еще быстрее, надежнее и экономичнее. Они использовали более совершенные микросхемы, высокоскоростные каналы связи с периферией и новые технологии управления памятью. Благодаря этому мейнфреймы 390-й серии могли выполнять сложные вычислительные задачи в корпоративной среде без серьезных узких мест.

С программной стороны была введена операционная система OS/390, которая поддерживала все новые возможности архитектуры, включая распределенные вычисления, усовершенствованное управление данными и высокую надежность работы. Программы, написанные для S/370, могли выполняться на S/390 практически без изменений, но при этом они автоматически получали доступ к новым возможностям и повышенной производительности.

Как и 370-й, S/390 был массово использован в банках, телекоммуникациях и других больших организациях, где критически важна надежность, непрерывная работа и обработка огромного объема данных.

IBM zSeries (начало 2000-х)

С появлением zSeries IBM ввела архитектуру, названную z/Architecture, которая впервые сделала мейнфреймы полностью 64-битными. Это был ключевой шаг: адресное пространство резко увеличилось, что позволило работать с огромными объемами памяти и хранить данные в размерах, которые ранее были невозможны. Для крупных банков, страховых компаний, телекомов и государственных структур это означало, что можно обрабатывать миллионы транзакций и одновременно работать с огромными базами данных без ограничений старой 31-битной архитектуры.

Архитектура zSeries встроила поддержку современных сетевых технологий и криптографических функций на аппаратном уровне. Кроме того, она позволяла виртуализировать всю систему – целиком, а не по отдельным процессам – что значительно повысило эффективность использования ресурсов и надежность. При этом сохранялась полная совместимость с программами S/390, что позволило клиентам обновлять железо без переписывания приложений.

IBM System z

Следующий шаг – серия System z – развивала концепцию масштабируемости и интеграции. Системы стали поддерживать сотни тысяч одновременных операций, десятки ядер и тысячи потоков. Вводились новые функции безопасности, поддержка современных операционных систем, включая Linux, и инструменты для интеграции с распределенными системами. Здесь мейнфрейм превращался из «локальной вычислительной машины» в «центр корпоративной вычислительной инфраструктуры», который мог взаимодействовать с внешними системами, облаками и высокоскоростными сетями.

Современные IBM Z – это вершина эволюции мейнфреймов. Они полностью 64-битные, рассчитаны на обработку огромных объемов данных и имеют встроенные аппаратные возможности для шифрования и защиты информации. Эти системы поддерживают контейнеры, облачные технологии и интеграцию с современными DevOps-процессами, сохраняя при этом полную совместимость с кодом, написанным еще для S/370, S/390 или z/Architecture.

Когда IBM представила zSeries в начале 2000-х, это был настоящий качественный рывок. Архитектура z/Architecture сразу перевела мейнфреймы в 64-битную эру. Почему это было важно? Потому что предыдущая система, S/390, ограничивалась 31-битной адресацией. На практике это означало, что физически и логически доступная память ограничивала программы и базы данных, а крупные предприятия все больше сталкивались с этим потолком. С переходом на 64 бита задачи вроде управления банковскими транзакциями, обработкой страховых портфелей или телеком-данных стали решаемыми на одном и том же физическом железе, без необходимости дробить нагрузки на несколько машин.

Аппаратная база zSeries была построена на микропроцессорах RISC POWER. Этот момент особенный: ранее процессоры мейнфреймов представляли собой сложные интегральные схемы, где каждая логика была «своя», но с POWER IBM смогла стандартизировать ядро, повысить тактовые частоты, добавить многопоточность и снизить энергопотребление. И при этом весь исторический код S/390 продолжал работать без изменений. Это, по сути, был мост между старым и новым: старые программы жили на новом железе, а новые задачи можно было решать уже эффективно и масштабируемо.

Программно zSeries принесла еще одно важное новшество – полноценную виртуализацию на уровне всей системы через z/VM. В старых системах виртуализация была локальной, ограниченной процессом или группой процессов. Теперь можно было запустить сотни и тысячи виртуальных машин на одном мейнфрейме, и каждая чувствовала себя полностью автономной. Это резко увеличило эффективность использования ресурсов и стало прямым ответом на требования корпоративной IT-инфраструктуры: нагрузка распределялась автоматически, и при этом сохранялась полная совместимость со старыми приложениями. Появилась также поддержка Linux, что открыло дверь для open-source и современных распределенных приложений.

Следующий шаг System z развивал эти идеи дальше. Если zSeries уже был «умным городом», то System z можно представить как город с развитыми транспортными магистралями, центрами управления и системами безопасности. Аппаратная база стала еще более плотной: процессоры стали более производительными, появилось больше ядер, улучшена работа кэшей и шины памяти. Системы могли одновременно управлять большим числом потоков, поддерживать высокоскоростные каналы ввода-вывода и интегрироваться с внешними распределенными системами. Программное обеспечение тоже эволюционировало: z/OS теперь умела управлять этой сложной многопроцессорной инфраструктурой, поддерживала виртуализацию, шифрование и распределенные вычисления. Linux на System z стал полноценным игроком: предприятия

получили возможность запускать современные приложения на мейнфрейме, не отказываясь от проверенной надежной среды для критичных данных.

Наконец, IBM Z – современная вершина эволюции. Тут мы видим уже не просто умный город, а целую глобальную инфраструктуру. Все принципы zSeries и System z сохраняются, но масштабы выросли в разы. Машины поддерживают десятки ядер с тысячами потоков, управляют огромными массивами памяти, криптографические функции встроены на аппаратном уровне, что делает работу с конфиденциальными данными мгновенно защищенной. Программы S/370, S/390, zSeries и System z по-прежнему запускаются без изменений – преемственность полностью сохранена, а возможности выросли настолько, что современные контейнеры, облачные сервисы и open-source приложения могут работать на том же железе.

Можно сказать, что IBM Z – это эволюция от надежного и мощного мейнфрейма к масштабируемой, полностью виртуализованной, безопасной и интегрируемой корпоративной платформе. Аппаратная база теперь – это многопроцессорные микропроцессоры с RISC-ядрами, плотной кэш-системой и поддержкой тысяч потоков одновременно. Программная база – это z/OS и Linux, способные управлять этими ресурсами как единым целым. Все это создает уникальное сочетание старой надежности и новых гибких возможностей.

Иными словами, IBM Z – это масштабируемая, высоконадежная и чрезвычайно мощная платформа, которая соединяет историческую преемственность мейнфреймов с современными корпоративными и облачными требованиями.

Когда IBM представила zSeries в начале 2000-х, это был не просто новый компьютер, а целый качественный сдвиг в подходе к мейнфреймам. Архитектура z/Architecture сделала машины 64-битными, и это стало переломным моментом. Раньше, на S/390, 31-битная адресация накладывала жесткие ограничения на количество одновременно обрабатываемых данных. Для банков, страховых компаний и телекоммуникационных операторов, которые ежедневно обрабатывали миллионы транзакций, это уже становилось узким местом. Переход на 64-битную архитектуру позволил использовать огромные объемы памяти и работать с базами данных и приложениями без ограничений прежних поколений.

Аппаратная база zSeries строилась на микропроцессорах POWER, что было серьезным изменением. RISC-ядра POWER обеспечивали высокую скорость выполнения команд, многопоточность и многопроцессорность, улучшали плотность размещения процессоров и снижали энергопотребление. При этом сохранялась логическая совместимость с S/390: старые приложения продолжали работать без изменений, что позволяло предприятиям внедрять новые системы без риска для критичных данных. Одновременно zSeries предложила полноценную виртуализацию на уровне всей системы через z/VM, что позволяло запускать сотни и тысячи виртуальных машин на одном мейнфрейме. Программы чувствовали себя полностью автономными, но при этом ресурсы распределялись максимально эффективно. Появление Linux на zSeries открывало доступ к современным open-source приложениям и контейнерным технологиям, соединяя надежность традиционных мейнфреймов с гибкостью современных решений.

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.