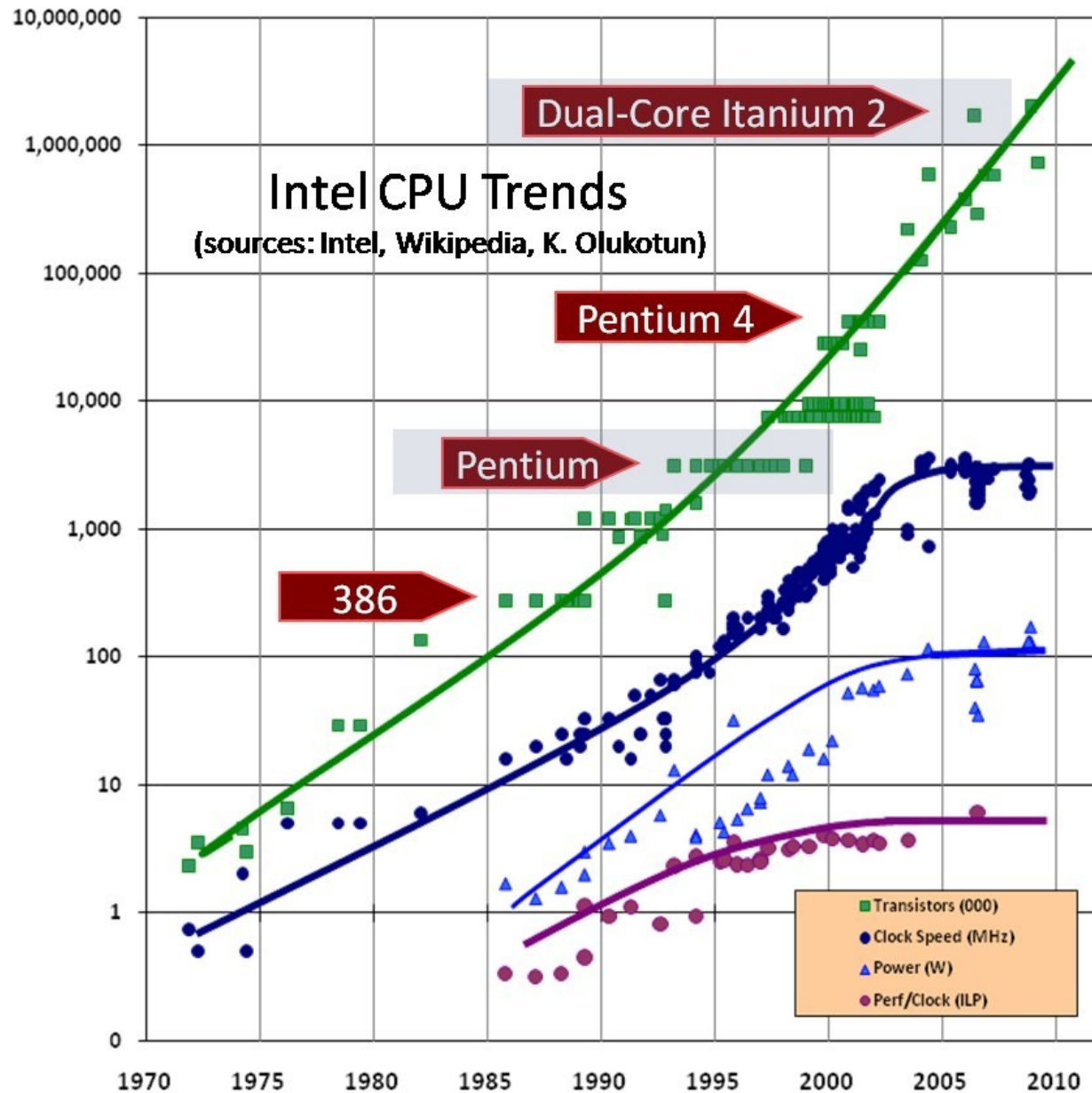


# Лекция 3

		voltage	power	clock khz	техн. Nm	trans. Cnt	die
1971	i4004	15V	1W	740	10000	2300	12
1974	i8080	+5V, ...	1,3W	2000	6000	6000	
1978	i8086	+5V	2,5W	5000	3000	29000	33
1982	i80286	+5V	3,3W	8000	1500	134000	
1985	i386	+5V	1,5W	16000	1000	275000	
1989	i486	+5V	3,5W	20000	1000	1000000	
1993	Pentium	+5V	14,6W	60000	800	3100000	294
1995	Pentium Pro	+3,3V	35W	166000	500	5500000	307
1997	Pentium II	+2,8V	33W	233000	350	7500000	195
1999	Pentium III	+2,0V		450000	250	9500000	128
2001	Pentium 4	+1,605- 1,75V	48,9W	1300000	180	42000000	217
2004	P 4 Prescott	+1,287- 1,400V	89W	2800000	90	125000000	112
2006	P Core Duo			1860000	65	291000000	143
2008	Core i7				45	731000000	263
2011	Sandy Bridge		130W	3300000	32	2270000000	434
2014	Ivy Bridge		150W	3300000	22	4310000000	541

# Закон Мура



# «Закон Мура»

- Эмпирическое наблюдение
- Число транзисторов на кристалле процессора удваивается каждые 24 месяца
- Действует с 70-х годов по наст. время, хотя постепенно выходит на насыщение
- Гордон Мур — сооснователь Intel

# RISC

- RISC (Reduced Instruction Set Computing) — противопоставление CISC (Complex Instruction Set Computing)
- Предпосылки (начало 80-х):
  - Традиционные архитектуры предлагали большое количество режимов адресации
  - Ориентировались на удобство написания программ на ассемблере человеком
  - Квинтэссенция CISC: VAX
    - 21 режим адресации
    - «Сложные инструкции», вплоть до работы со списками

# RISC

- Предпосылки (2) — в то же время:
  - Все больше ПО разрабатывается на языках высокого уровня
  - UNIX — ядро ОС написано на ЯВУ
  - Качество кода, генерируемого компиляторами, улучшается и становится ближе к качеству кода, написанного вручную
  - Компиляторы используют небольшое подмножество CISC-инструкций

# Мотивация RISC

- Оставить только «основные» инструкции
- Оставить только «основные» режимы адресации
- За счет этого упростить и ускорить работу процессора
- Исследовательский процессор Berkeley RISC показал отличные результаты

# Коммерциализация

- Середина 80-х: производители оборудования разрабатывают свои RISC-архитектуры
  - Berkeley RISC → Sun SPARC
  - DEC → Alpha
  - HP → PA-RISC
  - IBM → Power (PPC)
  - Stanford Univ → MIPS
  - Cambridge → ARM



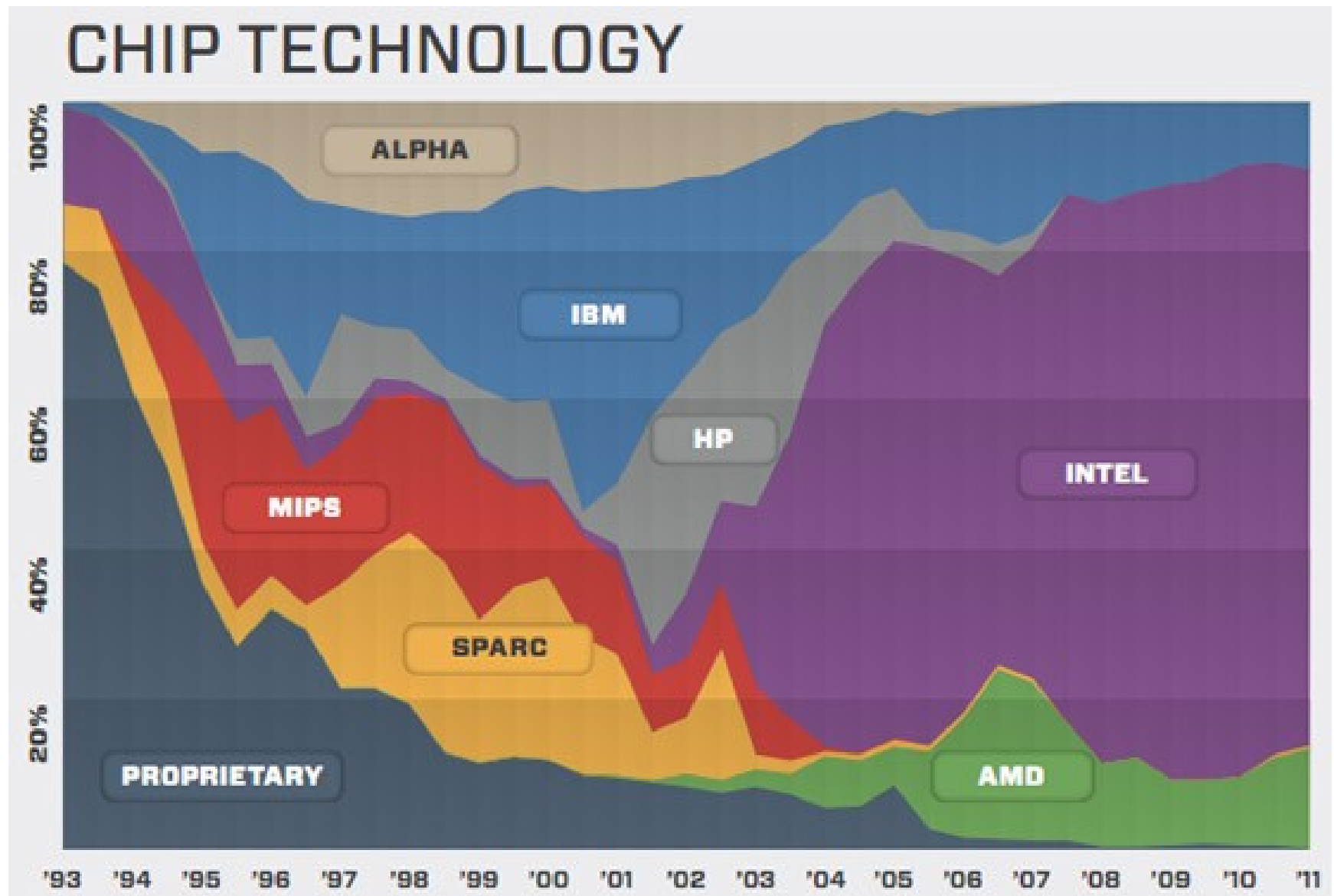
# Рабочие станции UNIX

- В итоге к 90-м годам каждый крупный производитель оборудования выпускал «workstation» на своей архитектуре со своей версией Unix
  - Sun: SPARC и Ultra SPARC, Solaris
  - HP: PA-RISC, HP-UX
  - IBM: ROMP, затем PPC, AIX
  - SGI: MIPS, IRIX

# Workstations vs PCs

- В начале 90-х годов мощность «персональных компьютеров» на процессорах x86 (486, Pentium, ...) нагнала мощность «рабочих станций»
- WinNT приближалась по возможностям к возможностям Unix
- Активно развивались {Free, Net, Open}BSD и Linux

# Market Share



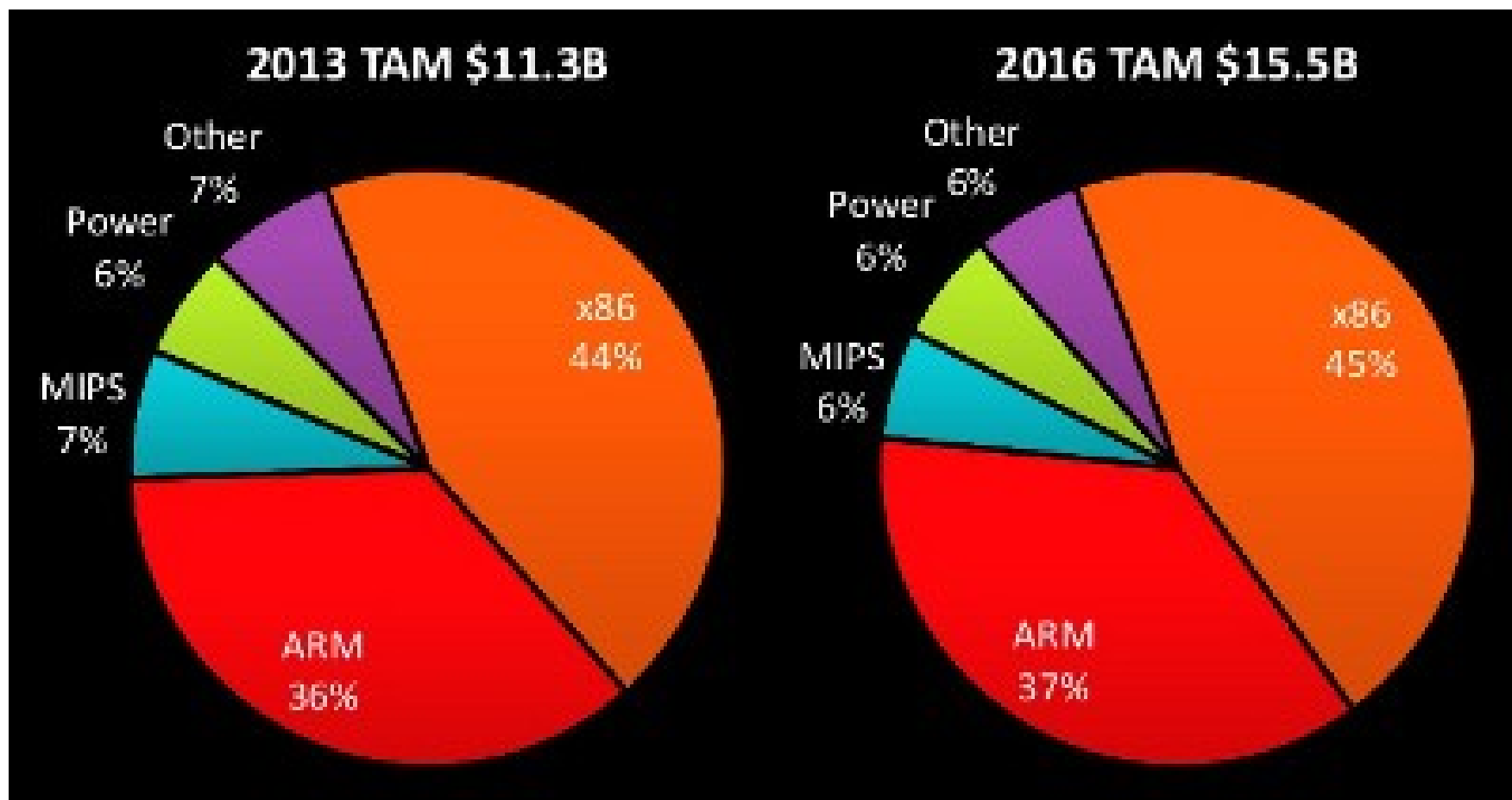
# «Гонка мегагерц» в 90-х

- В 90-х тактовая частота процессоров примерно удваивается каждые полтора года
- Примерное удвоение производительности каждые полтора года у x86 (т. н. закон Мура)
- В итоге — большинство RISC-архитектур рабочих станций теряют рынок
- Традиционная концепция: Unix обречен, Wintel завоюет все

# Смартфоны

- 2007 год — iPhone — использует iOS (производная от Darwin (BSD)), процессор ARM
- 2007 год — Android — ядро Linux, процессор ARM
- Далее бурный рост числа мобильных устройств

# Процессорные архитектуры

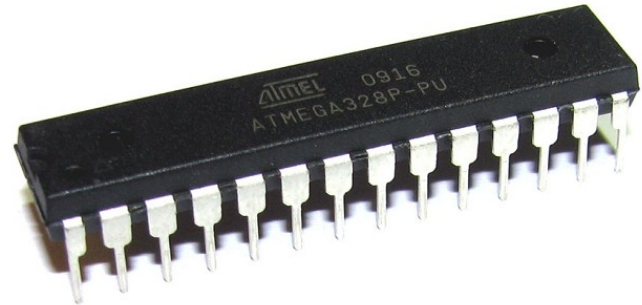


# Современный RISC

- ARM — мобильные устройства
- MIPS — Sony PlayStation, PS2, Nintendo 64, домашние маршрутизаторы
- Atmel AVR — микроконтроллеры
- SPARC - суперкомпьютеры

# Микроконтроллер

- Процессор
- ОЗУ
- ПЗУ (EEPROM, Flash)
- GPIO
- Коммуникационные интерфейсы (UART, I2C, SPI)
- Таймеры
- АЦП

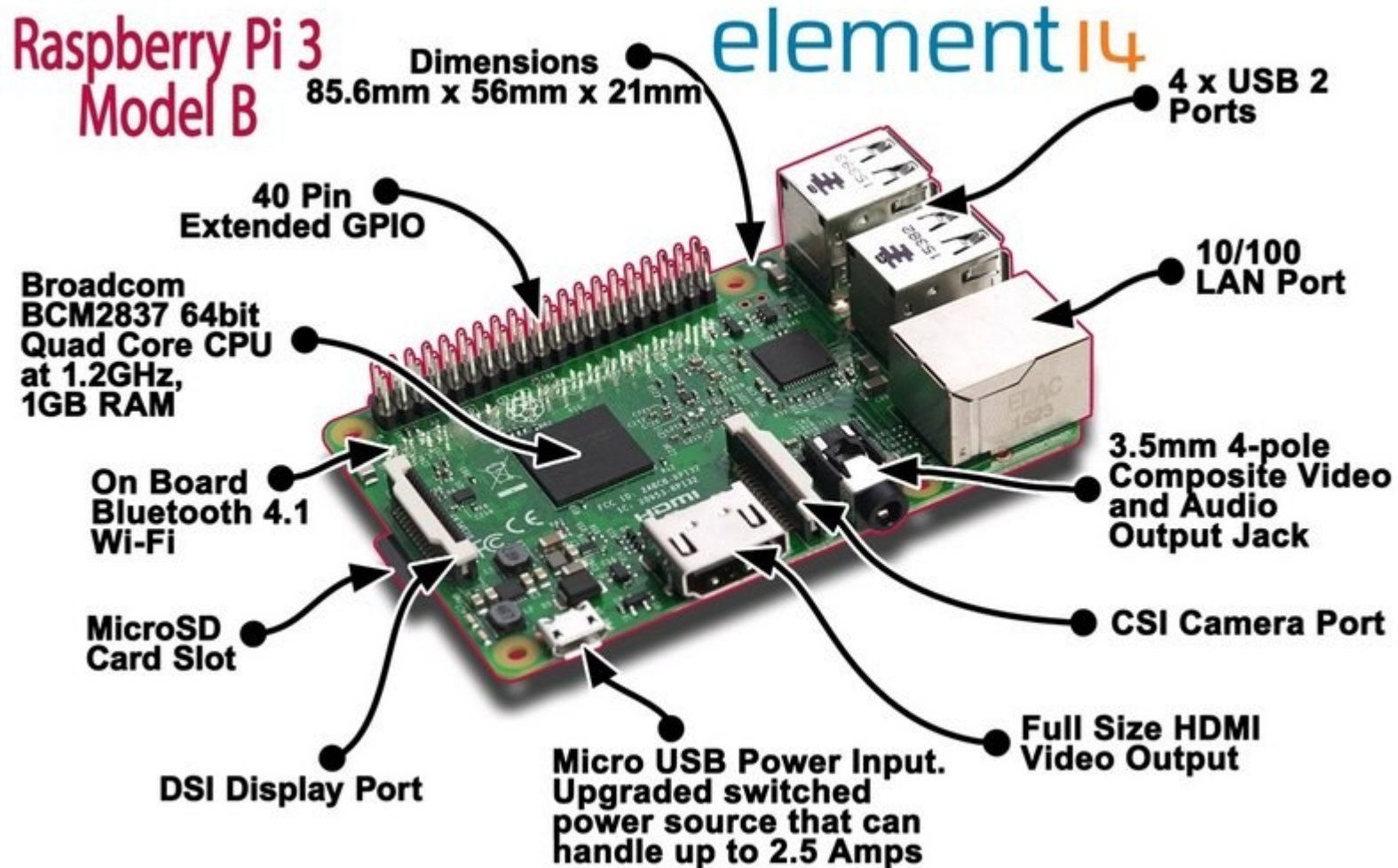




# System-On-Chip

- Микроконтроллер по характеристикам приближающийся к компьютерам:
  - 512 и более MiB RAM
  - Несколько ядер
  - Интегрированный GPU

# System-On-Chip — Raspberry Pi3



# Типы ОЗУ

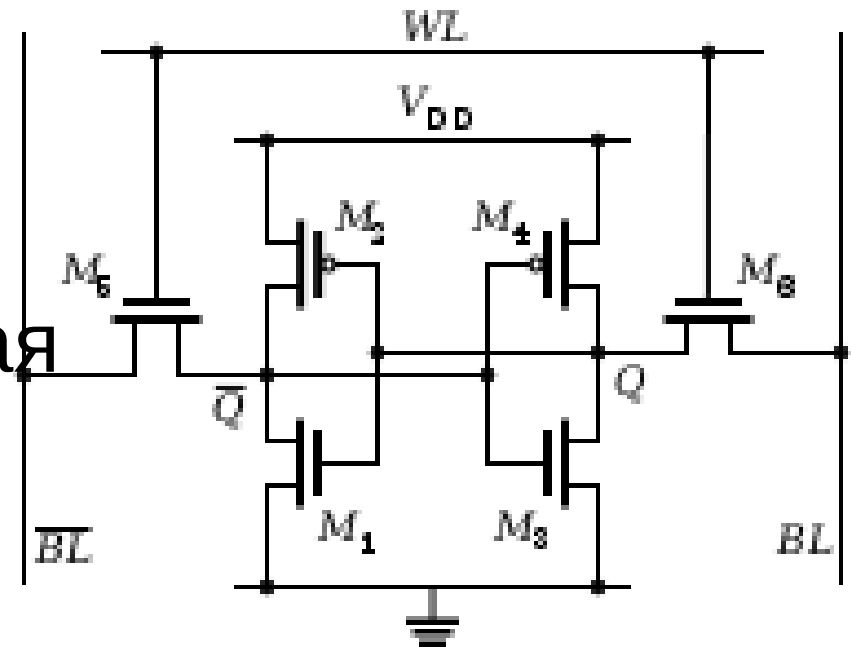
- Статическая память произвольного доступа (SRAM)
- Динамическая память произвольного доступа (DRAM)

# Временные характеристики ОЗУ

- Время чтения — время от начала операции чтения до появления значения на выходе
- Время цикла — время от начала операции чтения до готовности к следующей операции

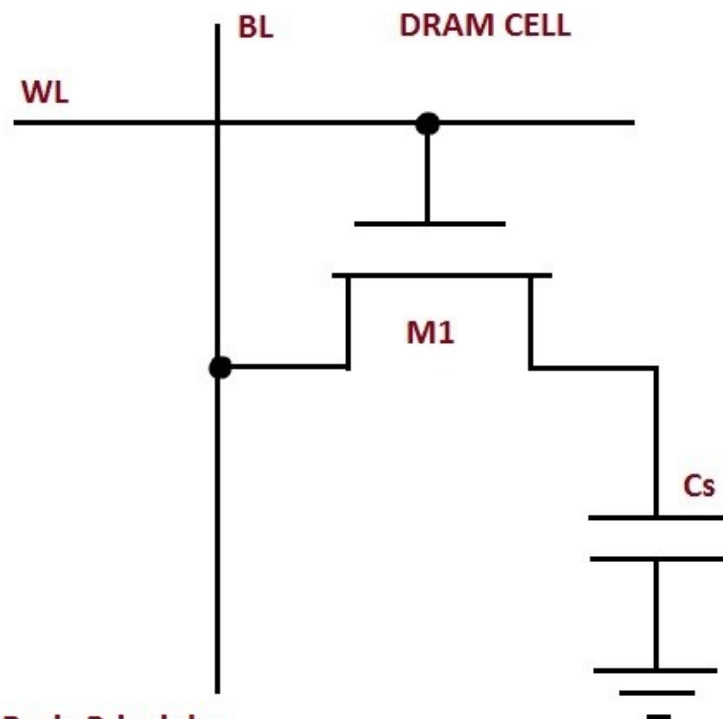
# Статическая ППД

- 6 транзисторов
- Время чтения:  $\sim 1$  такт
- Время цикла:  $\sim 1$  такт
- Время записи:  $\sim 2$  такта
- Тактовая частота - любая

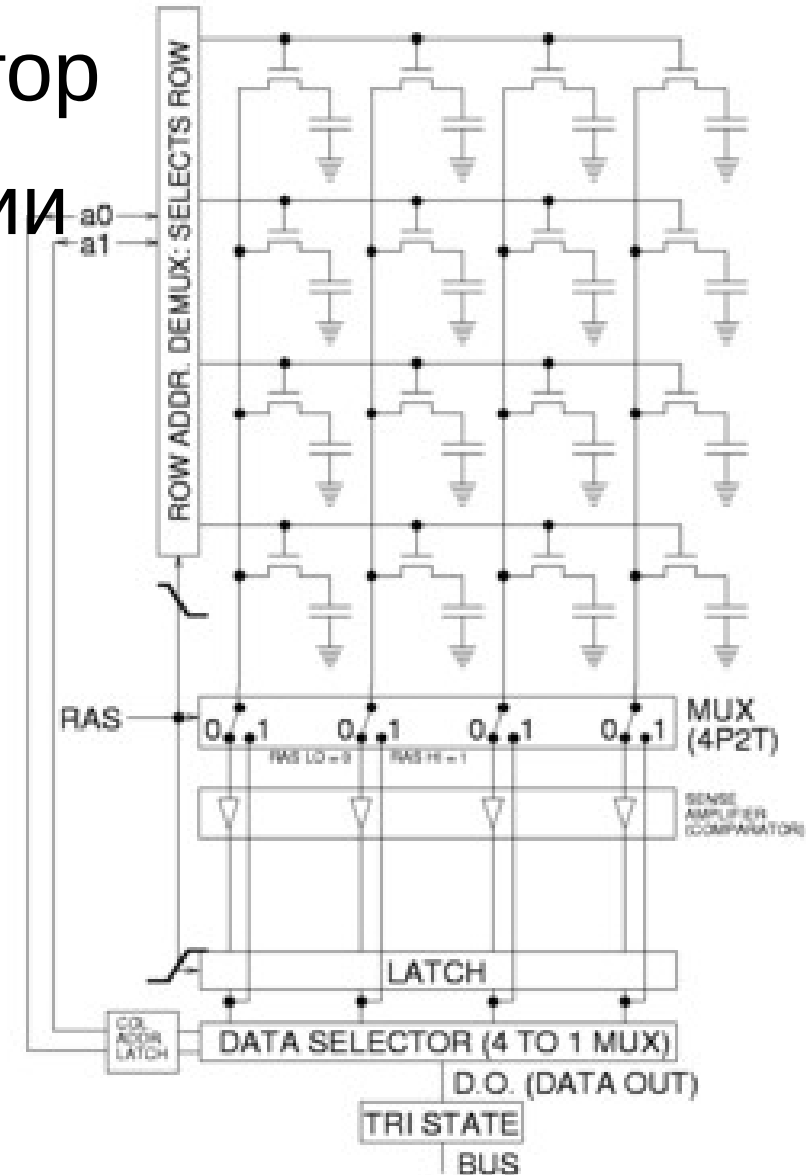


# Динамическая ППД

- 1 транзистор + 1 конденсатор
- Необходимость регенерации



Basic Principle:  
Capacitor stores Data  
MOSFET acts as the Control Switch



# Обновление DRAM

- Распознаваемый «единичный» заряд в конденсаторе может держаться до 1-10с
- Обновление — чтение и запись обратно каждой ячейки
- Современные поколения DRAM обновляют каждую ячейку каждые 64мс
- Во время обновления память недоступна процессору
- Накладные расходы на обновление — примерно 1% времени

# Модификации DRAM

- SDRAM (synchronous DRAM) — использует тактовые импульсы для синхронизации. Все временные характеристики задаются в тактах.
- DDR SDRAM (Double Data Rate) — передача/прием данных дважды за такт
- DDR, DDR2, DDR3



# Алгоритм чтения из DRAM

- ROW SELECT — на нужную строку подается напряжение  $V$ , транзисторы открываются
- Напряжение вертикальной линии  $B+$  слегка изменяется, разница  $B+$  и  $B-$  усиливается дифференциальным усилителем — одновременно считывается и фиксируется (latch) целая строка
- COL SELECT — выборка нужных битов из выбранной строки
- После выборки нужных бит строка закрывается
- PRECHARGE — на все битовые (вертикальные) линии  $B+$  и  $B-$  подается напряжение  $0.5 V$  (от рабочего)

# Memory Timings

- CAS Latency (Tcl) — время между выдачей адреса столбца и получением данных
- RAS to CAS Delay (Trcd) — время между выдачей адрес строки и адреса столбца
- Row Precharge Time (Trp) — время между выдачей команды precharge и открытием следующей строки
- Row Active Time (Tras) — время между выборкой строки и выдачей команды precharge
- 
- Row Cycle Time:  $Trc = Tras + Trp$

# Производительность памяти

- PC-3200 (DDR-400) - 2000
  - 3-4-4-8 (15ns-20ns-20ns-40ns)
- PC2-6400 (DDR2-800) - 2003
  - 5-5-5-16 (12.5-12.5-12.5-40)
- PC3-12800 (DDR3-1600) - 2007
  - 9-9-9-27 (11.25-11.25-11.25-33.75)

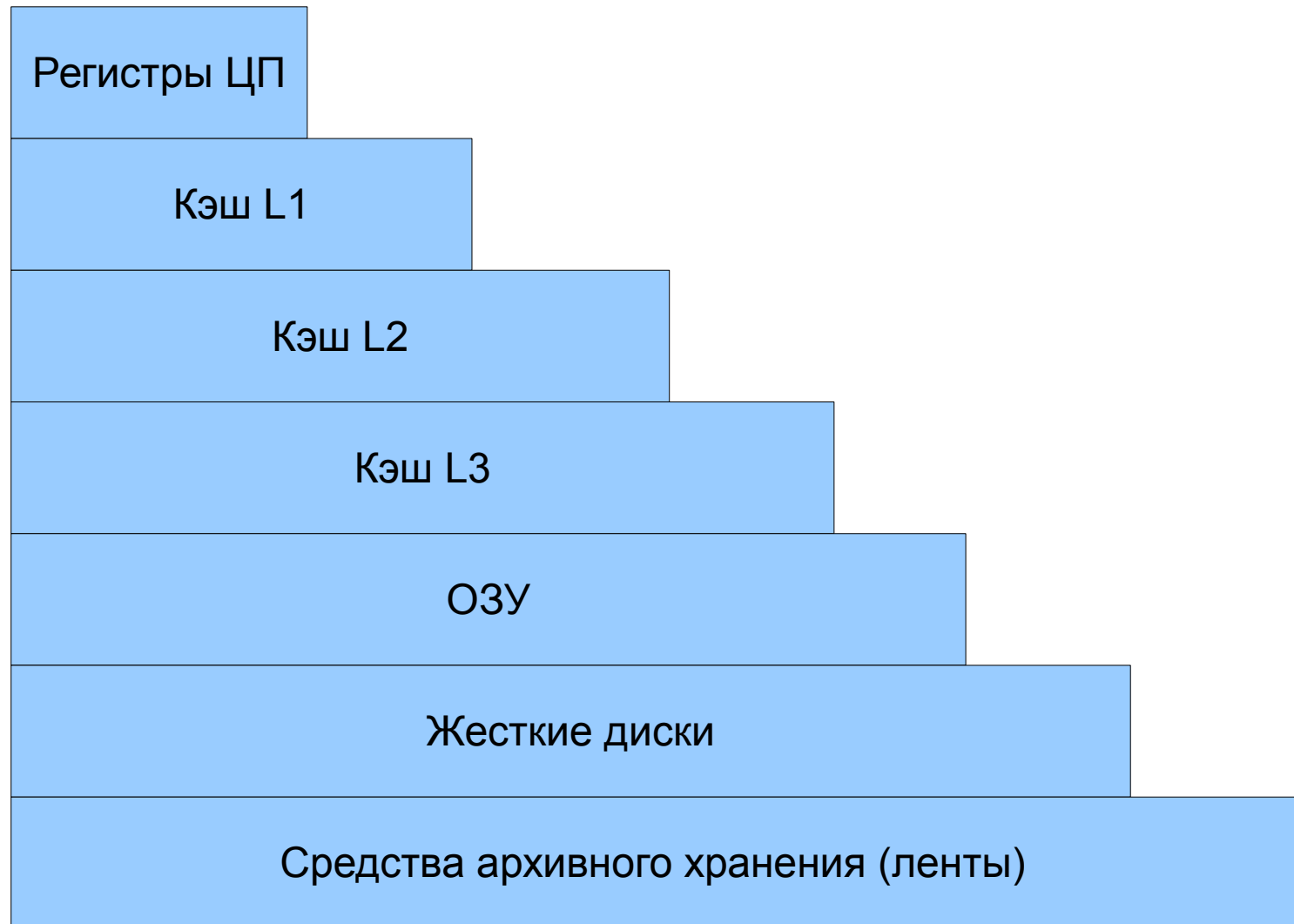
# Сравнение скорости ЦП и ОЗУ

- Предположим, тактовая частота процессора — 3.2ГГц
- Тактовая частота ОЗУ — 800 МГц ( $\frac{1}{4}$  от частоты процессора)
- Время чтения: 72 тактов процессора
- Время цикла: 108 тактов
- С учетом накладных расходов время чтения из ОЗУ: 100 — 200 тактов процессора

# Кэширование

- Кэш (cache) (зачка, тайник) — средство для сглаживания разности скоростей устройств. Некоторая часть данных с медленного устройства помещается в кэш, работающий со скоростью быстрого устройства.

# Иерархия памяти



# Кэш-память ЦП

- L1 — наименьший размер (до 64 KiB), делится на кэш инструкций и данных (L1I, L1D), отдельный для каждого ядра
- L2 — средний размер (1MiB), общая для данных и инструкций, отдельный
- L3 — наибольший размер (до 8MiB), общая для данных и инструкций, общая для всех ядер
- TLB (translation lookahead buffer) — для организации виртуальной памяти

# Работа кэш-памяти

- Попадание (hit) — значение берется из кэша, а не из ОЗУ
- Промех (miss) — требуемой ячейки в кэше нет
  - Обязательный промах (ячейка не загружена)
  - Промех из-за размера
  - Промех из-за конфликта (ячейка была в кэше, но оказалась выгруженной)



# Характеристики кэш-памяти

- Размер
- Время доступа
- Размер ряда (блока) кэша
- Ассоциативность
- Политика записи в основную память
- Политика замещения
- Политика обеспечения когерентности

# Размер ряда (блока)

- При чтении одного байта из ОЗУ загружается ряд (блок) ячеек
- Типичный размер: 32 или 64 байта

# Ассоциативность

- Предположим, что кэш позволяет разместить  $M$  блоков
- Рассмотрим загрузку блока  $B$  из памяти в кэш
- Если блок  $B$  может быть размещен в любой ячейке кэша — **полностью** ассоциативный
- Если блок  $B$  может быть размещен в одной ячейке — **прямое отображение**
- Если блок  $B$  может быть размещен в  $N$  ячейках — **кэш  $N$ -ассоциативный**