

Memory Hierarchy

彭伟桀

2020.11

目录

- 1 随机访问存储器 (RAM)
- 2 磁盘存储
- 3 固态硬盘
- 4 局部性
- 5 存储器层次架构

随机访问存储器 (RAM)

静态 RAM(SRAM)

- 每个单元用六个晶体管电路实现
- 双稳态特性，即使有干扰，干扰消失时电路就会恢复稳定

动态 RAM(DRAM)

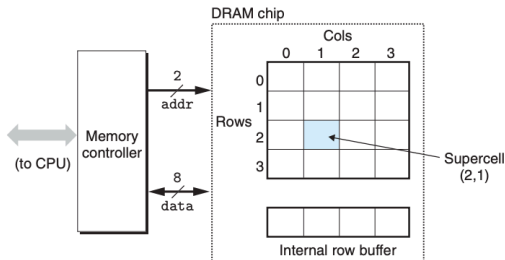
- 每个位储存为对一个电容 ($\sim 30 \times 10^{-15} F$) 的充电
- 容易漏电，需要周期性地读出、重写来刷新，或是使用纠错码

	Transistors per bit	Relative access time	Persistent?	Sensitive?	Relative cost	Applications
SRAM	6	1×	Yes	No	1,000×	Cache memory
DRAM	1	10×	No	Yes	1×	Main memory, frame buffers

Figure 6.2 Characteristics of DRAM and SRAM memory.

传统 DRAM

- 一块 DRAM 芯片 d 个超单元
- 一个超单元 w 个 DRAM 单元
- 超单元被组织为 $r \times c$ 矩阵
- 一块芯片储存 dw 位信息



DRAM 超单元的读取

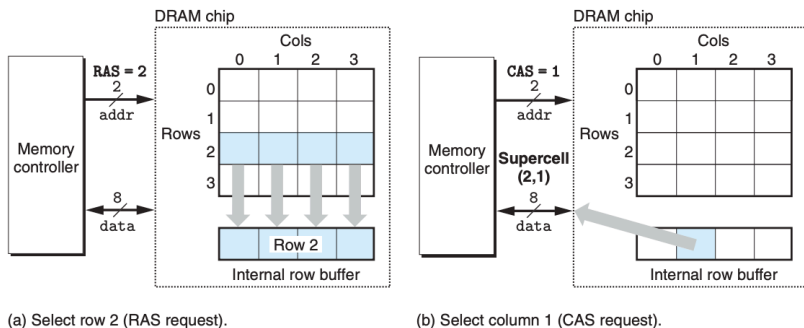


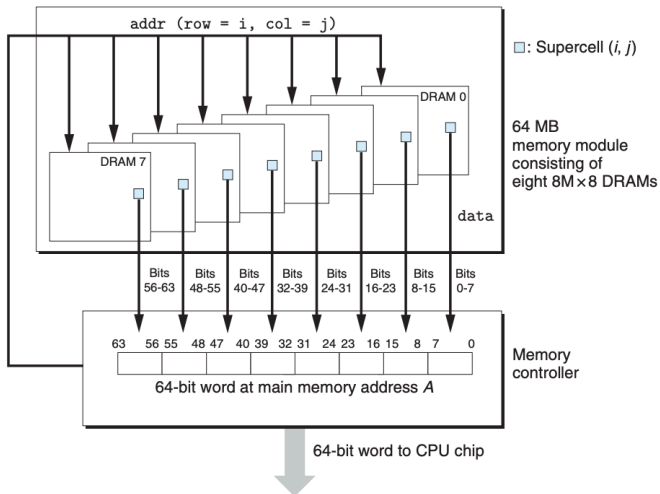
Figure 6.4 Reading the contents of a DRAM supercell.

RAS 和 CAS 分时复用相同的地址引脚
增加了访问时间，减少了引脚数目

内存模块

把多个 DRAM 芯片封装在一起就形成了内存模块
读取时，内存控制器将地址 A 转换成超单元地址 (i, j) 发送到内存模块
内存模块将 i 和 j 广播到每个 DRAM
再将所有 DRAM 的输出合并成最后的结果返回给内存控制器

读取内存模块



增强的 DRAM

- 快页模式 DRAM (FPM DRAM)
利用 DRAM 的行缓冲加速
读同一行中的超单元时，一个 RAS 请求 + 多个 CAS 请求
- 扩展数据输出 DRAM (EDO DRAM)
FPM DRAM 的加强版
- 同步 DRAM (SDRAM)
用和内存控制器相同的时钟信号代替控制信号
- 双倍速率同步 DRAM (DDR SDRAM)
同时利用上升沿和下降沿作为控制信号
DDR X \implies 预取缓冲区大小为 2^X
- 视频 RAM (VRAM)
输出由移位得到，允许并行读写

非易失性存储器

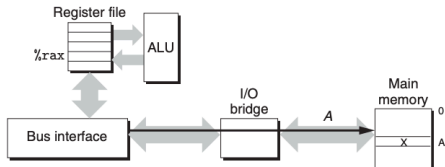
断电也能保存信息的存储器，由于历史原因也被称作 ROM

- ROM: 造好了就确定了
- PROM: 只能被编程一次（利用熔丝熔断）
- EPROM: 允许擦写，利用紫外光擦除，用特殊设备写入
- EEPROM: 类似 EPROM，但可以用电子的方法擦写
- Flash: 基于 EEPROM，但每次抹除的区块较大

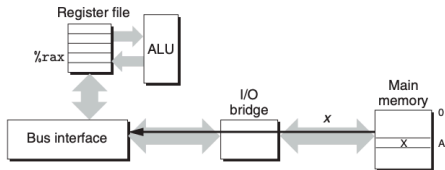
访问主存（读）

Figure 6.7

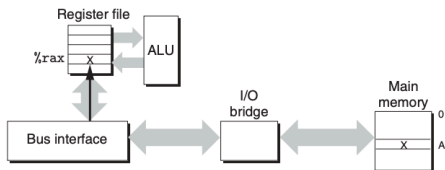
Memory read transaction
for a load operation: `movq`
`A,%rax`.



(a) CPU places address *A* on the memory bus.

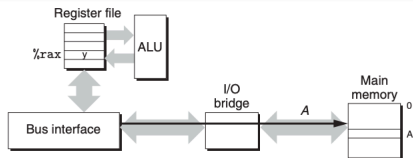


(b) Main memory reads *A* from the bus, retrieves word *x*, and places it on the bus.

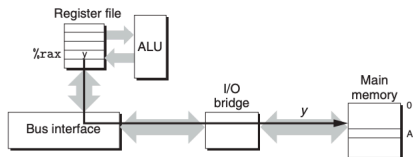


(c) CPU reads word *x* from the bus, and copies it into register `%rax`.

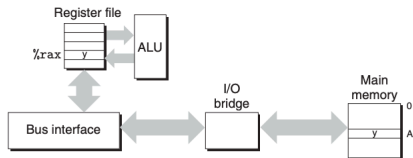
访问主存（写）



(a) CPU places address A on the memory bus. Main memory reads it and waits for the data word.



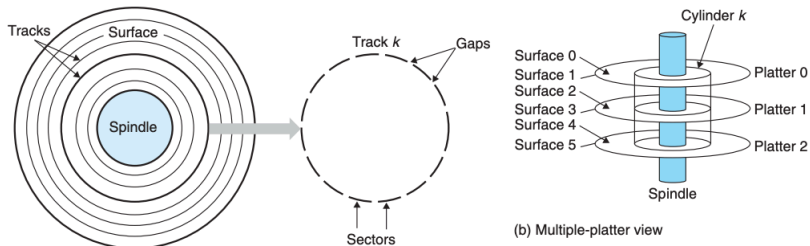
(b) CPU places data word y on the bus.



(c) Main memory reads data word y from the bus and stores it at address A .

Figure 6.8 Memory write transaction for a store operation: `movq %rax, A`.

磁盘构造



(a) Single-platter view

Figure 6.9 Disk geometry.

磁盘容量

磁盘容量决定于

- 记录密度：磁道单位长度能存储的位数
- 磁道密度：半径上单位长度的磁道数
- 面密度：记录密度 × 磁道密度

计算公式：

$$\text{Capacity} = \frac{\# \text{ bytes}}{\text{sector}} \times \frac{\text{average } \# \text{ sectors}}{\text{track}} \times \frac{\# \text{ tracks}}{\text{surface}} \times \frac{\# \text{ surfaces}}{\text{platter}} \times \frac{\# \text{ platters}}{\text{disk}}$$

磁盘操作

盘片逆时针旋转，传动臂径向移动，读写头就能访问到每个扇区。

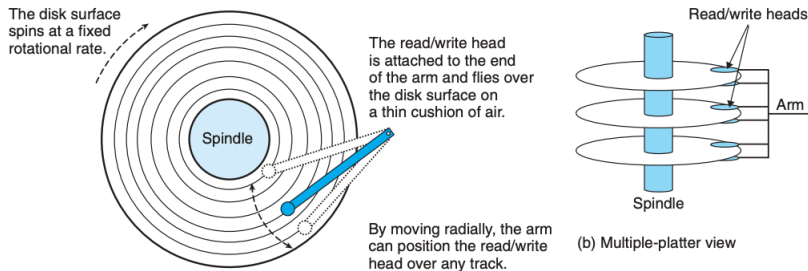
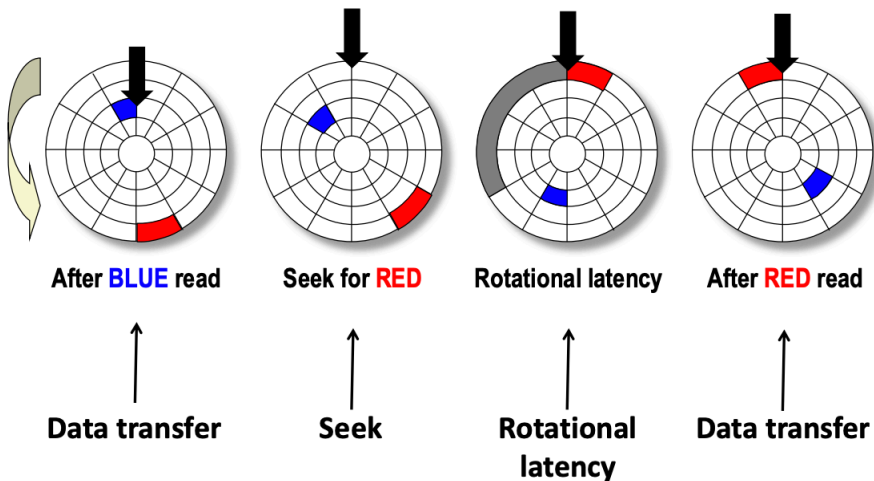


Figure 6.10 Disk dynamics.

磁盘访问



访问时间

- 寻道时间: $T_{avg\ seek} = 3 \sim 9\text{ms}$, $T_{max\ seek}$ up to 20ms
- 旋转时间: $2T_{avg\ rotation} = T_{max\ rotation} = \frac{1}{RPM} \times \frac{60s}{1min}$
- 传送时间: $T_{avg\ transfer} = \frac{1}{RPM} \times \frac{1}{\# \text{ sectors/track}} \times \frac{60s}{1min}$
- 访问时间: $T_{access} = T_{avg\ seek} + T_{avg\ rotation} + T_{avg\ transfer}$

寻道时间和旋转延迟是延迟的主要部分

(可以用寻道时间 $\times 2$ 来估计访问时间)

访问时间比较

64bit word: SRAM $\sim 4\text{ns}$, DRAM $\sim 60\text{ns}$

512Byte: SRAM $\sim 256\text{ns}$, DRAM $\sim 4000\text{ns}$, Disk $\sim 10\text{ms}$

磁盘所花时间大约是 SRAM 的 40000 倍, DRAM 的 2500 倍。

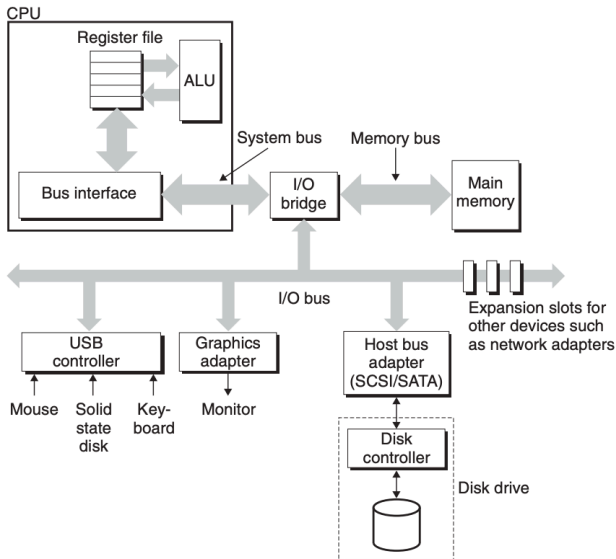
逻辑磁盘块

现代磁盘构造复杂，为了简化，我们将磁盘抽象为一个有 B 个扇区的逻辑块序列，编号 $0, 1, \dots, B-1$ 。至于将逻辑块号映射到物理扇区的工作，则由磁盘里的磁盘控制器来完成。

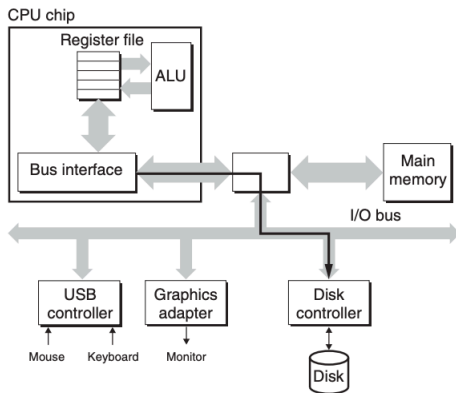
系统执行 I/O 操作，仅需把逻辑块号发给磁盘控制器；
控制器上的固件会查表把逻辑块号转化为（盘面，磁道，扇区）三元组；
控制器上的硬件会控制读写头到合适位置进行读写；
读出的数据被放在缓冲区里，然后被复制到主存。

磁盘控制器可以在每个记录区保留一些空间
可能表现为最大容量和格式化容量的不一致

I/O 总线

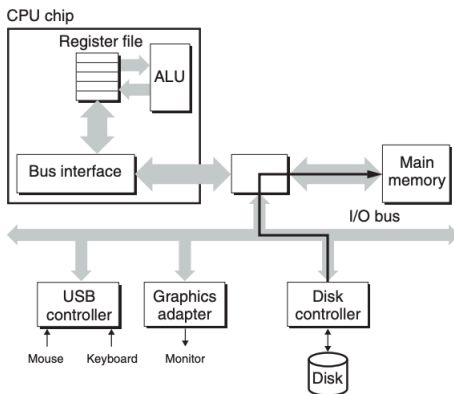


访问磁盘



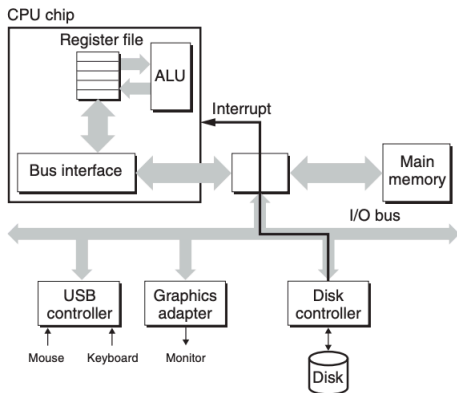
(a) The CPU initiates a disk read by writing a command, logical block number, and destination memory address to the memory-mapped address associated with the disk.

访问磁盘



(b) The disk controller reads the sector and performs a DMA transfer into main memory.

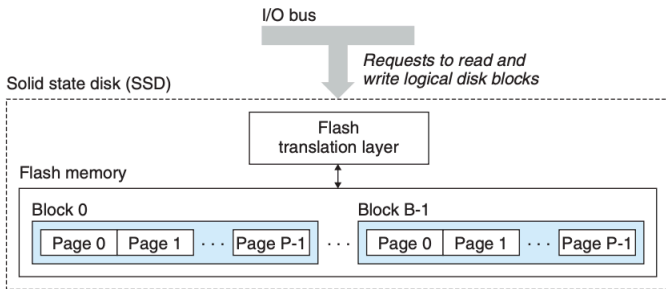
访问磁盘



(c) When the DMA transfer is complete, the disk controller notifies the CPU with an interrupt.

固态硬盘

基于闪存的存储技术



- Page: 512B to 4KB; Block: 32 128 pages
- 以页为单位读写
- 页只有当它所属的块整个被擦除之后才能写
- 一个块的寿命大约是 100,000 次重复写

固态硬盘的特性

Reads		Writes	
Sequential read throughput	550 MB/s	Sequential write throughput	470 MB/s
Random read throughput (IOPS)	89,000 IOPS	Random write throughput (IOPS)	74,000 IOPS
Random read throughput (MB/s)	365 MB/s	Random write throughput (MB/s)	303 MB/s
Avg. sequential read access time	50 μ s	Avg. sequential write access time	60 μ s

- 顺序访问比随机访问快
- 随机写较慢，因为
 - 擦除块耗时较长
 - 擦除一个块前需要把数据复制到其他块上

SSD 与旋转磁盘对比

- 优点

- 没有移动部件。更快，能耗更低，不容易坏（物理上）。

- 缺点

- 闪存容易磨损
- 更加昂贵

存储技术趋势

- 价格和性能之间权衡。越快越贵。
- DRAM 和磁盘的性能滞后于 CPU。

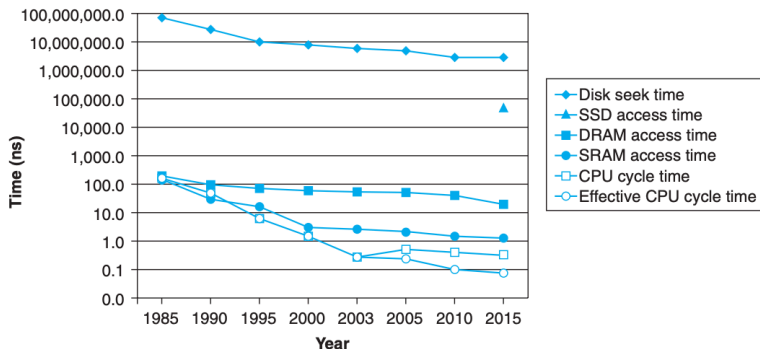


Figure 6.16 The gap between disk, DRAM, and CPU speeds.

局部性

- 时间局部性：引用过一次的内存位置，在不远的将来再被多次引用
- 空间局部性：一个位置被引用，在不远的将来引用它附近的位置

好例子

```
1 int sumarrayrows(int a[M][N])
2 {
3     int i, j, sum = 0;
4
5     for (i = 0; i < M; i++)
6         for (j = 0; j < N; j++)
7             sum += a[i][j];
8     return sum;
9 }
```

Address	0	4	8	12	16	20
Contents	a_{00}	a_{01}	a_{02}	a_{10}	a_{11}	a_{12}
Access order	1	2	3	4	5	6

(b)

(a)

坏例子

```
1  int sumarraycols(int a[M][N])
2  {
3      int i, j, sum = 0;
4
5      for (j = 0; j < N; j++)
6          for (i = 0; i < M; i++)
7              sum += a[i][j];
8      return sum;
9  }
```

Address	0	4	8	12	16	20
Contents	a_{00}	a_{01}	a_{02}	a_{10}	a_{11}	a_{12}
Access order	1	3	5	2	4	6

(b)

(a)

评价局部性的原则

- 重复引用相同变量 → 良好的时间局部性
- 寻址的步长越小，空间局部性越好
- 对于指令，循环的时间局部性和空间局部性都很好，但是循环体越小局部性越好

存储器层次架构

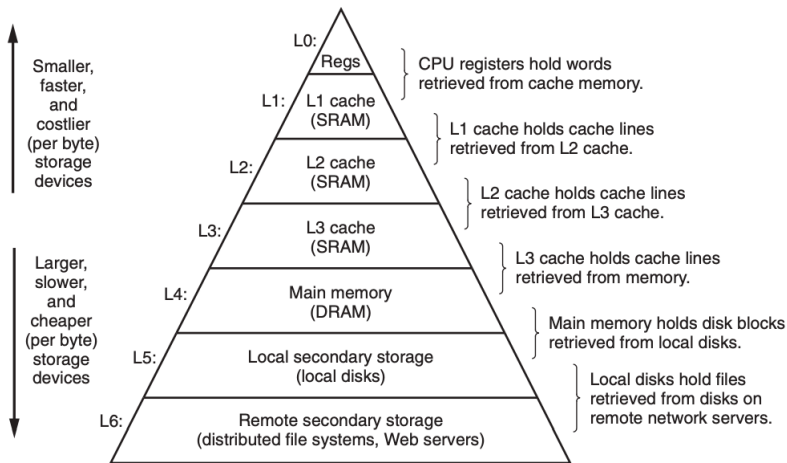


Figure 6.21 The memory hierarchy.

缓存

缓存是一个小而快的存储设备

存储器层次架构的核心思想：用 k 层的存储器来做 $k + 1$ 层的缓存

- 缓存命中
- 缓存不命中
 - 冷不命中
 - 放置策略
 - 冲突不命中
 - 容量不命中

总结

- CPU、存储器之间的速度差距越来越大
- 编写良好的程序具有局部性
- 基于缓存的存储器层次架构利用局部性来缩小差距

Thanks for listening!