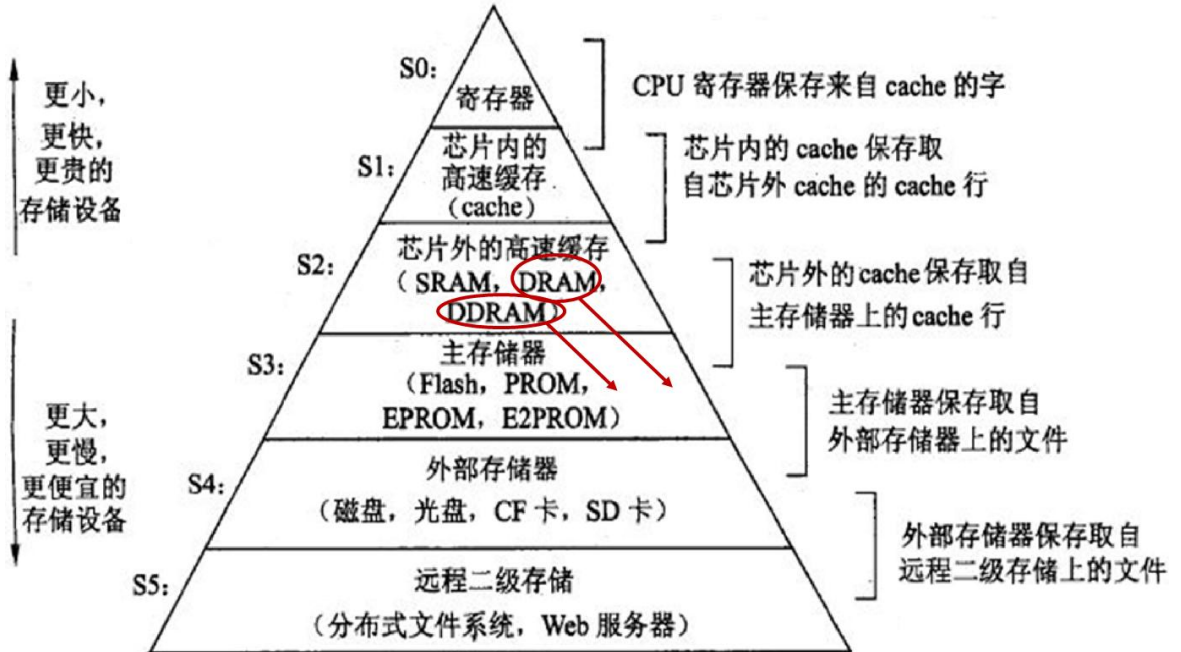
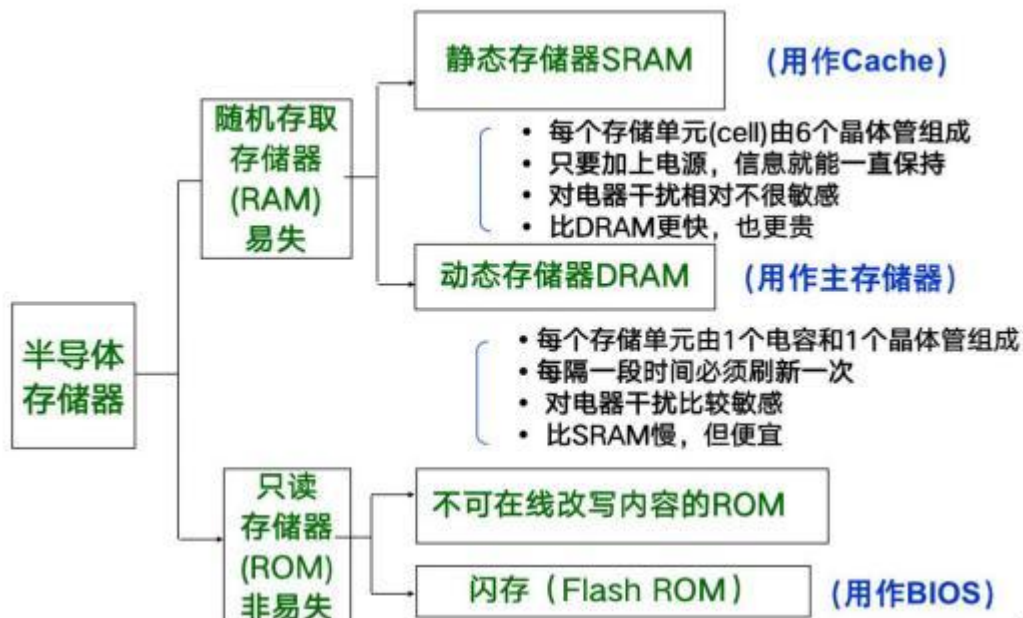


第六章知识点

1. 存储器层级如下图，注意DRAM和DDRAM应该属于S3层主存储器，原图有点小问题。



2. 不同种类半导体存储器的作用，SRAM，DRAM，ROM。



3. 存储器读写事务的基本过程。

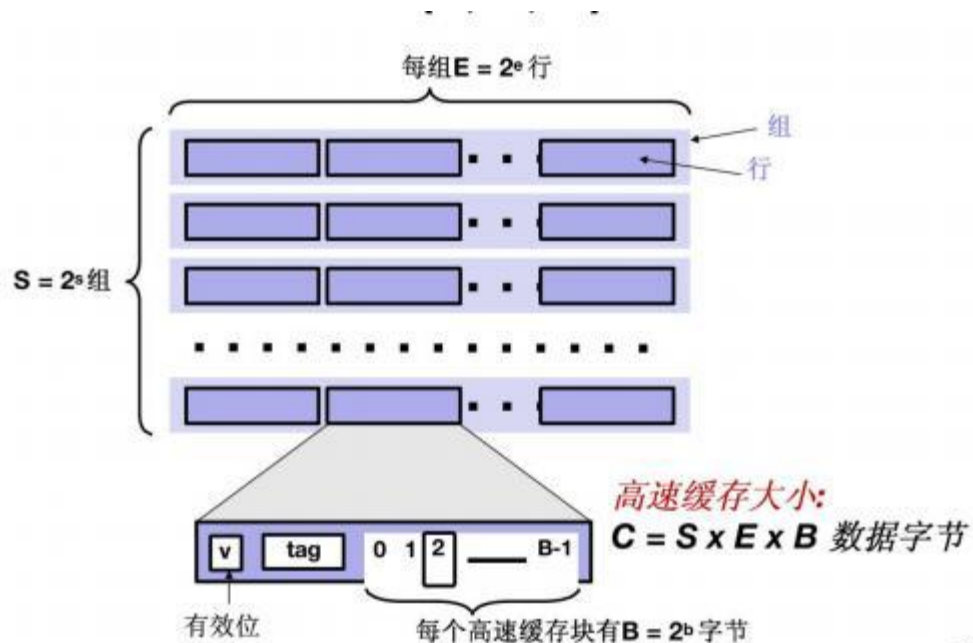
读事务：CPU 把地址放在总线上，主存读地址，然后取出字 X，然后把字放在总线上，然后 CPU 从总线上读取对应的字 X，放入寄存器。

写事务：CPU 把地址放在总线上，主存读地址，然后对应的地址等待写入数据。CPU 把写入的数据放在总线上，主存读到需要写入的数据，然后写入对应的地址处。

4. 磁盘容量的计算方法 容量 = 每个扇区可记录的字节数 x 扇区总数

5. 磁盘访问的平均读取时间 访问时间 = 寻道时间 + 平均旋转延迟 + 数据传输时间
6. 局部性原理
 - a) 程序倾向于使用最近一段时间，距离其较近地址的指令和数据
 - b) 时间局部性(Temporal locality): 当前被访问的信息近期可能还会被再次访问
 - c) 空间局部性(Spatial locality): 在最近的将来将用到的信息很可能与现在正在使用的信息在空间地址上是临近的

让程序局部性尽可能地好的方法：让最内的循环，每次数组索引改变的步长尽可能为 1。
7. 高速缓存的组织形式



根据组索引位和有效位，tag，块内偏移进行查找对应地址元素。

Cache 地址的构成: Tag+SetIndex+Offset

$E=1$ 直接映射高速缓存

$S=1$ 全相联高速缓存

他们之间的比较

Cache的组相联映射和直接映射的对比

组相联映射

cache利用率较高
cache冲突率较低(减少了冲突不命中)
淘汰算法较复杂
应用场合:
小容量cache

直接映射

cache利用率很低
cache冲突率很高
淘汰算法很简单
应用场合:
大容量

cache

全相联映射(特殊的组相联映射)

cache利用率很高
cache冲突率为零(无冲突不命中)
淘汰算法较复杂
应用场合: 小容量cache

看似最好: 全相联映射。
但是全相联映射也有缺点:
硬件结构很复杂, 实现难度和价格都很高, 因此实际应用中往往采取组相联映射。

写策略：直写和写回策略。