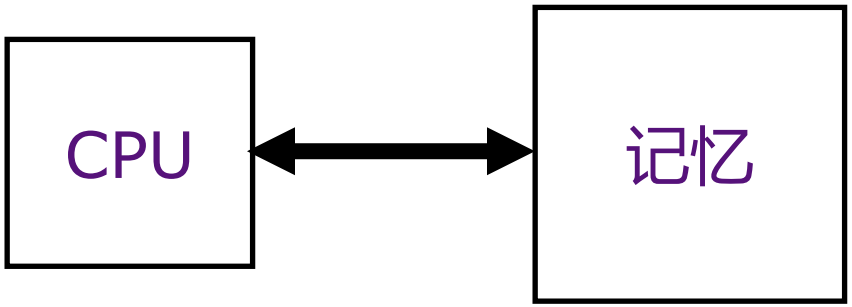
CSE 486/586分布式系统
缓存一致性

史蒂夫-高
计算机科学与工程
布法罗大学

存储到内存

- 我们已经研究了存储的一致性。
- 同样的一致性模型也同样适用于记忆。
  - 认为多个线程访问相同的内存地址
- 但是内存系统可以有另一种形式的一致性，主要用于管理缓存。我们今天就来看看这个。
  - 在一个多核系统中，有许多缓存，它们需要同步进行。

缓存基础知识。CPU-内存瓶颈

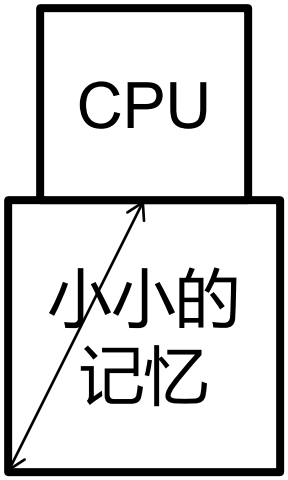


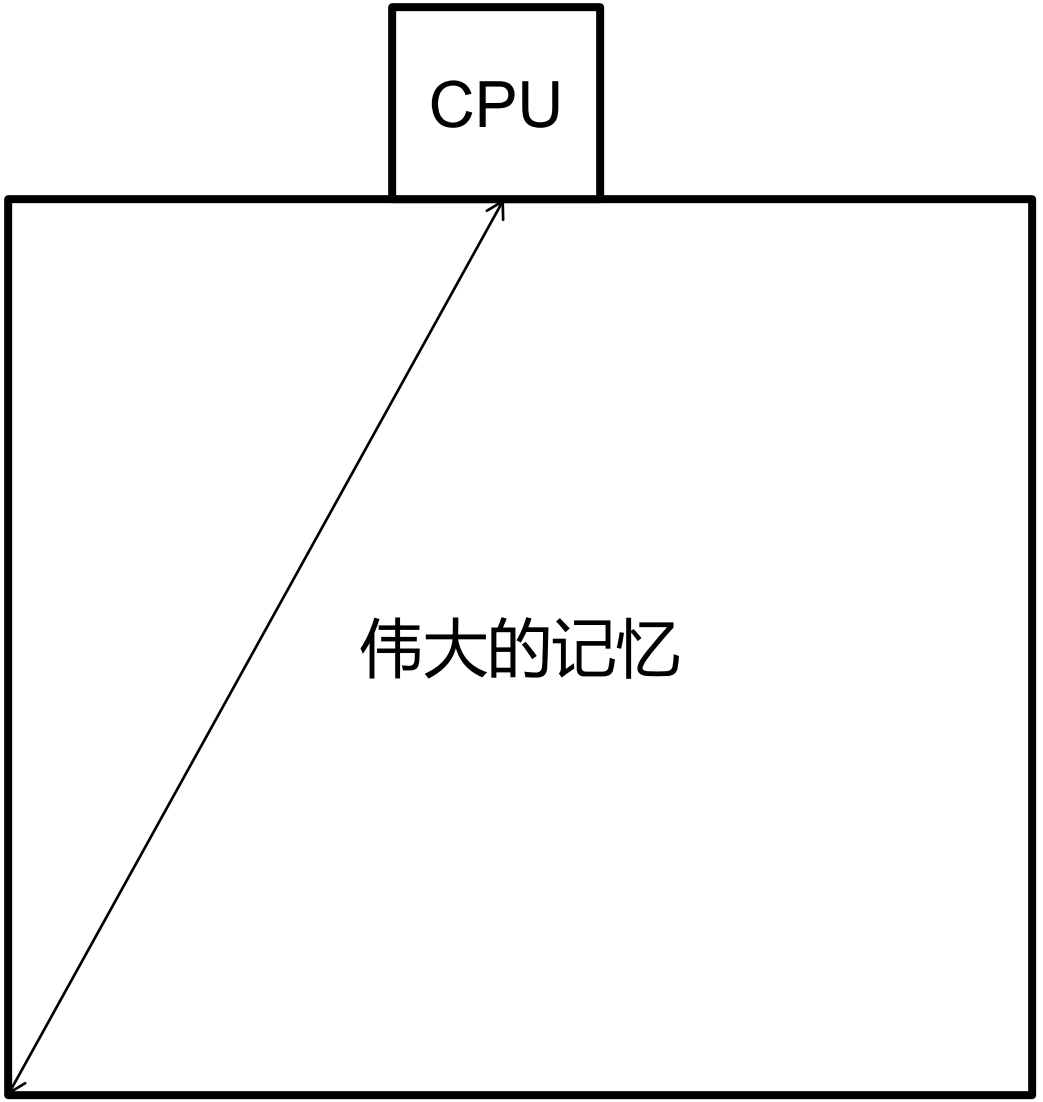
高速计算机的性能通常是
受限于内存*带宽*和*延迟*

- 延迟（单次访问的时间）
  内存访问时间 >> 处理器周期时间
  有问题的

- 带宽（单位时间内的访问次数）。
  增加总线的大小，等等。
  通常情况下是可以的

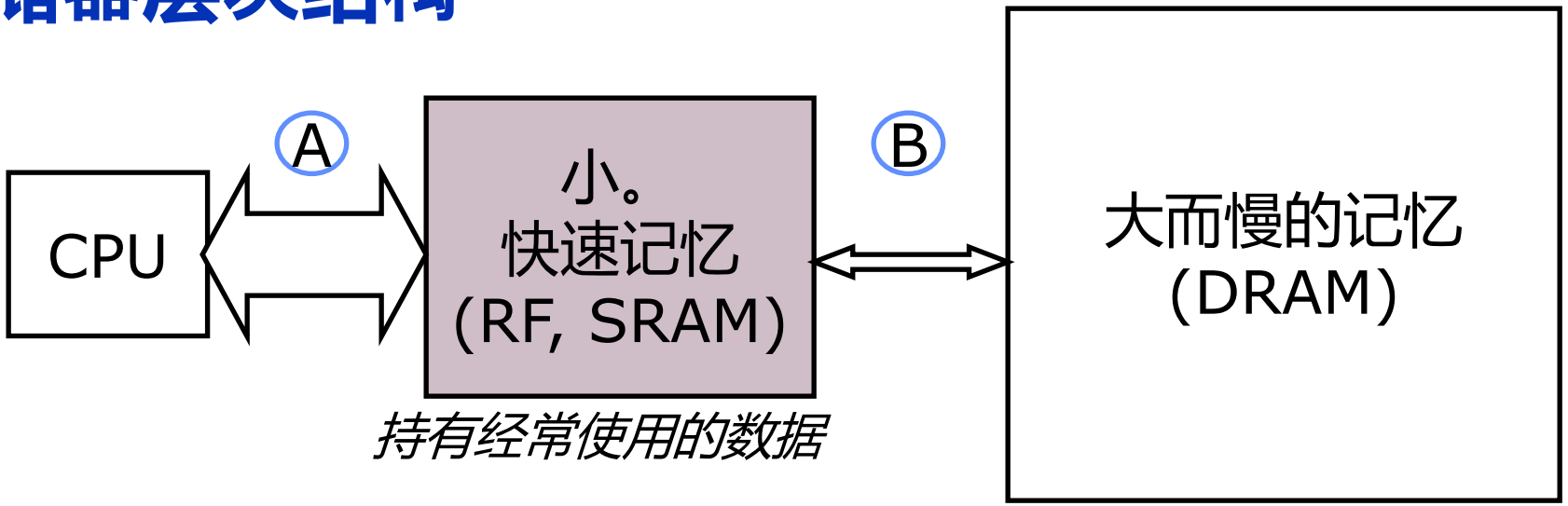
物理尺寸影响延时





- 信号要走得更远
- 扩展到更多的地点

存储器层次结构

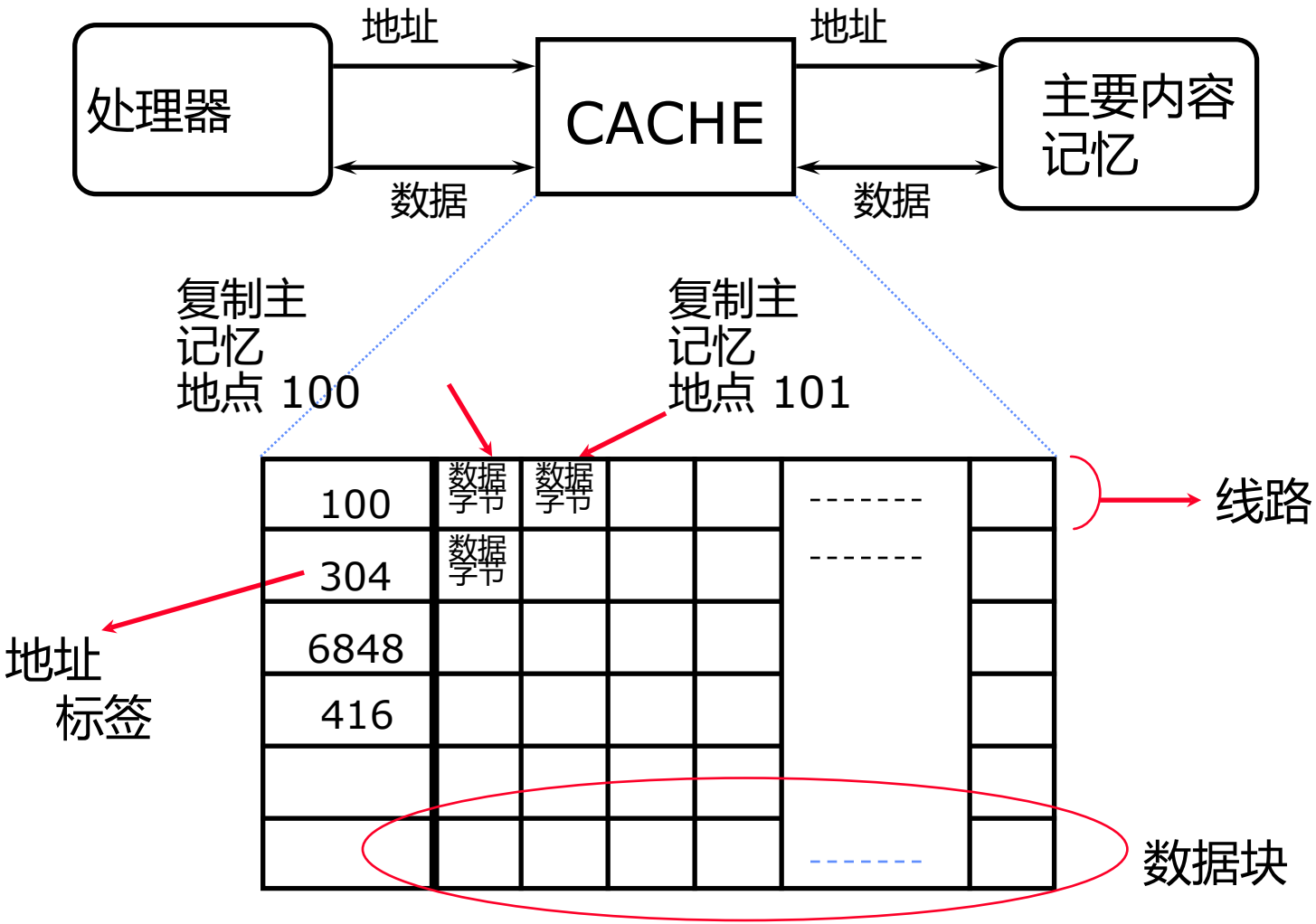


- *容量*：寄存器 << SRAM << DRAM *(成本)*
- *延迟*：寄存器 << SRAM << DRAM *(size)*
- *带宽*：片上>>片外 *(延迟)*。

在一个数据访问中。

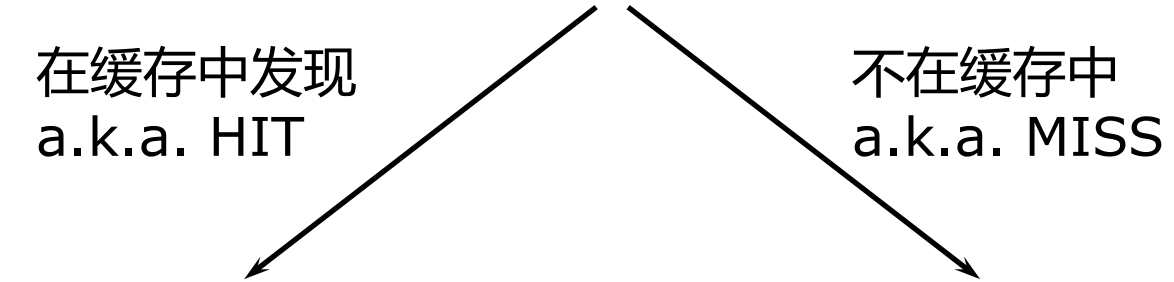
如果数据™快速存储低延迟访问 (SRAM)。
如果数据∉快速内存长延迟访问(DRAM)

缓存内部


处理器
地址
数据
CACHE
地址
数据
主要内容
记忆
复制主
记忆
地点 100
复制主
记忆
地点 101
100
304
6848
416
数据
字节
数据
字节
数据
字节
线路
地址
标签
数据块

缓存读取

看一下处理器地址，搜索缓存标签以找到匹配。 然后，要么



返回副本
的数据来自
缓存

读取来自的数据块
主存储器

等等...

将数据返回给处理器
并更新缓存
(使用一个替换算法
来选择要替换的行）。

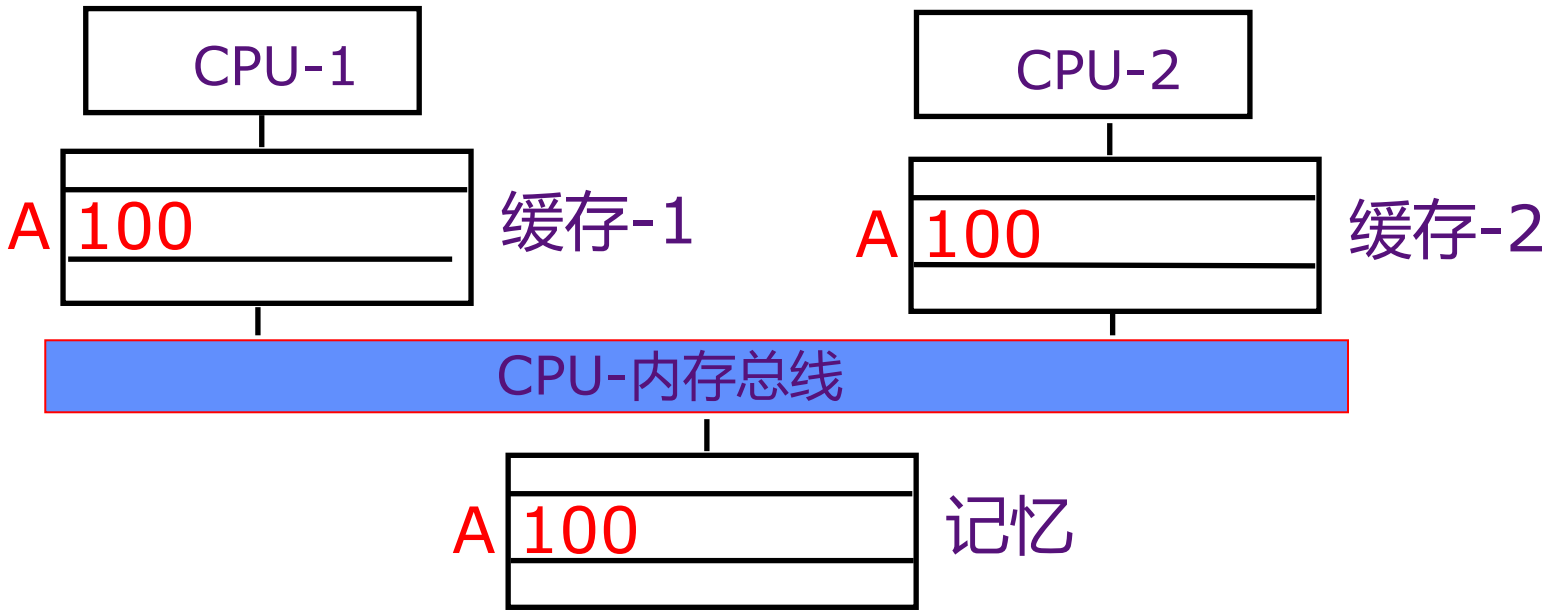
缓存写入

- 缓存击中。
  - ***写透:***同时写入高速缓存和内存
  - ***回写:***只写缓存，只有当条目被驱逐时才会写入内存。
- 缓存丢失。
  - ***不写分配:***只写给主内存
  - ***写入分配（又称写时取物）:***取物到缓存中
- 共同的组合。
  - 写通和不写分配
  - 用写入分配的方式写回

行政琐事

- PA3分级仍在进行中
- 本周五，没有背诵，本科生办公时间为下午2点至4点，普通办公时间为下午4点至5点。

SMP中的内存一致性



假设CPU-1将A更新为200.

回写：内存和cache-2有陈旧的值

Write-through：cache-2有一个陈旧的值

这些陈旧的价值观是否重要?

你能得到什么样的保证?

缓存的一致性

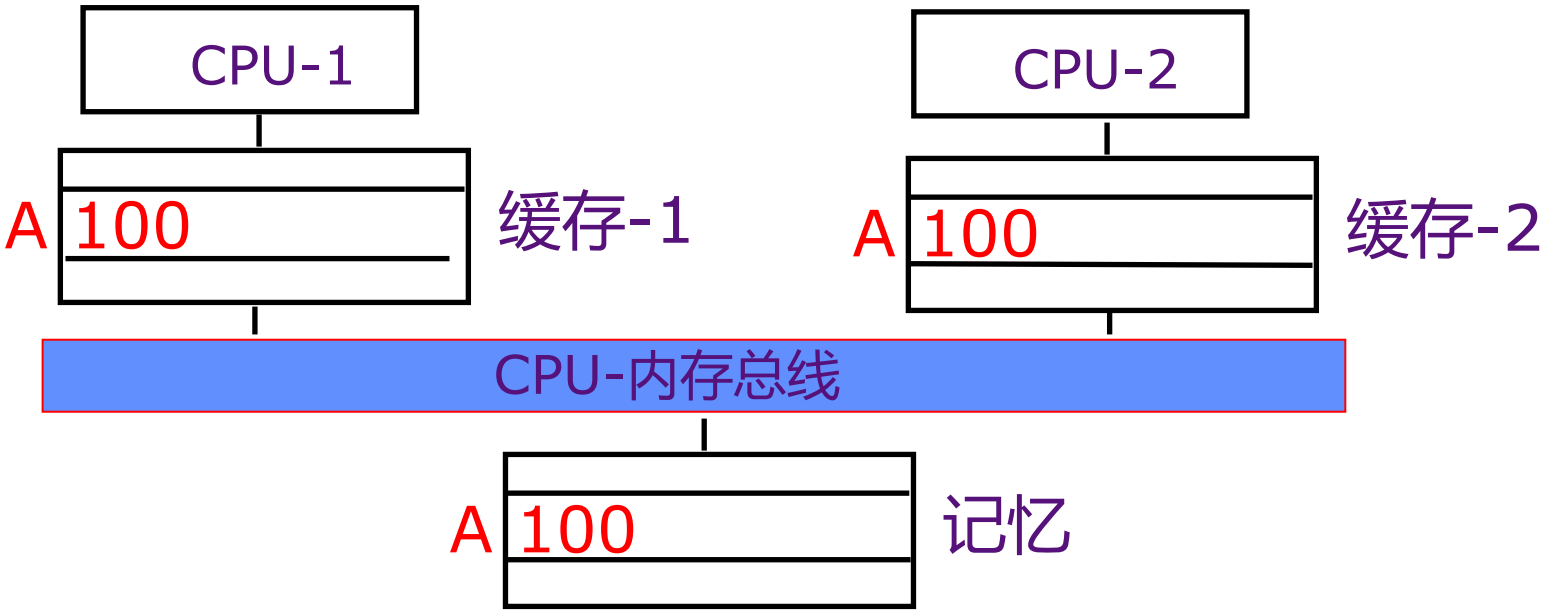
- 缓存一致性协议确保一个处理器的所有写操作最终对其他处理器可见。
  - 即，更新不会丢失
  - 你可以认为这是一种基于硬件的分布式缓存的更新传播机制。

- 需要有硬件支持，以便
  - 每次只有一个处理器对一个位置有写入权限
  - 任何处理器都不能在写入后加载一个陈旧的位置副本

缓存的一致性

- 一个记忆系统是连贯的，如果。
- 一个处理器P对一个位置X的读取是在P对X的写入之后进行的，在P的写入和读取之间没有另一个处理器对X的写入，总是返回P的写入值。
- 如果一个处理器对位置X的读取是在另一个处理器对X的写入之后进行的，那么，如果读取和写入在时间上有足够的间隔，并且在两次访问之间没有发生对X的其他写入，那么就会返回写入的值。
- 对同一位置的写入是串行化的；也就是说，任何两个处理器对同一位置的两次写入在所有处理器中都以相同的顺序看到。
- (相干性提供了每个位置的顺序一致性)。

一个设计。史努比缓存



- 缓存控制器一起工作以保持缓存的一致性。
- 缓存控制器向总线发送命令。
- 每个高速缓存控制器对总线流量进行窥探，以捕捉各种命令并跟踪它们。

史努比缓存相干性协议

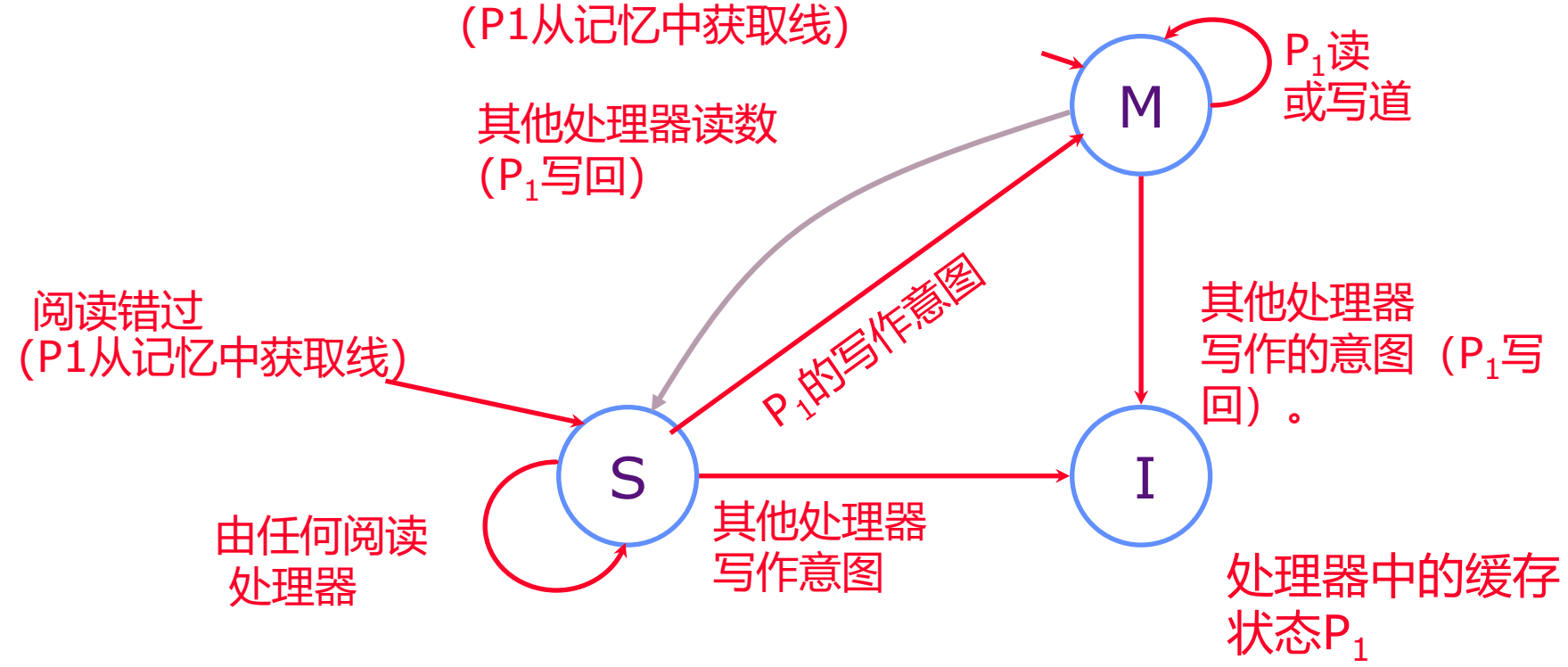
- 每个缓存行都有一个状态。
  - M (modified): 没有其他缓冲区有副本，处理器可以写到它。
  - S（共享）：其他缓存可能也有一个副本。
  - I（无效）：该数据不再有效。
- 如果一个缓存行在S中，那么就只能读。
- 如果一个缓存行在M中，那么写也是可能的。
- 写入高速缓存行
  - 如果是M，缓存控制器就会进行写入。
  - 如果它不是M，它就向其他缓存发送一个无效请求，将状态切换到M，并进行写入。
  - 其他缓存控制器将状态切换为I。
- 读取一个内存地址
  - 如果它是一个热门，就读它。
  - 如果不是命中，就从内存中读取，其他缓存控制器将状态切换为S。

缓存状态转换图
MSI协议

*每个缓存*行都有状态位

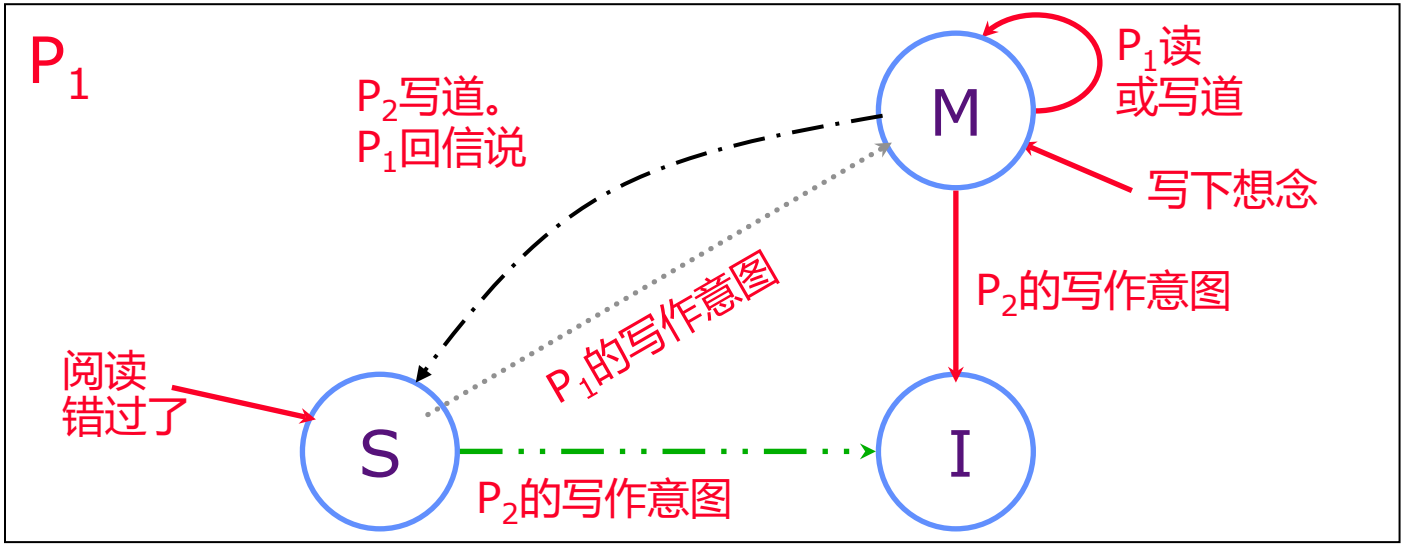
M: 修改过的
S: 共享的
I:无效

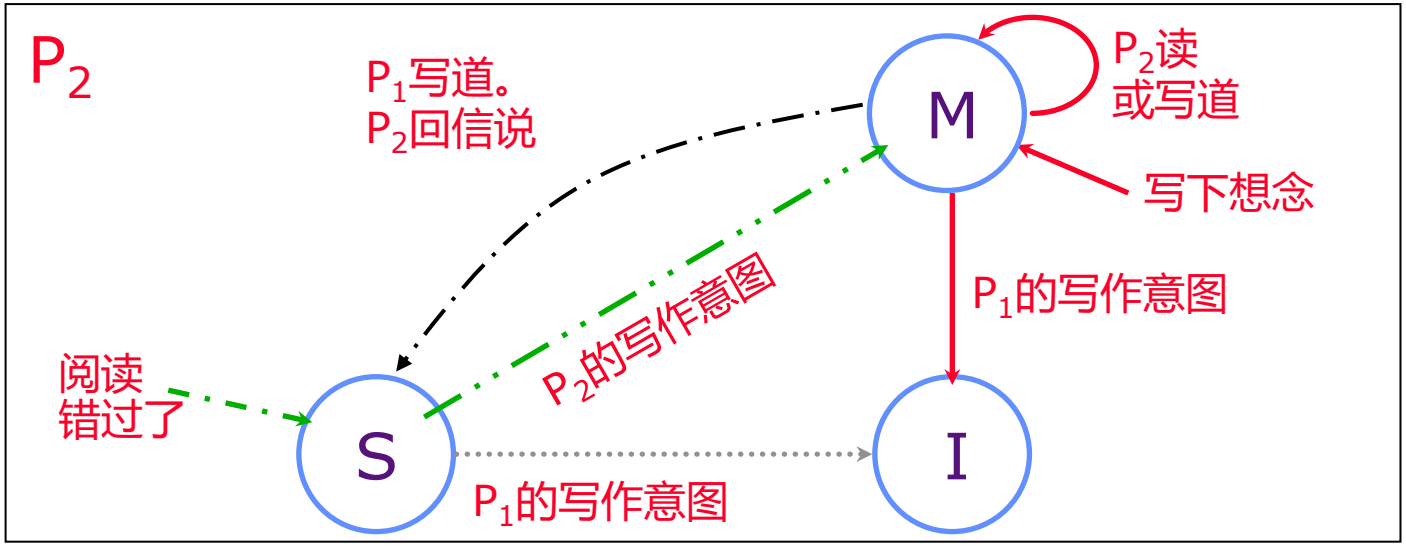
写下想念
(P1从记忆中获取线)



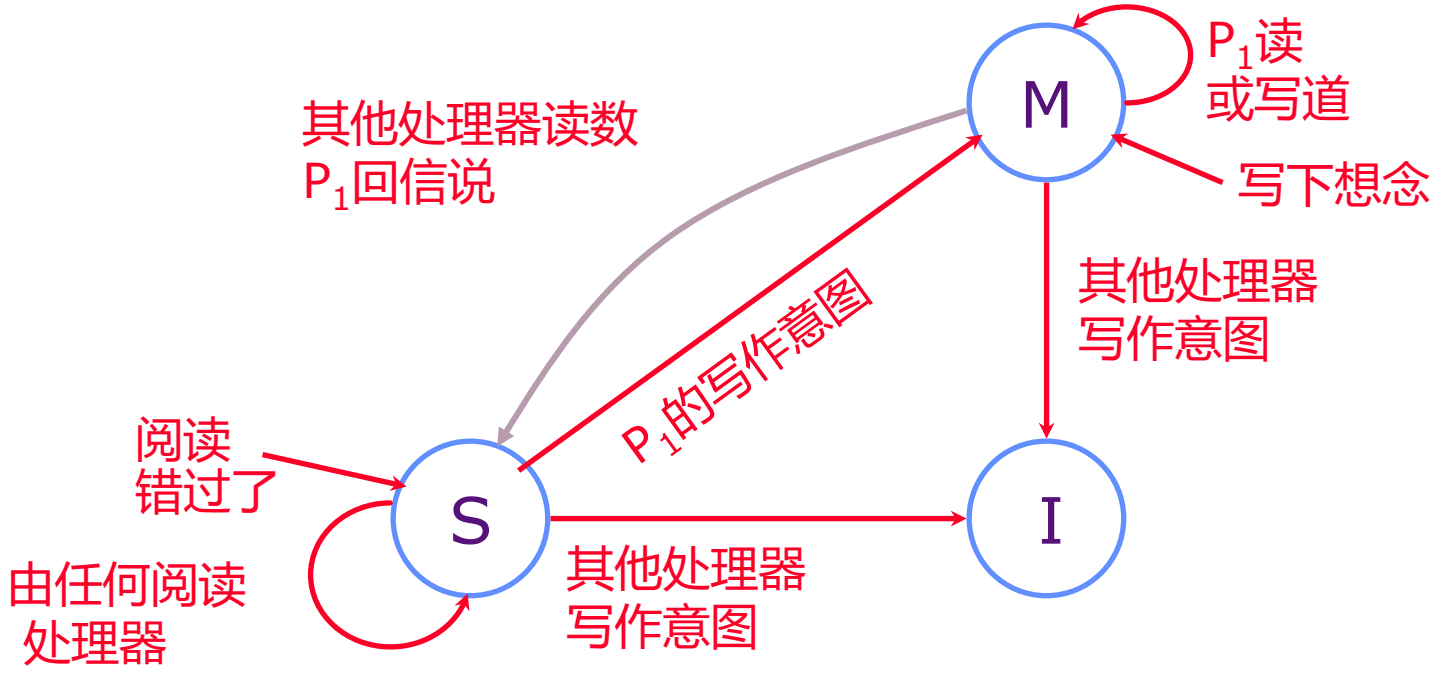
两个处理器的例子
(读取和写入相同的缓存行)。

1 垫子
1 铅笔
2 垫子
2 铅笔
1 垫子
1 铅笔
2 铅笔
1 铅笔





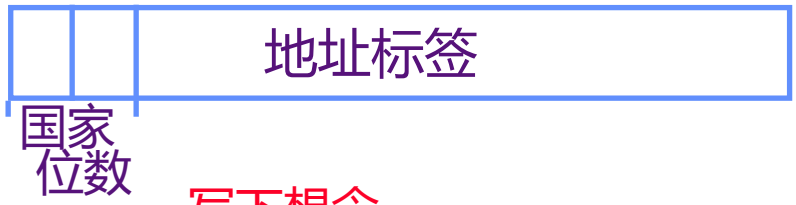
观察到的情况



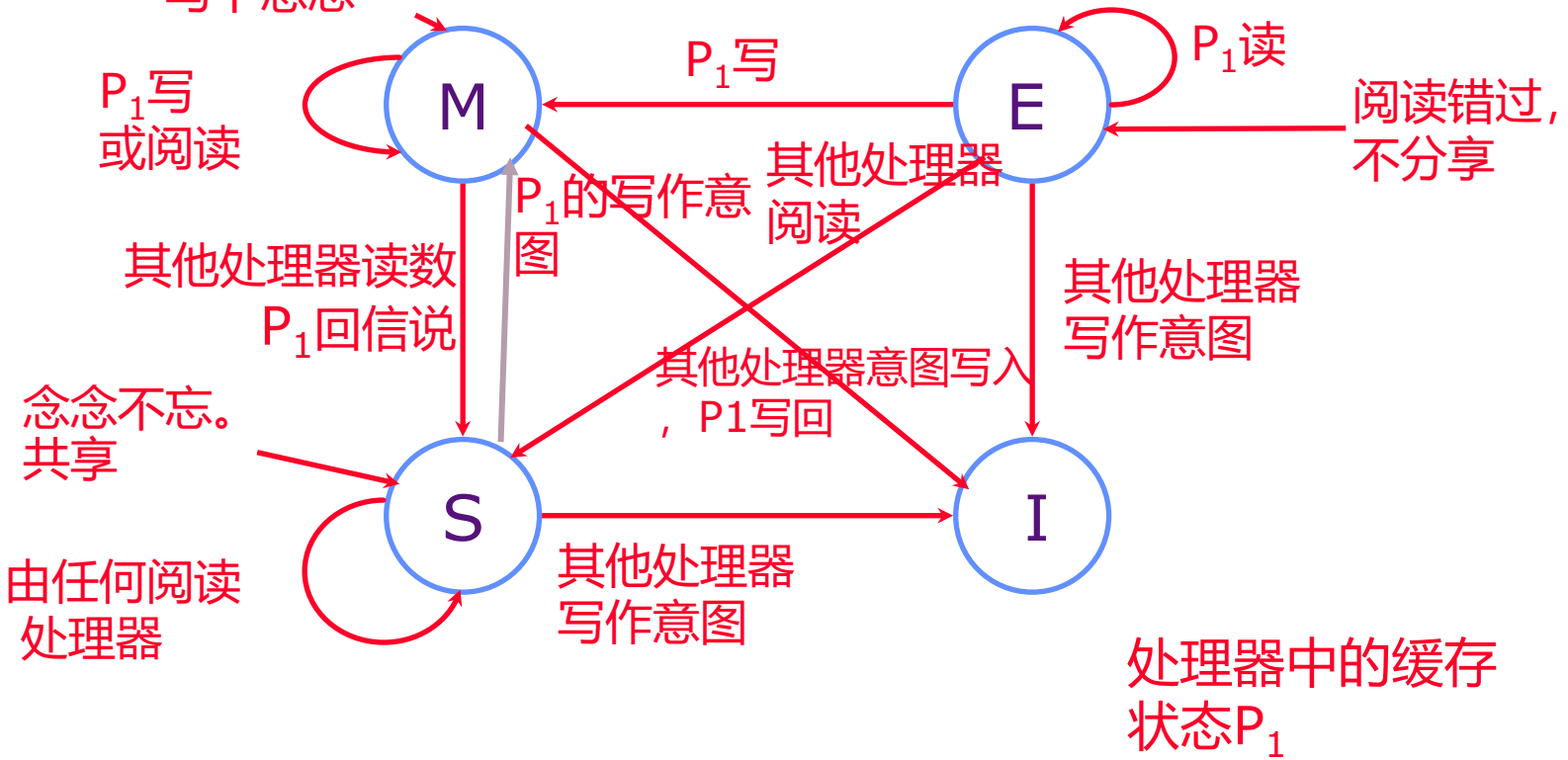
- 2位用于3个状态
  - 还有一个州的空间
- S表示分享是可能的，但不是确定的。
  - 从S到M，总是有无效的请求，即使实际上没有分享。

MESI: 一个增强的MSI协议
提高私人数据的性能

每个缓存行都有一个标签

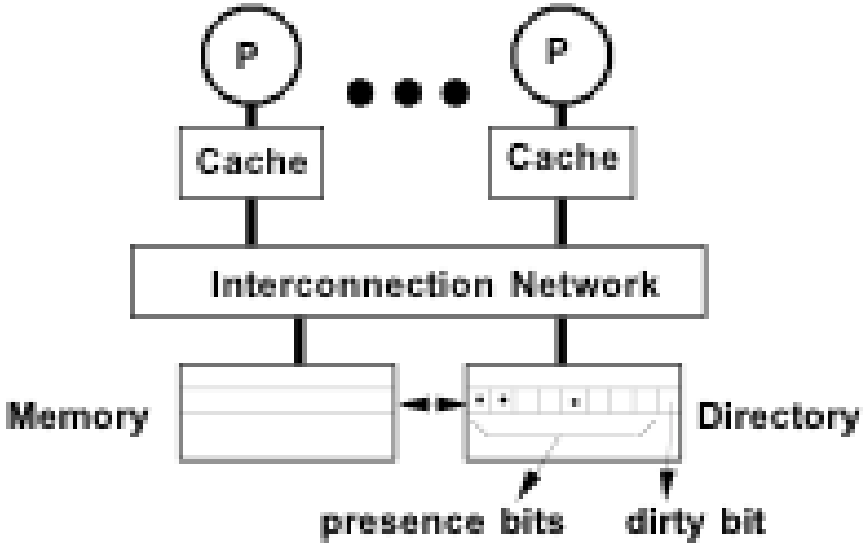


M: 修改后的专属
E: 独占但未修改
S: 共享的
I:无效



可扩展的方法。目录

- 每个内存块都有相关的目录信息
  - 跟踪缓存区块的副本和它们的状态
  - 在错过的情况下，找到目录条目，查找它，并在必要时只与有副本的节点通信。
  - 在可扩展的网络中，与目录和副本的通信是通过网络交易进行的。
  - 组织目录信息的许多备选方案



- k处理器。
- 在内存中的每个缓存块。
  k个存在位，1个脏位
- 在缓存中的每个缓存块。
  1个有效位，和1个脏（所有者）位

摘要

- 缓存一致性
  - 确保缓存不包含陈旧的副本。
- 斯努比缓存的一致性
  - MSI
  - MESI
- 基于目录的
  - 每个内存块使用一个目录

鸣谢

- 这些幻灯片大量包含了由以下机构开发并拥有版权的材料
  - Krste Asanovic (MIT/UCB)
  - 大卫-帕特森（UCB）
- 而且还由。
  - Arvind (MIT)
  - Joel Emer (英特尔/麻省理工学院)
  - James Hoe (CMU)
  - John Kubiatowicz (UCB)

- 麻省理工学院的材料来自于课程6.823
- UCB的材料来自于CS252课程