

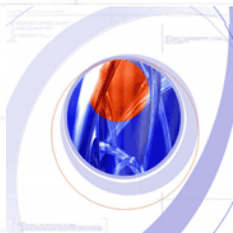
第8章 多处理器

8.1 多处理器概念

8.2 对称式共享存储器系统结构

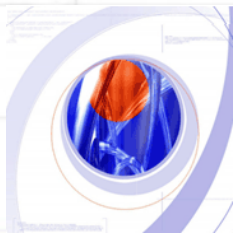
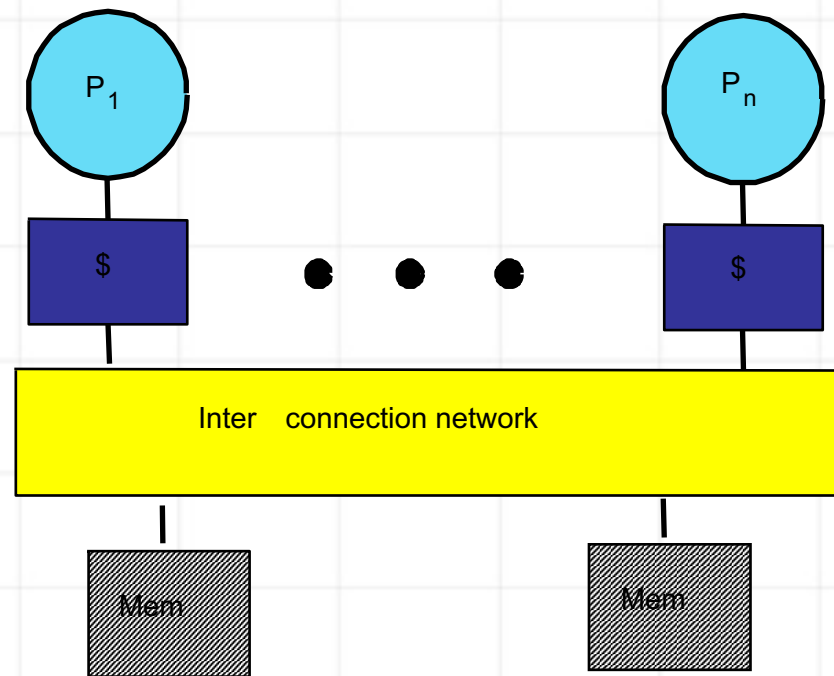
8.3 同步

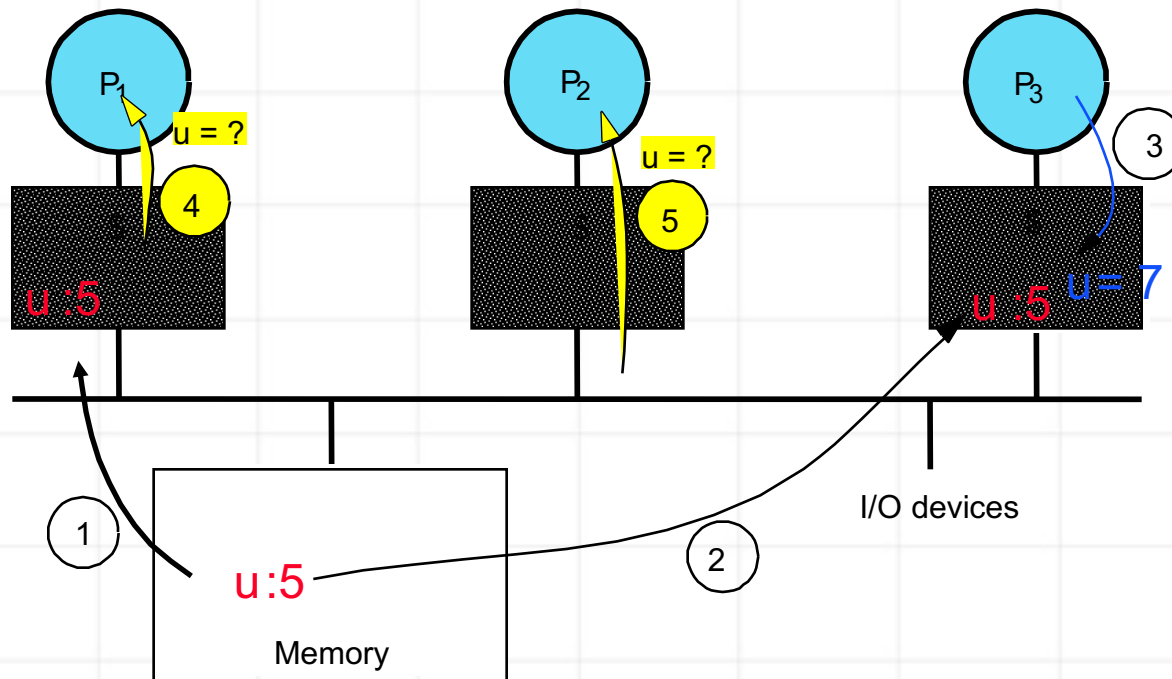
8.4 同步性能问题



8.2 多处理机Cache一致性

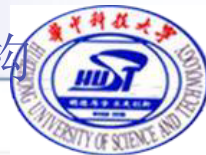
- 多个处理器共享一个存储器。





- 允许共享数据进入Cache，就可能出现多个处理器的Cache中都有同一存储块的副本，
- 当其中某个处理器对其Cache中的数据进行修改后，就会使得其Cache中的数据与其他Cache中的数据不一致





2. 存储器的一致性

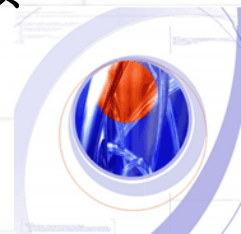
如果对某个数据项的任何读操作均可得到其最新写入的值，则认为这个存储系统是一致的。

➤ 存储系统行为的两个不同方面

- **What:** 读操作得到的是什么值
- **When:** 什么时候才能将已写入的值返回给读操作

➤ 需要满足以下条件

- 处理器P对单元X进行一次写之后又对单元X进行读，读和写之间没有其它处理器对单元X进行写，则P读到的值总是前面写进去的值。

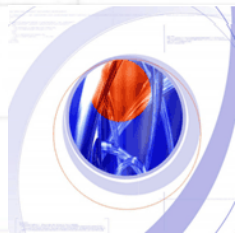




- 处理器P对单元X进行写之后，另一处理器Q对单元X进行读，读和写之间无其它写，则Q读到的值应为P写进去的值。
- 对同一单元的写是串行化的，即任意两个处理器对同一单元的两次写，从各个处理器的角度看来顺序都是相同的。（写串行化）

➤ 在后面的讨论中，我们假设：

- 直到所有的处理器均看到了写的结果，这个写操作才算完成；
- 处理器的任何访存均不能改变写的顺序。就是说，允许处理器对读进行重排序，但必须以程序规定的顺序进行写。

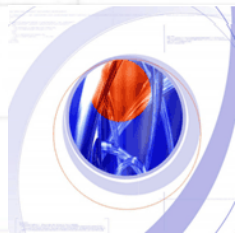




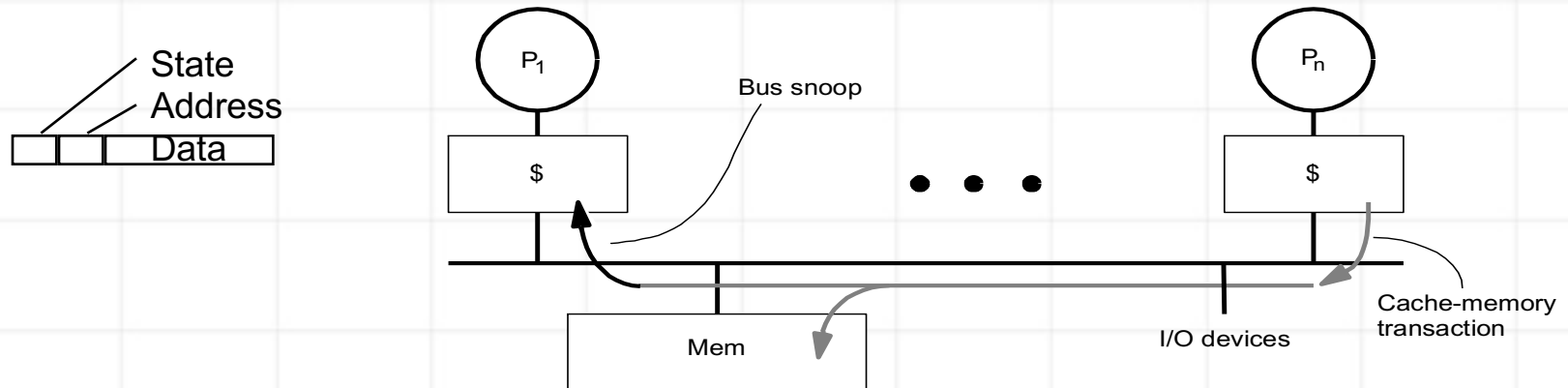
1. Cache一致性协议

在多个处理器中用来维护一致性的协议。

- 关键：跟踪记录共享数据块的状态
- 两类协议（采用不同的技术跟踪共享数据的状态）
 - 监听式协议（snooping）
 - 目录式协议（directory）

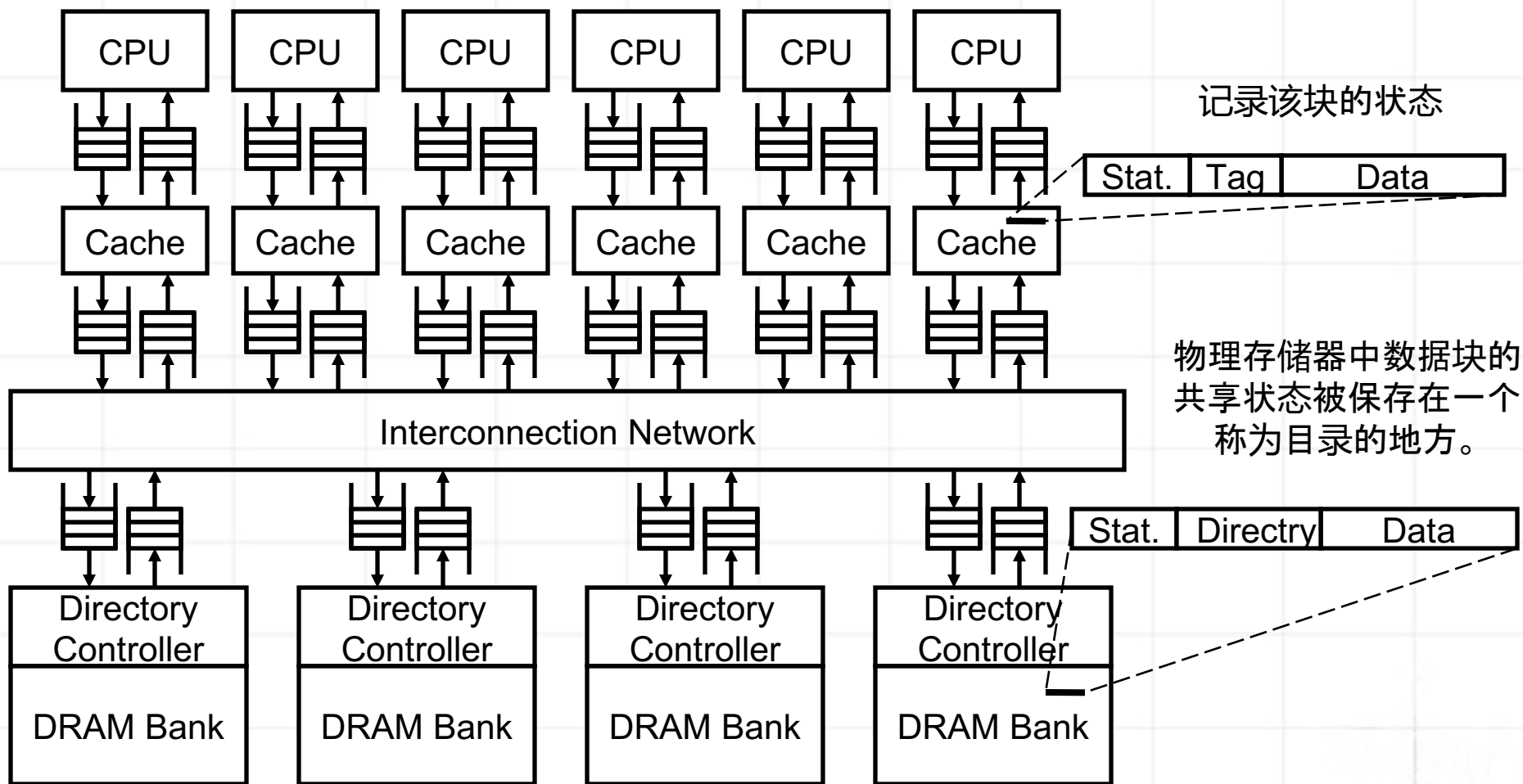
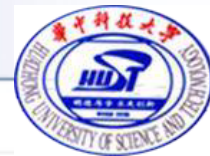


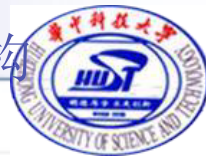
监听式协议 (snooping)



- **Cache**通常连在共享存储器的总线上，当某个**Cache**需要访问存储器时，它会把请求放到总线上广播出去，其他各个**Cache**控制器通过监听总线（它们一直在监听）来判断它们是否有总线上请求的数据块。如果有，就进行相应的操作。
- 每个**Cache**除了包含物理存储器中块的数据拷贝之外，也保存着各个块的共享状态信息。
-

目录式协议(directory)





多处理器写更新操作处理。

➤ 写作废协议

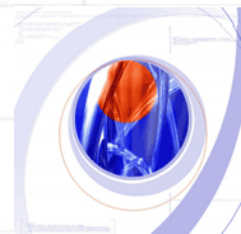
在处理器对某个数据项进行写入之前，保证它拥有对该数据项的唯一的访问权。（作废其它的副本）

➤ 写更新协议

当一个处理器对某数据项进行写入时，通过广播使其它
Cache中所有对应于该数据项的副本进行更新。

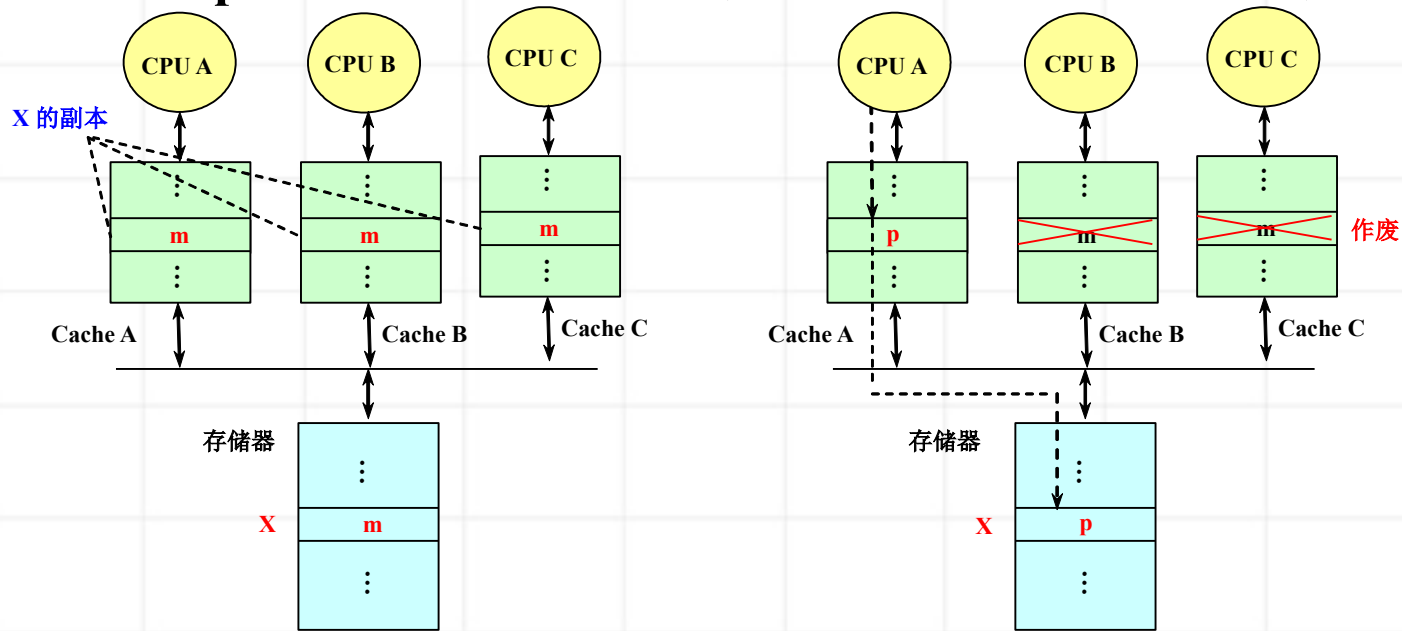
存储器写策略

写直达操作和写回操作



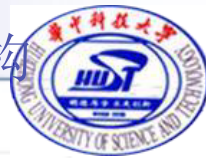
例 监听总线、写作废协议举例（采用写直达法）

初始状态： CPU A、CPU B、CPU C 都有 X 的副本。在 CPU A 要对 X 进行写入时，需先作废 CPU B 和 CPU C 中的副本，然后再将 p 写入 Cache A 中的副本，同时用该数据更新主存单元 X。



(a) CPU A 写入前

(b) CPU A 将 p 写入 X 后，作废其他 Cache 中的副本



➤ 写更新协议

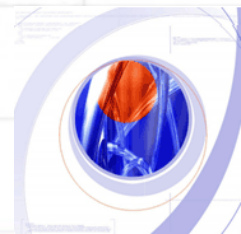
当一个处理器对某数据项进行写入时，通过广播使其它Cache中所有对应于该数据项的副本进行更新。

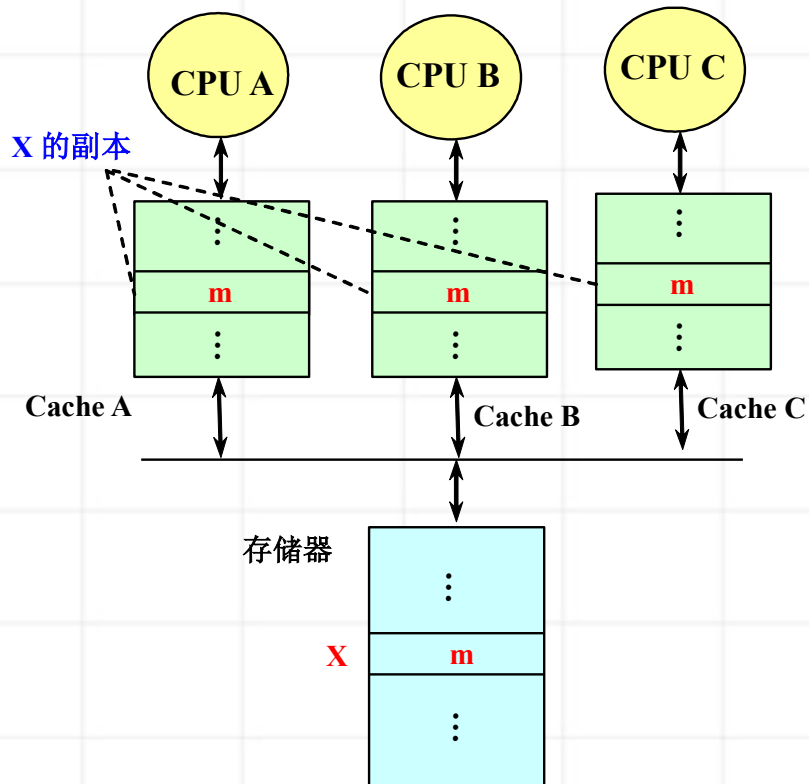
例 监听总线、写更新协议举例（采用写直达法）

假设：3个Cache都有X的副本。

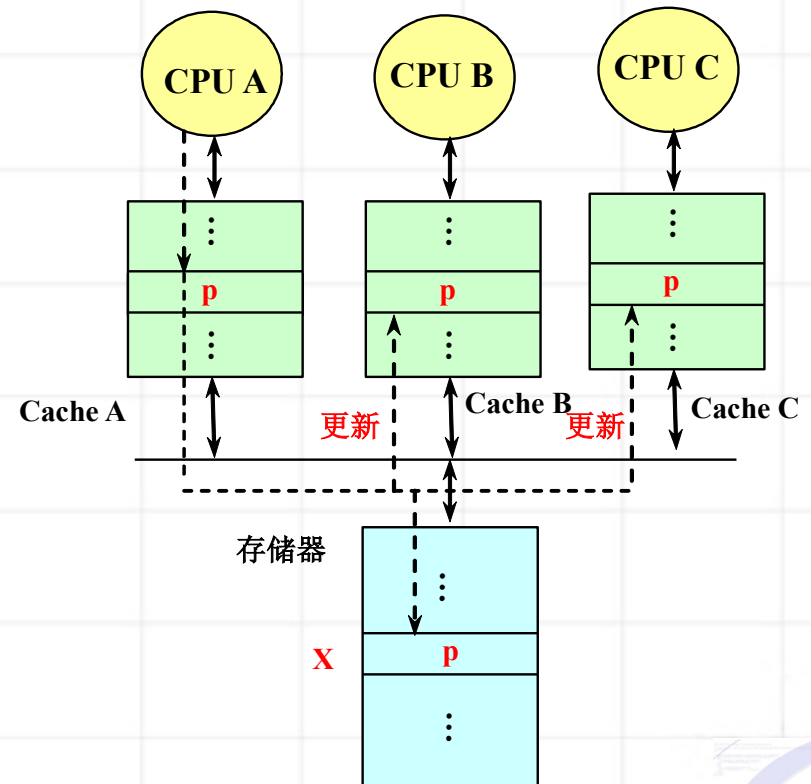
当CPU A将数据p写入Cache A中的副本时，将p广播给所有的Cache，这些Cache用p更新其中的副本。

由于这里是采用写直达法，所以CPU A还要将p写入存储器中的X。如果采用写回法，则不需要写入存储器。

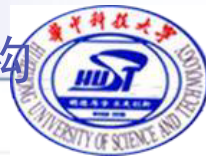




(a) CPU A 写入前



(b) CPU A 将 p 写入 X 后, 更新其他 Cache 中的副本



➤ 写更新和写作废协议性能上的差别主要来自：

- 在对同一个数据进行多次写操作而中间无读操作的情况下，写更新协议需进行多次写广播操作，而写作废协议只需一次作废操作。
- 在对同一Cache块的多个字进行写操作的情况下，写更新协议对于每一个写操作都要进行一次广播，而写作废协议仅在对该块的第一次写时进行作废操作即可。

写作废是针对Cache块进行操作，而写更新则是针对字（或字节）进行。

- 考虑从一个处理器A进行写操作后到另一个处理器B能读到该写入数据之间的延迟时间。

写更新协议的延迟时间较小。

