

计算机系统结构作业七

10.7 在基于总线的小型多处理器系统上，采用写直达 Cache。将图 10.6 与 10.7 中的基于监听 Cache 一致性协议改为写直达 Cache，且采用不按写分配。画出状态转换图。

解答：每个数据块的状态为如下两种状态中的一种：

- (1) 无效 (Invalid, I)：表示 Cache 中该块的内容无效。显然，所要访问的块尚未进入 Cache
- (2) 有效 (Valid, V)：表示当前 Cache 中概括的内容有效。

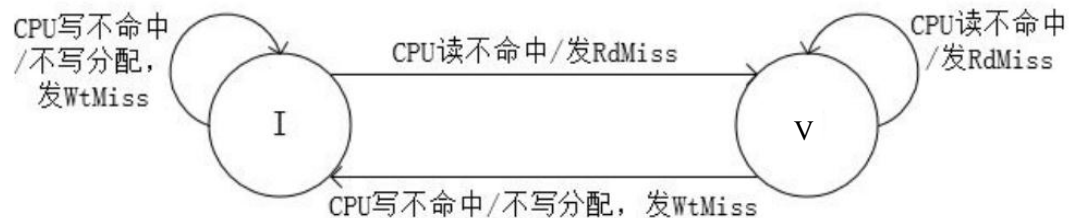
因为采用写直达的方式，存储器中的内容始终为最新的，即与有效 Cache 中的数据一致。

1) 响应来自处理器的请求

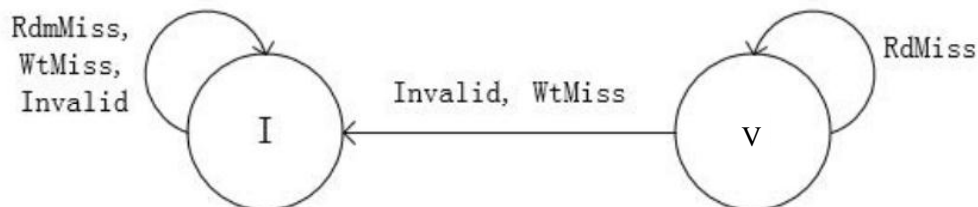
① 不发生替换



② 发生替换



2) 响应来自总线的请求



(1) 当前状态为 I:

当 CPU 要进行读访问时，由于所要的块尚未调入 Cache，所以发生读不命中，需要向总线发送 RdMiss 消息，调入该块后，将其状态改为有效状态 V。当 CPU 要进行写访问时，由于所要访问的块尚未调入 Cache，所以发生写不命中，需要向总线发 WtMiss 消息，因为采用不按写分配，此时无需调入该块，直接修改内存中的值，向总线发送 Invalidate 消息，作废所有其他 Cache 中的

副本，此时仍然处于状态 I。

(3) 当前状态为 V:

当 CPU 要进行读访问时，如果命中，则状态不变；如果读不命中，那么向总线发送 RdMiss 消息，调入该块，状态仍为 V。当 CPU 要进行写访问时，如果命中，需要向总线发送 Invalidate 消息，作废所有其他 Cache 中的副本，状态不变；如果写不命中，则向总线发送 WtMiss 消息，不调入块，作废其他副本，状态不变。如果总线发出 RdMiss 消息，状态不变。如果总线发出 WtMiss 或 Invalidate，作废副本，状态变为 I。

PS: 本题重点在于要明确写直达和不按写分配的概念。其实也有很多同学仍旧按照教材中的实例来进行写，即采用 M/S/I 三种状态，其实也是完全可以的，但是需要说明的是，必须标明 M 状态的意义，因为在写直达的情况下，不存在这 Cache 与存储器中内存不一致的情况。在这时，M 应该指的是只有当前处理器 Cache 和存储器中的数据为最新的，S 仍指多个处理器(≥ 2)共享数据。

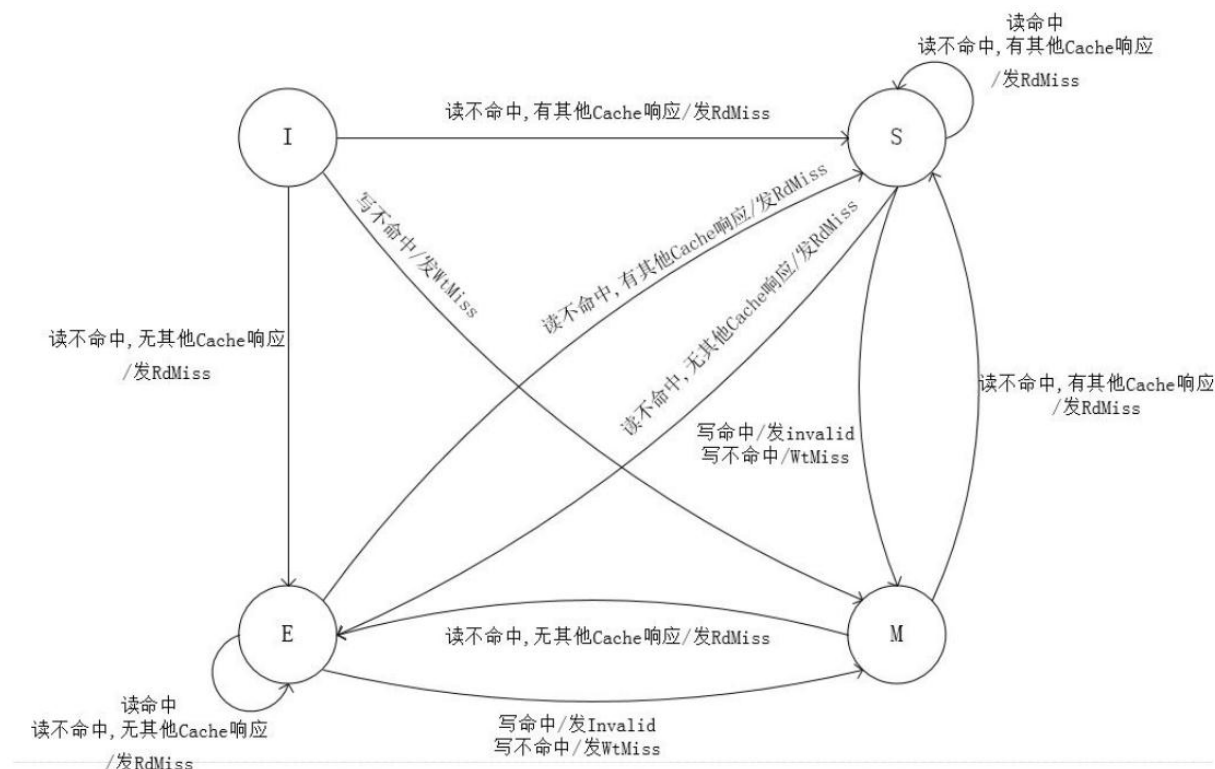
2. Suppose there is a 3-state protocol for write back cache. The states include (1) Modified (one-cache, dirty); (2) Shared (possibly many caches, clean); and (3) Invalid (no caches). You will be modifying the 3-state protocol by splitting the Shared state into two states: Shared, which is a shared, clean state; and Exclusive, which is a private, clean state.

1. Draw the 4-state protocol diagrams

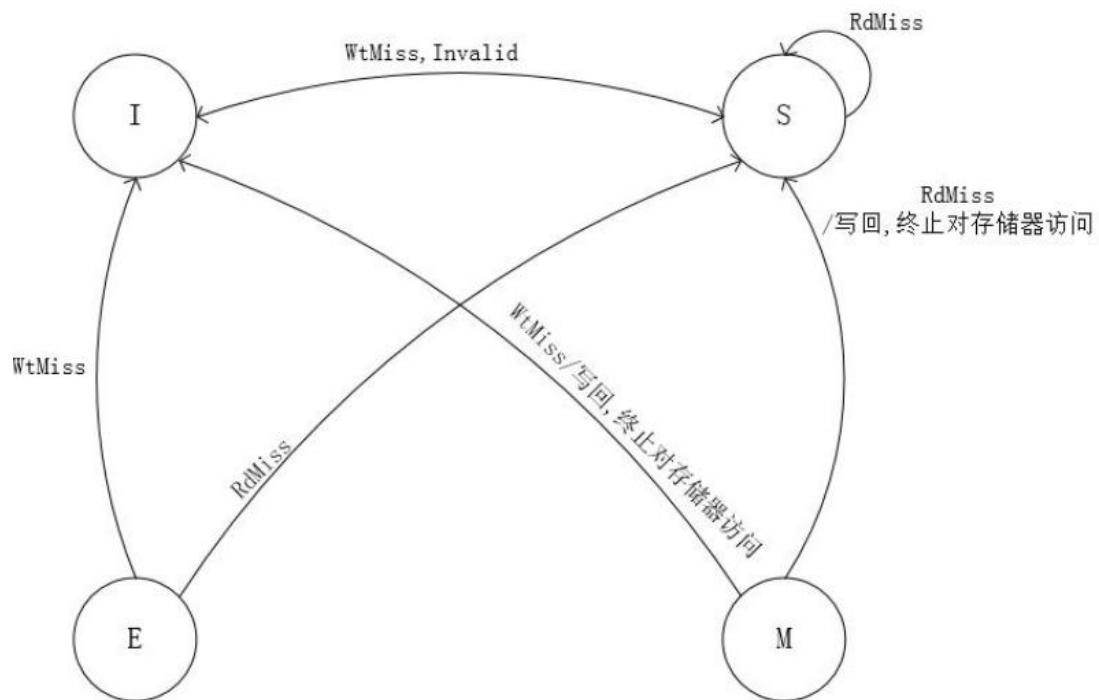
2. Describe why the split provides performance benefit.

解答:

(1) 来自处理器请求



来自总线请求



- (2) 4-状态协议的好处在于区分了独享和共享的状态。有点是出于独享状态写命中时，不用向总线发送信号，节省了 Cache 间通信的代价；但独享状态 Cache 较少而且写命中较少时没有性能上的提高，反而因为多了一个状态，状态转移时需要增加一次判断，即当读不命中发送 **RdMiss** 之后，要根据是否处于独享状态来进行状态转移，增加了开销，对性能不利。