北京大学计算机学院本科生课程

计算机组成与系统结构实习 Review 6

北京大学微处理器研发中心

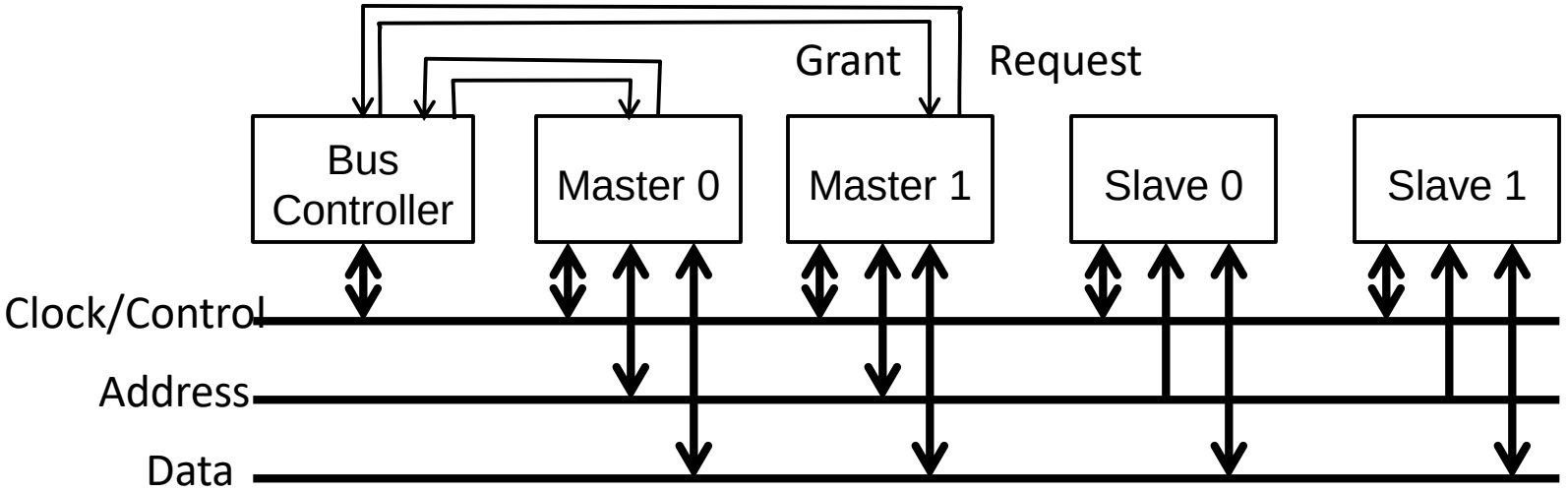
易江芳

2022-11-14

主要内容

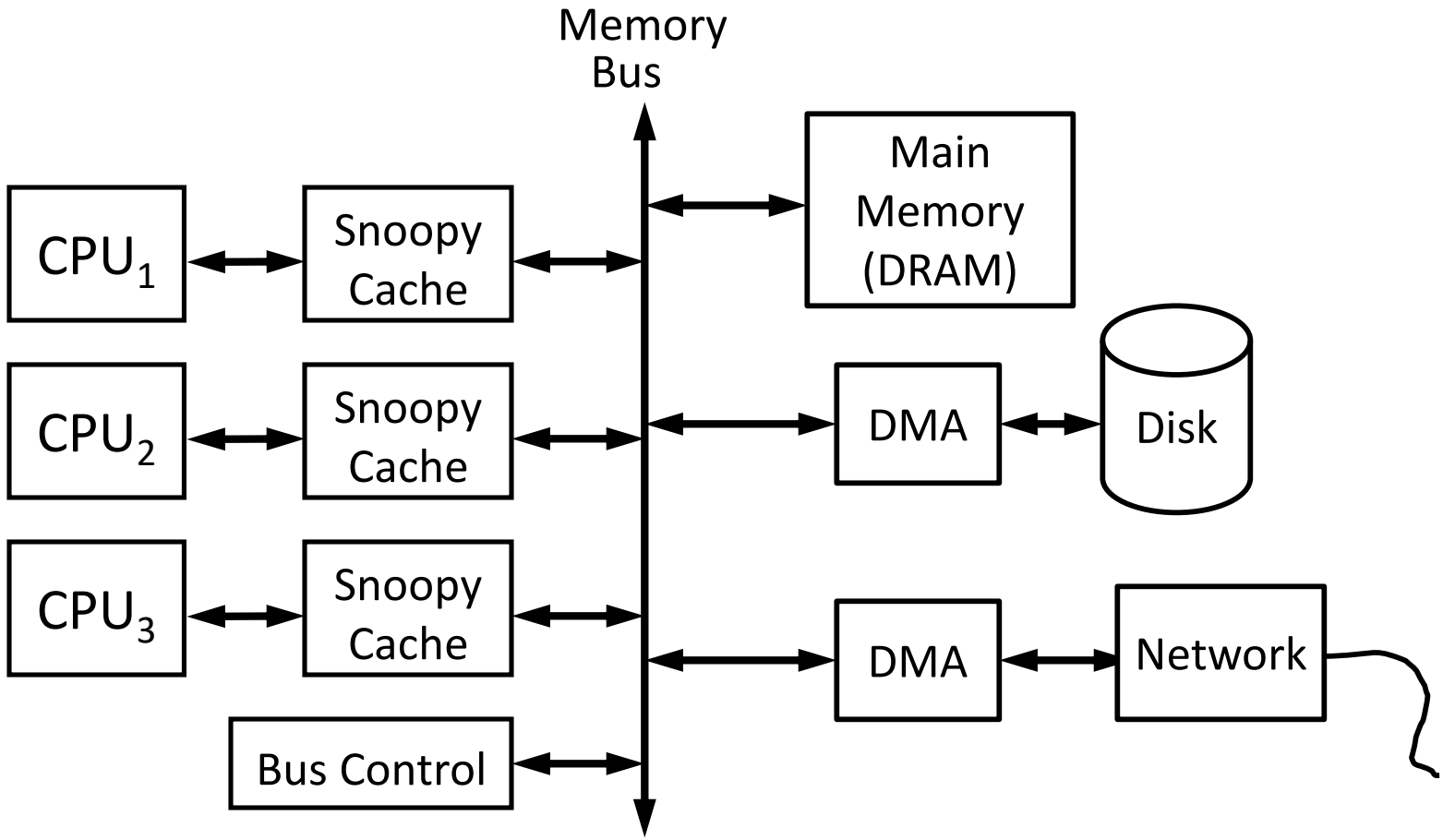
- 缓存一致性
  - 共享存储多处理器结构中存在多个Cache，不同Cache之间可能存在不一致，和内存之间也可能存在不一致
  - 造成同一时刻不同处理器核访问的数据内容不一致
  - 采用监听总线的方式保证缓存之间的数据一致性

总线互连



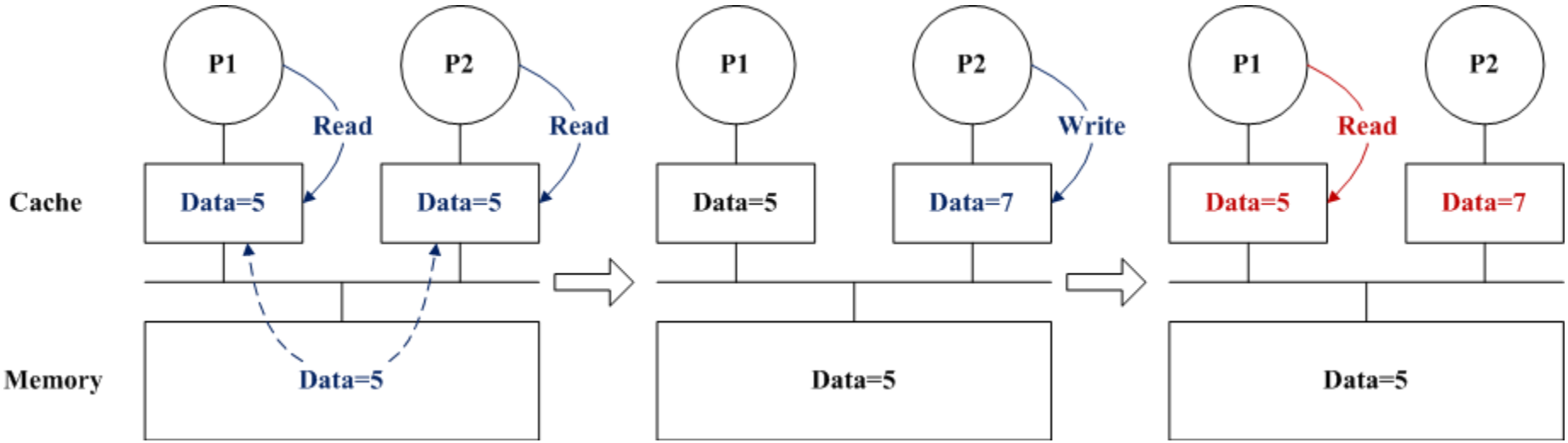
- 总线是一组共享信息的线的总和
- 任何一个时刻，只有一个master可以在总线上发送请求
- 任何一个时刻，所有的slave都可以监测到总线请求，并选择性的对其做出响应
- 通过Bus Controller 来判断是否响应Master的请求

共享存储多处理器



- Shared-Memory Multiprocessor
- 通过snoopy（监听总线）来保持高速缓存的数据一致性

Cache一致性


P1
P2
Read
Read
Cache
Data=5
Data=5
Memory
Data=5
P1
P2
Write
Data=5
Data=7
Data=5
P1
P2
Read
Data=5
Data=7
Data=5

Cache一致性协议

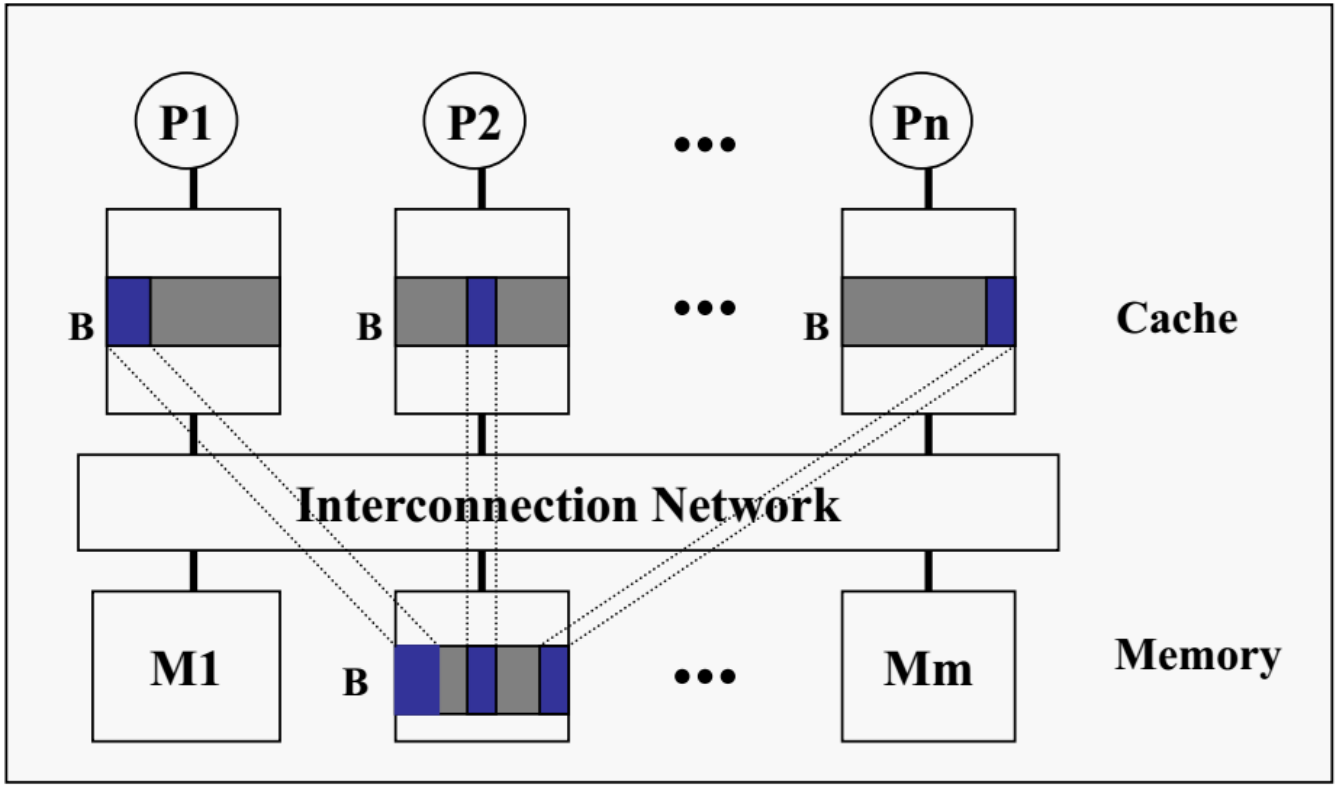
- 通俗点说，就是一种把新值传播到其他处理器的机制

- 涉及到的问题
  - 谁可以产生新值： single writer vs. multiple writers
  - 如何传播新值：write update vs. write invalidate
  - 如何相互通信：snoopy vs. directory

- Cache一致性协议决定系统为维护一致性所做的具体动作，因而直接影响系统性能

单写和多写

- 单写：任一时刻只有一个处理器能写某共享资源
- 多写：多个处理器同时写某共享资源



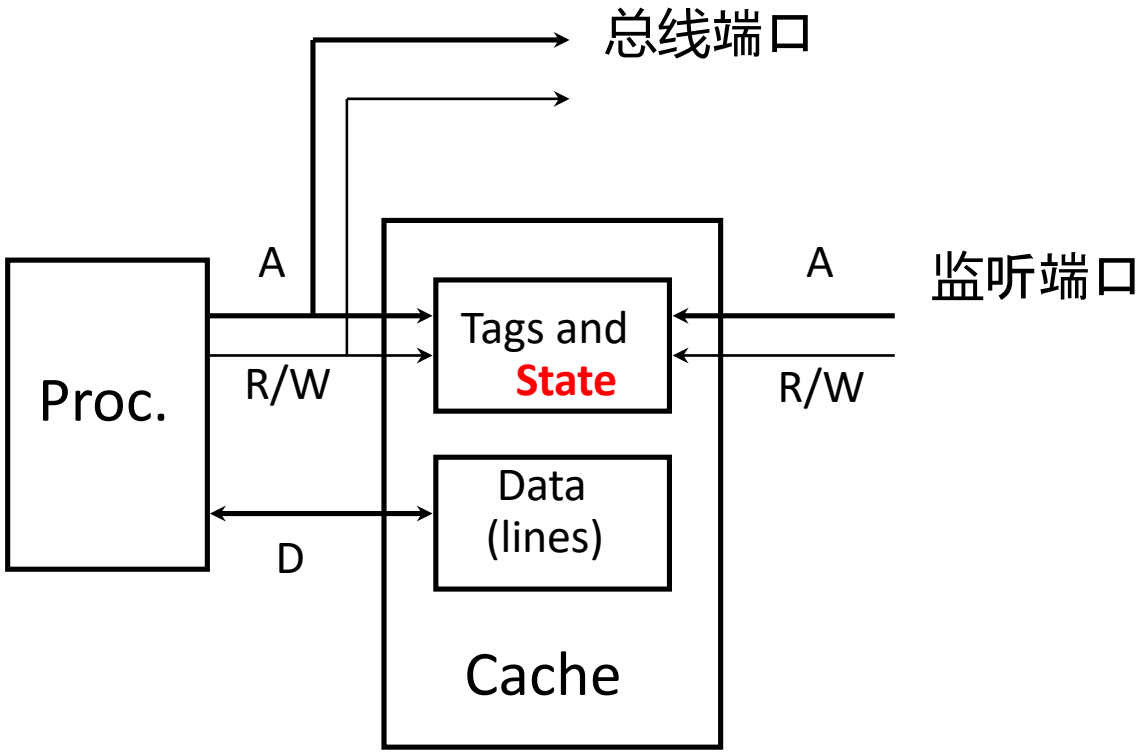
写无效 vs. 写更新

- Write invalidate
  - 当一个处理器更新某共享资源（如存储块或存储页）时，通过某种机制使该共享资源的其它备份无效
  - 当其它处理器访问该共享资源的备份时，访问失效，需要到内存中取得有效数据

- Write update
  - 当一个处理器更新某共享资源（如存储块或存储页）时，会把更新内容传播给所有拥有该共享资源备份的处理器
  - 当其它处理器访问该共享资源的备份时，读取的就是最新的数据

Snoopy Cache, *Goodman 1983*

- 动机：如果每个Cache都能看到其他Cache的访存请求，它们就能做正确的事情
- Snoopy Cache 需要支持双端口

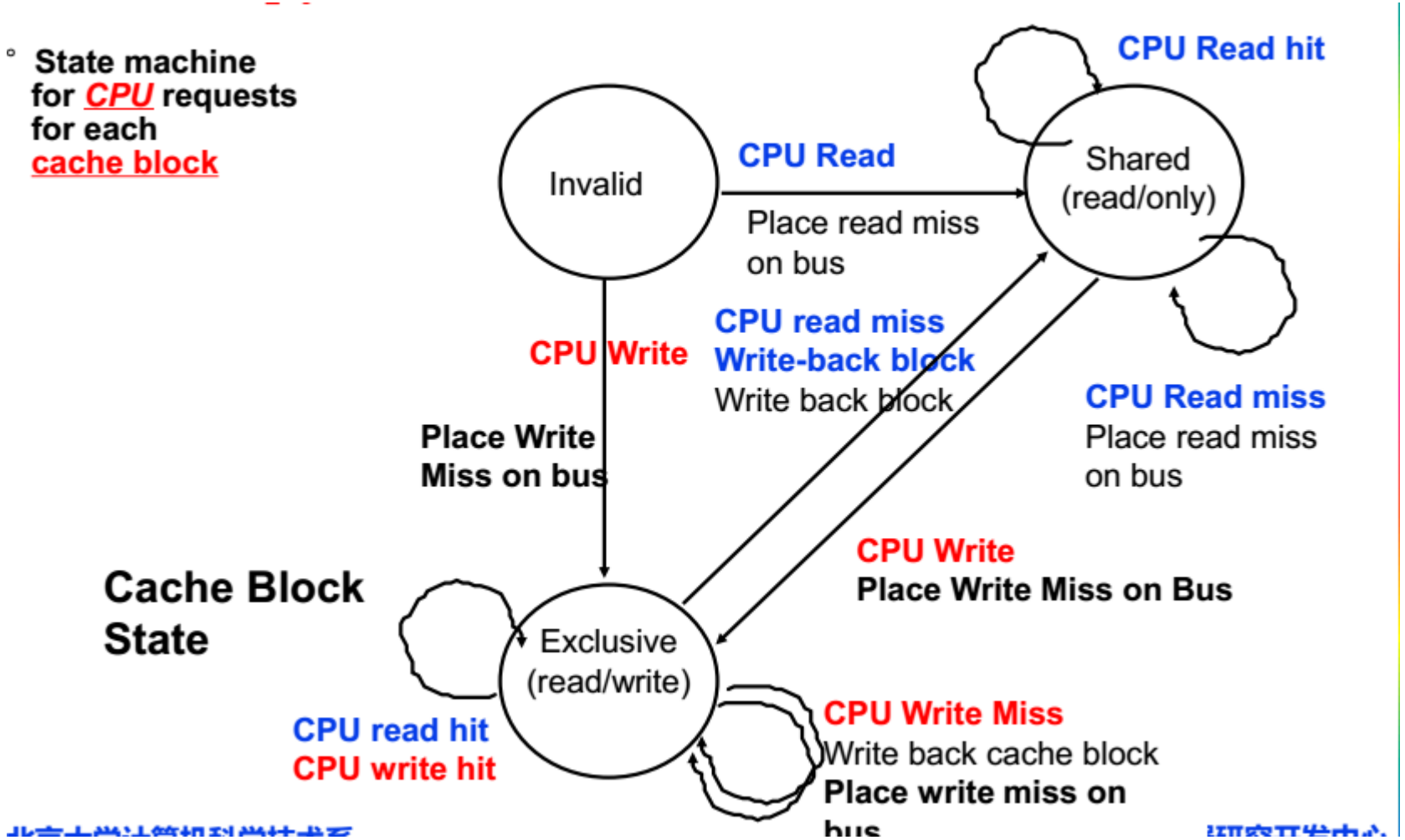


Snoopy 协议的三种状态（MSI）

- 每个Cache行 的一致性状态
  - M (modified或exclusive)，表明对应cache行的数据已被当前处理器核修改
  - S (shared)，表明对应cache行数据为clean，与内存中数据保持一致，目前被多个处理器核共享，可以读取
  - I (invalid)，表明对应cache行的数据是无效的

snoopy cache 状态机 I

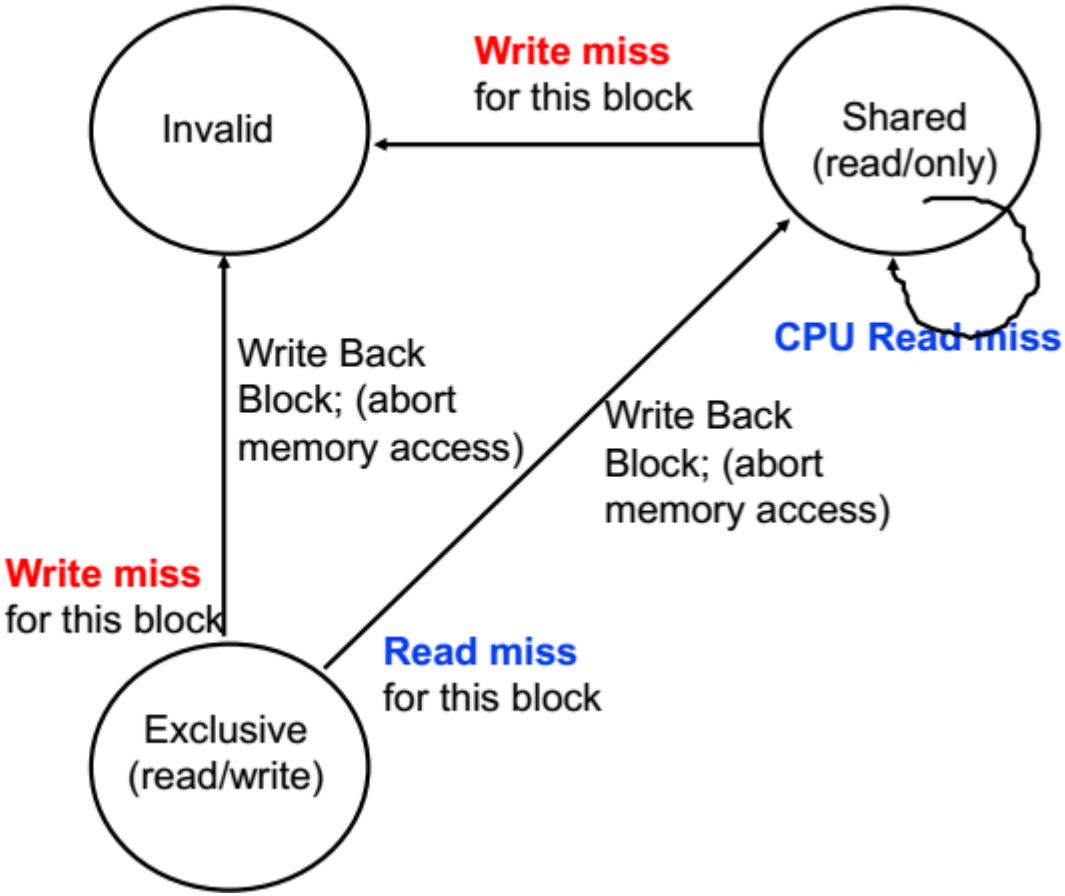
本地 CPU 对 Cache 数据块进行访问，Cache 数据块的状态变化



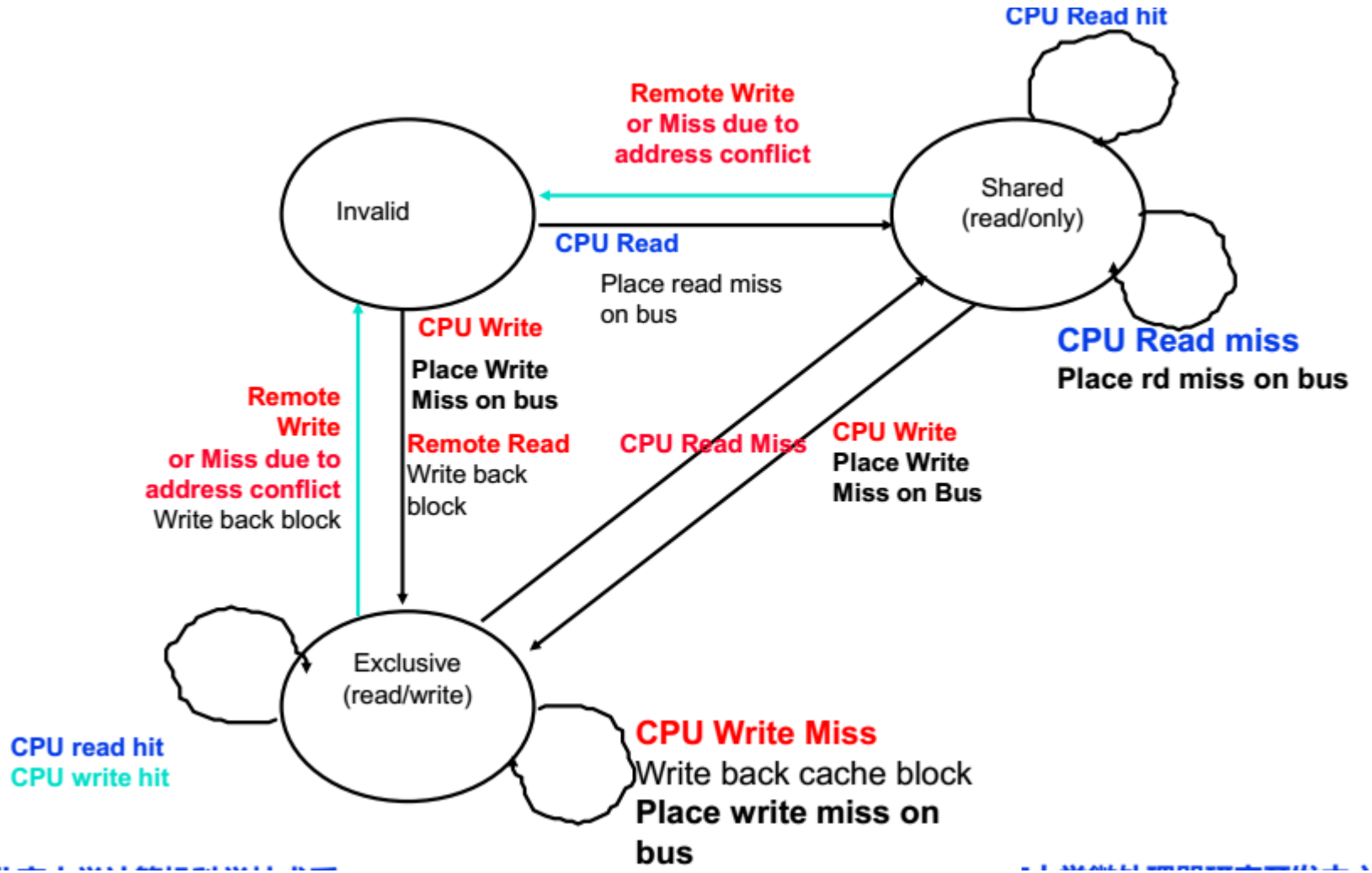
snoopy cache 状态机 II

通过总线上广播消息后，其他处理器中数据状态变化

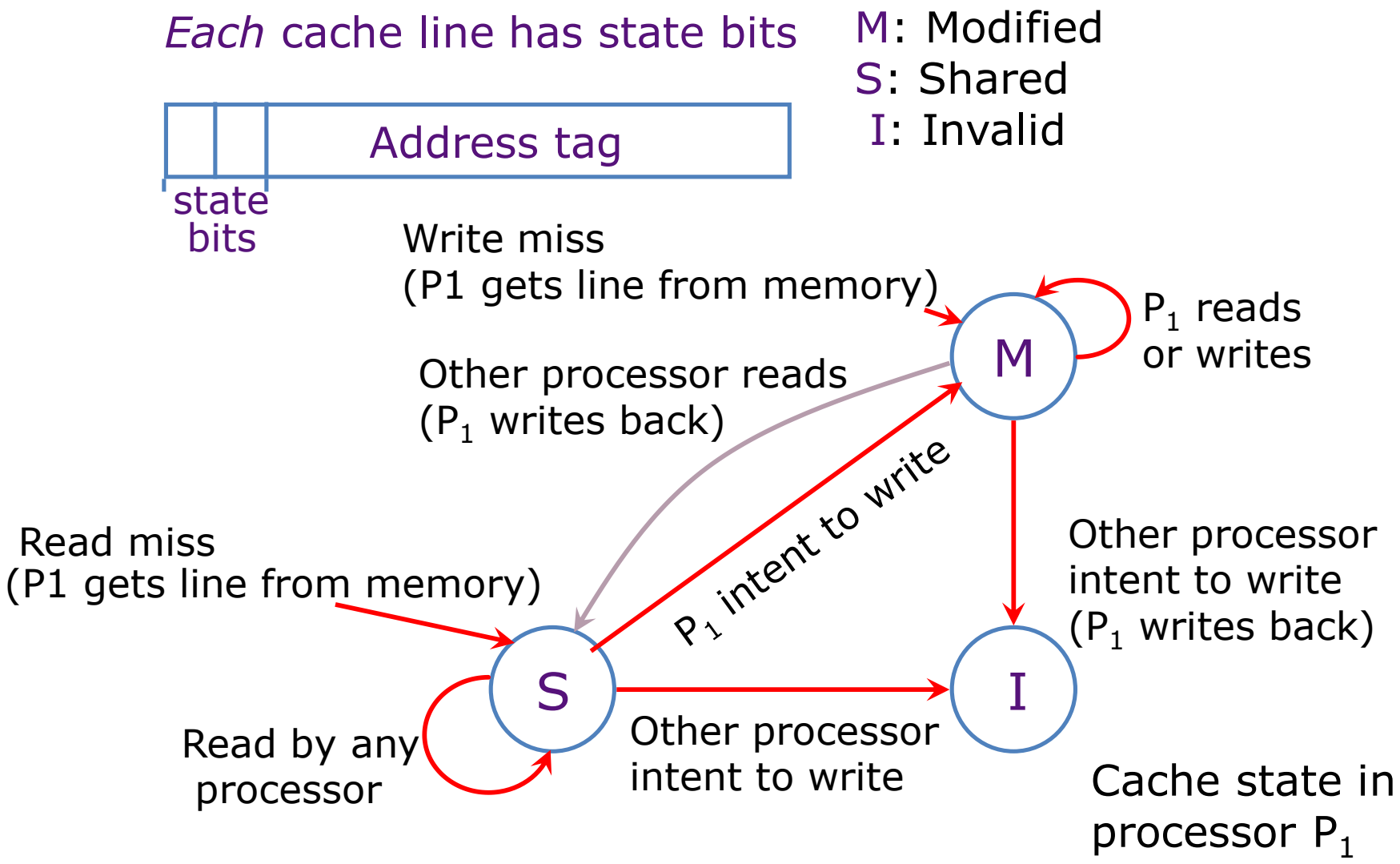
- State machine for *bus* requests for each cache block



合并上述状态图


CPU Read hit
Remote Write
or Miss due to
address conflict
Invalid
Shared
(read/only)
CPU Read
Place read miss
on bus
CPU Write
Place Write
Miss on bus
CPU Read miss
Place rd miss on bus
Remote
Write
or Miss due to
address conflict
Write back block
Remote Read
Write back
block
CPU Read Miss
CPU Write
Place Write
Miss on Bus
Exclusive
(read/write)
CPU read hit
CPU write hit
CPU Write Miss
Write back cache block
Place write miss on
bus

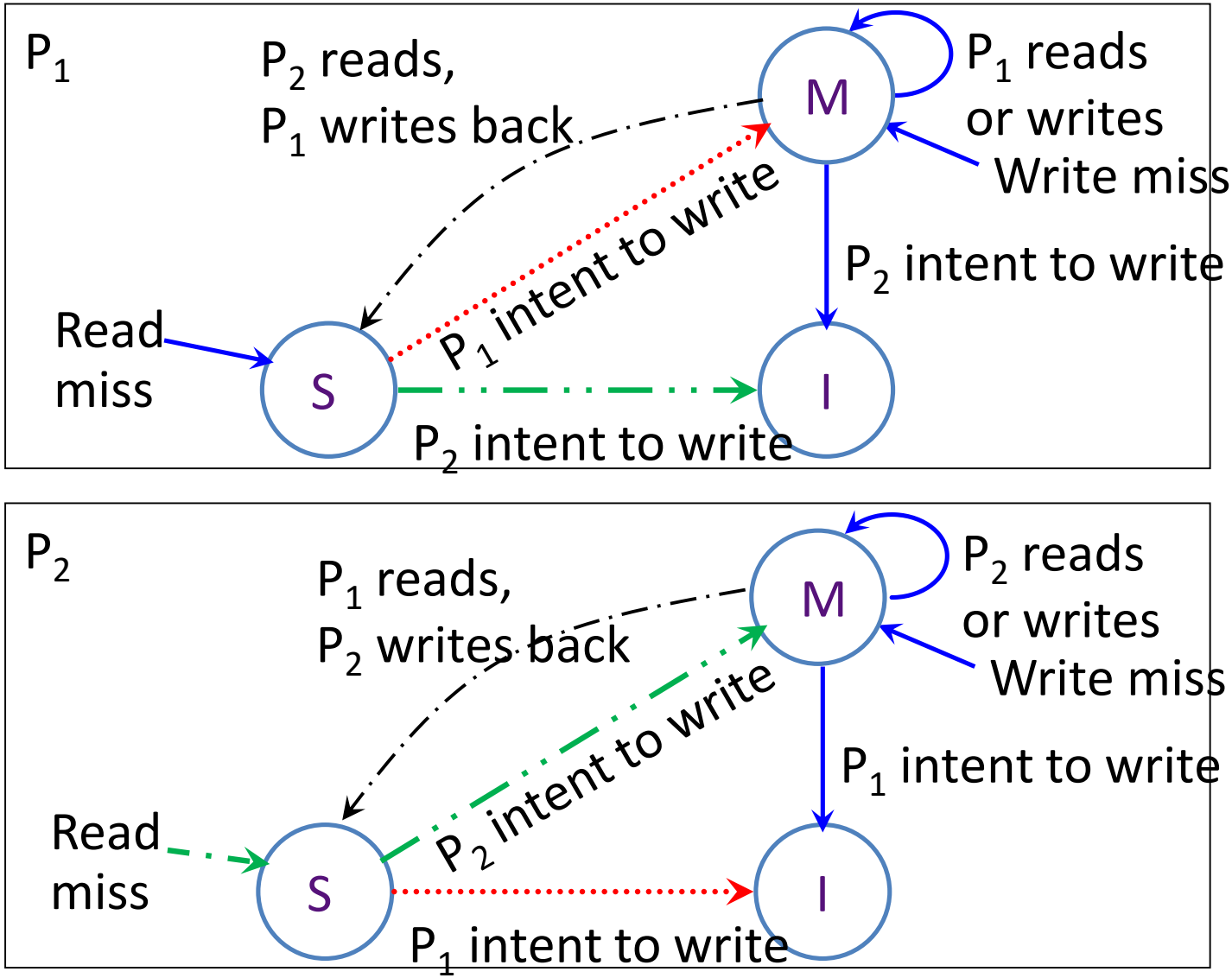
另一种画法


Each cache line has state bits
M: Modified
S: Shared
I: Invalid
Address tag
state bits
Write miss
(P1 gets line from memory)
M
P1 reads or writes
Other processor reads
(P1 writes back)
P1 intent to write
Read miss
(P1 gets line from memory)
Other processor intent to write
(P1 writes back)
S
I
Read by any processor
Other processor intent to write
Cache state in processor P1

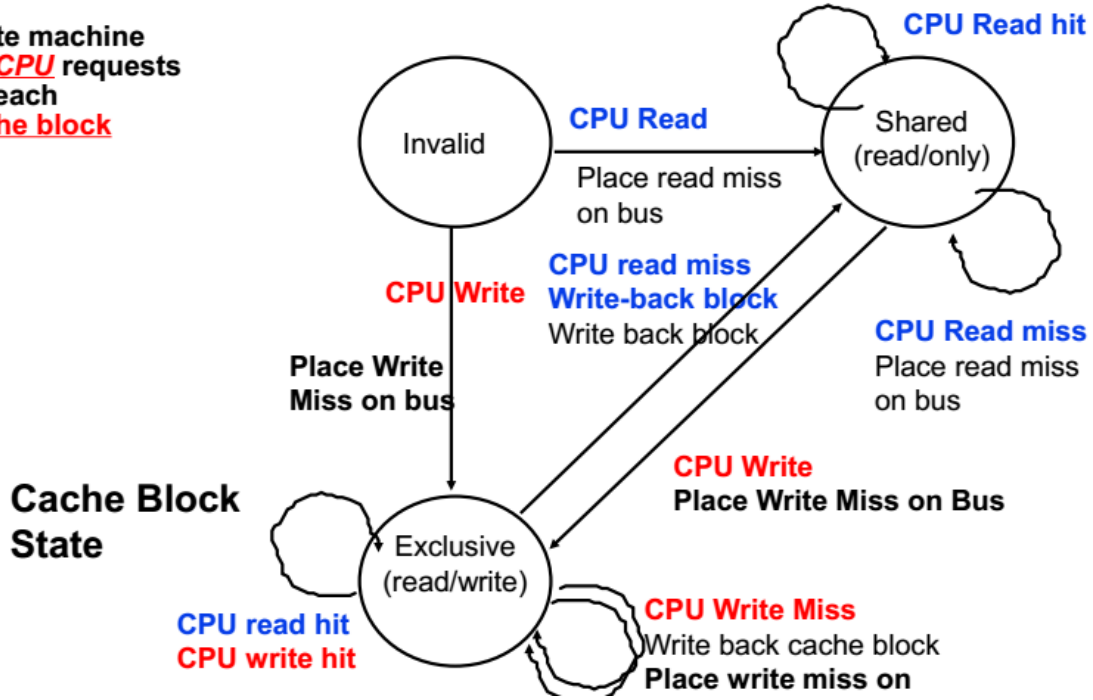
举例：读写都是同一个cache line

读写序列如下

P_1 reads
P_1 writes
P_2 reads
P_2 writes
P_1 reads
P_1 writes
P_2 writes
P_1 writes

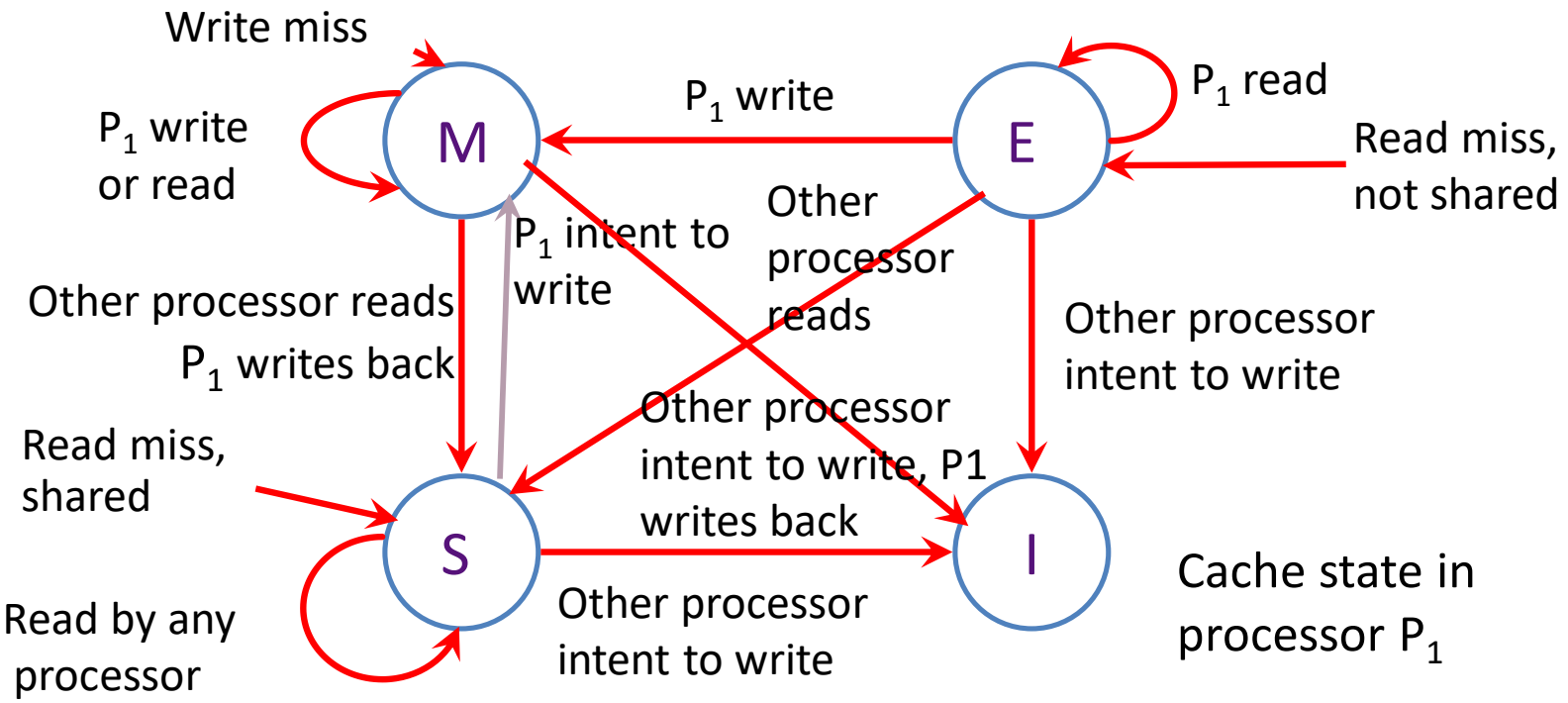


观察与思考



- I态 -> S态：存在只有唯一Cache存有副本的情况
- 增加一个状态E（Exclusive），表示当前数据有效，并且数据和内存中的数据一致，且数据只存在于本Cache中

MESI



- E 和 S 的区别：只有唯一Cache存有副本
- E 和 M 的区别：数据是否为dirty
- E态下，无需通过miss通知其他cache；数据无需写回内存
- AMD的Opteron处理器从MESI中演化出的MOESI协议
- Intel的core i7处理器使用从MESI中演化出的MESIF协议

False Sharing

- Cache line 中包含了多个 word
- Cache 一致性协议是以 line 为粒度进行数据共享，而非以 word 为粒度
- 假设 M1 对 $word_i$ 进行写操作，M2 对 $word_j$ 进行写操作，$i \neq j$，但 $word_i$ 和 $word_j$ 在同一个 line 中

- 会发生什么问题吗？^v^

Cache miss in SMP

- Cache 性能的影响因素：miss rate，miss penalty, hit time
  - 单核下的 cache miss (3C)
    - 义务失效 (Compulsory miss)
    - 容量失效 (Capacity miss)
    - 冲突失效 (Conflict miss)
  - 一致性协议导致的 cache miss（the 4th C）
    - 有时也称为通信失效 (Communication miss)
  - 由通信失效带来的性能损失
    - 数据被置为无效 (invalidated) 后，引起后续的cache miss（这在单核下可能是不存在的）

Coherency Misses

- 由一致性协议引发的 true sharing miss
  - 由于改写共享数据块导致其他Cache中被置无效
  - 其他核心需要读取该共享数据块
  - 即使cache line容量为1 word，也应引起 cache miss

- 由一致性协议引发的 false sharing miss
  - 也是因为改写数据块导致被置无效，但这并不是需要共享的那个数据引起的
  - 因此，这样的置无效操作，并不会传播新值（被改写的数值），还增加了额外的 cache miss
  - 如果 cache line容量为1 word，这样的 cache miss不会发生

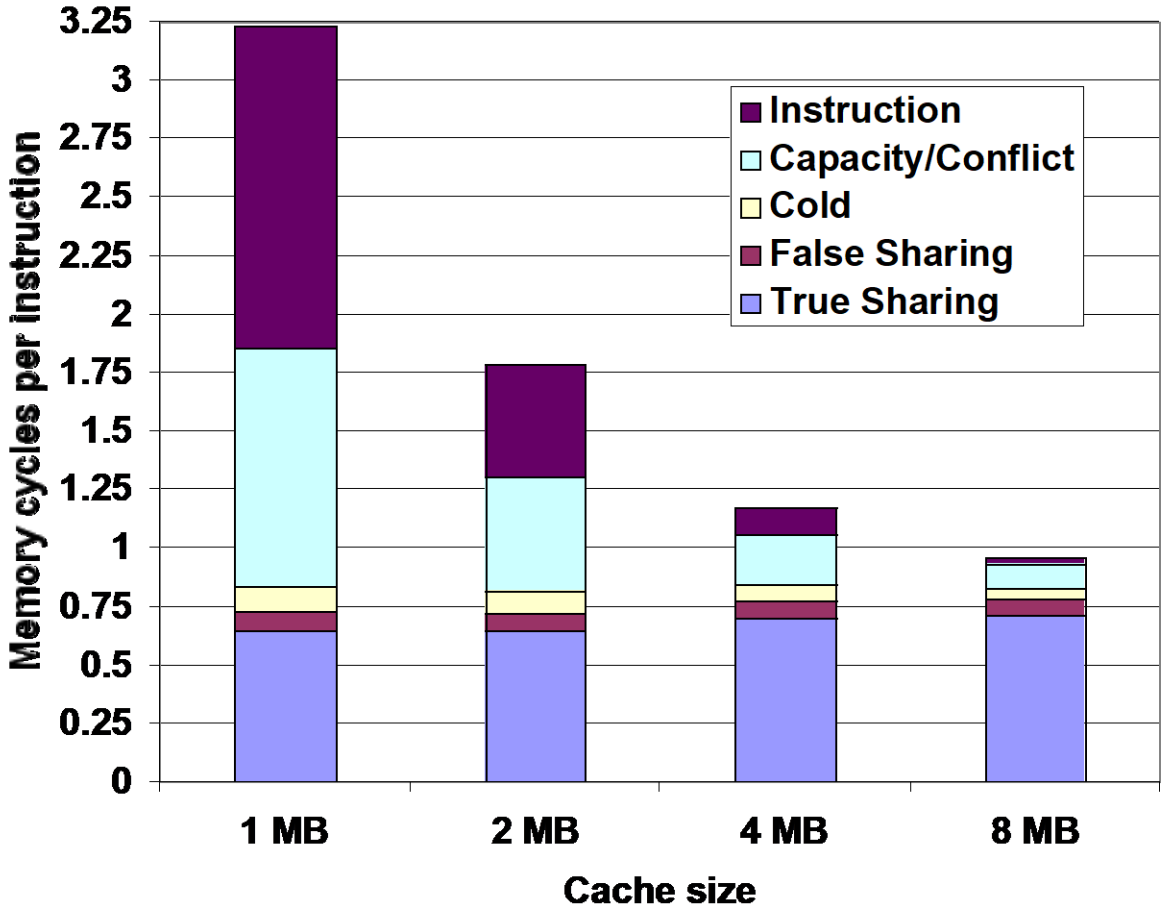
Example: True v. False Sharing v. Hit?

- MSI protocol
- 假设数据x1 和 x2 在同一个数据块中，p1和p2之前都读过x1 和 x2

Time	P1	P2	True, False, Hit? Why?
1	Write x1		True miss; x1 invalidate in P2
2		Read x2	False miss; x1 irrelevant to P2
3	Write x1		False miss; x1 irrelevant to P2
4		Write x2	True miss; x2 not writeable
5	Read x2		True miss; x2 invalidate in P1

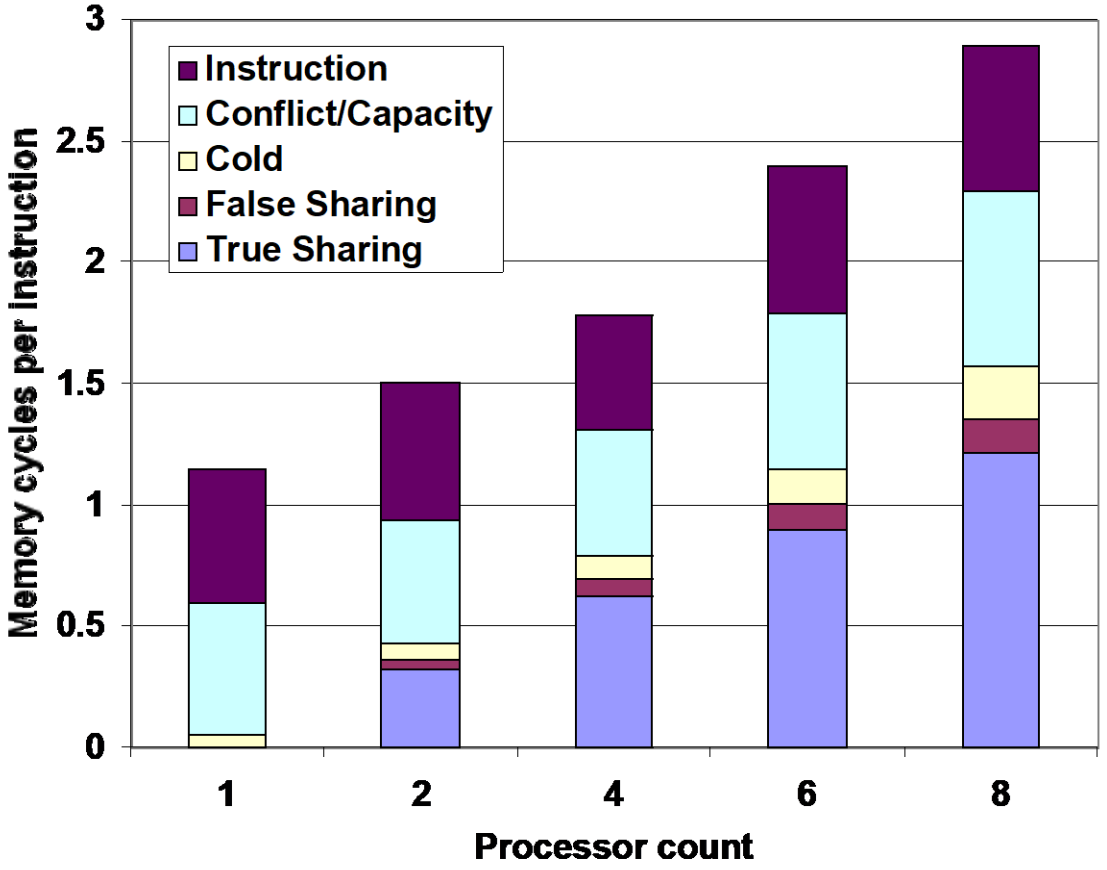
使用商业workload对4核心处理器进行性能评估：OLTP, Decision Support (Database), Search Engine

- 单核的cache miss随着cache容量的增大而改善
- Coherence miss并不随cache容量的变化而变化
- L3 Cache：1MB~8MB



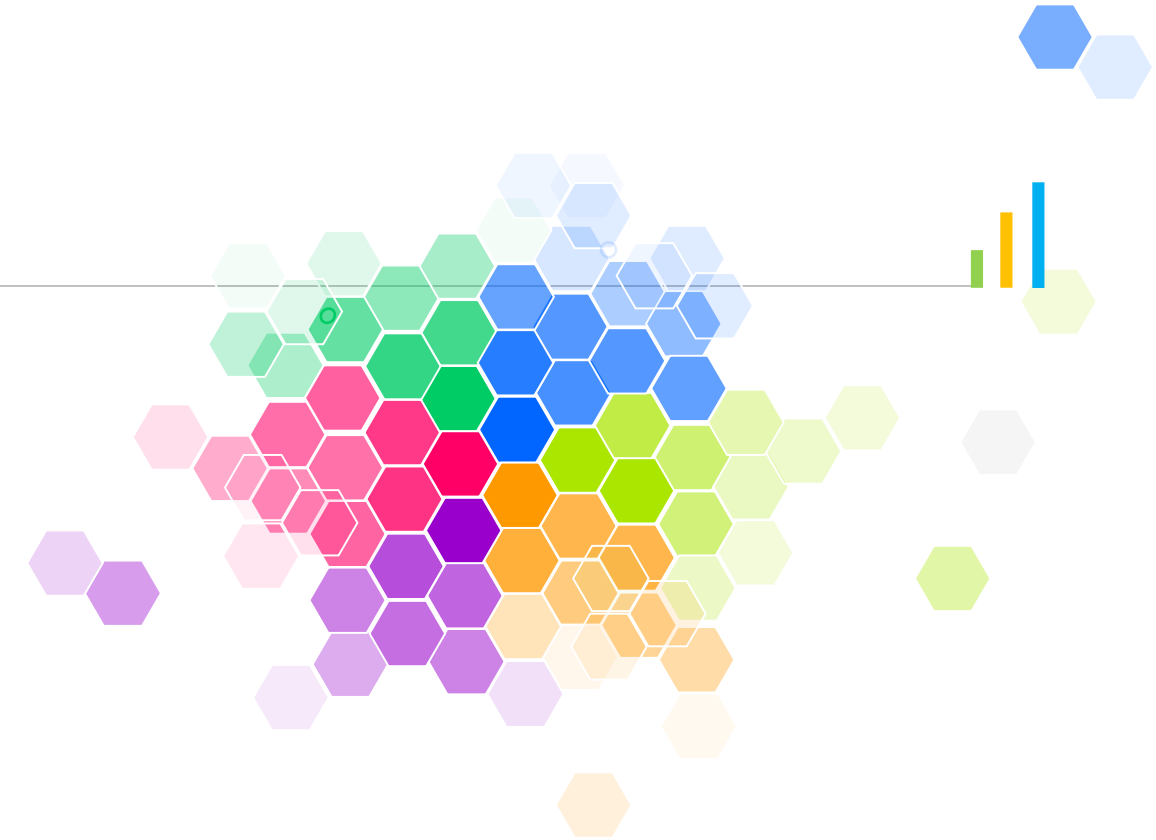
使用商业workload对固定2MB cache 处理器进行性能评估：OLTP, Decision Support (Database), Search Engine

- 随着核数的增加，coherence miss在不断增大
- True sharing miss和 false sharing miss都增大



小结

- 缓存一致性是在共享存储多处理器(SMP)中，由于缓存结构的存在引起的数据不一致
- 采用缓存一致性协议来保持数据的一致性
- 缓存一致性协议的分类
- 侦听（snoopy）协议的原理
  - MSI
  - MESI
- 由于引入一致性协议，cache 性能的影响因素变化
  - Coherence miss
  - True sharing miss 和 false sharing miss
  - 随核数的变化而变化，与 cache 容量没有关系



欢迎提问

侦听（snoopy）协议

- Write Invalidate 协议
  - 单个写者，多个读者
  - 本地核对共享数据进行改写：其他核的cache中的数据副本被置为invalidate
- Write update 协议（也叫write broadcast）
  - 本地核对共享数据进行改写：在总线上进行广播，其他核侦听总线，并更新相应的数据副本
  - 通常都是write through cache，内存总是最新的