

HOMEWORK SOLUTION № 7

Problem 1

对于表 5-12 的理解需要参考 5.2 题。

当 P0 发生缺失而其它处理器 P1 有该块时：

1. P1 处于 M 态，则 P0 从 P1 获取数据；
2. P1 处于 S 态，则 P0 从 memory 获取数据。

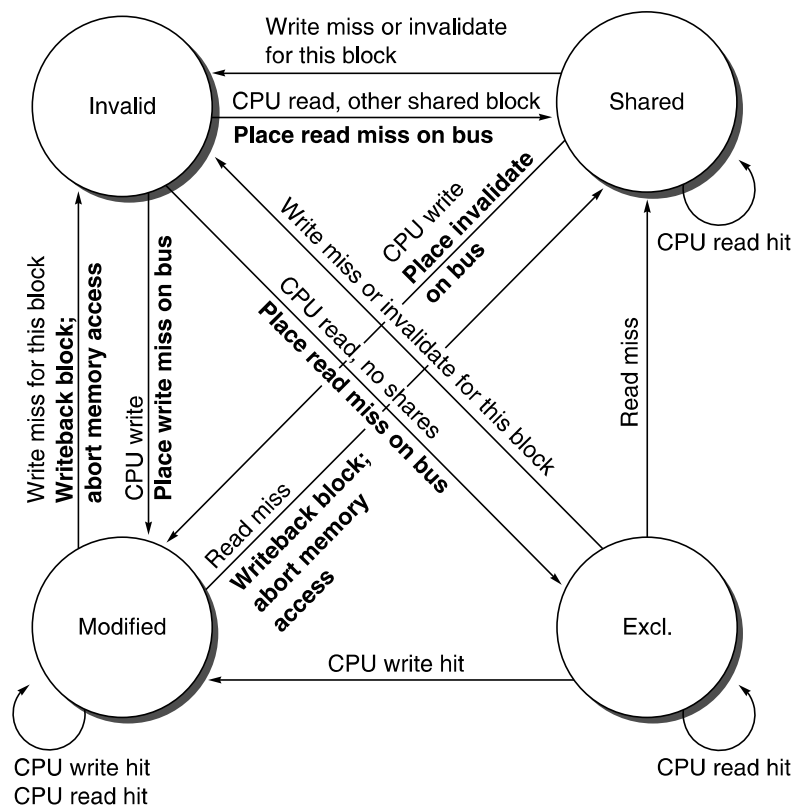


图 1: MESI 状态图

a

P0: read 100, 读缺失, 从内存中读取, 代价 N_{memory} , MSI 进入 S 态, MESI 进入 E 态。

P0: write 100, 写命中, MSI 发送 invalidate 信号, 代价 $N_{invalidate}$, MESI: E \rightarrow M, 代价 0。

MSI: $100 + 15 = 115$ stall cycles

MESI: $100 + 0 = 100$ stall cycles

b

P0: read 120, 读缺失, P3 处于 S 态, 从 memory 获取数据, 代价为 N_{memory} , MSI 和 MESI 进入 S 态

P0: write 120, 写命中, MSI 和 MESI 都发送 invalidate 信号, 代价 $N_{invalidate}$

MSI 和 MESI: $100+15 = 115$ stall cycles

c

P0: read 100, 问题 a;

P0: read 120, 问题 b.

MSI 和 MESI: $100+100=200$ stall cycles

d

P0: read 100, 问题 a;

P1: write 100, 写缺失, 从内存读取, 代价 N_{memory} , (写缺失产生 BusRdX 信号, 会将其它 cache 上的块置为 I 态, 无需发送 invalidate 信号)

MSI 和 MESI: $100+100=200$ stall cycles

e

P0: read 100 && P0: write 100, 问题 a

P1: write 100, 写缺失, 从 P0 的 cache 读取脏块的代价为 N_{cache} , P0 写回代价 $N_{writeback}$

MSI: $100 + 15 + 40 + 10 = 165$ stall cycles

MESI: $100 + 0 + 40 + 10 = 150$ stall cycles

误区

1. 在 MESI 中如果某个块只存在于某个 cache 中且未修改时它理所当然的处于 E 态?

也有可能处于 S 态: 如果原先在其它 cache 中存在这个块 (此时只能是 S 态), 随后其它 cache 中该块被替换, 就会出现这种情况。在本题中, 题目标明 P3 的 120 处于 S 态就不要自作聪明的认为它处于 E 态。

2. 在总线事务为 BusRd 或 BusRdX, 所请求的块在其它 cache 上存在且处于 S 态时可以直接进行 cache-to-cache 传输?

针对 BusRd 和 BusRdX 事务, 只有请求块处于 M 态 (MESI 的 E 态应该也可以) 时才会存在 cache-to-cache 的传送优化。如果 S 态也进行 cache-to-cache 传输的话需要额外的电路来选择某个 cache 负责传送。cache-to-cache 的基本假设是这种传输比直接从 memory 读取快, 如果加入选择电路, 这一假设不一定成立。

3. 在 P0 write miss 而其它 cache 对应块处于 S 态时，需要先发送 invalidate 信号？

P0 write miss 时向总线放置 BusRdX 信号，其它 cache 探测到这个信号时会自己将对应的块置为 I 态，所以不需要再发送 invalidate 信号。