

《 计算机体系结构 》

实验报告

|  |  |
| --- | --- |
| 班 级： | **计211** |
| 学 号： | **19001531** |
| 姓 名： | **陈正江** |
| 指导教师： | **梁建宁** |

信息科学与工程学院

2023年11月

**实验名称** Cache一致性──监听协议

**实验地点** 信息楼418  **实验日期** 2023.12.13

1. **实验目的**
2. 加深对多Cache一致性的理解；
3. 进一步掌握解决多Cache一致性的监听协议的基本思想；
4. 掌握在各种情况下，监听协议是如何工作的，能给出要进行怎么样的操作及状态变化情况。
5. **实验设备**

实验平台采用多Cache一致性监听协议模拟器。

1. **实验原理**

**1、Cache一致性**

在共享存储器系统中，维持数据在存储器和多个处理器或多个核的私有Cache中的数据副本一致。

**2、Cache一致性协议**

（1）作用：把某个处理器核新写的值传播给其他处理器核以确保所有处理器核看到一致的共享存储内容。

（2）种类：监听协议（将写无效信号通过信道广播给所有的处理器），目录协议（只向持该数据备份的处理器发出写无效信号，避免广播）。

**3、监听协议**

（1）监听协议下Cache块的状态

·无效I：Cache块内容无效，空闲块；

·共享S：Cache块内容有多个处理器副本；特殊情况：存储器块首次载入Cache，存储器内容与Cache内容一致，且只有一个数据副本，Cache块也设为S；

·修改M：Cache内容被修改，数据为最新值，存储器尚未写回，保存的是旧值，最新值由当前处理器独占。

（2）串行化写操作

总线控制权获得的顺序性保证多个处理器对同一数据写操作的串行化，所有一致性协议都需要保证对同一Cache写操作的串行化。

（3）总线上的两类消息

要写/读的数据不在Cache中时，Cache发送到总线上有两类消息：写不命中和读不命中。一般是从存储器或者其他处理器的Cache中获得最新副本；对于写回Cache，为了尽快获得最新副本，马上启动存储器访问；如果最新副本在其他Cache中，由该Cache提供数据最新值，中断存储器访问。

（4）写回Cache

1）每个 Cache块设置一个有限状态控制器，根据本地CPU和总线的请求以及Cache状态做出响应，转换Cache块状态；2）某个Cache写操作，让其他Cache副本作废，不引发替换；3）Cache控制器监听总线上传送的地址，如果拥有某数据最新值，将最新值提供给需要的Cache。写回Cache比写直达通信量少，需要带宽少。

（5）监听实现

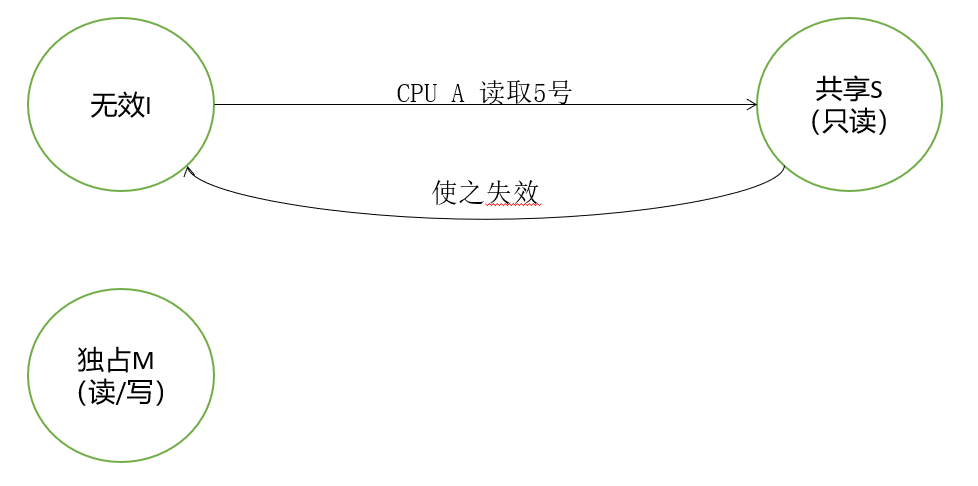
1）利用Cache的Tag标记来比较总线上传送的地址，判定消息是否与本Cache相关；

2）有效位：方便实现作废机制；共享位：1表示数据有多个副本，0表示被独占

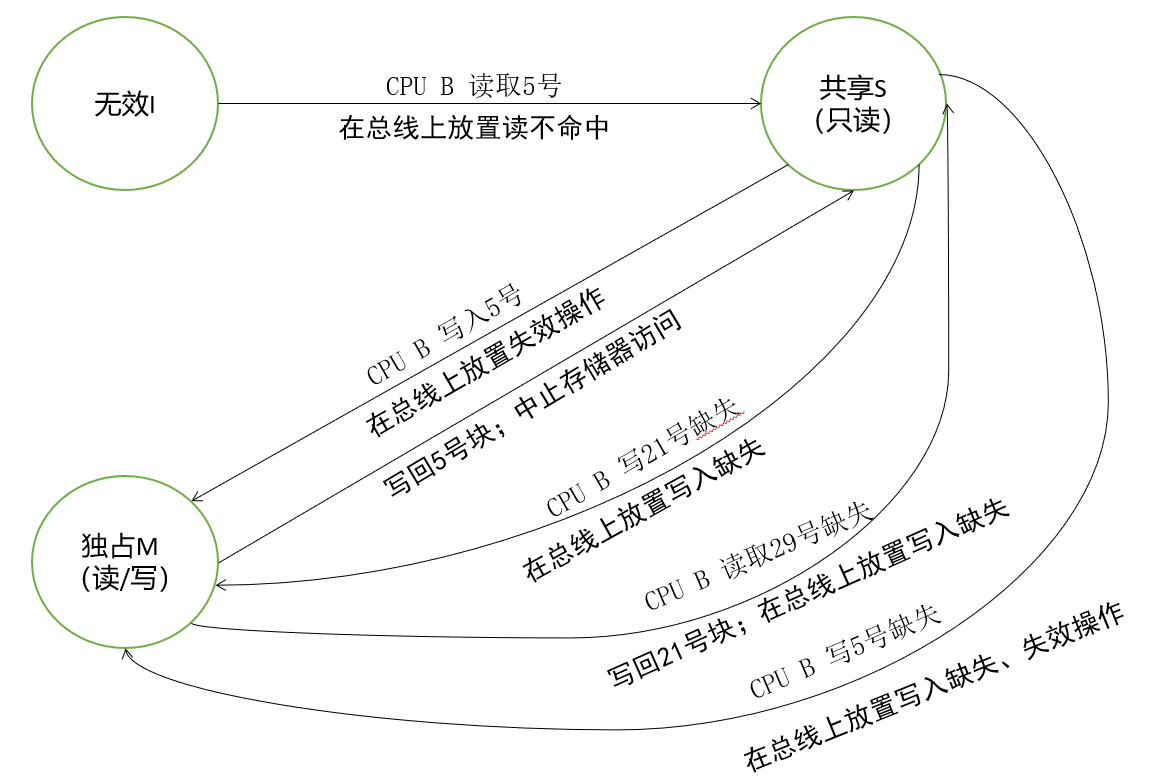
3）写操作时，如果共享位0，表示数据只有唯一副本，不需要将作废消息发到总线，减少带宽，加快执行速度；写操作时，如果共享位1，需要发送作废消息，将其他Cache数据副本作废；同时被写Cache共享位设为0，表示为拥有者；如果独占数据后面被复制，共享位变为1。

1. **实验操作及运行结果**
2. **掌握模拟器使用方法（参考模拟器帮助菜单）。**
3. **对以下访存序列，写出监听协议所做的操作。**

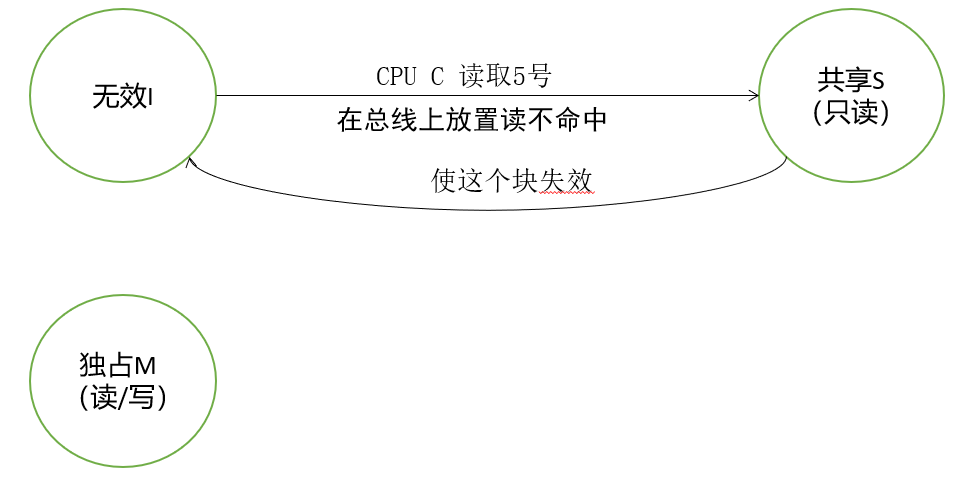
|  |  |  |  |
| --- | --- | --- | --- |
| **所进行的访问** | **是否发生替换** | **是否发生写回** | **监听协议所进行的操作** |
| CPU A读第5块 | 否 | 否 | 将存储器号为5的块送到Cache A的1中，再送到CPU A中。 |
| CPU B读第5块 | 否 | 否 | 将存储器号为5的块送到Cache B的1中，再送到CPU B中。 |
| CPU C读第5块 | 否 | 否 | 将存储器号为5的块送到Cache C的1中，再送到CPU C中。 |
| CPU B写第5块 | 是 | 否 | CPU B将新内容写入Cache B的1中替换掉原本的内容，其他Cache中原5号块的内容作废。 |
| CPU D读第5块 | 否 | 是 | 将Cache B的1中内容写回存储器号为5的块，然后将该块内容送到Cache D的1中，再送到CPU D中。 |
| CPU B写第21块 | 是 | 否 | 将存储器号为21的块送到Cache B的1中，替换原5号块，CPU B再将要写的内容写入Cache B的1中。 |
| CPU A写第23块 | 否 | 否 | 将存储器号为23的块送到Cache A的3中，CPU A再将要写的内容写入Cache A的3中。 |
| CPU C写第23块 | 否 | 是 | 将Cache A的3中内容写回存储器号为23的块，然后将该块送到Cache C的3中，CPU C再将要写的内容写入Cache C的3中。 |
| CPU B读第29块 | 是 | 是 | 将Cache B的1中内容写回存储器号为21的块；然后将存储器号为29的块送到Cache B的1中，替换掉原21号块，再送到CPU B中。 |
| CPU B写第5块 | 是 | 否 | 将存储器号为5的块送到Cache B的1中，替换掉原有内容， Cache D中原5号块的内容作废；CPU B再将要写的内容写入Cache B的1中。 |

1. **根据上述结果，写出相关的状态转换图**
2. Cache A的1号块

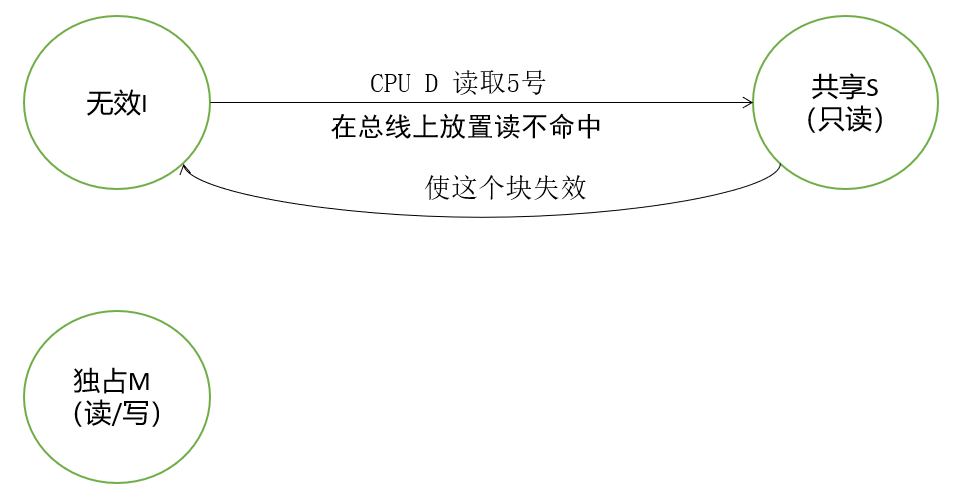
（2）Cache B的1号块

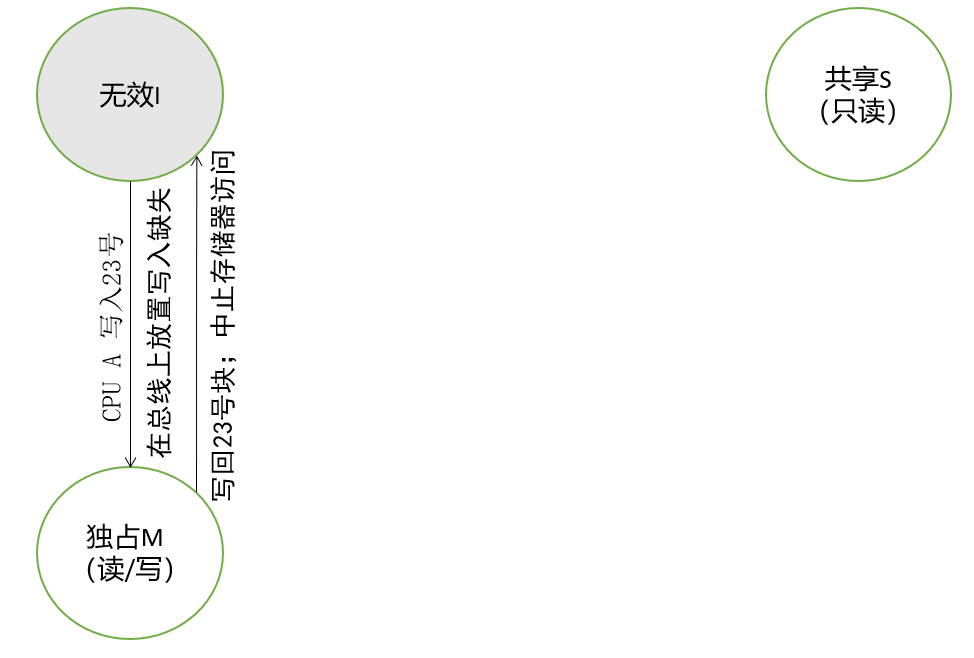


（3）Cache C的1号块



（4）Cache D的1号块



（5）Cache A的3号块

（6）Cache C的3号块



1. **自己编写一个访问序列，写出监听协议所做的操作。**

| **所进行的访问** | **是否发生替换** | **是否发生写回** | **监听协议所进行的操作** |
| --- | --- | --- | --- |
| CPU A读第7块 | 否 | 否 | 将存储器号为7的块送到Cache A的3中，再送到CPU A中 |
| CPU B读第7块 | 否 | 否 | 将存储器号为7的块送到Cache B的3中，再送到CPU B中 |
| CPU C读第7块 | 否 | 否 | 将存储器号为7的块送到Cache C的3中，再送到CPU C中 |
| CPU D读第7块 | 否 | 否 | 将存储器号为7的块送到Cache D的3中，再送到CPU D中 |
| CPU A写第7块 | 是 | 否 | CPU A将新内容写入Cache A的3中**替换**掉原本的7号块内容，同时其他Cache中原本缓存的7号块的内容均作废。 |
| CPU B写第17块 | 否 | 否 | 将存储器号为17的块送到Cache B的2中，CPU B再将要写的内容写入Cache B的2中。 |
| CPU A写第20块 | 否 | 否 | 将存储器号为20的块送到Cache A的1中，CPU A再将要写的内容写入Cache A的1中。 |
| CPU C写第20块 | 否 | 是 | 将Cache A的1中内容**写回**存储器号为20的块，然后将该块送到Cache C的2中。 |
| CPU B读第17块 | 否 | 否 | 将Cache B的2中内容送到CPU B中。 |
| CPU B写第6块 | 是 | 是 | 将Cache B的2中内容**写回**存储器号为17的块，然后将存储器号为6的块送到Cache B的2中，**替换**掉原来存的17号块，最后CPU B再将新的6号块写入Cache B的2中。 |