

《计算机体系结构实验报告》



实验题目：Tomasulo 和Cache 一致性

学生姓名：柯志伟

学生学号：PB20061338

完成时间：2022年6月21日

【实验题目】

Tomasulo 和Cache 一致性

【实验目的】

- 1.熟悉Tomasulo模拟器和cache一致性模拟器（监听法和目录法）的使用
- 2.加深对Tomasulo算法的理解，从而理解指令级并行的一种方式-动态指令调度
- 3.掌握Tomasulo算法在指令流出、执行、写结果各阶段对浮点操作指令以及load和store指令进行什么处理；给定被执行代码片段，对于具体某个时钟周期，能够写出保留站、指令状态表以及浮点寄存器状态表内容的变化情况。
- 4.理解监听法和目录法的基本思想，加深对多cache一致性的理解
- 5.做到给出指定的读写序列，可以模拟出读写过程中发生的替换、换出等操作，同时模拟出cache块的无效、共享和独占态的相互切换

【实验过程】

• Tomasulo算法

使用模拟器进行以下指令流的执行并对模拟器截图、回答问题

L.D F6, 21 (R2)

L.D F2, 0 (R3)

MUL.D F0, F2, F4

SUB.D F8, F6, F2

DIV.D F10, F0, F6

ADD.D F6, F8, F2

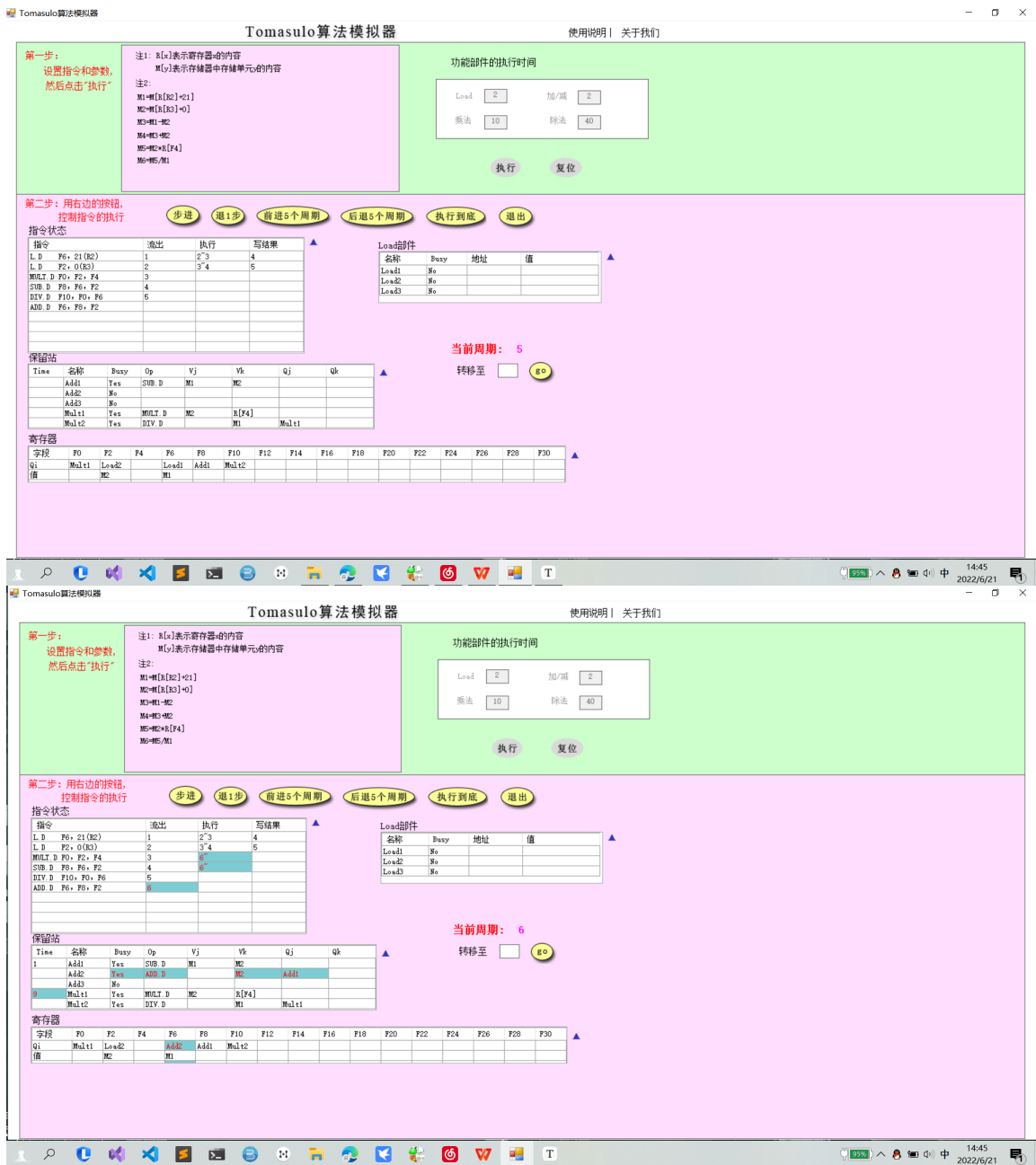
假设浮点功能部件的延迟时间：加减法2个周期，乘法10个周期，*load/store*2个周期，除法40个周期。

1. 分别截图（当前周期**2**和当前周期**3**），请简要说明**load**部件做了什么改动



Load部件中, Load2地址由0变为R[R3]+0, Load1的值由空变为M[R[R2]+21]

2. 请截图 (**MUL.D**刚开始执行时系统状态), 并说明该周期相比上一周期整个系统发生了哪些改动 (指令状态、保留站、寄存器和**Load**部件)



指令状态改动： MUL.D、SUB.D指令进入执行状态

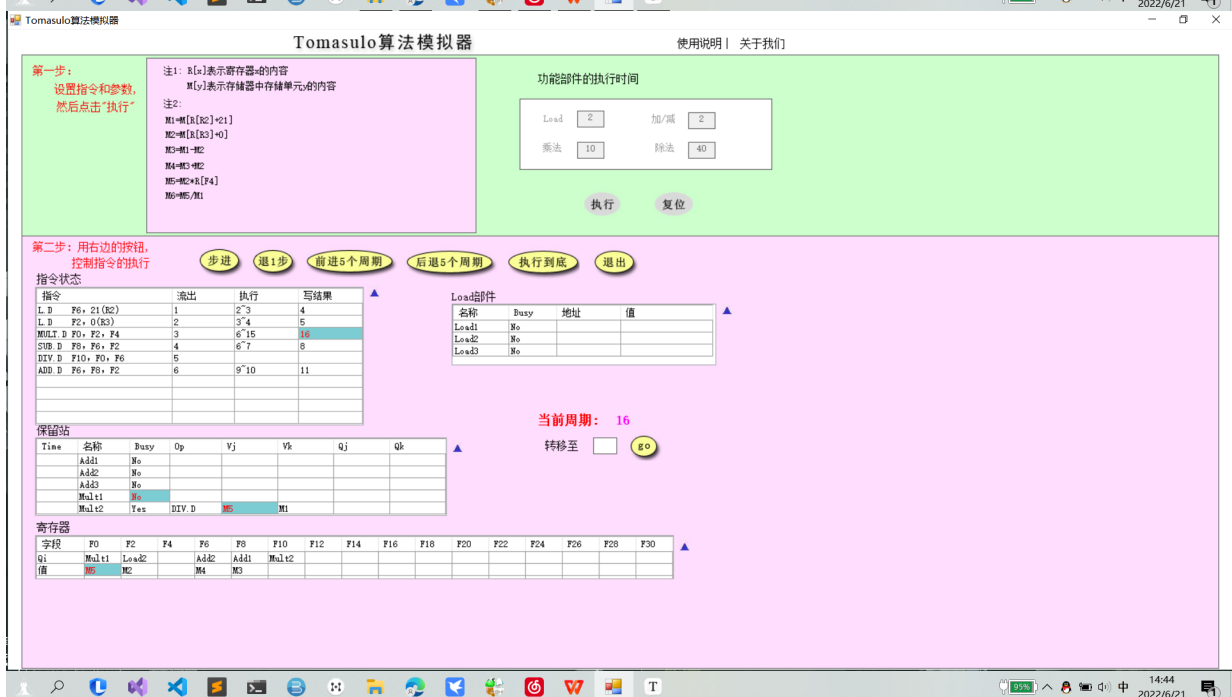
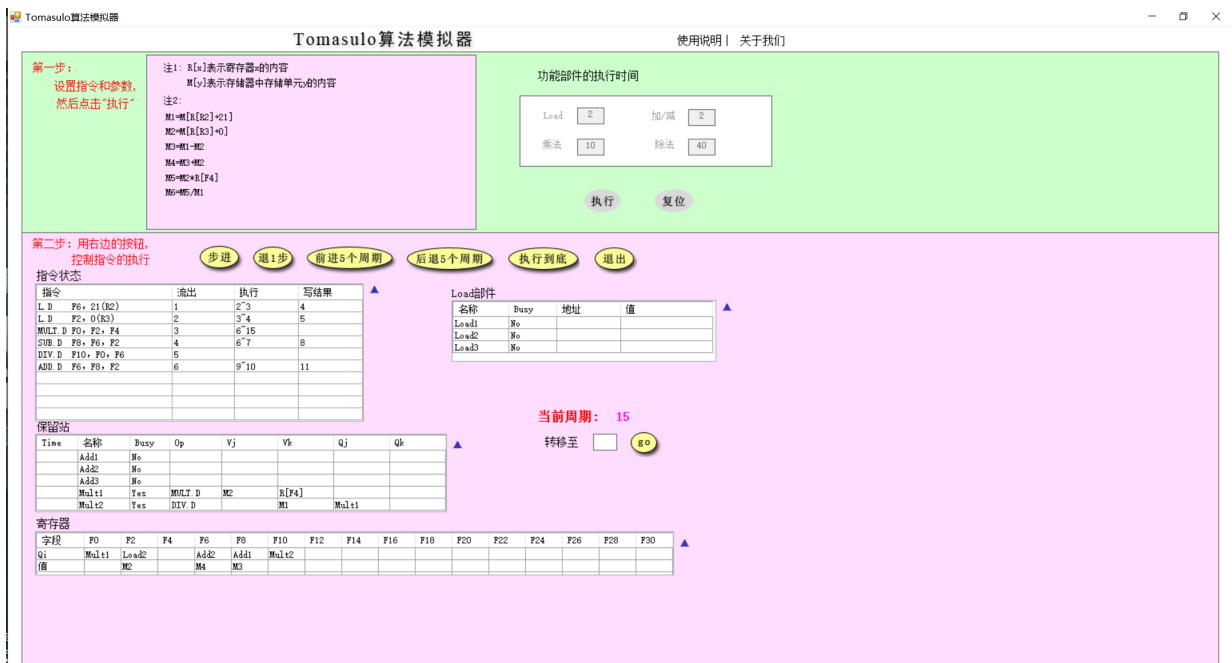
保留站改动： Addr1的Time域变为1, Addr2的Busy域变为Yes, OP域变为ADD.D, Vs域变为M2, Qj域变为Addr1, Mult1的Time域变为9

寄存器改动： F6的Qi域变为Addr2, Busy域变为Yes

3. 简要说明是什么相关导致MUL.D流出后没有立即执行

MUL.D的操作数来自L.D指令的结果, 即存在写后读相关。

4. 请分别截图（15周期和16周期的系统状态），并分析系统发生了哪些变化



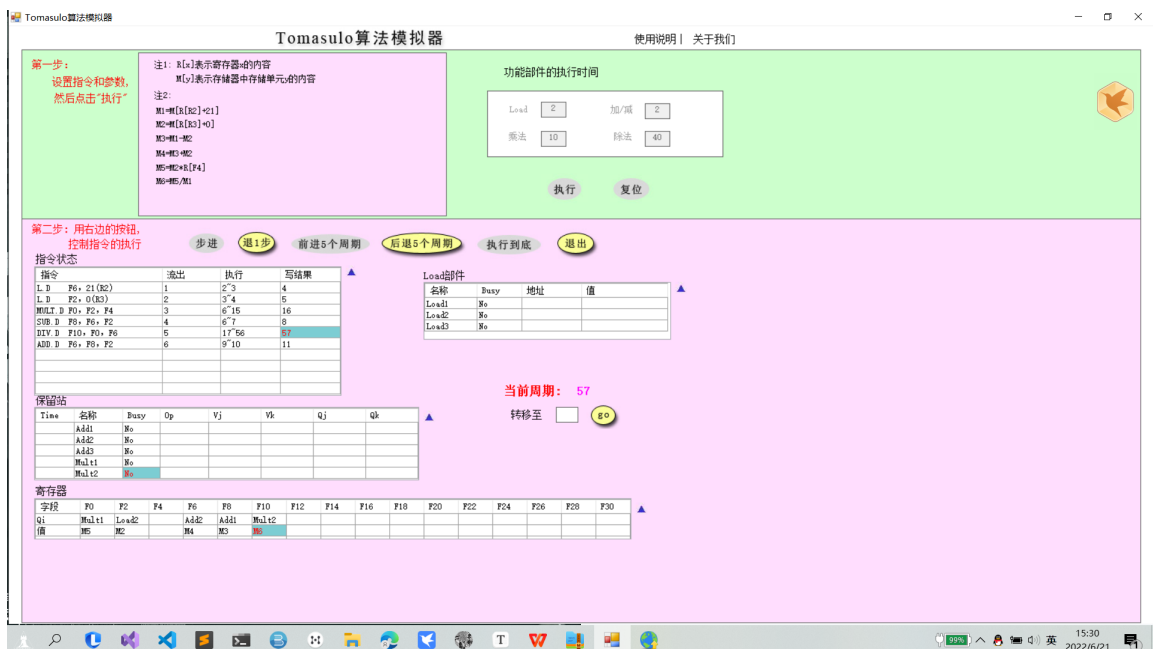
指令状态改动: MUL.D指令执行完毕，进入写结果阶段

保留站改动: Mult1的Busy域变为No，之前保存的信息被清空。Mult2的Vj域变为M5

寄存器改动: F0的值域变为M5，Busy域被清空。

Load部件改动: 无改动。

5. 回答所有指令刚刚执行完毕时是第多少周期，同时请截图（最后一条指令写CBD时认为指令流执行结束）



- 第一条指令 (L.D F6, 21 (R2))
4
- 第二条指令 (L.D F2, 0 (R3))
5
- 第三条指令 (MUL.D F0, F2, F4)
16
- 第四条指令 (SUB.D F8, F6, F2)
8
- 第五条指令 (DIV.D F10, F0, F6)
57
- 第六条指令 (ADD.D F6, F8, F2)
11

• 多cache一致性算法-监听法

1.利用模拟器进行下述操作，并填写下表

所进行的访问

是否发生了替换?

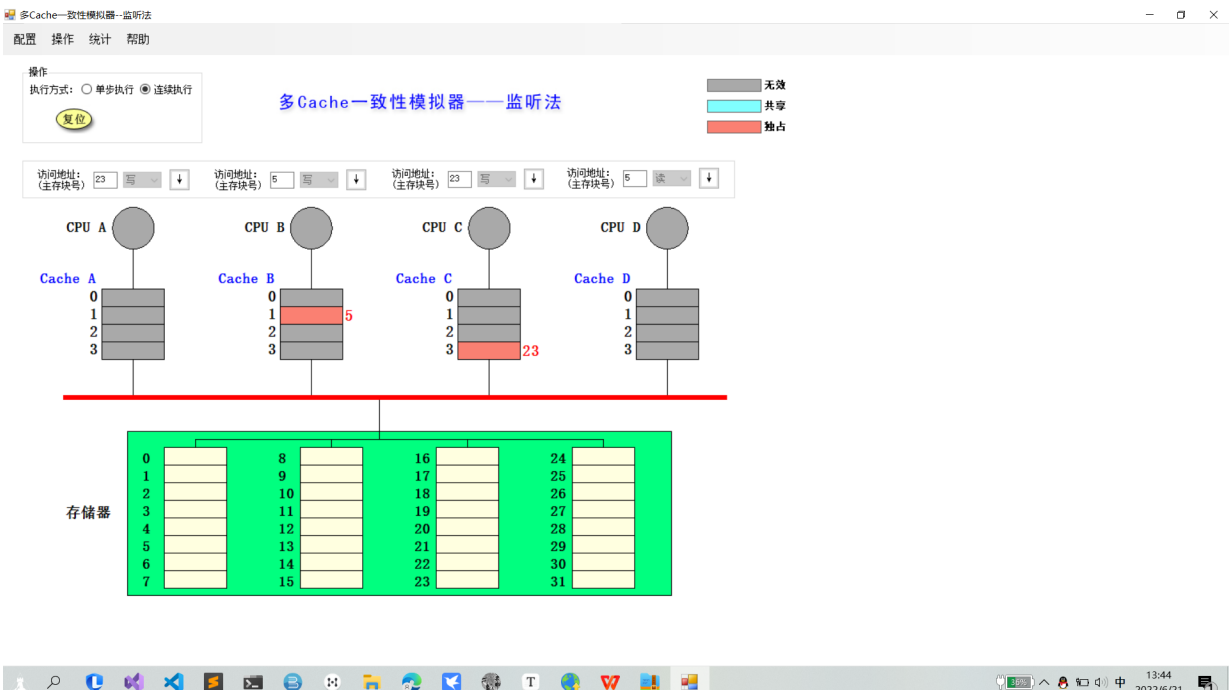
是否发生了写回?

监听协议进行的操作与块状态改变

所进行的访问	是否发生了替换？	是否发生了写回？	监听协议进行的操作与块状态改变
CPU A 读第5块	否	否	A向总线发送读不命中信号，从存储器读第五块到cache，CPU A中该块状态由无效变为共享
CPU B 读第5块	否	否	B向总线发送读不命中信号，从存储器读第五块到cache，CPU B中该块状态由无效变为共享
CPU C 读第5块	否	否	C向总线发送读不命中信号，从存储器读第五块到cache，CPU C中该块状态由无效变为共享
CPU B 写第5块	否	否	B向总线发送作废信号，CPU A、CPU C中第5块状态变为无效，CPU B中第5块状态变为独占
CPU D 读第5块	否	是	D向总线发送读不命中信号，Cache B持有的第5块写回存储器，且状态变为共享。接着存储器将第5块发送给cache D，状态变为共享。
CPU B 写第21块	是	否	B向总线发送写不命中信号，存储器将第21块发送给Cache B，Cache B直接将第21块覆盖第5块，第21块在Cache B中状态变为独占。
CPU A 写第23块	否	否	A向总线发送写不命中信号， 存储器将第23块发送给Cache A，第23块在Cache A中状态变为独占。
CPU C 写第23块	否	是	C向总线发送写不命中信号，Cache A将第23块写回存储器，并置该块状态为无效。存储器将第23块发送给Cache C，第23块在Cache C中状态变为独占。

所进行的访问	是否发生了替换？	是否发生了写回？	监听协议进行的操作与块状态改变
CPU B 读第29块	是	是	Cache B将第23块写回存储器，然后B向总线发送读不命中信号，存储器将第29块发送给Cache B，第29块在Cache B中状态变为独占。
CPU B 写第5块	是	否	B向总线发送写不命中信号，存储器将第5块发送给Cache B，第5块在Cache B中状态变为独占。然后B向总线发送写不命中信号，Cache D中第5块状态变为无效。

2.请截图，展示执行完以上操作后整个cache系统的状态。



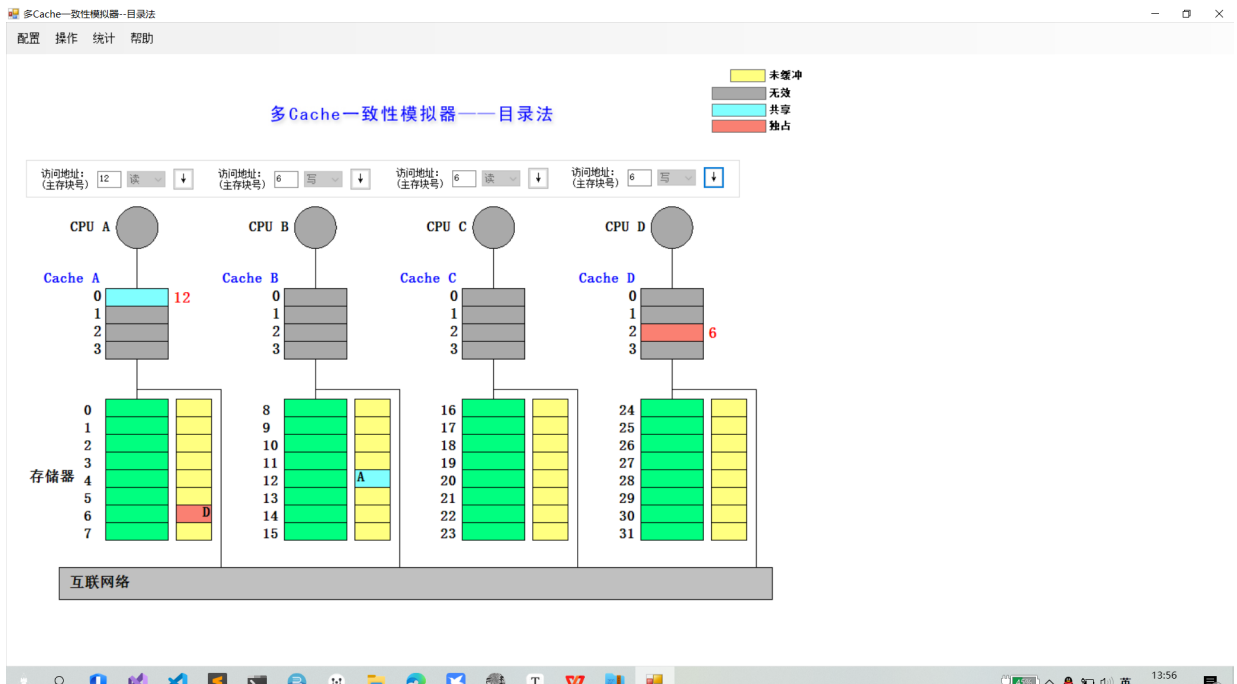
• 多cache一致性算法-目录法

1. 利用模拟器进行下述操作，并填写下表

所进行的访问	监听协议进行的操作与块状态改变		
CPU A 读第6块	CPU A	向宿主节点发送读不命中信号，存储器将第6块发送给Cache A，第6块在Cache中状态变为共享，在目录中状态也变为共享，共享者有A	

所进行的访问	监听协议进行的操作与块状态改变
CPU B读第6块	CPU B向宿主节点发送读不命中信号，存储器将第6块发送给Cache B，第6块在Cache中状态变为共享，在目录中状态也变为共享，共享者有A、B
CPU D读第6块	CPU C向宿主节点发送读不命中信号，存储器将第6块发送给Cache B，第6块在Cache中状态变为共享，在目录中状态也变为共享，共享者有A、B、C
CPU B写第6块	CPU B向宿主节点发送写命中信号，然后宿主向远程节点A、C分别发送作废信号。第6块在Cache A和Cache C中状态变为无效，在目录中状态变为由B独占，在Cache B中状态变为独占
CPU C读第6块	CPU C向宿主节点发送读不命中信号，然后宿主给远程节点B发送第6块的信息，B将第6块发送给宿主节点，宿主节点再将第6块发送给Cache C。第六块在Cache B、Cache C中状态均变为共享，在目录中状态也变为共享,共享者有：B、C
CPU D写第20块	CPU D向宿主节点发送写不命中信号， 宿主节点将第20块发送给Cache D。第20块在Cache D中状态变为独占，在目录中状态也变为独占
CPU A写第20块	CPU A向宿主节点发送写不命中信号，宿主向CPU D发送取并作废信号，CPU D将第20块发送给宿主节点，并将Cache D中该块状态置为无效。然后宿主将第20块发送给Cache A。第20块在Cache A中状态变为独占，在目录中状态也变由A独占
CPU D写第6块	CPU D向宿主节点发送写不命中信号，宿主向远程节点B、C分别发送作废信号，第6块在Cache B和Cache C中状态变为无效。然后宿主将第6块发送给Cache D。第6块在Cache D中状态变为独占，在目录中状态变为由D独占
CPU A读第12块	CPU A向第20块的宿主节点发写回并修改共享集的信息，第20块在目录中状态变为无效。然后向第12块的宿主节点发送读不命中消息，宿主将数据块发送给Cache A。第12块在Cache A中状态变为共享，在目录中状态也变为共享，共享者有：A

2. 请截图，展示执行完以上操作后整个**cache**系统的状态。



• 综合问答

1. 目录法和监听法分别是集中式和基于总线，两者优劣是什么？

相比之下, 监听法优点在于, 小型体系中, 总线的交流很少, 不会把太多时间浪费在不同宿主的沟通之间, 缺点在于对于CPU数量多的架构, 总线压力会很大, 性能下降很快; 目录法优点在于, 多个CPU的大型分布式cache可以稳定运行, 缺点在于小型体系使用目录法会较为复杂

2. Tomasulo算法相比ScoreBoard算法有什么异同？（简要回答两点：1. 分别解决了什么相关，2. 分别是分布式还是集中式）

1. tomasolu算法解决了结构、RAW、WAW、WAR相关。Score Board算法解决了由真相关引起的数据冒险停顿
2. tomasolu算法具有分布式的阻塞检测机制，而Score Board算法则是集中式的

3. Tomasulo算法是如何解决结构、RAW、WAR和WAW相关的？

结构相关：多功能部件或功能部件流水化，当计算资源可用时才将指令发射
WAW、WAR：使用保留站进行寄存器重命名
RAW：延迟指令执行，直到所有操作数可用为止

【总结与思考】

通过Tomasulo算法模拟器, Cache一致性的监听式和目录式协议的模拟器, 具体了解了Tomasulo算法, Cache一致性的监听式和目录式协议的细节, 明白了Tomasulo算法是如何处理各种乱序发射时的相关问题, 体会到两种Cache一致性协议在应对Cache命中, 不命中, 写回时可能存在的cache不一致情况的处理, 对这三种算法有了更深刻的理解