USTC体系结构LAB5实验报告

谭泽霖 PB18010454

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

周期2:Load部件：占用Load2部件，Busy置位；R2就绪，将地址保存在Load1部件的地址寄存器

周期三：Load1部件将从存储器读到的值保存在Load1部件寄存器；R3就绪，将地址保存在Load2部件地址寄存器

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）

周期五是“上一周期”

第二步：用右边的按钮，
控制指令的执行

步进 | 退1步 | 前进5个周期 | 后退5个周期 | 执行到底 | 退出

指令状态

指令		流出	执行	写结果
L.D	F6，21(R2)	1	2~3	4
L.D	F2，0(R3)	2	3~4	5
MULT.D	F0，F2，F4	3		
SUB.D	F8，F6，F2	4		
DIV.D	F10，F0，F6	5		
ADD.D	F6，F8，F2			

Load部件

名称	Busy	地址	值
Load1	No		
Load2	No		
Load3	No		

当前周期： 5

转移至 go

保留站

Time	名称	Busy	Op	Vj	Vk	Qj	Qk
	Add1	Yes	SUB.D	M1	M2		
	Add2	No					
	Add3	No					
	Mult1	Yes	MULT.D	M2	R[F4]		
	Mult2	Yes	DIV.D		M1	Mult1	

寄存器

字段	F0	F2	F4	F6	F8	F10	F12	F14	F16	F18	F20	F22	F24	F26	F28	F30
Qi	Mult1	Load2		Load1	Add1	Mult2										
值		M2		M1												

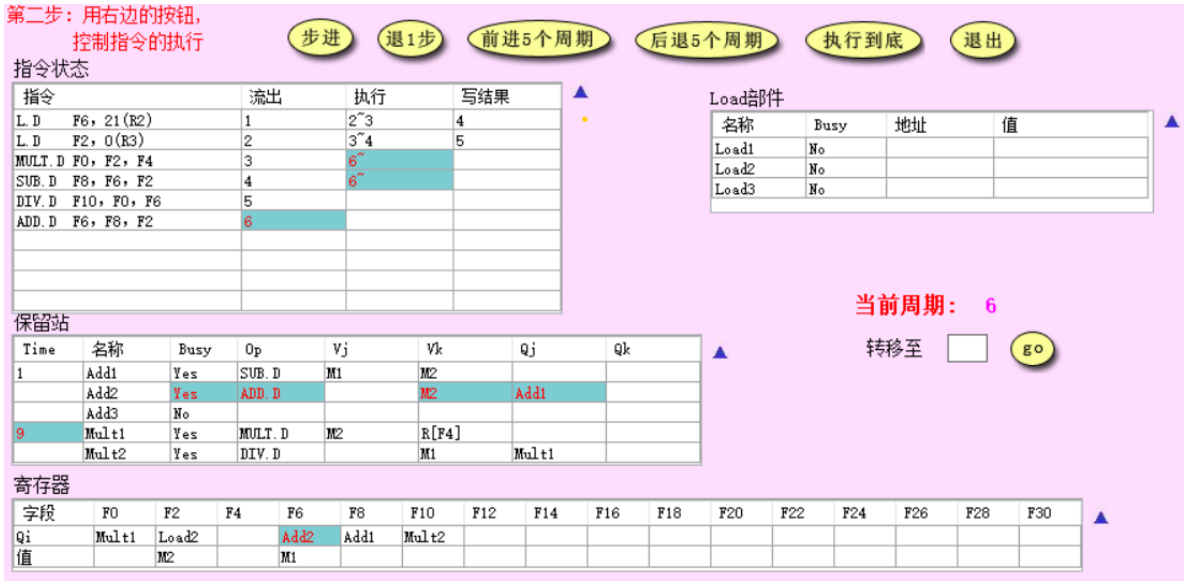
MUL.D刚开始执行时：第六周期

变化如下：
指令状态：发射第6条指令；第三条、第四条指令进入执行状态

Load部件：无变化

保留站：新发射的ADD.D指令占用Add2保留站，进入执行的指令MUL.D和SUB.D开始执行，时间开始倒计时

寄存器：新发射的指令ADD.D指令等待F8寄存器

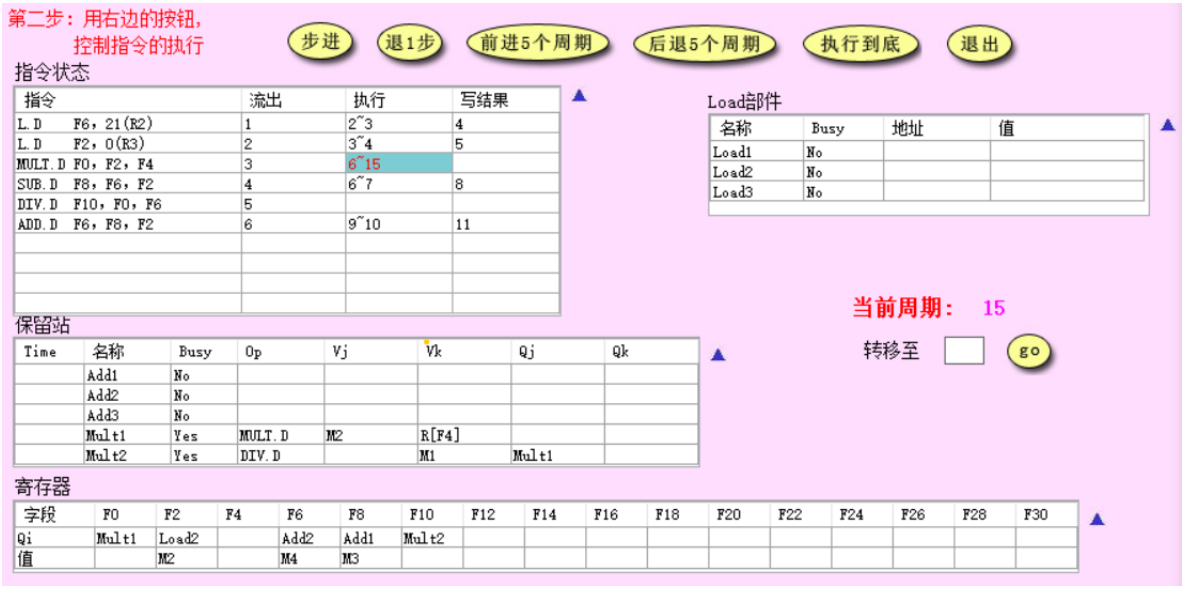


3.简要说明是什么相关导致MUL.D流出后没有立即执行

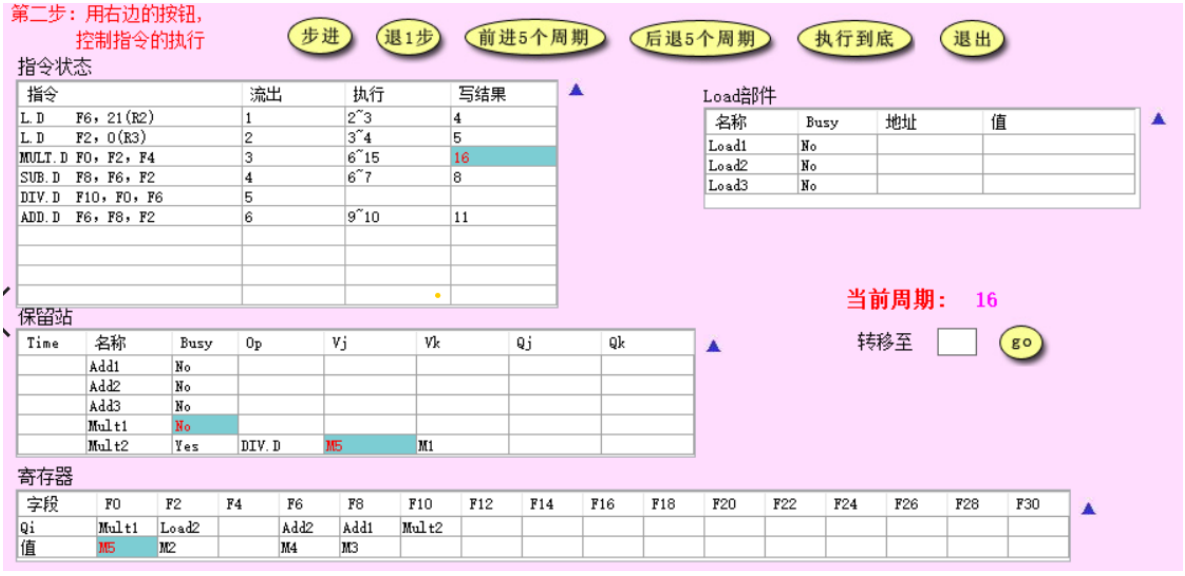
源操作数F2为写回，直到第五周期M2写入后才就绪

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

第15周期：指令ADD.D和指令SUB.D在周期7~15周期内执行完毕，将结果写回，释放相应的保留站和寄存器；此时MULT.D指令执行了10个周期。



第16周期：指令MUL.D写回结果，释放保留站CBD将结果广播到寄存器和指令DIV.D对应的保留站。



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）

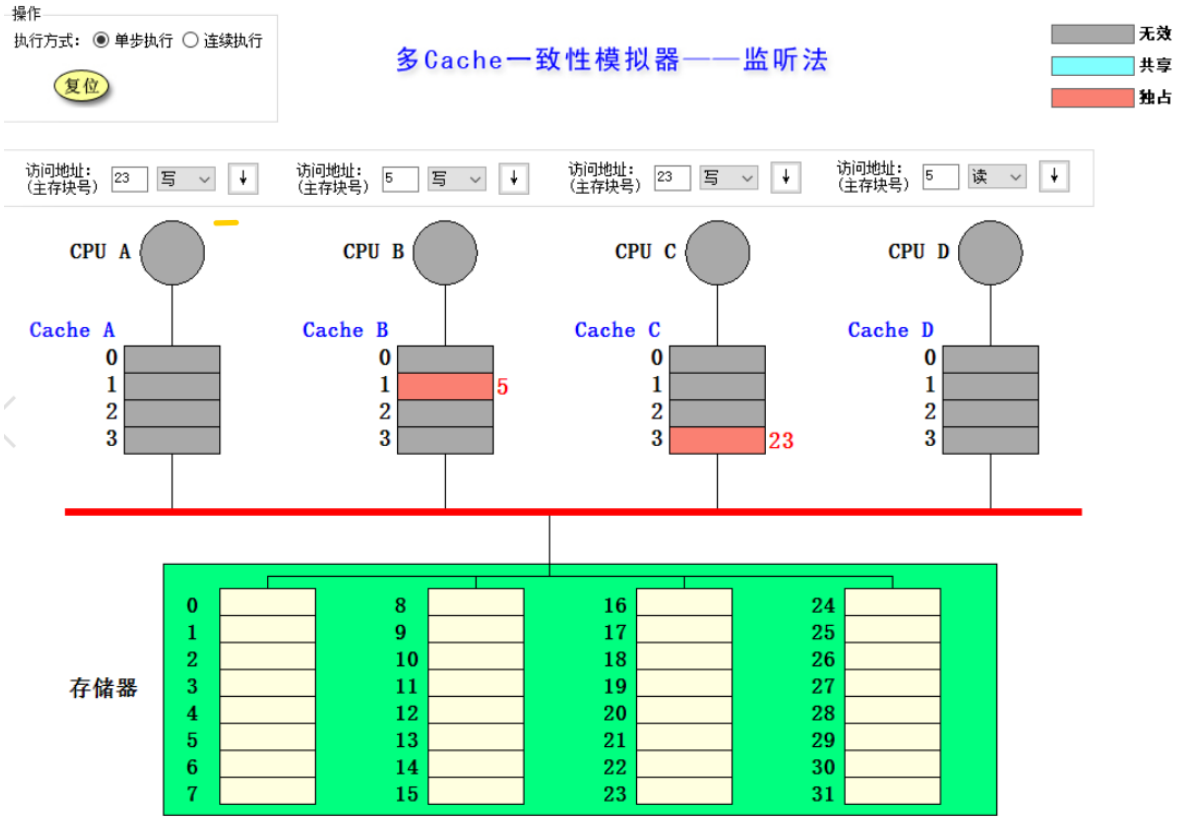
所有指令执行完毕需要57个周期

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
CPU A 读第5块	替换 Cache A的块1	0	CacheA发射Read Miss，存储器传输第5块到CacheB，CacheA的块1从状态I转为S
CPU B 读第5块	替换 CacheB 的块1	0	CacheB发射Read Miss，存储器传输第五块到CacheB，CacheB的块1从状态I转为S
CPU C 读第5块	替换 CacheC 的块1	0	CacheC发射Read Miss，存储器传输第5块到CacheC，CacheC的块1从状态I转换为S
CPU B 写第5块	0	0	CacheB发射Invalidate，CacheA的块1从状态S转换到I，CacheC的块1从状态S转换到I，CacheB的块1从S转换到M
CPU D 读第5块	替换 CacheD 的块1	CacheB 的块1 写回	CacheD发射Read Miss，CacheB写回第5块，存储器传输第5块到CacheD，CacheB的块1从状态M转换到S，CacheD的块从状态I转换到S
CPU B 写第21块	替换 CacheB 的块1	0	CacheB发射Write Miss，存储器传输第21块到CacheB，CacheB的块1从状态S转换为M
CPU A 写第23块	替换 CacheA 的块3	0	CacheA发射Write Miss，存储器传输第23块到CacheA，CacheA的块1从状态I转换到M
CPU C 写第23块	替换 CacheC 的块3	CacheA 的块3 写回	CacheC发射Read Miss，CacheA写回第23块，存储器传输第23块到CacheC，CacheA的块3从状态M转换到I，CacheC的块3从状态I转换到M
CPU B 读第29块	替换 CacheB 的块1	CacheB 的块1 写回	CacheB写回第21块，CacheB发射Read Miss，存储器传输第29块到CacheB，CacheB的块1从状态M转换到S
CPU B 写第5块	替换 CacheB 的块1	0	CacheB发射Write Miss，存储器传输第5块到CacheB，CacheB的块1从状态S转换到M，CacheD的块1从状态S转换带I

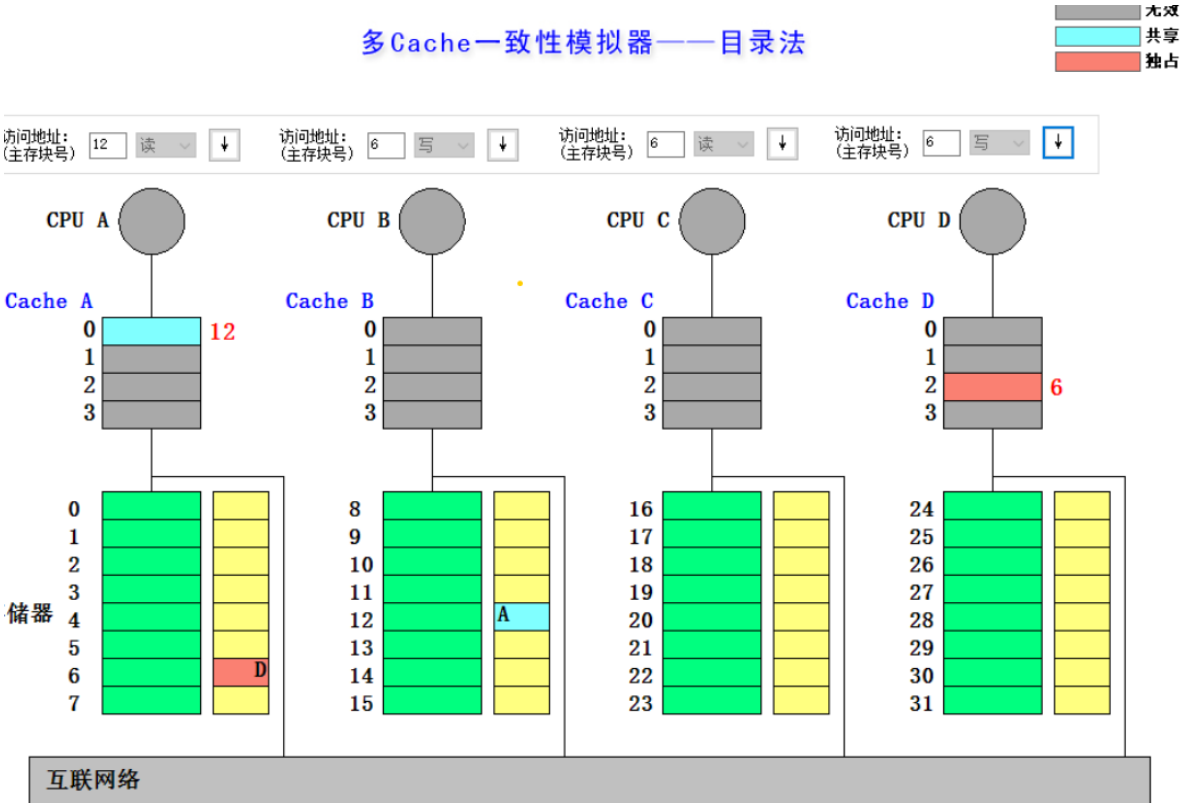
执行完成的时候：



三.多cache一致性-目录法

所进行的访问	监听协议进行的操作	块状态改变
CPU A 读第6块	Cache A发送Read Miss<A,6>到Memory A，Memory A传输第6块到Cache A，Cache A的块2从状态I转换为S	Memory A的块6，状态：U->S，Presence bits：0000->0001，共享集合{A}
CPU B 读第6块	Cache B发送Read Miss<B,6>到Memory A，Memory A传输第6块到CacheB，Cache B的块2从状态2转为S	Memory A的块6，状态：S->S，Presence bits：0001->0011，共享集合{A, B}
CPU D 读第6块	Cache D发送Read Miss<D,6>到Memory A，Memory A传输第6块到Cache D，Cache D的块2从状态I转为S	Memory A的块6，状态：S->S，Presence bits: 0011->1011，共享集合{A,B,D}
CPU B 写第6块	Cache B发送Write Hit<B,6>到Memory A，Memory A发送Invalidate(6)到Cache A，Cache A的块2从状态S转为I，Memory A发送Invalidate(6)到到Cache D，Cache D的块2从状态S转换到I，Cache B的块2从状态S转换到M	Memory A到块6，状态：S->M，Presence bits：1011->0010，共享集合{B}
CPU C 读第6块	Cache C发送Read Miss<C,6>到Memory A，Memory A发送Fetch(6)到Cache B，Cache B传输第6块到Memory A，Cache B的块2从状态M转为S，Memory A传输第6块到Cache C，Cache C的块2从状态I转为S	Memory A的块6，状态：M->S，Presence bits：0010->0110，共享集合：{B,C}
CPU D 写第20块	Cache D发送Write Miss<D,20>到Memory C，Memory C传输第20块到Cache D，Cache D的块0从状态I转为M	Memory C的块20，状态：U->M，Presence bits：0000->1000，共享集合{D}
CPU A 写第20块	Cache A发送Write Miss<A,20>到Memory C，Memory C发送Fetch&Invalidate(20)到Cache D，Cache D传输第20块到Memory C，Cache D的块0从状态M转为I，Memory C传输第20块到Cache A，Cache A的块0从状态I转换到M	Memory C的块20，状态：M->M，Presence bits：1000->0001，共享集合{A}
CPU D 写第6块	Cache D发送Write Miss<D,6>到Memory A，Memory A发送Invalidate(6)到Cache B，Cache B的块2从状态S转为I，Memory A传输第6块到Cache D，Cache D的块2从状态I转为M	Memory A的块6，状态：S->M，Presence bits：0110->1000，共享集合{D}
CPU A 读第12块	Cache A发送Write Back<A,20>到Memory C，Cache A的块0从状态M转为I，Cache A发送Read Miss<A,12>到Memory B，Memory B传输第12块到Cache A，Cache A的块0从状态I转为S	Memory C的块20，状态：M->U，Presence bits：0001->0000，共享集合{}；Memory B的块12，状态：U->S，Presence bits：0000->0001，共享集合：{A}

执行完成：



四.综合回答

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？（言之有理即可）

监听法

优势：保证了Cache的一致性，实现了写互斥和写串行

劣势：

扩展性差，总线上能够连接的处理器数目有限

存在总线竞争问题

总线的带宽会带来一些限制

在非总线和或环形网络上监听困难

总线事务多，通信开销大

目录法

优势：

拓展性强，可以连接的处理器数目更多

降低了对于总线带宽的占用

可以有效地适应交换网络进行通信

劣势：

需要额外的空间来存储Presence Bits，当处理器数目较多的时候会有很大的存储开销

总线竞争

存储器接口通信压力大，存储器速度成为限制

2.Tomasulo算法相比Score Board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）（参考第五版教材）

Tomasulo

特点：分布式；指令状态、相关控制和操作数缓存分布在各个部件中(保留站)

WAR相关：使用RS的寄存器或指向RS的指针代替指令中的寄存器-寄存器重命名

WAW相关：使用RS中的寄存器值或指向RS的指针代替指令中的寄存器

RAW相关：检测到寄存器就绪即没有冲突再读取操作数，进入执行阶段

结构相关：有结构冲突不发射

结果Forward：从FU广播结果到RS和寄存器

Score Board

特点：集中式；指令状态和相关控制都在记分牌处理

WAR相关：对操作排队，仅在读操作数阶段读寄存器

WAW相关：检测到相关后，停止发射前一条指令，直到前一条指令完成

RAW相关：检测到没有冲突(寄存器就绪)再读取操作数，进入执行阶段

结构相关：有结构相关不发射

结果Forward：写回寄存器接触等待

3.Tomasulo算法是如何解决结构、RAW、WAR和WAW相关的？（参考第五版教材）

结构相关：有结构冲突不发射

RAW：检测到没有冲突，即存储器就绪再读取操作操作数，进入执行阶段

WAW：使用RS中的寄存器值或指向RS的指针代替指令中的寄存器-寄存器重命名

WAR：使用RS中的寄存器值或指向RS的指针代替指令中的寄存器-寄存器重命名