

指令的相关性与冒险



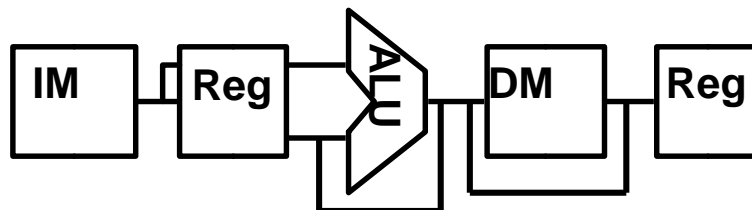
上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

本节主要内容

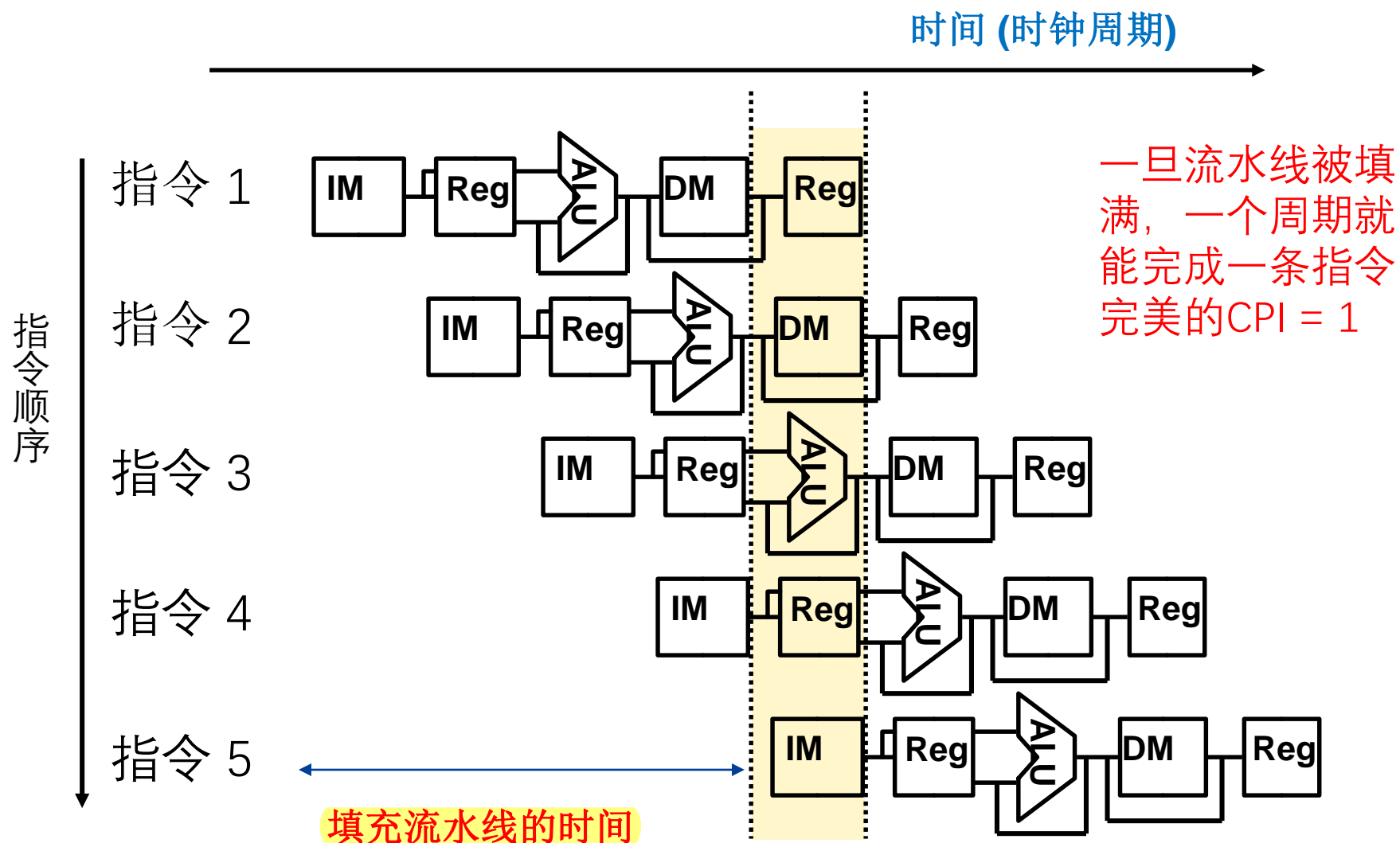


- 指令流水线中的相关性、冒险
 - 结构相关性
 - 数据相关性
 - 控制相关性
- 结构冒险
 - 解决方案

MIPS 流水线简化表达



流水线可以提高吞吐率



流水线中的相关性和“冒险”

- 流水线中的相关性：相邻或相近的指令之间因存在某种依赖性，或称为相关性，使得指令的执行可能受到影响。
- 这些相关性，可能会影响指令的执行，也可能不影响，因此又称为冒险（hazard）
 - 结构冒险（structural hazards）
 - 资源相关性：所需的硬件部件正在为之前的指令工作
 - 数据冒险（data hazards）
 - 数据依赖性需要等待之前的指令完成数据的读写
 - 控制冒险（control hazards）
 - 转移指令引起：需要根据指令的结果决定下一步

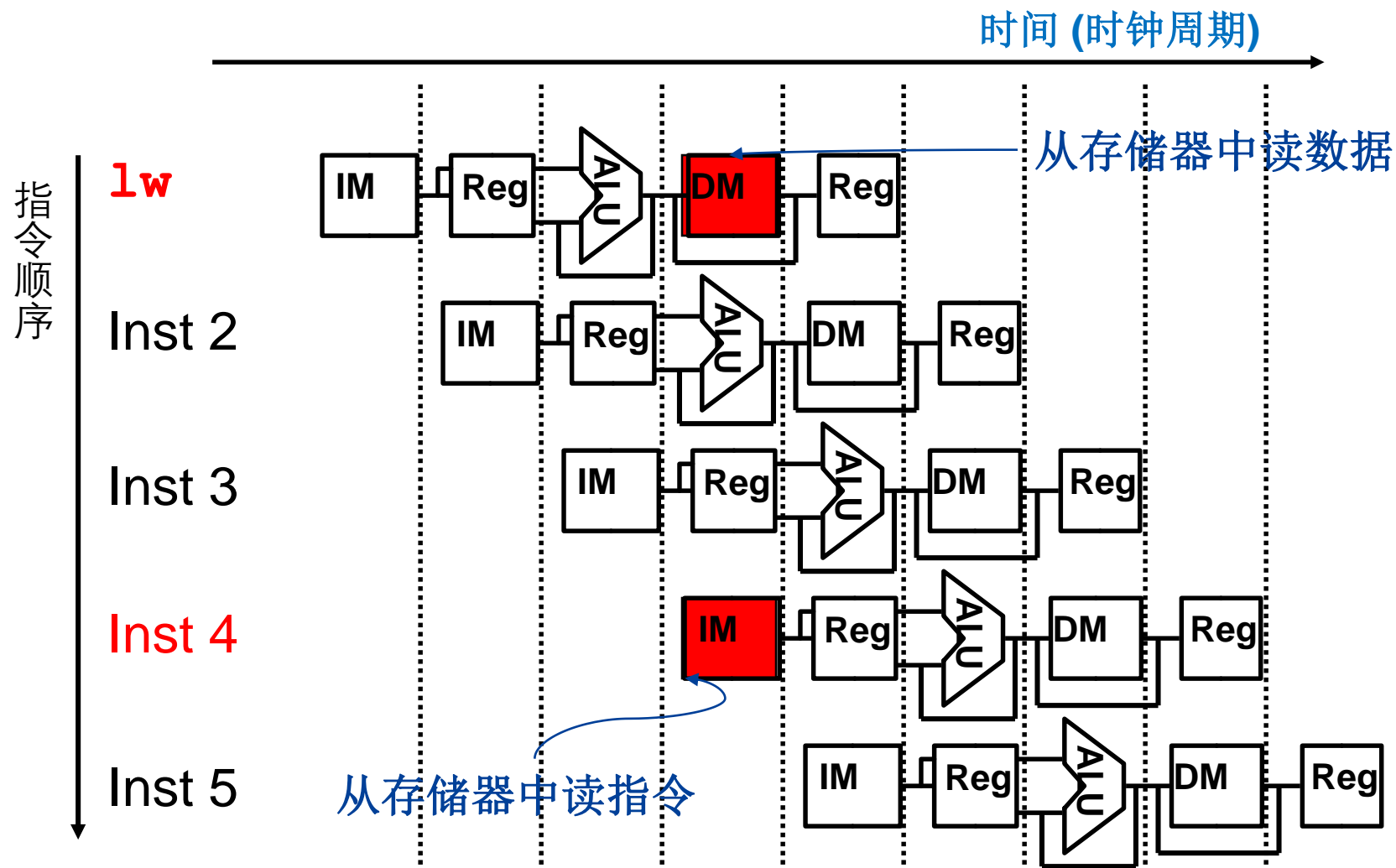


结构冒险

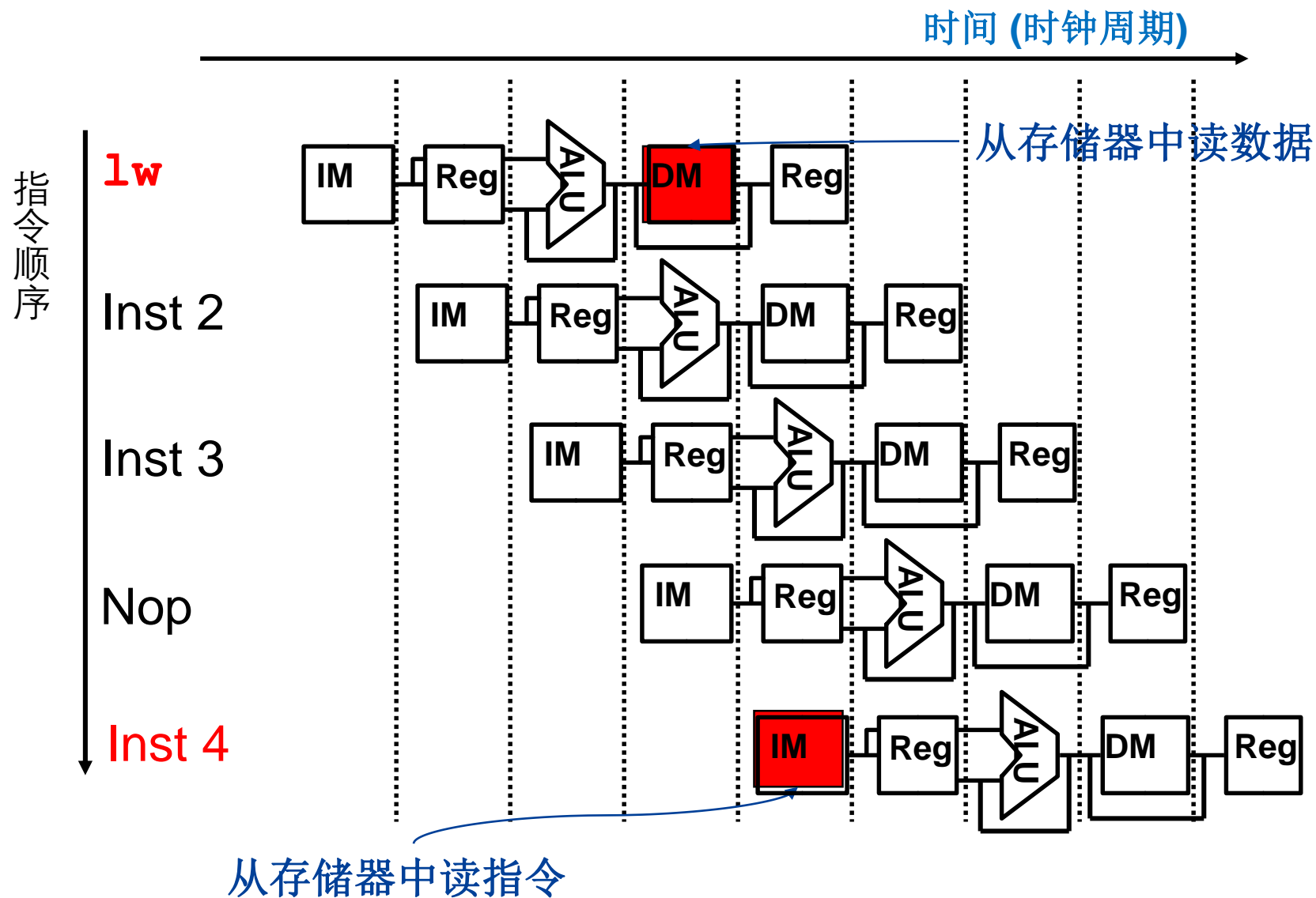


- 结构冒险 (structural hazards)

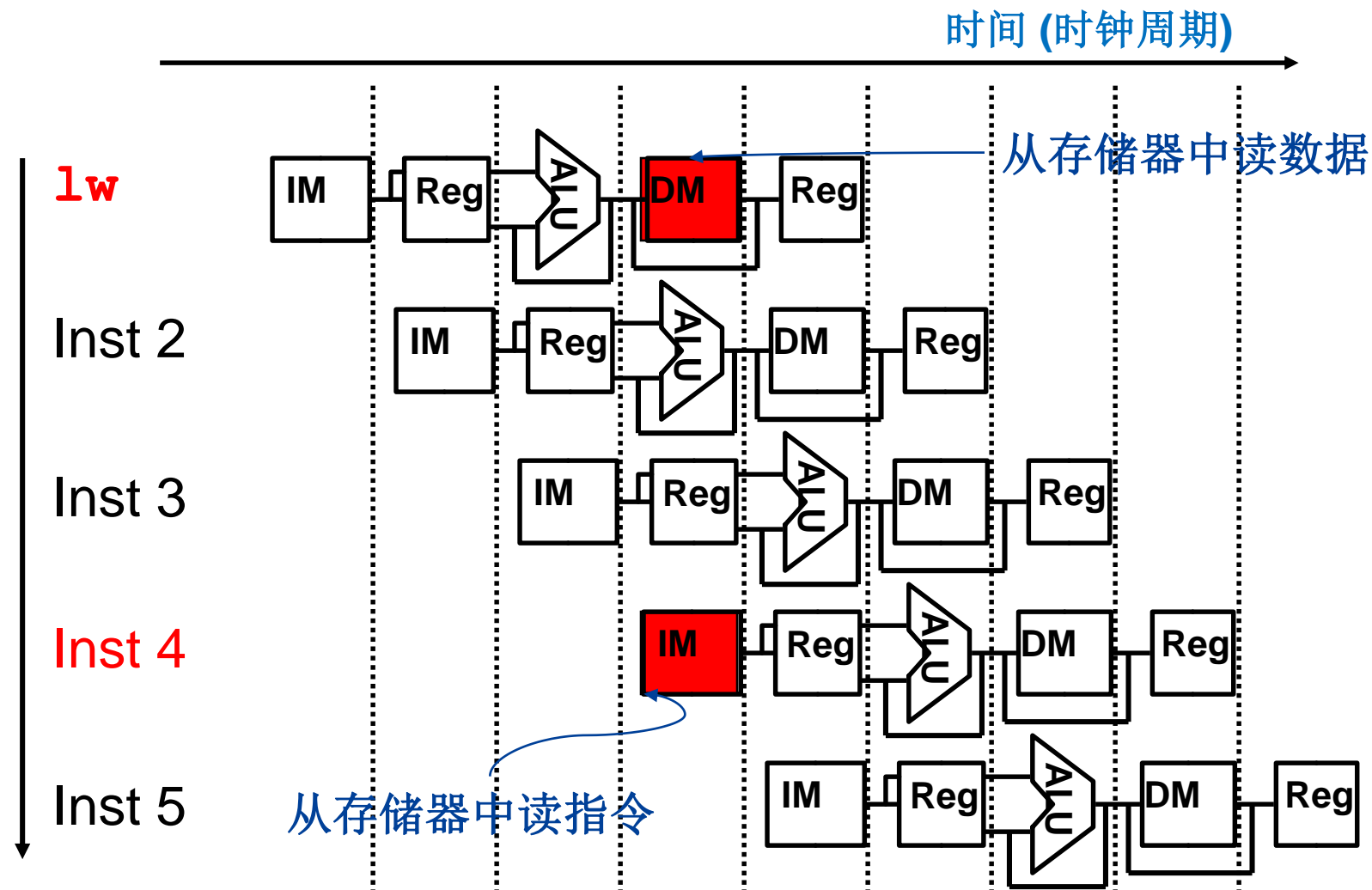
结构冒险：如果只有一个存储器



方案1：插入空指令



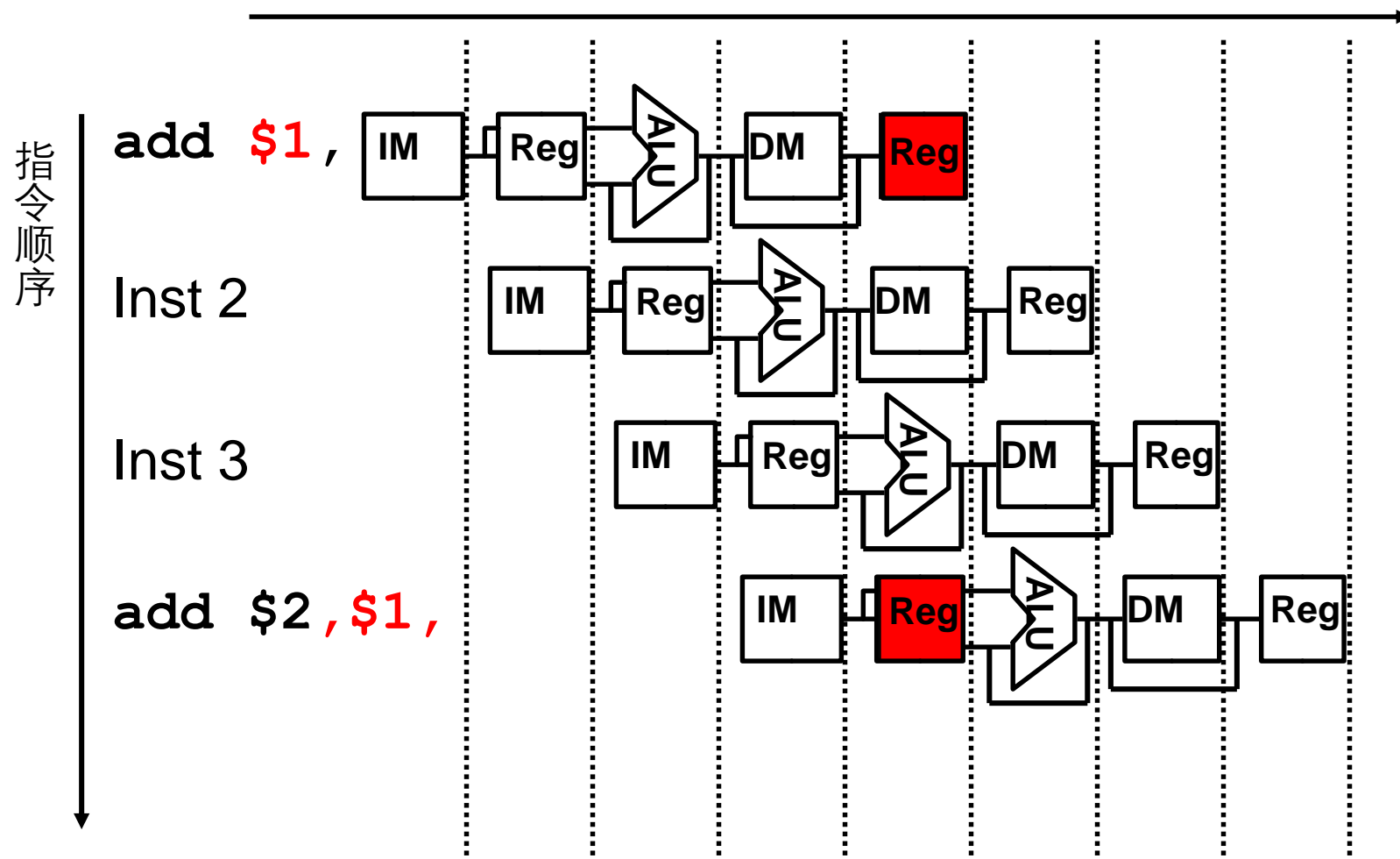
方案2：指令存储器和数据存储器相互独立



- 例如：CPU中分离的一级高速缓存：指令cache 数据cache

寄存器文件的读写是否存在结构冒险?

时间 (时钟周期)



方案：读写口独立的寄存器文件

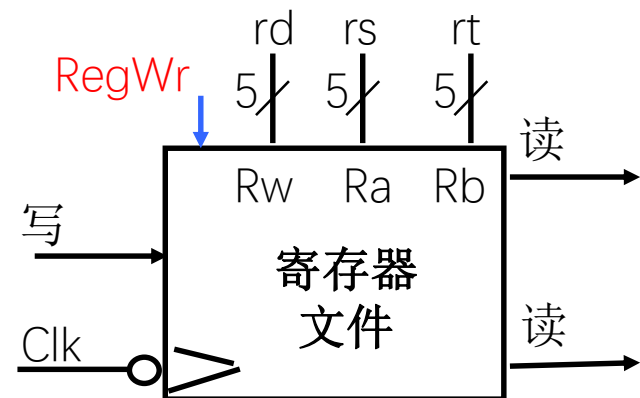
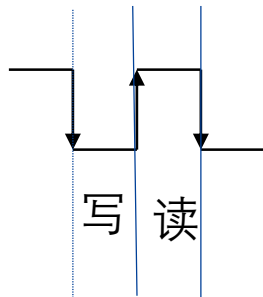


- 寄存器文件的读写速度较快
- 假设
 - 读或写寄存器的延迟为100ps（半个周期）
 - ALU、MEM等部件的延迟为200ps（一个周期）

寄存器文件设置相互独立的读口和写口

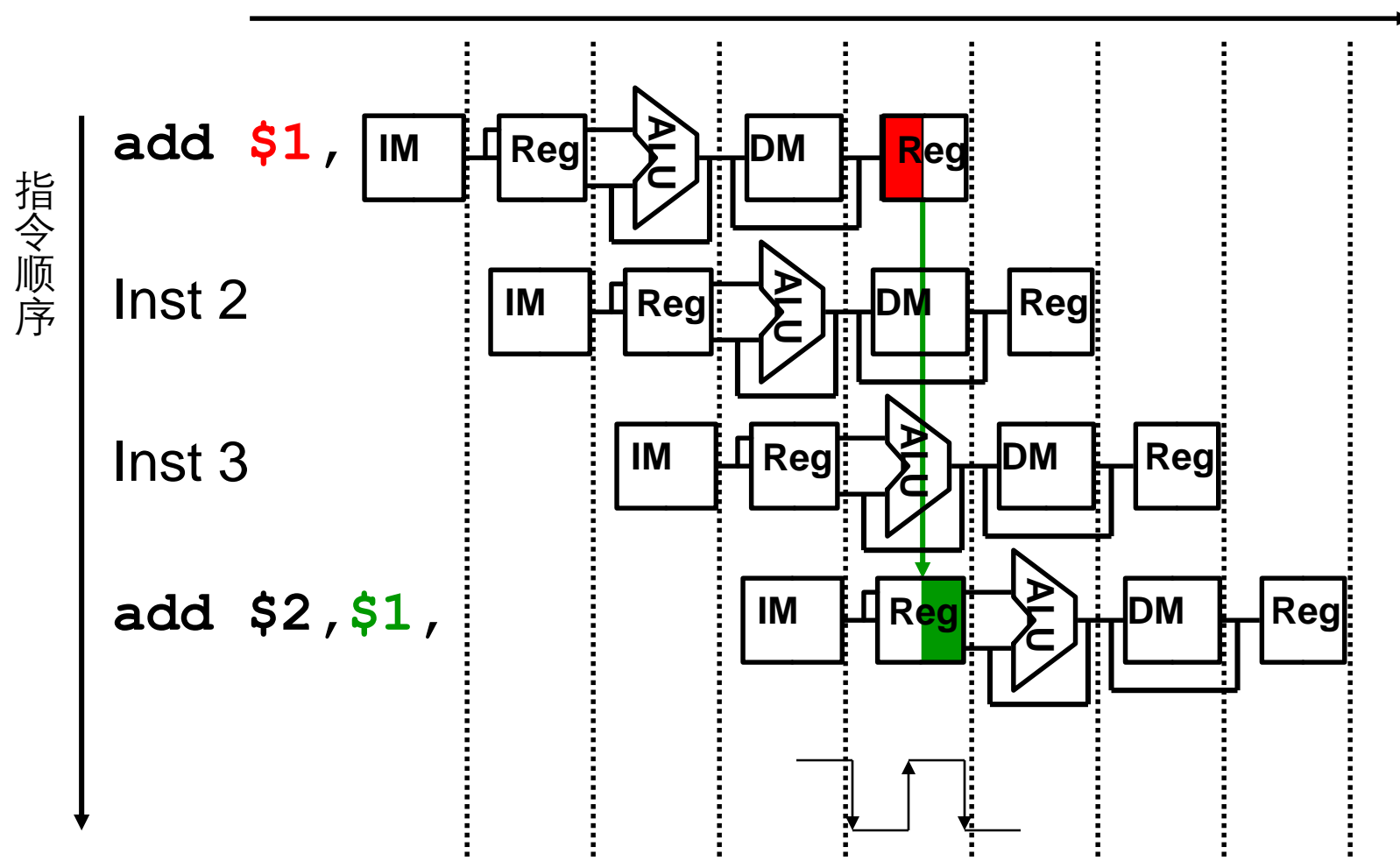
写寄存器：在一个周期的前半段完成

读寄存器：在一个周期的后半段完成



寄存器文件的读写是否存在结构冒险？

时间 (时钟周期)





避免结构冒险



- 避免结构冒险
 - 增加硬件资源
 - 功能单元流水化

小结



- 指令相关性、冒险
- 结构冒险的解决方案

谢谢！

