

# 处理器设计

## 流水线中的冒险

主讲人: 邓倩妮

上海交通大学

部分内容来自:

Computer Organization and Design, 4<sup>th</sup> Edition, Patterson & Hennessy



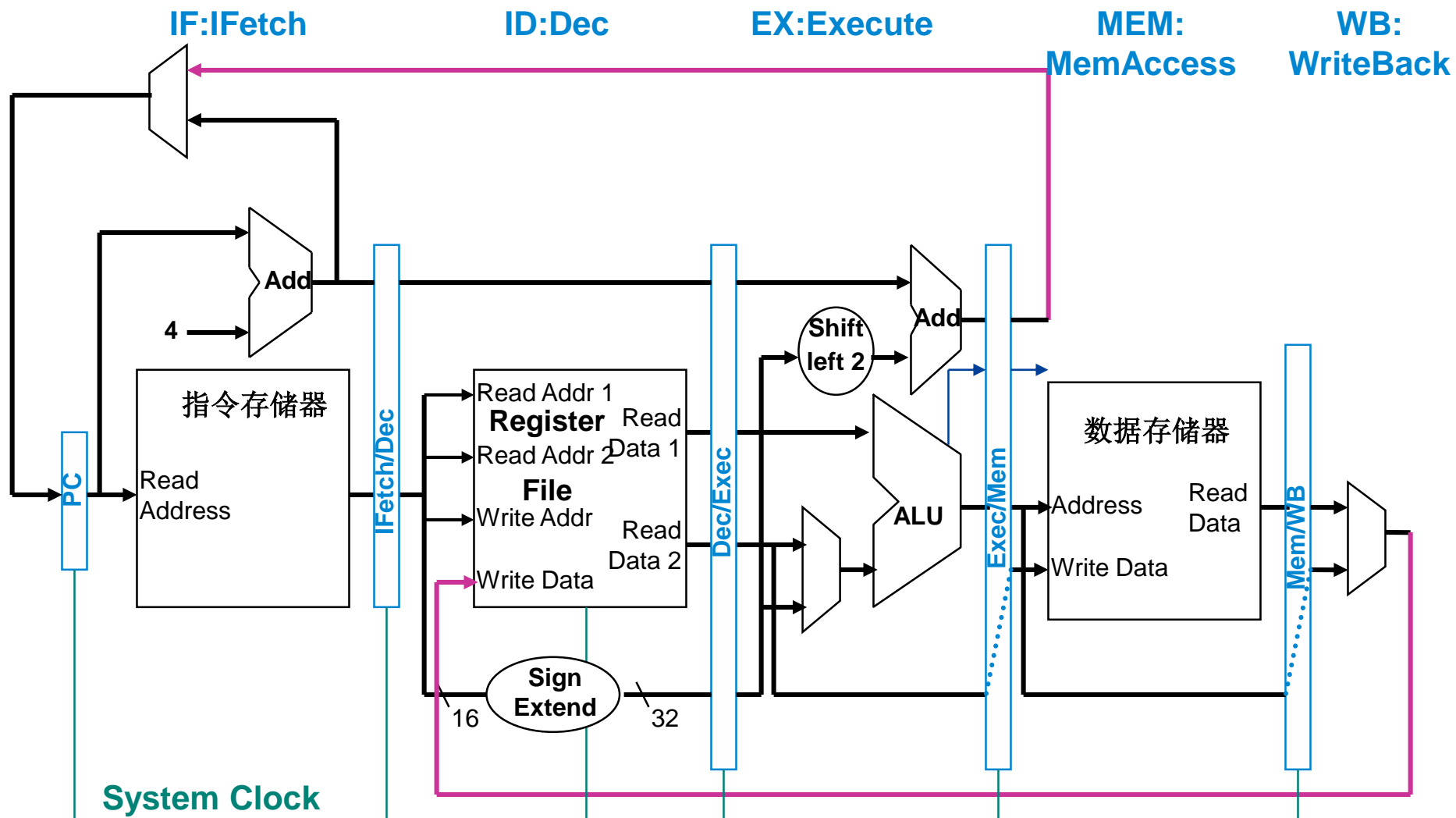
上海交通大学  
SHANGHAI JIAO TONG UNIVERSITY

# 本节主要内容



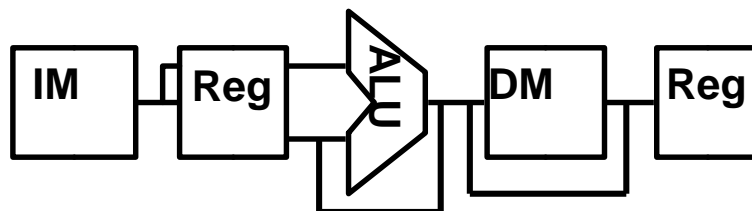
- 流水线冒险的定义
  - 结构冒险
  - 数据冒险
  - 控制冒险
- 结构冒险
  - 结构冒险解决方案
- 数据冒险
  - 数据冒险解决方案

# MIPS 流水线数据通路



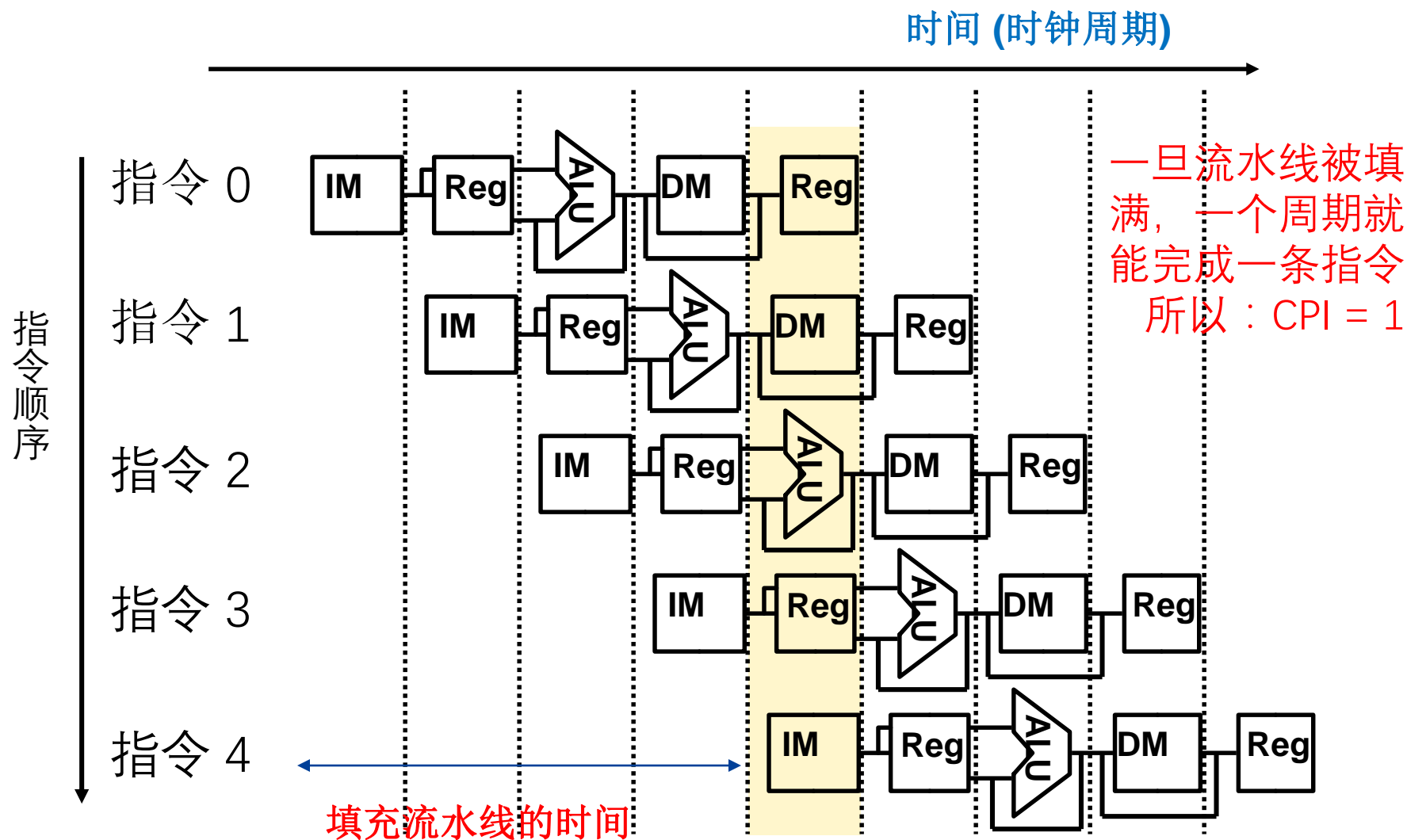
- 在MIPS 的数据通路上增加**段寄存器**，将各流水段隔离开

# MIPS 流水线简化表达



- 用简化的图形流水线可以分析:
  - 运行一段代码所需要的周期数?
  - 流水线中是否存在冒险, 如果存在, 如何解决

# 流水线可以提高吞吐率



# 流水线会因为冒险而停顿

流水线中的冒险（[Pipeline Hazards](#)）：相邻或相近的两条指令因存在某种关联，后一条指令不能在原指定的时钟周期开始执行

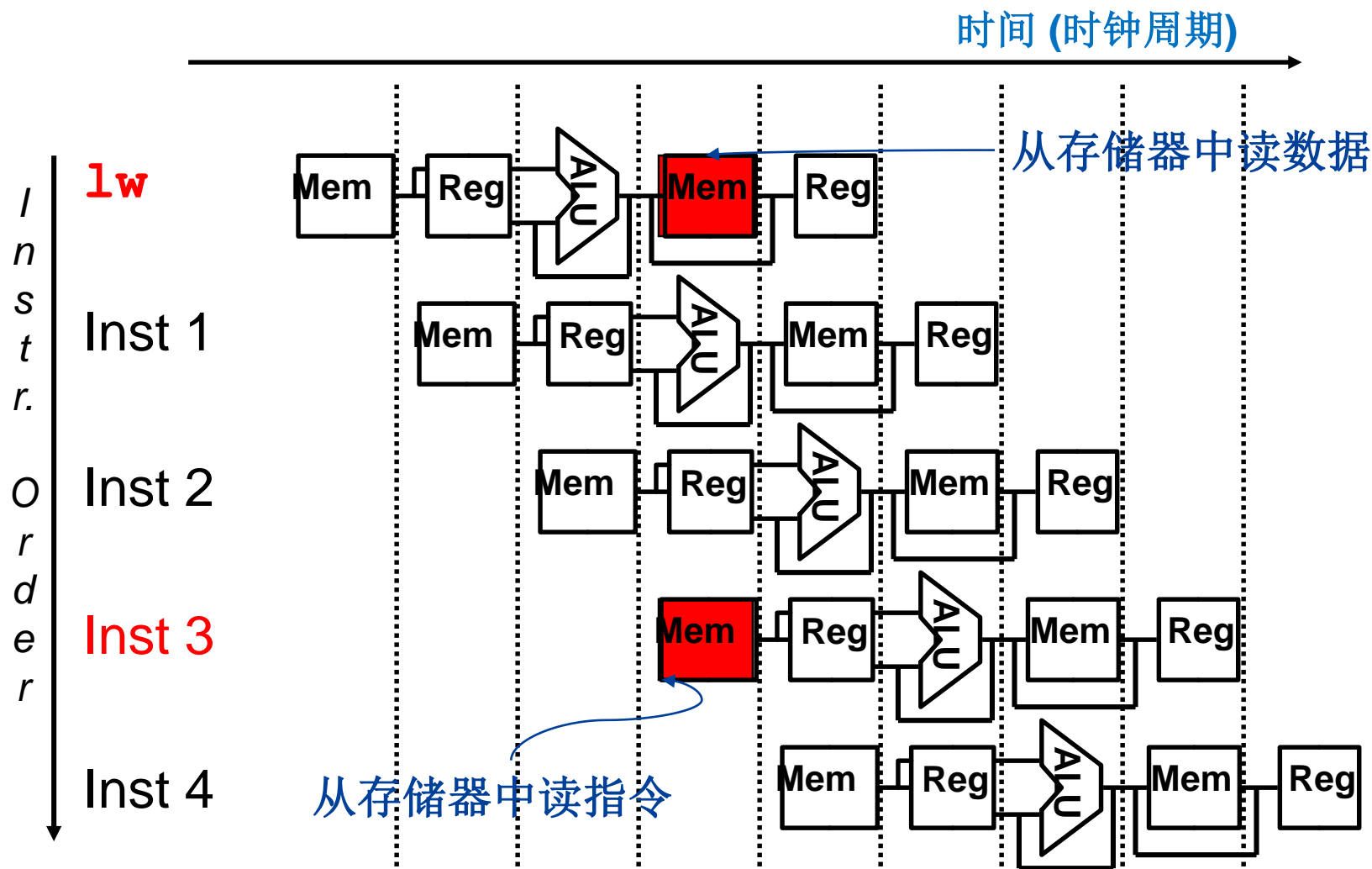
- [结构冒险](#)（[structural hazards](#)）：当硬件资源满足不了同时重叠执行的指令的要求，而发生资源冲突时，就发生了结构冒险。
- [数据冒险](#)（[data hazards](#)）当一条指令需要用到前面某条指令的结果，从而不能重叠执行时，就发生了数据冒险
- [控制冒险](#)（[control hazards](#)）当流水线遇到分支指令和其他能够改变 PC 值的指令时，就会发生控制冒险。
  - 例如 Branch、jump 指令

# 结构冒险



- 结构冒险（ structural hazards ） :当硬件资源满足不了同时重叠执行的指令的要求，而发生资源冲突时，就发生了结构冒险。

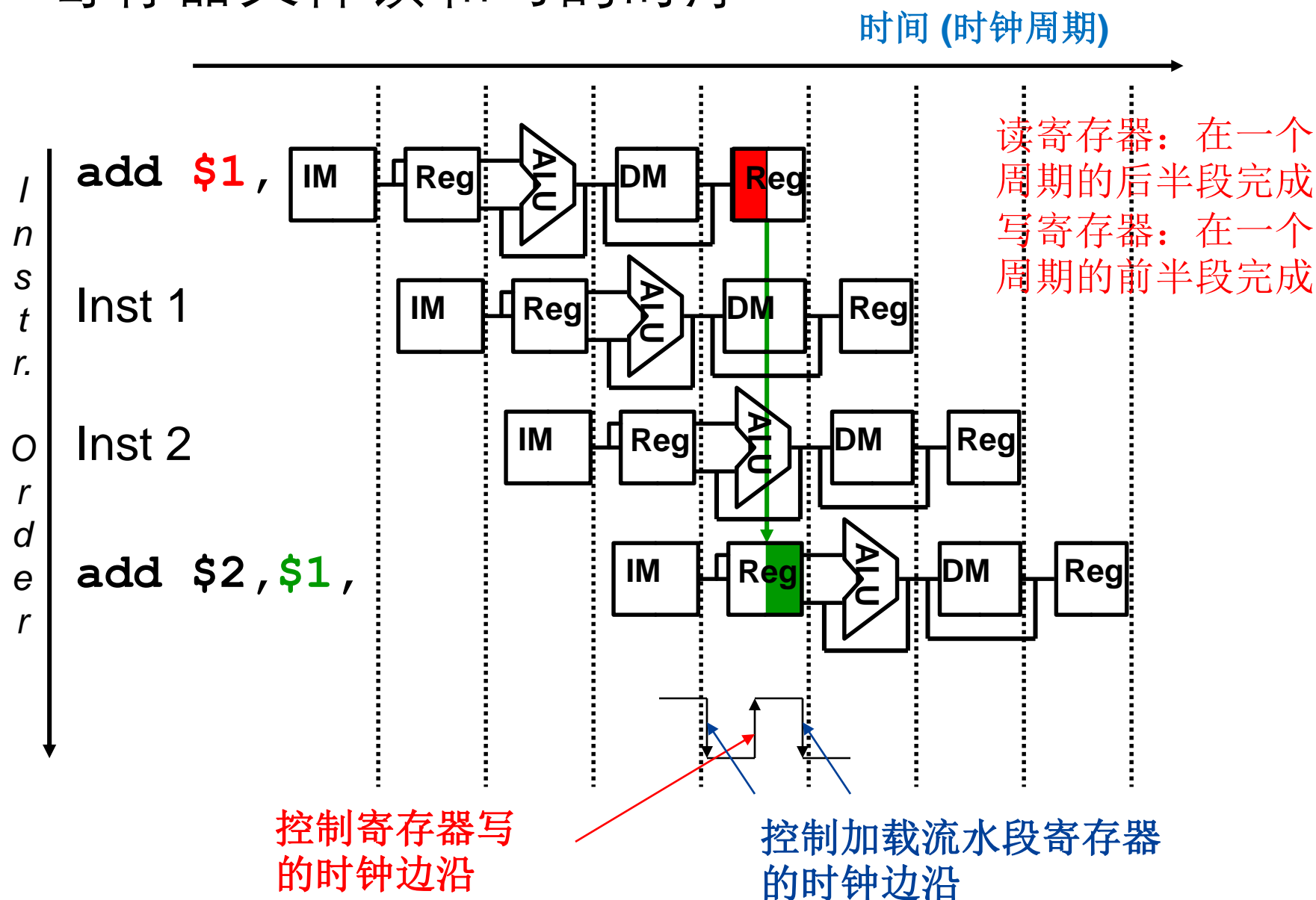
# 如果只有一个存储器：结构冒险



- 解决方案：指令存储器和数据存储器相互独立
- 指令cache 数据cache



# 寄存器文件读和写的时序



# 避免结构冒险



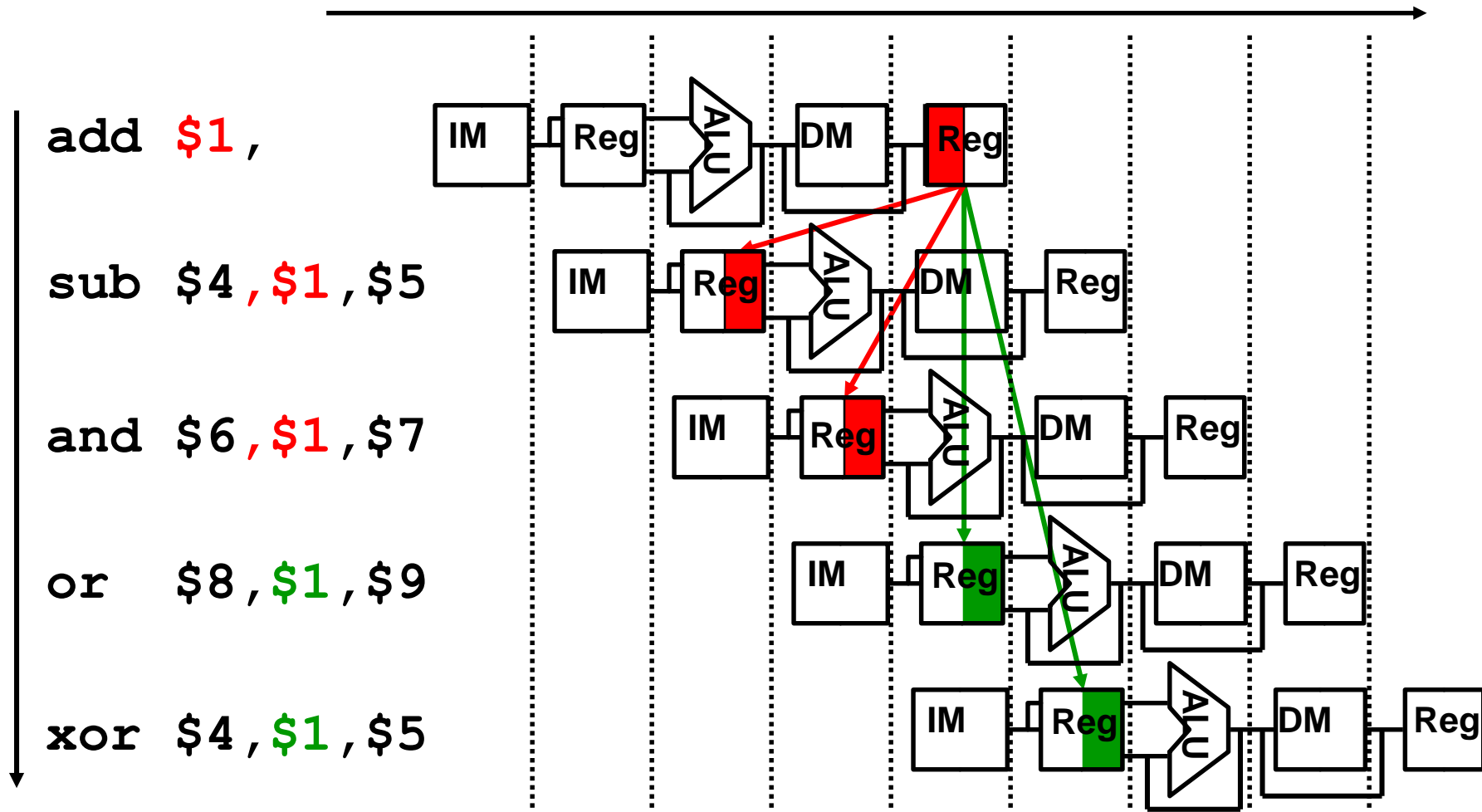
- 结构冒险（structural hazards）：当硬件资源满足不了同时重叠执行的指令的要求，而发生资源冲突时，就发生了结构冒险。
- 避免结构冒险
  - 所有功能单元完全流水化
  - 设置足够的硬件资源
  - 硬件代价很大

# 数据冒险



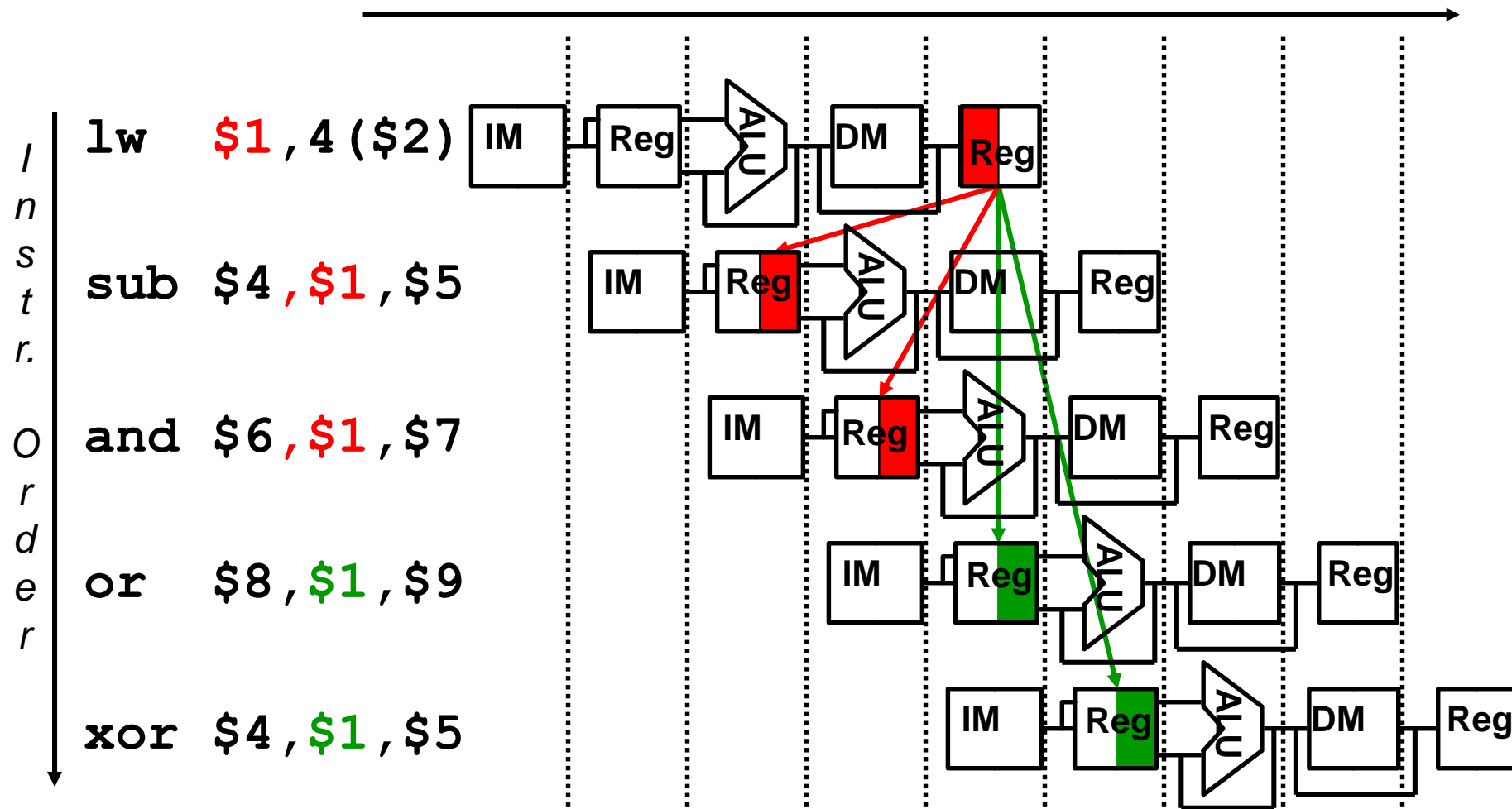
- 数据冒险（data hazards）当一条指令需要用到前面某条指令的结果，从而不能重叠执行时，就发生了数据冒险

# 寄存器的使用导致的数据冒险



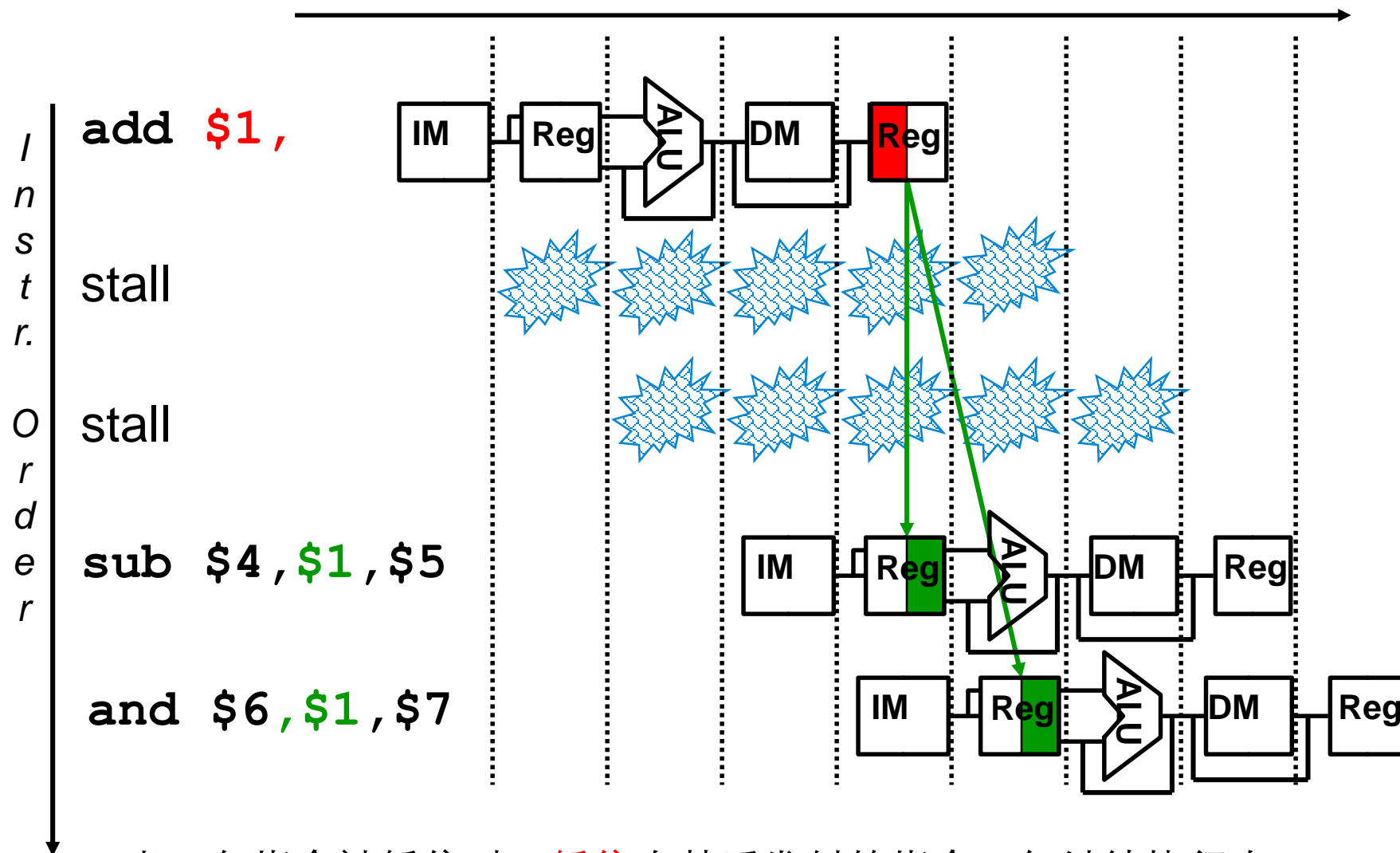
❑ 写后读数据冒险 ( Read before write data hazard )

# 读存储器导致数据冒险



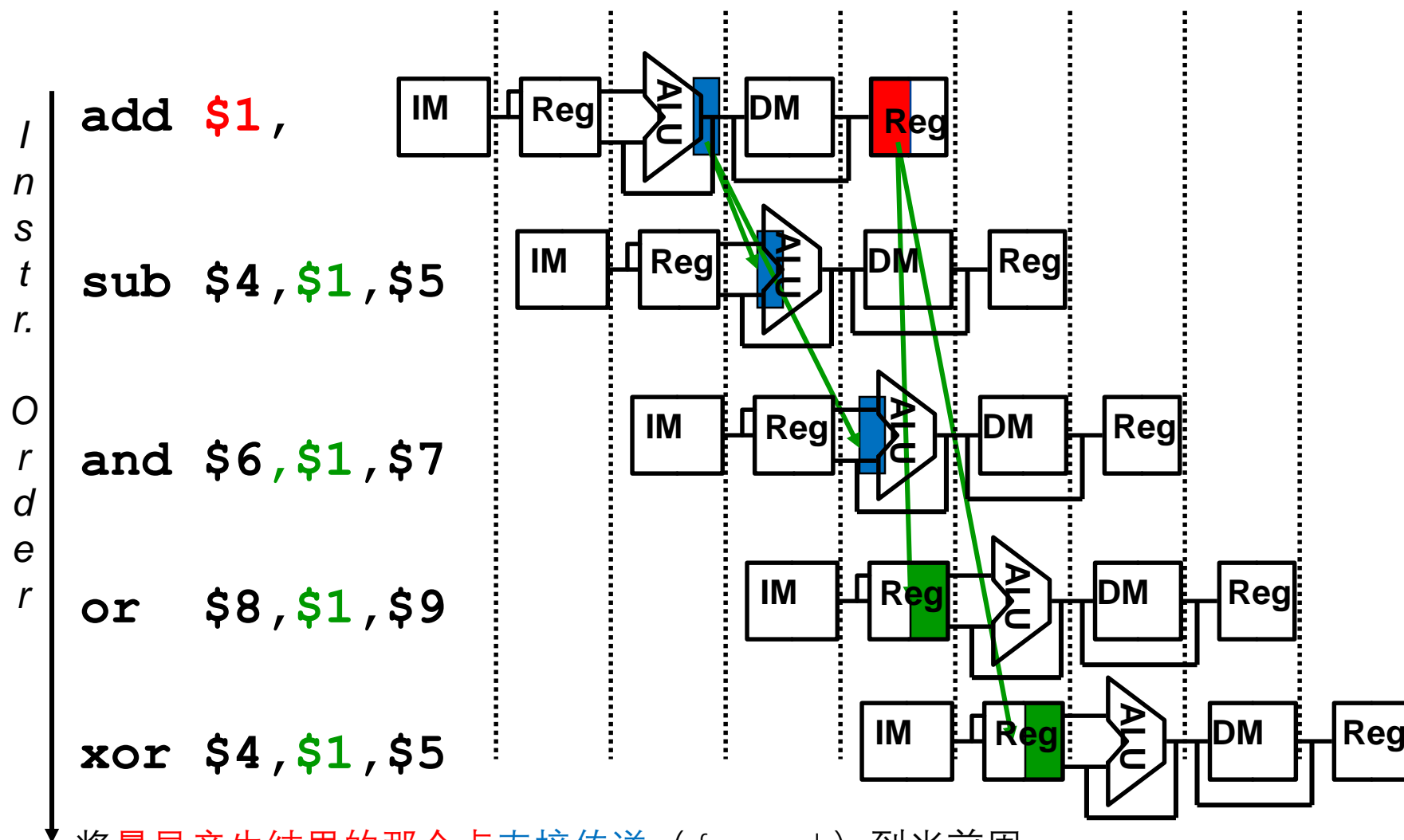
❑ “读存储器-使用” 冒险 ( Load-use data hazard )

# 一种解决办法：停顿



当一条指令被暂停时，**暂停**在其后发射的指令，但继续执行在其前发射的指令。

# 另一种解决方案：前向通路（forwarding）



将最早产生结果的那个点直接传送（forward）到当前周期需要结果的功能单元（functional units）（例如：ALU）

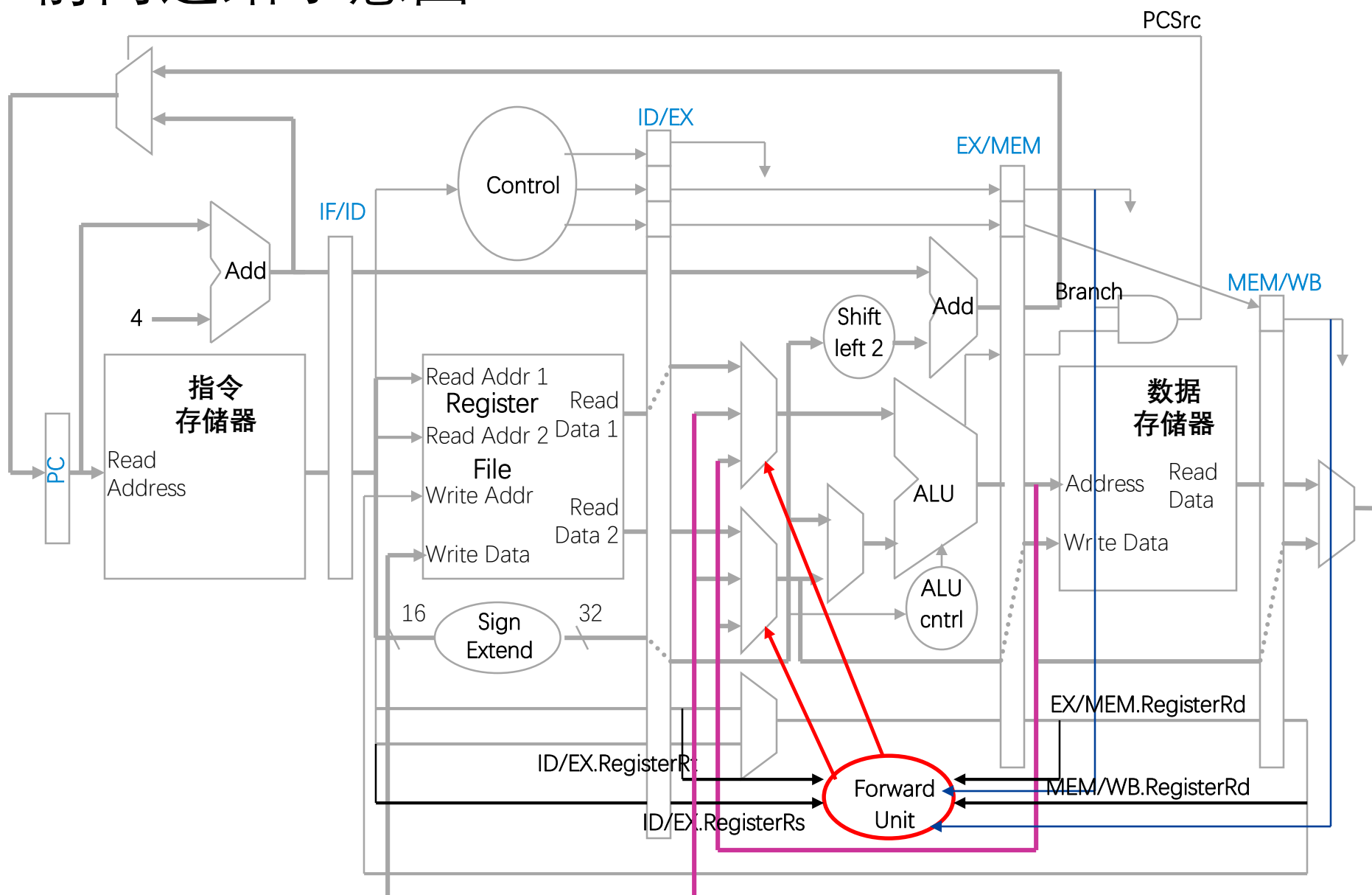
# 前向通路 (Forwarding)



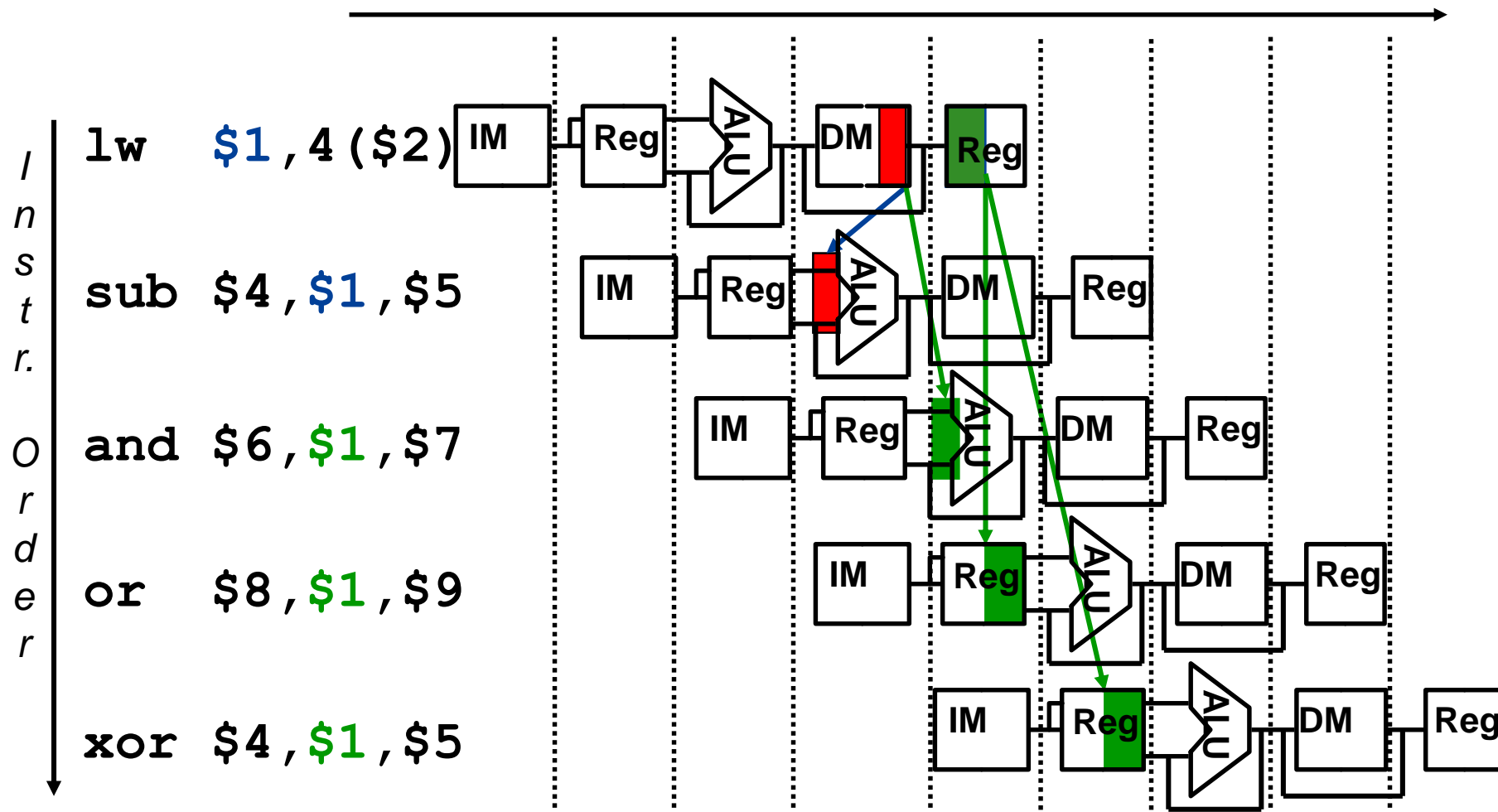
- 当硬件检测到前面某条指令的结果寄存器就是当前指令的源寄存器时，控制逻辑会将前面那条指令的结果直接从其产生的地方定向到当前指令所需的位置。
- 一个功能单元的输出生不仅可以定向到其自身的输入，而且还可以定向到其它单元的输入
- 任何流水寄存器到任何功能单元的输入都可能需要前向通路



# 前向通路示意图

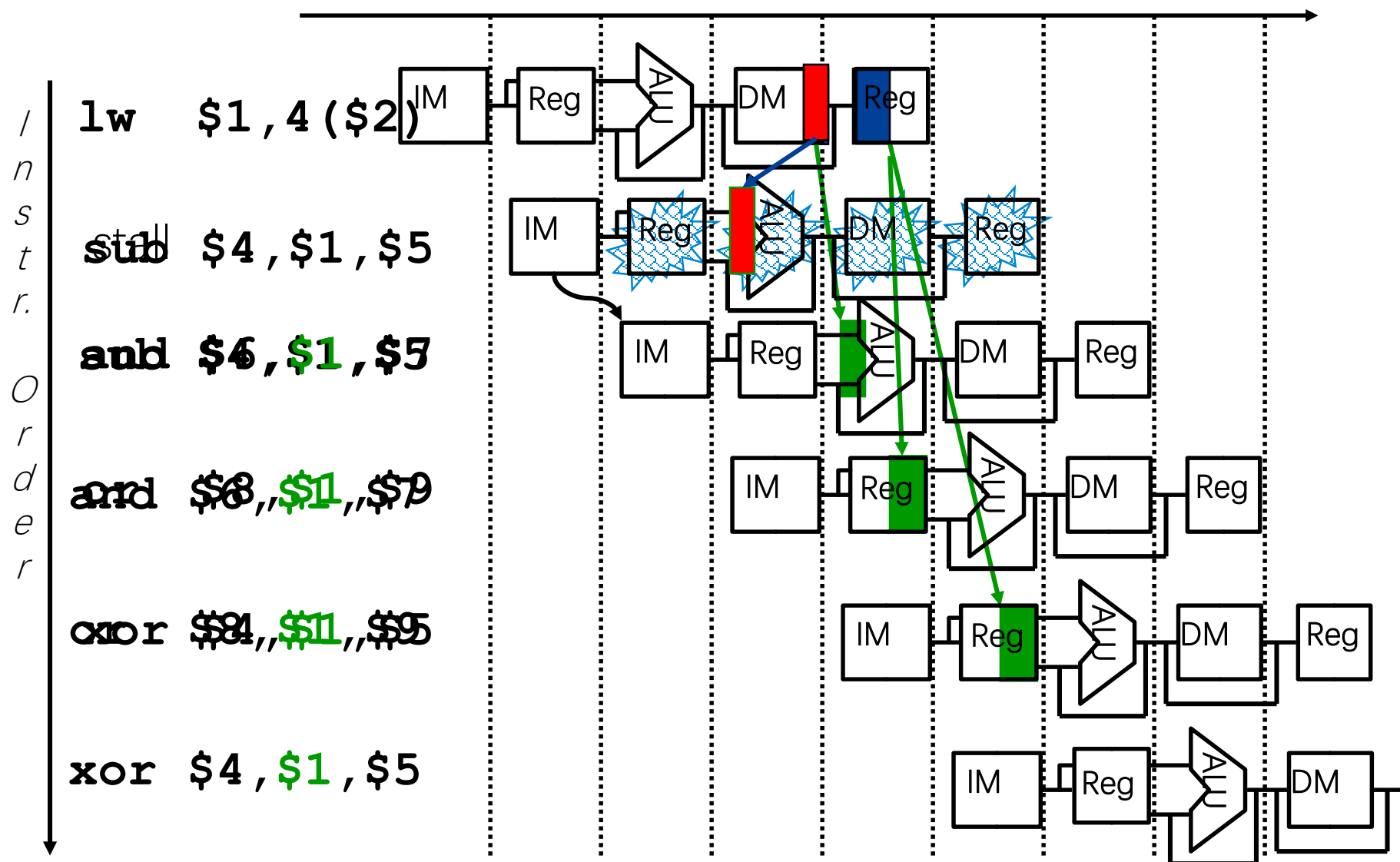


# 用前向通路能解决“读存储器-使用”冒险吗？



- 仍然需要一周期的停顿

# 前向通路不能解决“读存储器-使用”冒险



□ 仍然需要一个周期的停顿

# 流水线“时空图”



插入一个周期停顿的流水线时空图

LW R1, 0 (R2)	IF	ID	EX	MEM	WB					
SUB R4, R1, R5		IF	ID	stall	EX	MEM	WB			
AND R6, R1, R7			IF	stall	ID	EX	MEM	WB		
OR R8, R1, R9				stall	IF	ID	EX	MEM	WB	



## 流水线数据冒险（相关） 检测硬件检测的各种情况

相关情况	指令序列范围	动作
没有相关	LW R1, 45(R2) ADD R5, R6, R7 SUB R8, R6, R7 OR R9, R6, R7	这三条指令与R1无关, 故不可能出现相关。
需要暂停的相关	LW R1, 45(R2) ADD R5, R1, R7 SUB R8, R6, R7 OR R9, R6, R7	比较器检测到ADD指令中使用R1作为源寄存器, 并在ADD指令进入EX段之前将之暂停。(同时也将SUB和OR指令暂停)
通过定向消除的相关	LW R1, 45(R2) ADD R5, R6, R7 SUB R8, R1, R7 OR R9, R6, R7	比较器检测到SUB指令中使用R1作为源寄存器, 并及时的在SUB指令进入EX段之前将Load指令的结果定向到ALU。
按顺序访问的相关	LW R1, 45(R2) ADD R5, R6, R7 SUB R8, R6, R7 OR R9, R1, R7	LW指令在前半周期将结果写入R1 OR指令在后半周期读R1

# 小结



- 流水线冒险的定义
  - 结构冒险
  - 数据冒险
  - 控制冒险
- 数据冒险
  - 使用寄存器引发的数据冒险
  - “读存储器-使用”引发的数据冒险
- 数据冒险解决方案
  - 停顿
  - 前向通路

谢谢！

