

实验四 流水线CPU设计

2024春季

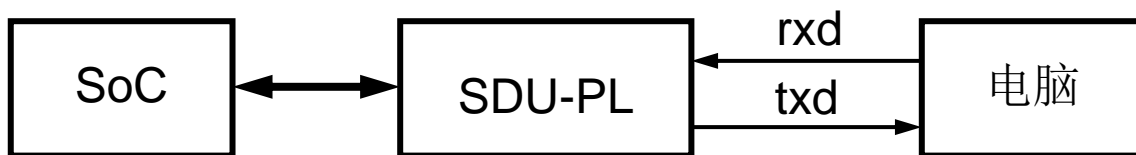
zjx@ustc.edu.cn

实验目标

- 理解流水线LA32R CPU的结构和工作原理
- 掌握流水线LA32R CPU的设计和调试方法，特别是流水线中的数据相关和控制相关的处理
- 掌握查看生成电路及其性能和资源使用情况

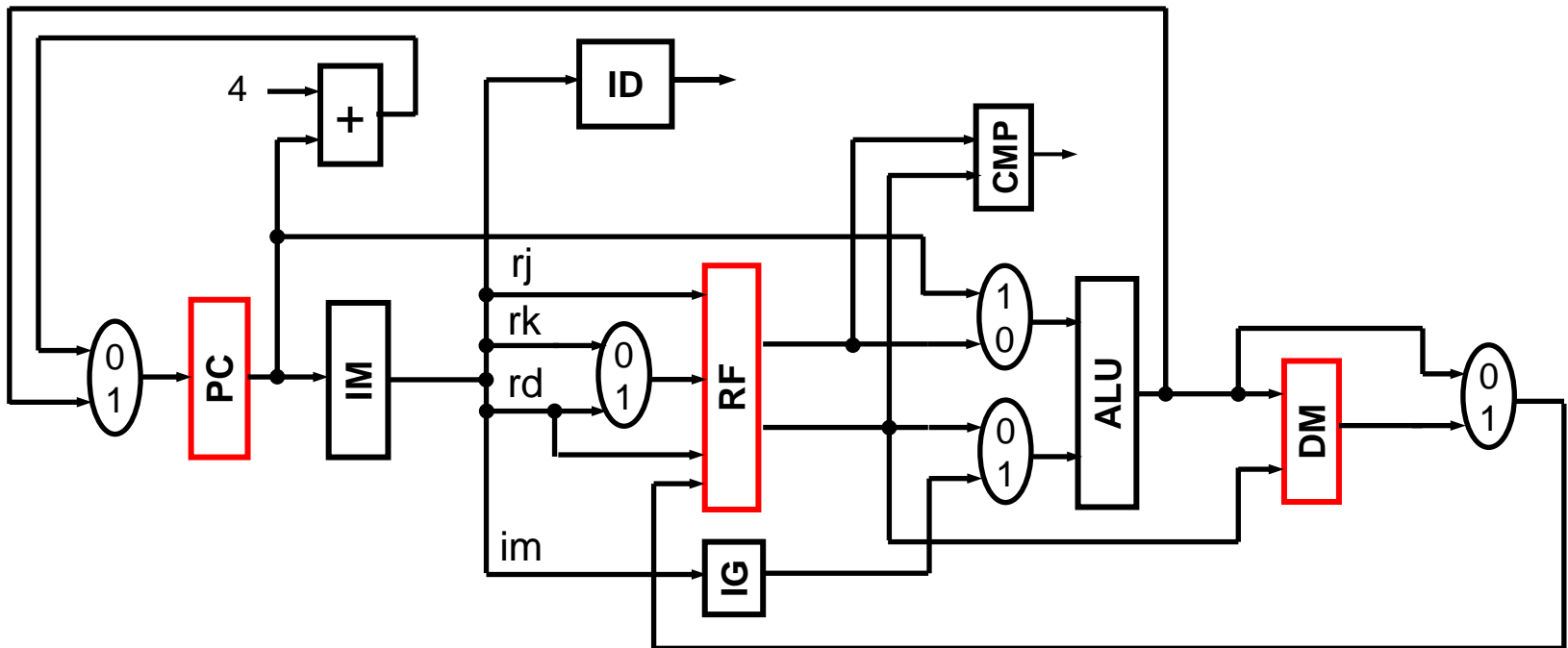
实验内容

- 设计流水线 **LA32R CPU**，可以执行如下指令：
 - 算术/比较运算： **add.w**, **addi.w**, **lu12i.w**, sub.w, pcaddu12i, slt, sltu, slti, sltui
 - 逻辑运算： and, or, nor, xor, andi, ori, xori
 - 移位运算： slli.w, srli.w, srai.w, sll.w, srl.w, sra.w
 - 访存： **ld.w**, **st.w**, ld.b, st.b, st.h, ld.h, ld.hu, ld.bu
 - 转移： **bne**, beq, b, bl, jirl, blt, bge, bltu, bgeu
- 构建包含**LA32R CPU**的片上系统(**SoC**)，连接流水线版的串行调试单元(**SDU-PL**)上板验证



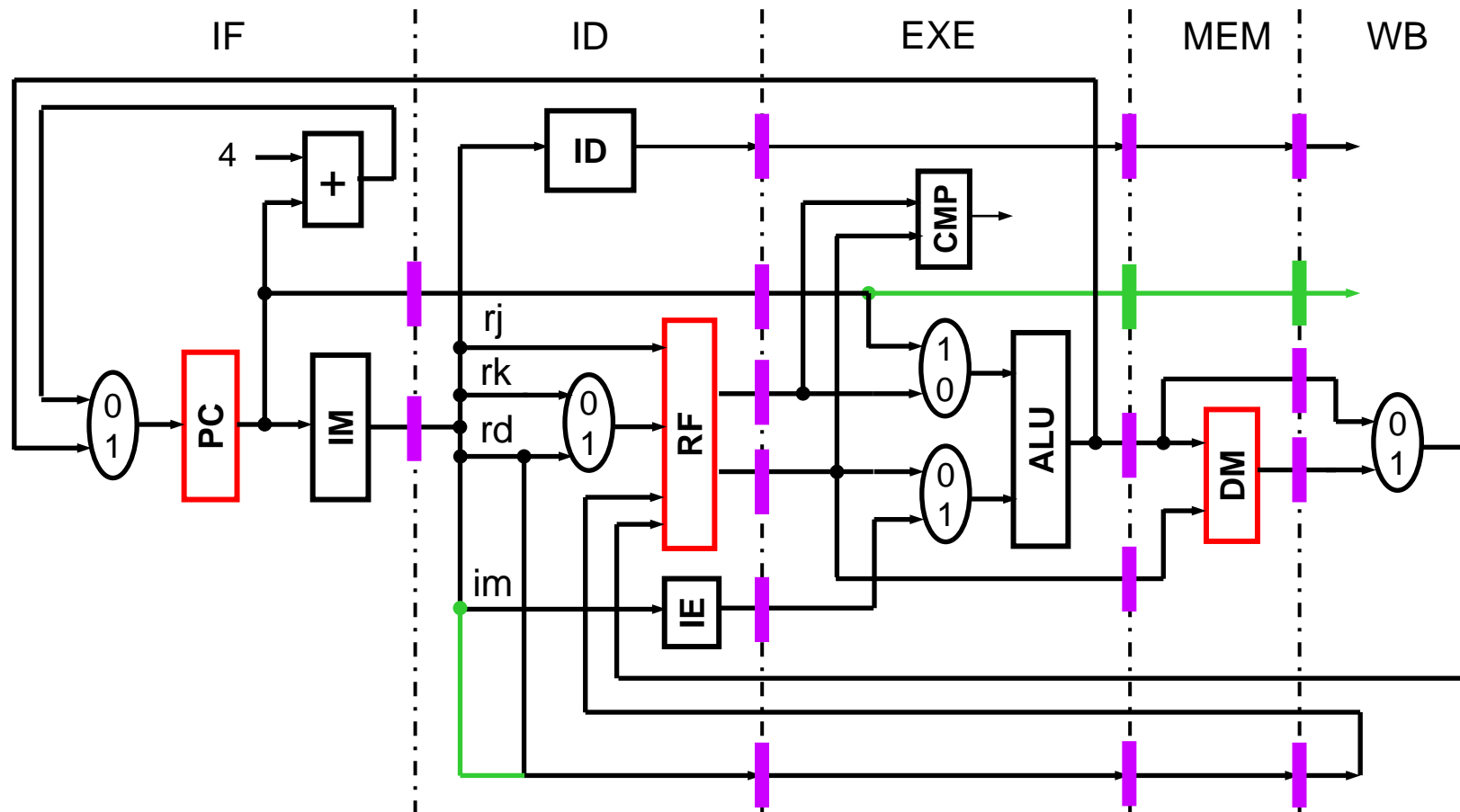
单周期数据通路

- LA32R C1-CPU的数据通路
 - add.w, addi.w, lui2i.w, ld.w, st.w, bne

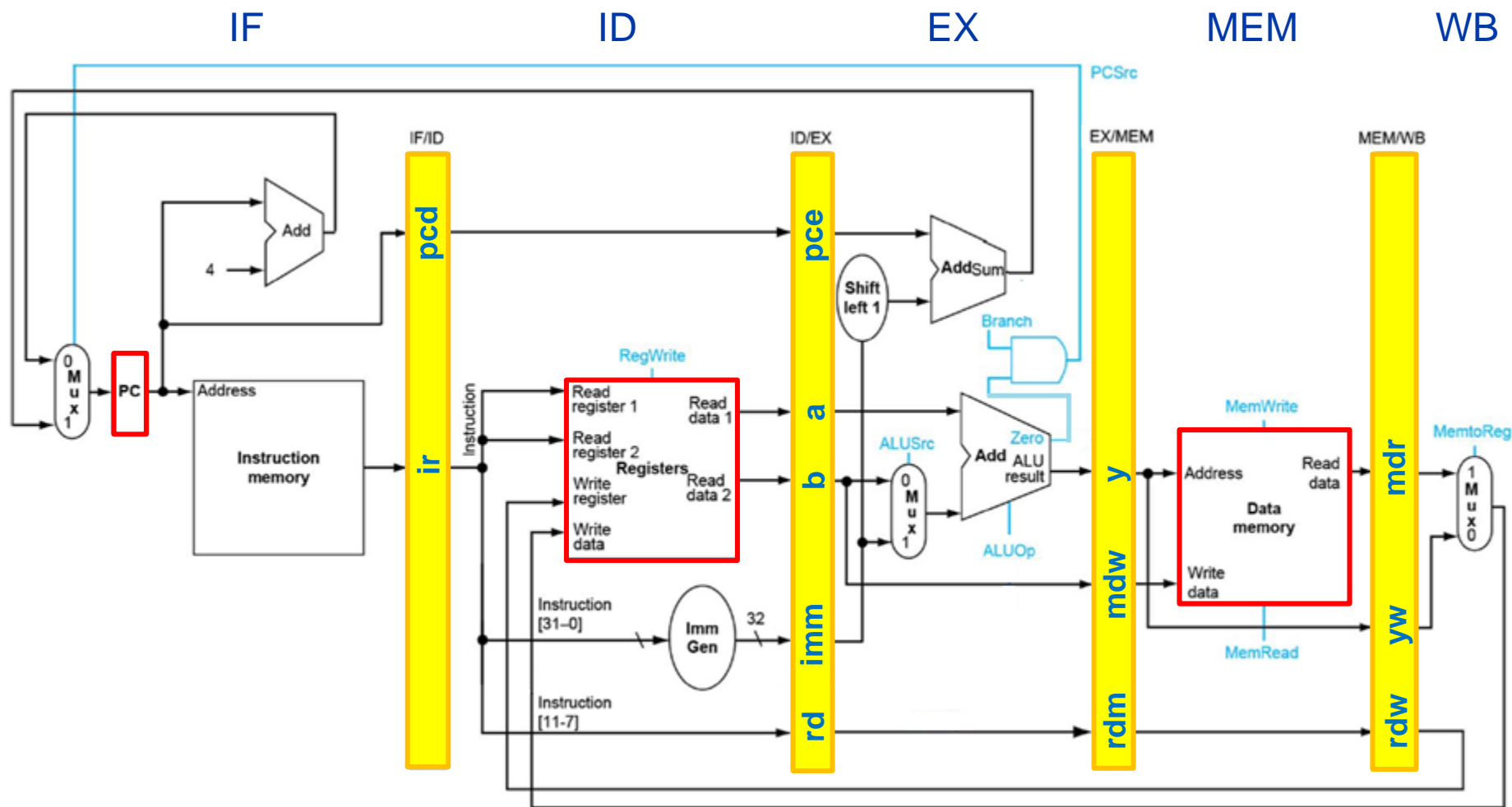


流水线数据通路

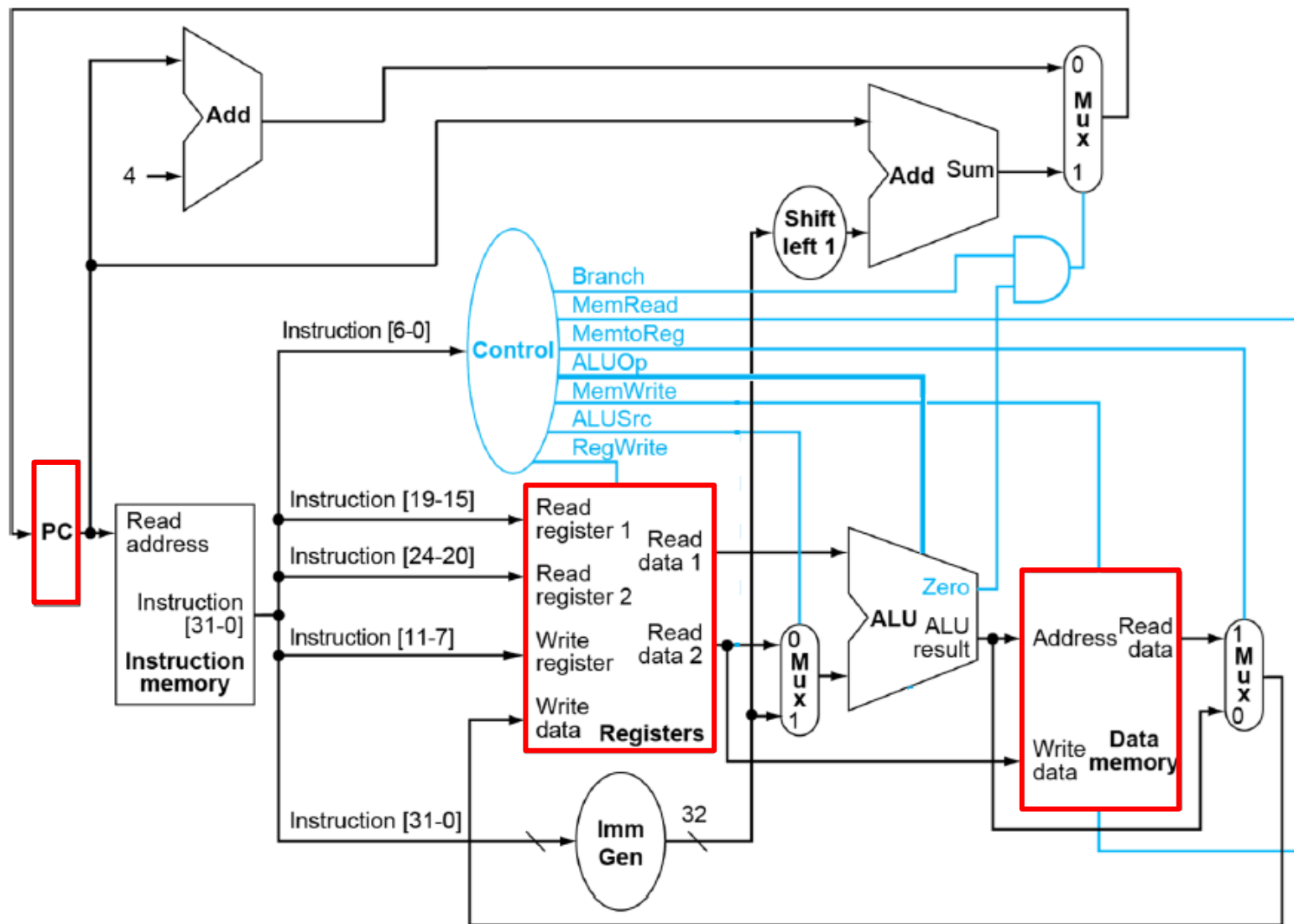
- LA32R C1-CPU的数据通路



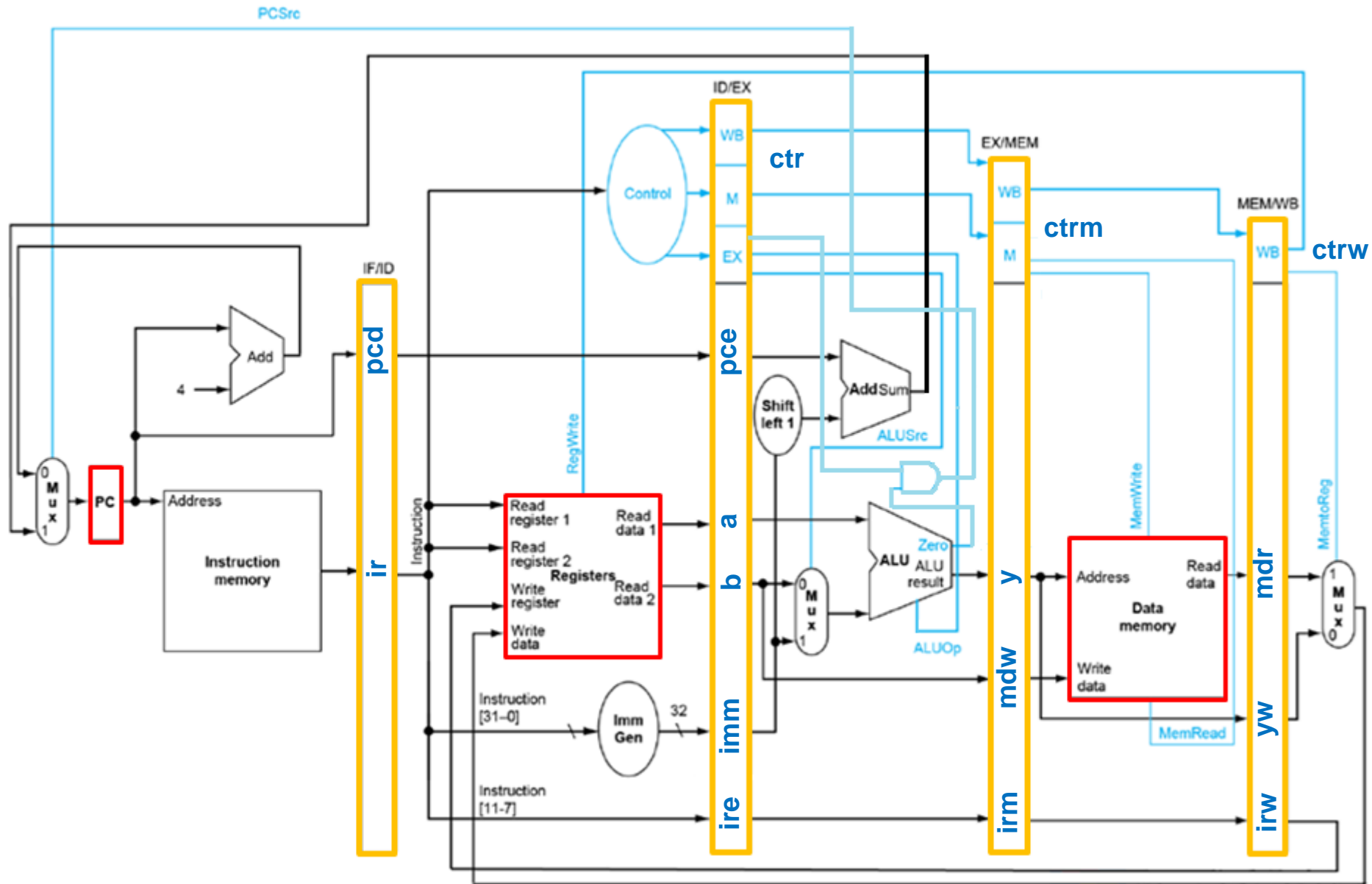
流水线CPU数据通路



单周期CPU数据通路+控制器



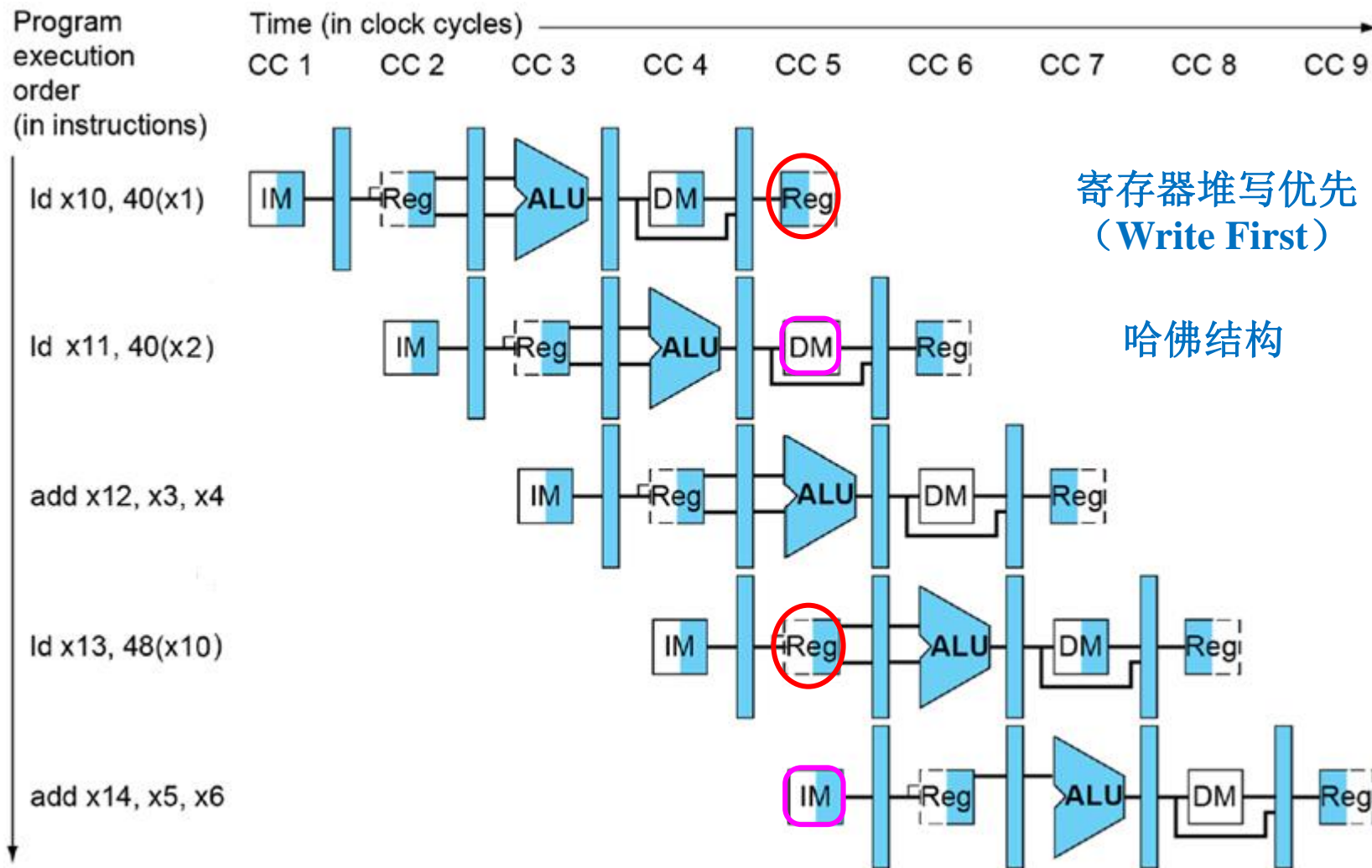
流水线CPU数据通路+控制器



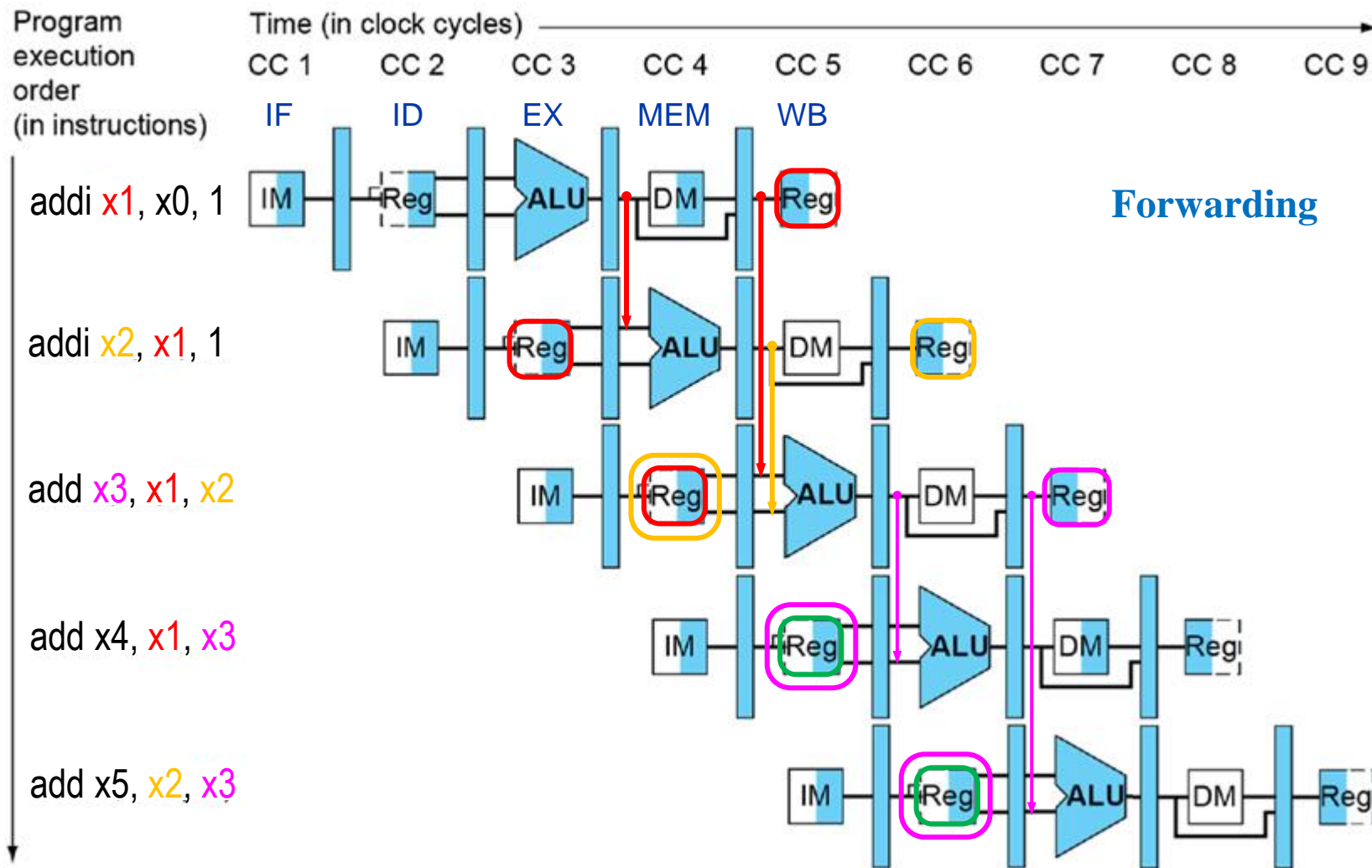
流水线相关及其处理

- **结构相关：当多条指令执行时竞争使用同一资源时**
 - 存储器相关处理：哈佛结构（指令和数据存储器分开）
 - 寄存器堆相关处理：同一寄存器读写时，写优先（Write First）
- **数据相关：当一条指令需要等待前面指令的执行结果时**
 - 数据定向（Forwarding）：将执行结果提前传递至之前流水段
 - 加载-使用相关（Load-use hazard）：阻止紧随Load已进入流水线的指令流动（Stall），向后续流水段插入空操作（Bubble）
- **控制相关：当遇到转移指令且不能继续顺序执行时**
 - 清除（Flush）紧随转移指令已进入流水线的指令
 - 从转移目标处取指令后执行

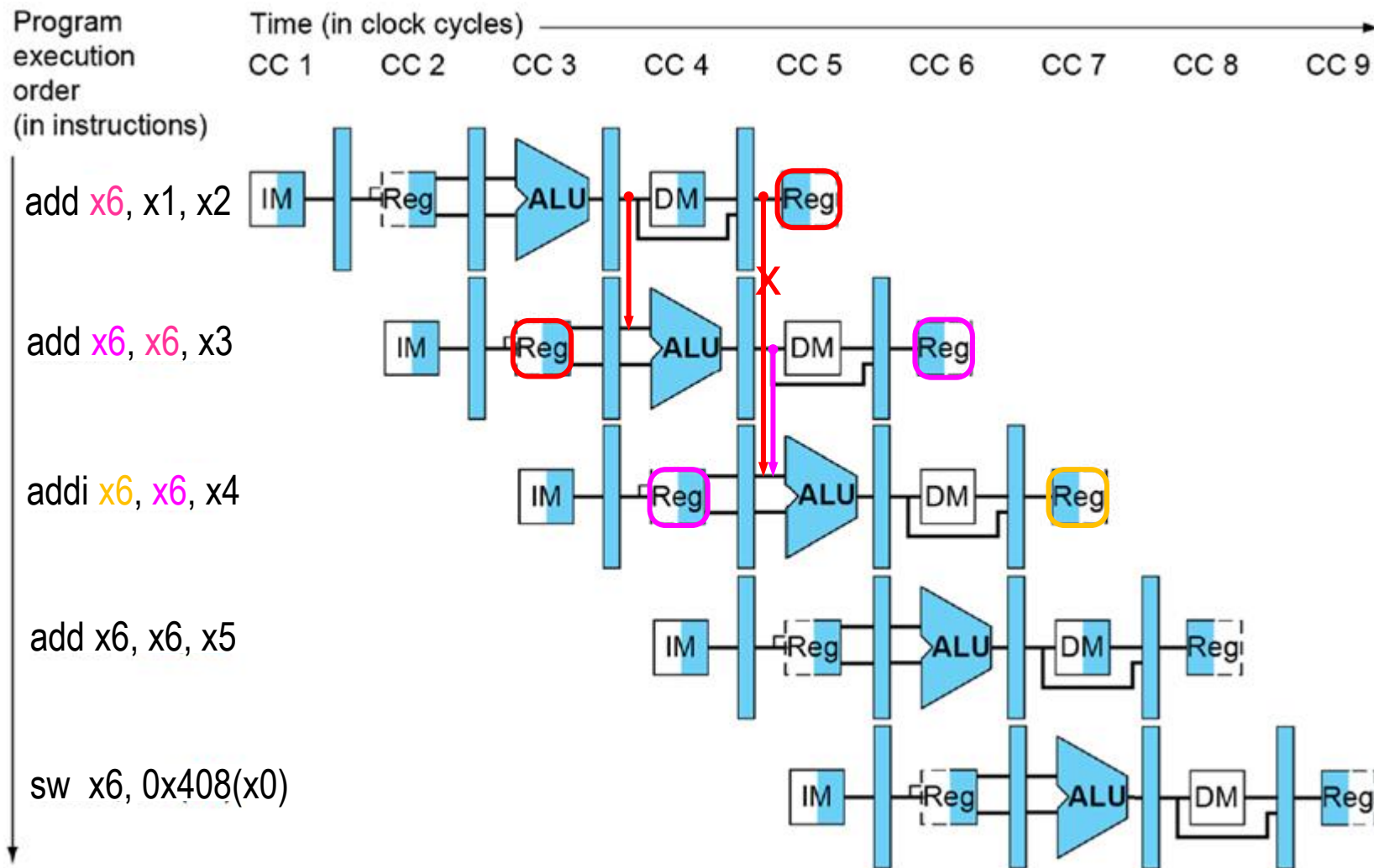
流水线相关：结构相关



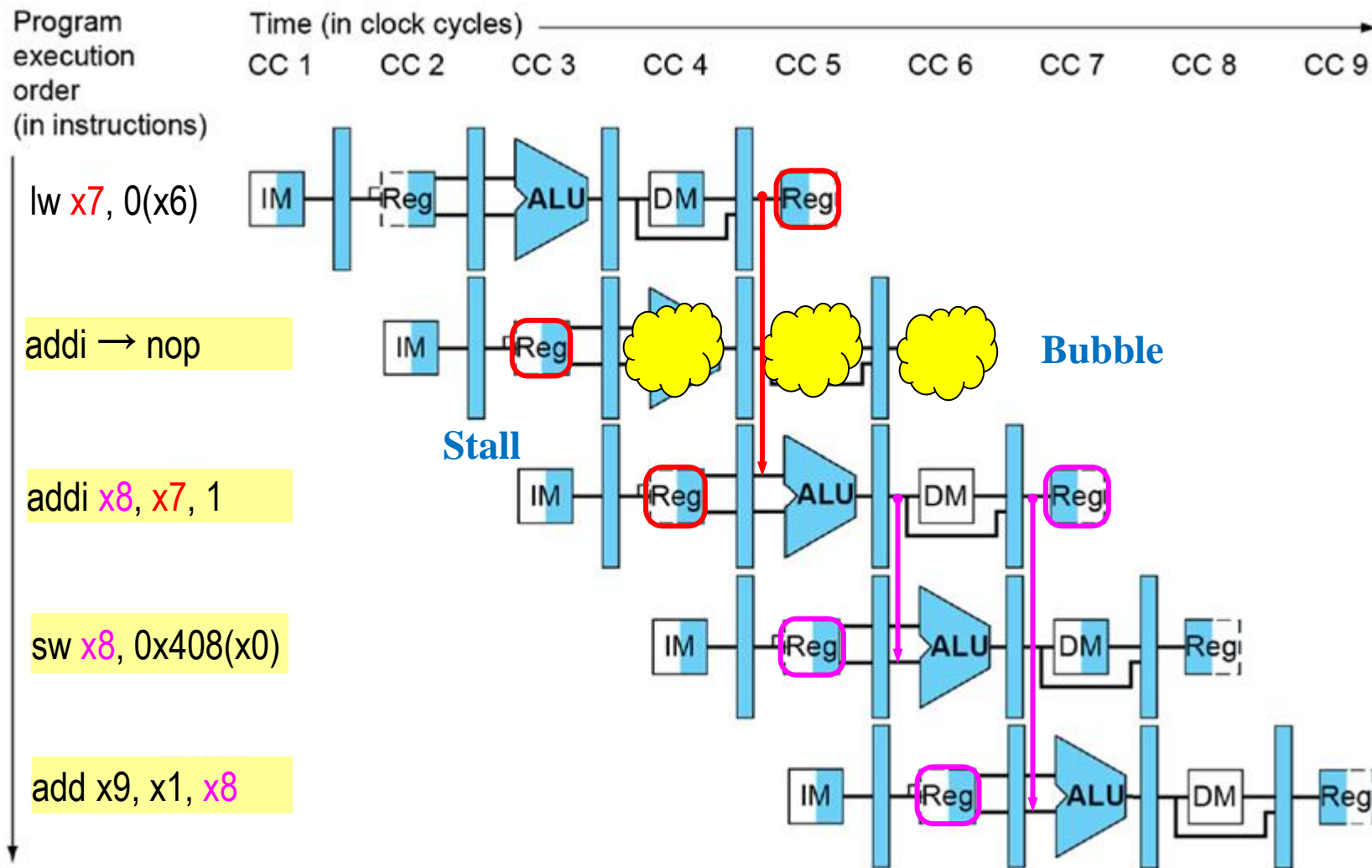
流水线相关：数据相关



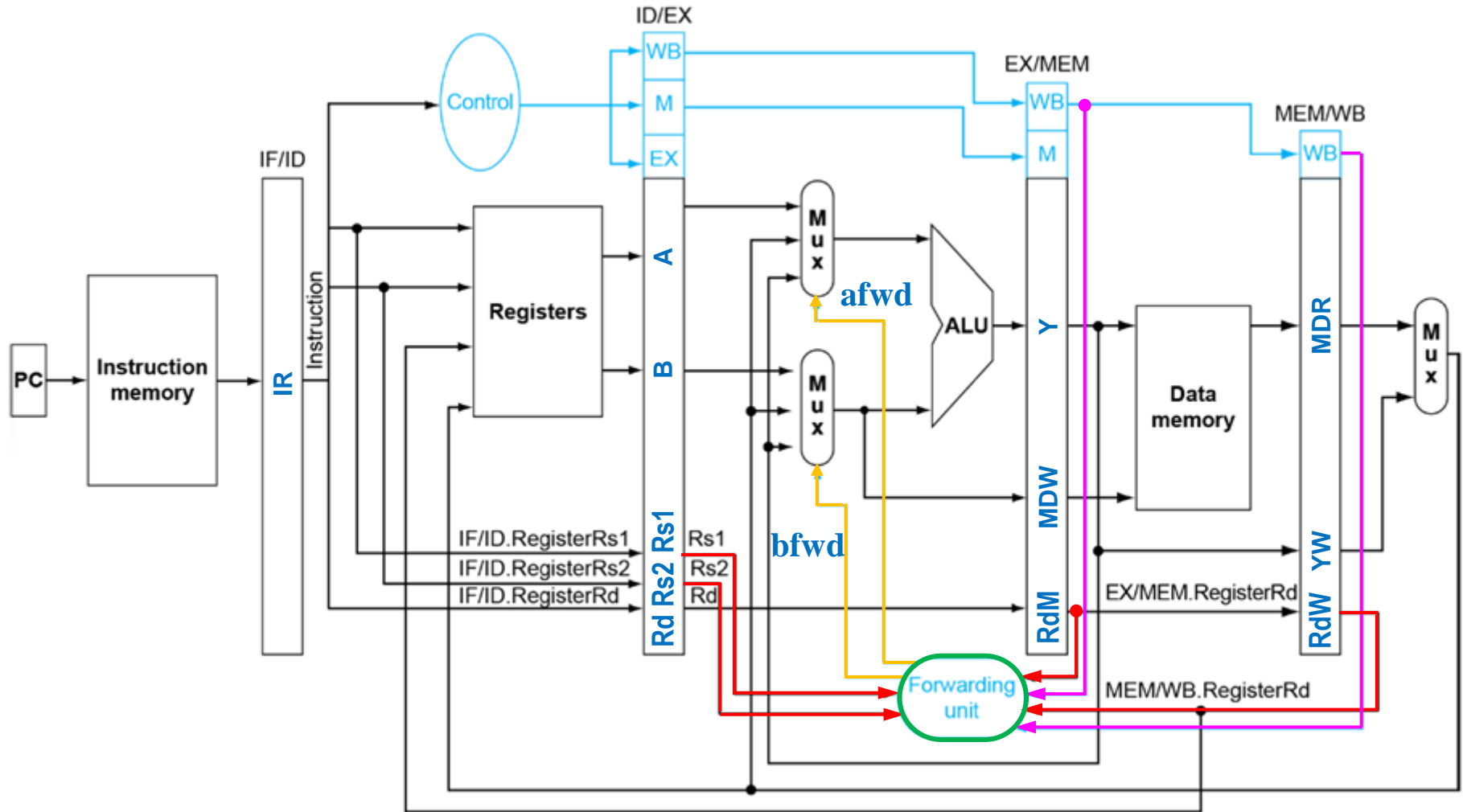
流水线相关：数据相关 (续1)



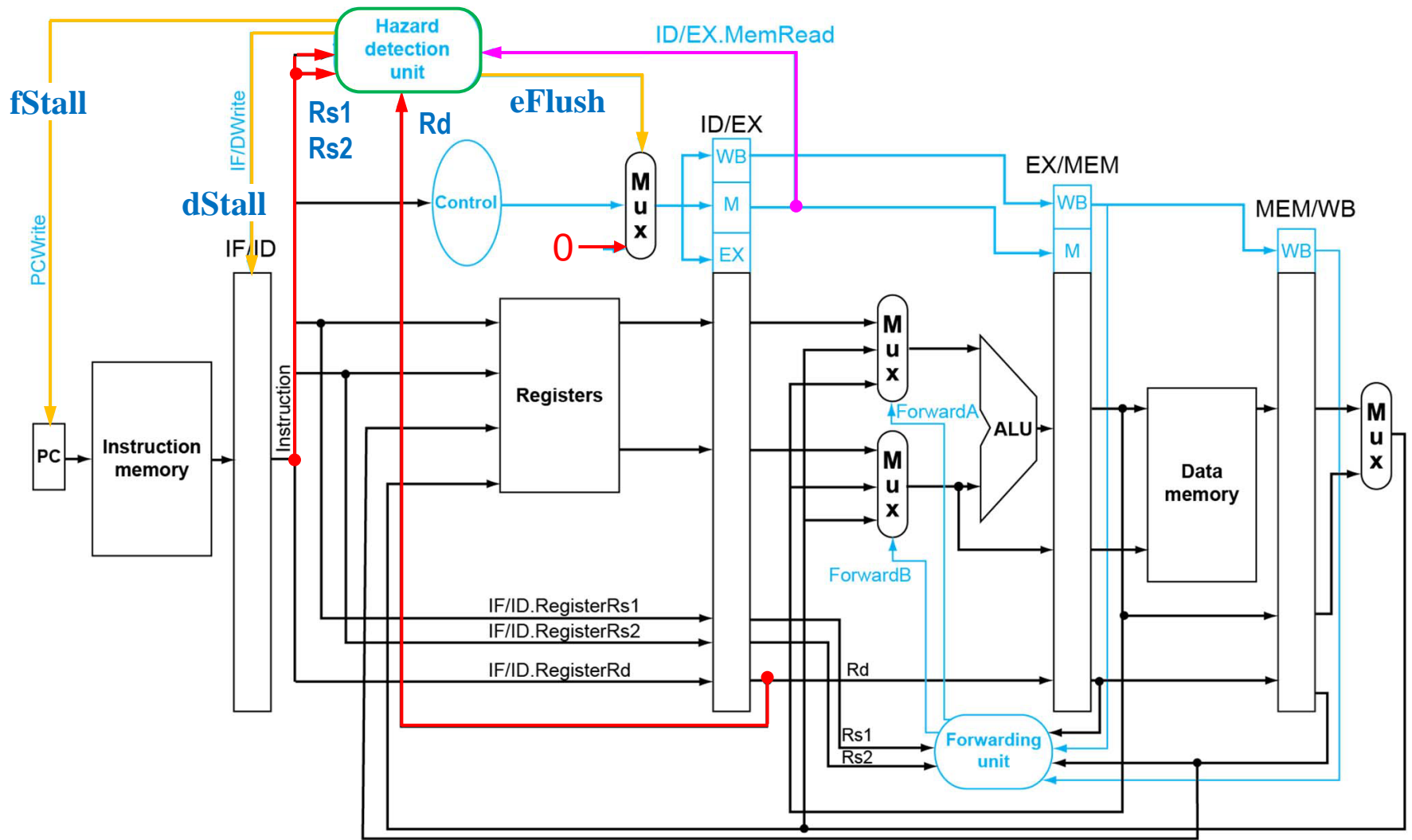
数据相关 (续2)



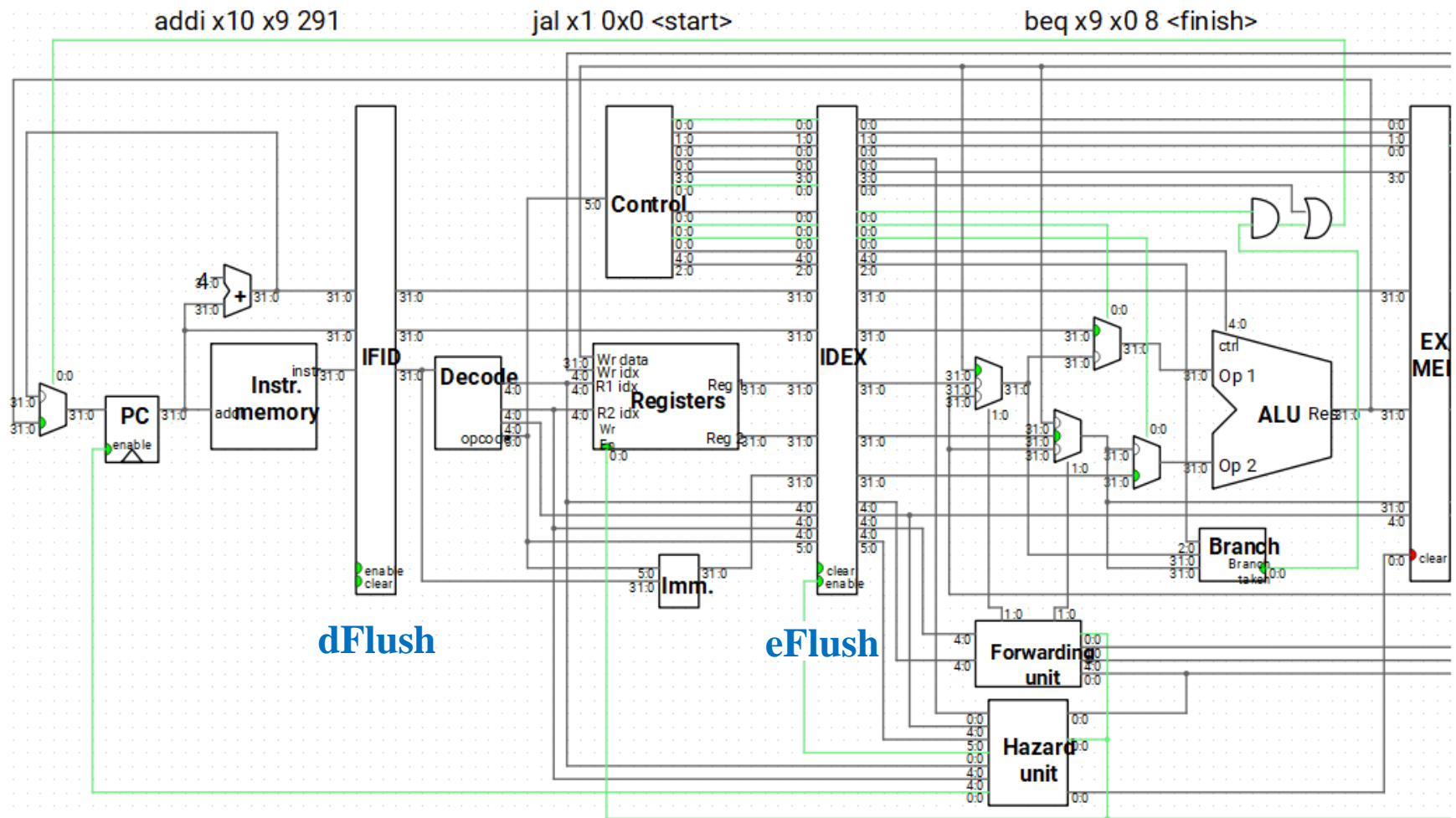
Forwarding



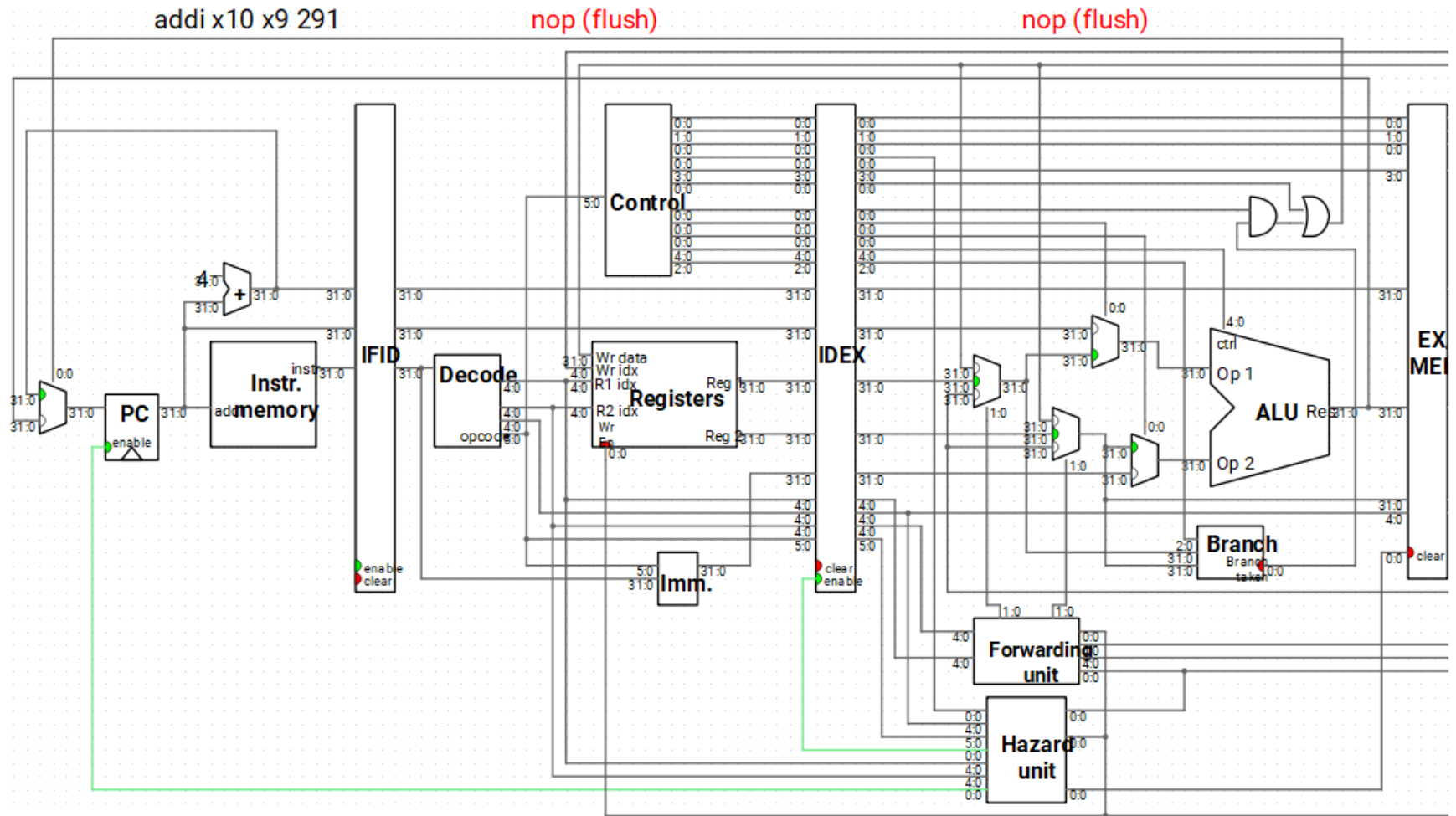
Load-Use Hazard



Branch Hazard: Ripes

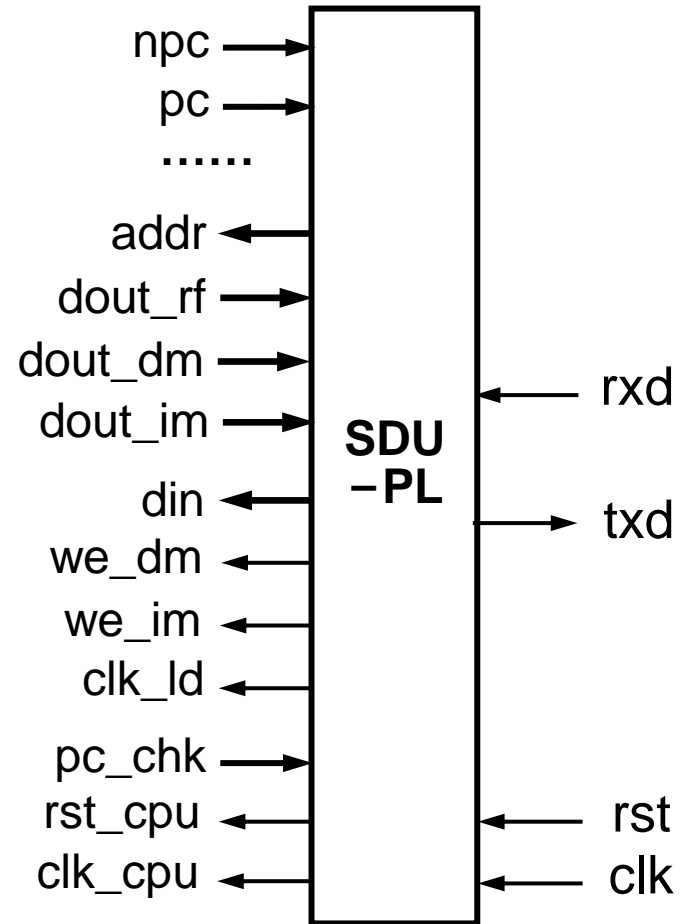


Branch Hazard: Ripes



串行调试单元—流水线

- 与调试单周期CPU的SDU不同点在于
 - P命令时查看的数据通路信息
IF, ID: npc, pc, ir, pcd
EXE: ire, imm, a, b, pce, ctr
MEM: irm, mdw, y, ctrm
WB: irw, yw, mdr, ctrw
 - G命令时检查的执行指令指针
pc_chk = pce



实验要求

- **设计流水线LA32R CPU，构建SoC并上板验证**
 - 指令和数据存储器均为8K x 32位的分布式存储器
 - 查看电路资源和性能
 - 运行指令测试程序
 - 运行排序程序
- **选项：增添乘/除法运算指令**
 - mul.w, mulh.w, mulh.wu, div.w, div.wu, mod.w, mod.wu
 - 查看电路资源和性能
 - 运行指令测试程序

The End