山东大学 计算机科学与技术 学院

计算机体系结构 课程实验报告

学号：202200130048	姓名： 陈静雯	班级： 6
实验题目：实验四：结构相关		
实验学时：2	实验日期： 2025.5.16	
实验目的： 通过本实验，加深对结构相关的理解，了解结构相关对 CPU 性能的影响		
硬件环境： Windows		
软件环境： Otvdm		

实验程序：structure_d.s

代码解读：

1. LHI R2, (A>>16)&0xFFFF
   将数组 A 的高 16 位地址加载到寄存器 R2 的高 16 位，低 16 位清零。
2. ADDUI R2, R2, A&0xFFFF
   将数组 A 的低 16 位地址加到 R2，形成完整的 32 位基地址。
3. LHI R3, (B>>16)&0xFFFF
   类似地，加载数组 B 的高 16 位地址到 R3。
4. ADDUI R3, R3, B&0xFFFF
   补全 B 的 32 位基地址到 R3。
5. ADDU R4, R0, R3
   将 R3（B 的基地址）复制到 R4，用于循环终止条件。
6. loop: LD F0, 0(R2)
   从 R2 指向的 A 数组加载双精度数到浮点寄存器 F0。
7. LD F4, 0(R3)
   从 R3 指向的 B 数组加载双精度数到 F4。
8. ADDD F0, F0, F4
   计算 A[i] + B[i]，结果存入 F0。
9. ADDD F2, F0, F2
   累加 F0 到 F2（此处存在结构相关导致的暂停）。
10. ADDI R2, R2, #8
    指针 R2 递增 8 字节（指向 A 的下一个元素）。
11. ADDI R3, R3, #8
    指针 R3 递增 8 字节（指向 B 的下一个元素）。
12. SUB R5, R4, R2
    计算剩余元素数（R4 - R2）。
13. BNEZ R5, loop
    若 R5 不为零，继续循环。
14. TRAP #0

退出程序。

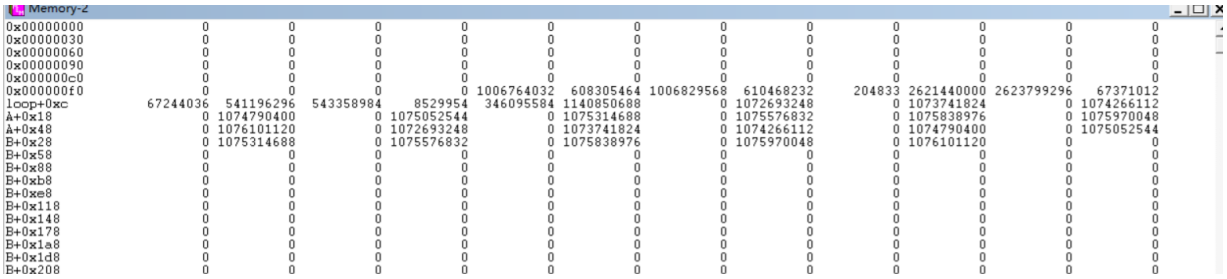
15. A: .double 1, ..., 10
    定义数组 A，包含 10 个双精度数。
16. B: .double 1, ..., 10
    定义数组 B，与 A 相同。

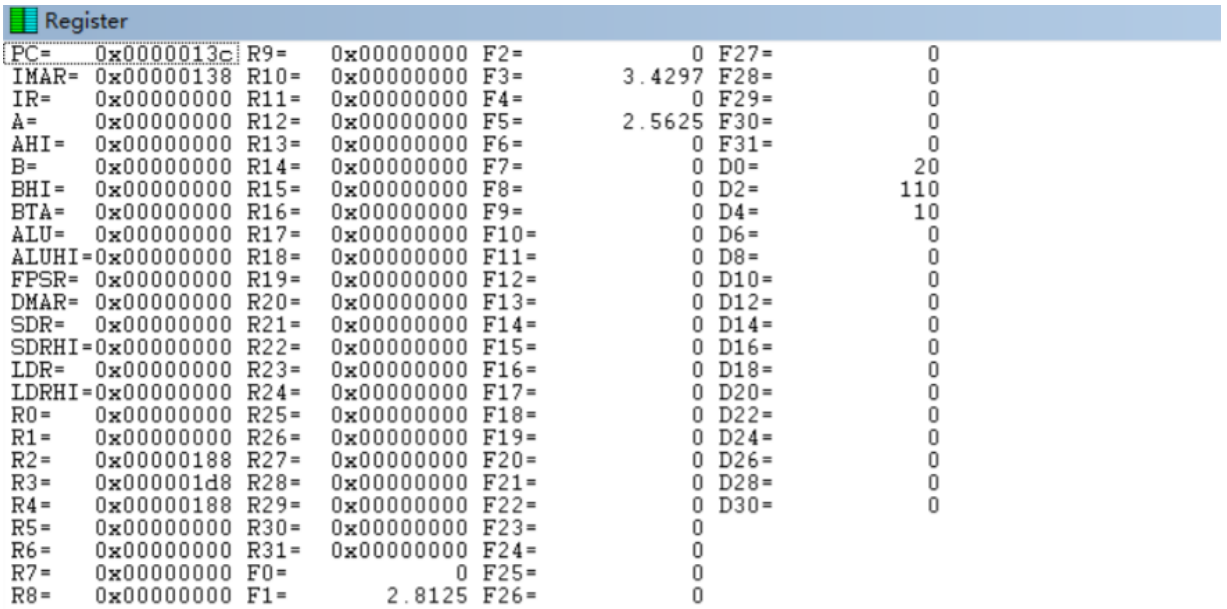
实验步骤：

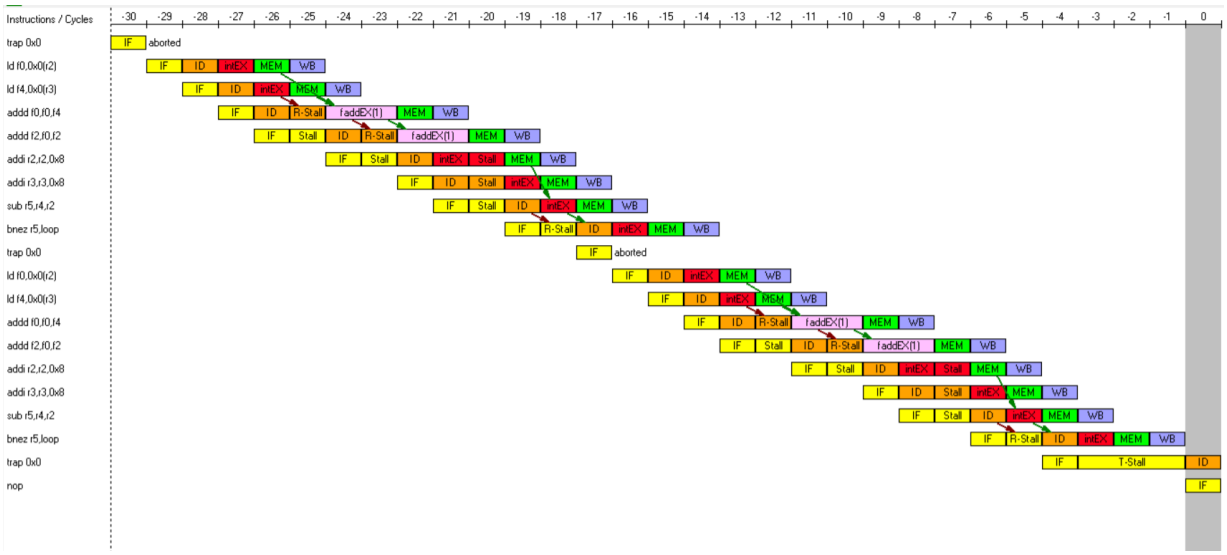
(1). 用 WinDLX 模拟器运行程序 structure_d.s 。
(2). 通过模拟，找出存在结构相关的指令对以及导致结构相关的部件。
(3). 记录由结构相关引起的暂停时钟周期数，计算暂停时钟周期数占总执行周期数的百分比。
(4). 论述结构相关对 CPU 性能的影响，讨论解决结构相关的方法。

实验内容：

（1） 实验结果

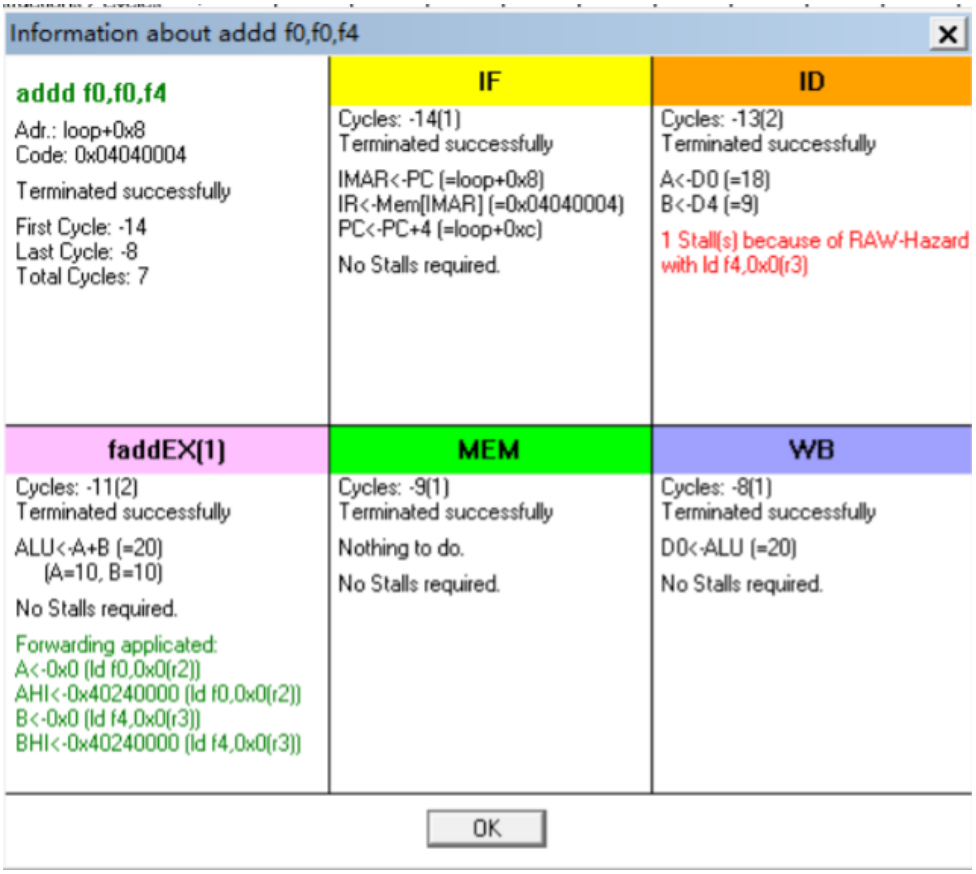


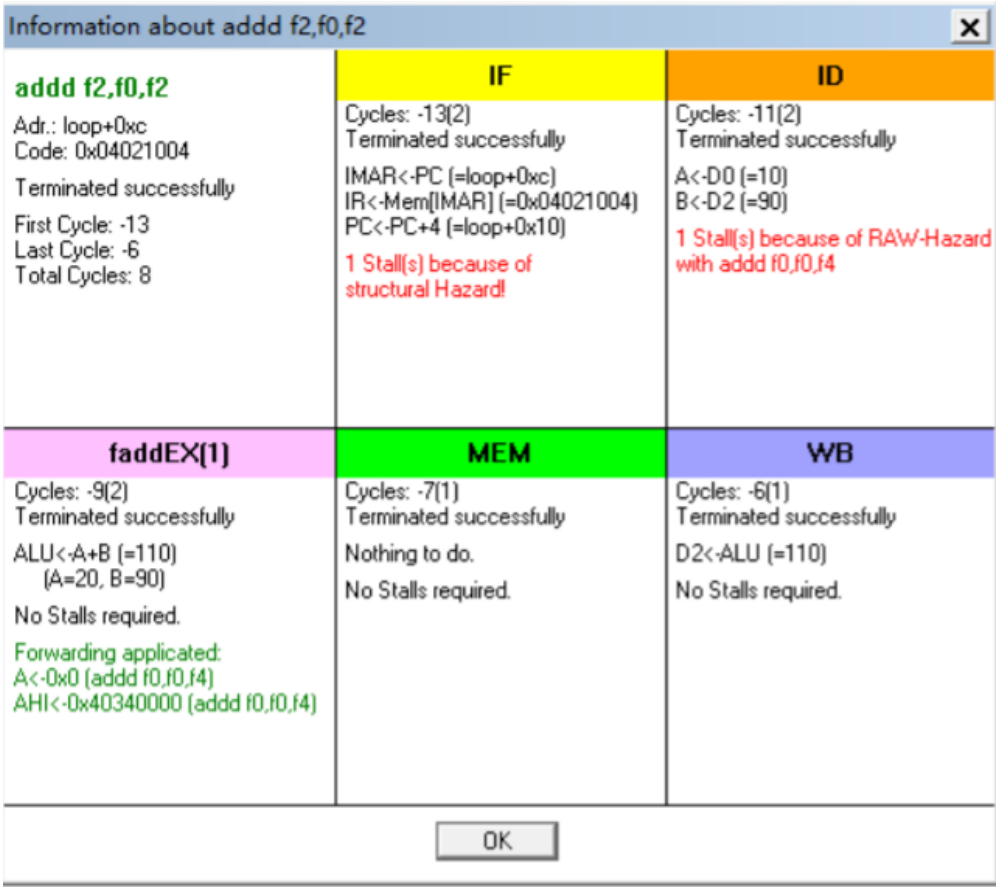




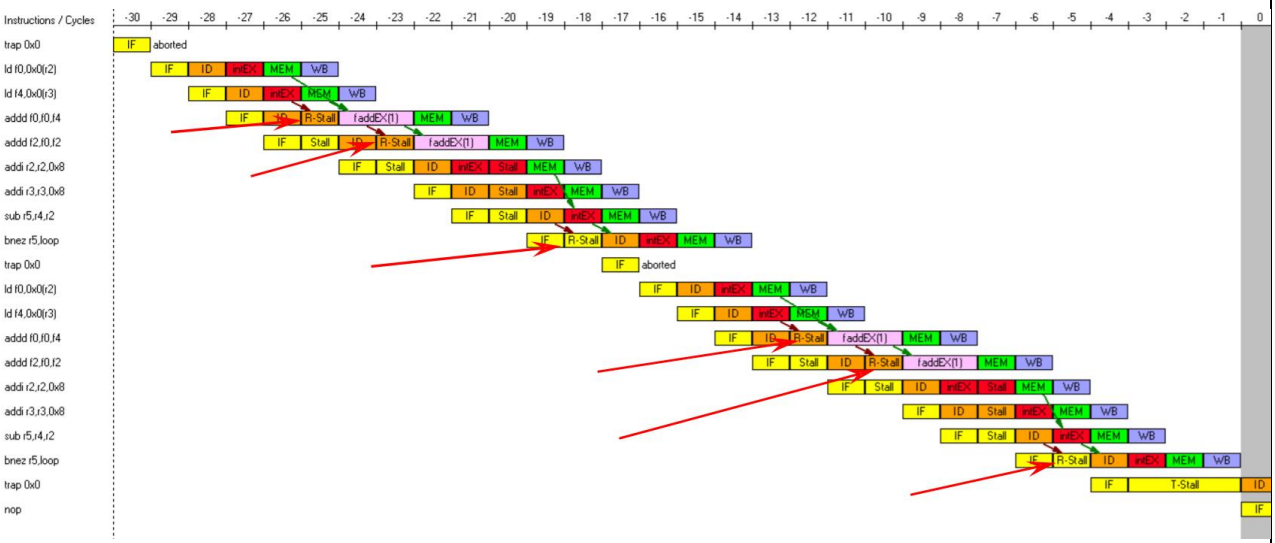
（2）存在的结构相关指令和部件

ADDD F2, F0, F2 和 ADDI R2, R2, #8 都用到了浮点数加法器，就要等前一条指令用完下一条指令才能用。





(3) 记录由结构相关引起的暂停时钟周期数，计算暂停时钟周期数占总执行周期数的百分比。



每次循环，总共有箭头指向的 6 处结构相关，每处暂停了一个时钟周期，总共暂停了 6 个周期，总执行周期 30，百分之 20%。

(4) 论述结构相关对 CPU 性能的影响，讨论解决结构相关的方法。

影响：

- 性能下降：流水线因资源冲突暂停，降低指令吞吐量。
- 资源争用：功能部件（如浮点加法器）成为瓶颈，限制并行性。

解决方法：

1. 增加硬件资源：添加多个浮点加法器，支持并发执行。

2. 流水线化部件：将功能部件划分为多级流水，允许指令重叠执行。
3. 指令调度：调整指令顺序，插入无关指令或使用编译器优化减少冲突。

结论分析与体会：

1. 结构相关的本质
   结构相关（Structural Hazard）源于硬件资源的冲突。例如，两条连续的浮点加法指令（ADDD F0, F0, F4 和 ADDD F2, F0, F2）争用非流水线化的浮点加法器，导致第二条指令必须等待第一条指令完成才能执行，引发流水线暂停。
2. 现代 CPU 的解决方案
   （1）动态调度（Tomasulo 算法）：通过寄存器重命名和保留站（Reservation Station）动态分配功能部件，缓解资源冲突。
   （2）超标量架构：支持同时发射多条指令到不同功能单元，结合乱序执行（Out-of-Order Execution）提升资源利用率。
3. 结构相关是流水线设计中不可忽视的性能瓶颈，其影响在资源受限或指令密集的场景下尤为显著。硬件增强（如多部件、流水线化）和软件优化（指令调度、循环展开）的结合是解决结构相关的关键。