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：202222130195	姓名：阎发祥	班级：22.3
实验题目：结构相关		
实验学时：2	实验日期：2024.5.7	

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：

WinDLX(一个基于 Windows 的DLX 模拟器)

软件环境：

VMware Workstation 16 Player

Windows 7

实验步骤与内容：

实验内容：

(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。

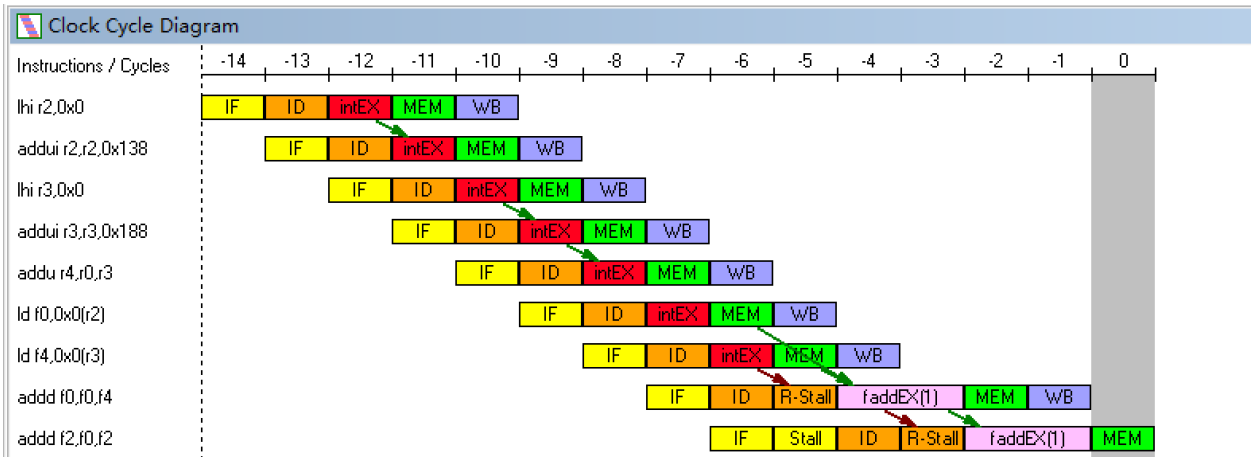
实验步骤：

（1）阅读汇编代码

```
1                LHI     R2, (A>>16)&0xFFFF      ;; R2的高16位设为A地址的高16位
2                ADDUI   R2, R2, A&0xFFFF        ;; R2加上A地址的低16位，R2中存储A的完整地址
3                LHI     R3, (B>>16)&0xFFFF      ;; R3的高16位设为B地址的高16位
4                ADDUI   R3, R3, B&0xFFFF        ;; R3加上B地址的低16位，R3中存储B的完整地址
5                ADDU    R4, R0, R3              ;; R4 = R3，保存数组B起始地址用于后续比较
6    loop:
7                LD      F0, 0(R2)               ;; 从R2指向的地址加载一个double到F0（A数组元素）
8                LD      F4, 0(R3)               ;; 从R3指向的地址加载一个double到F4（B数组元素）
9                ADDD    F0, F0, F4              ;; F0 = F0 + F4，执行A[i] + B[i]
10               ADDD    F2, F0, F2              ;; F2 = F2 + F0，累加结果到F2中（F2初始为0）
11                                               ;; <- 存在结构冒险（stall），因为F0刚被修改
12               ADDI    R2, R2, #8              ;; R2加8，指向A数组下一个double元素
13               ADDI    R3, R3, #8              ;; R3加8，指向B数组下一个double元素
14               SUB     R5, R4, R2              ;; R5 = R4 - R2，用于判断是否到达数组末尾
15               BNEZ    R5, loop                ;; 如果R5 ≠ 0，说明未遍历完，继续循环
16               TRAP    #0                      ;; 系统调用：程序结束
17           A:  .double 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
18           B:  .double 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
```

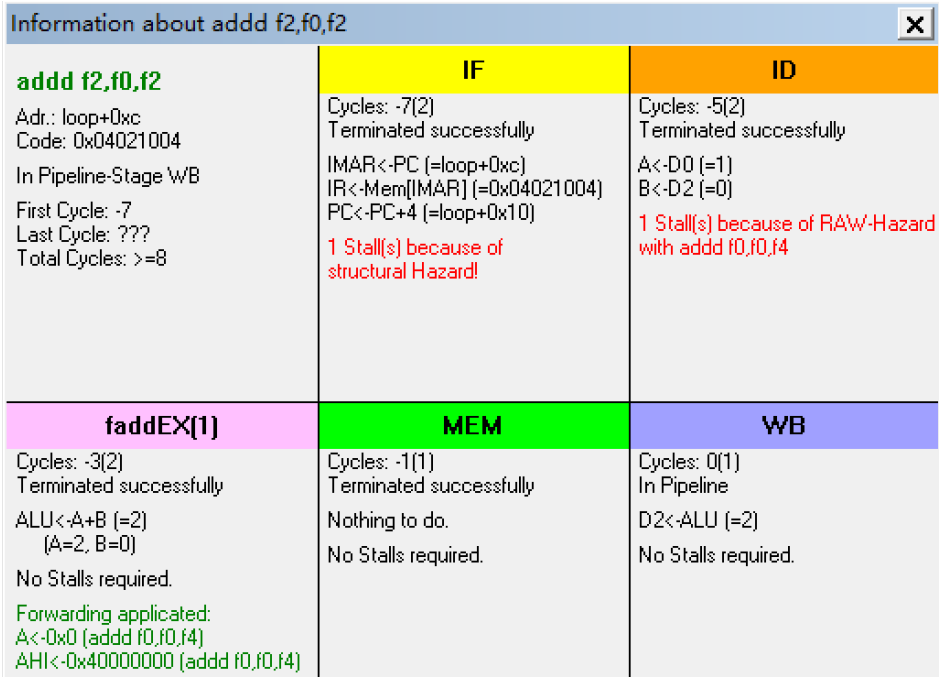
本程序是将两个浮点数数组对应位置逐个相加。

（2） F7 单步执行，观察结构相关：

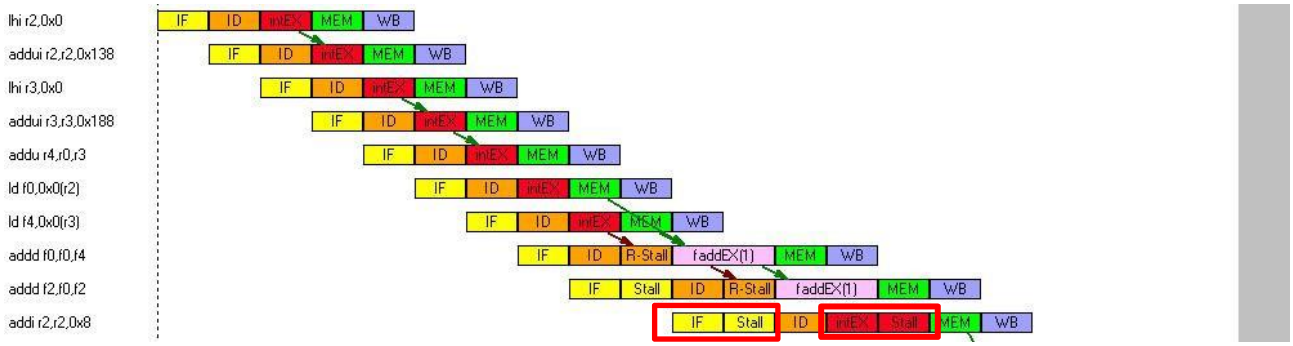


遇到第一个结构相关，此时对应的指令为 addd f2，f0，f2。此处的结构相关是由于之前的数据相关所导致的，具体原因是在上一条指令中 f0 的计算依赖于 f4，而 f4 在上一条指令执行阶段尚未计算产生，所以不能正常进入执行阶段（需要等待上一条指令访存阶段以后才能进入执行阶段），因此上一条指令会停留在译码阶段，而不能正常进入执行阶段。而此时对于执行 addd f2，f0，f2 来说，由于译码部件被上一条指令占用，导致本条指令产生译码器的结构相关，将该指令阻塞在取值阶段。

双击该行，可从弹出的窗口看见下图：

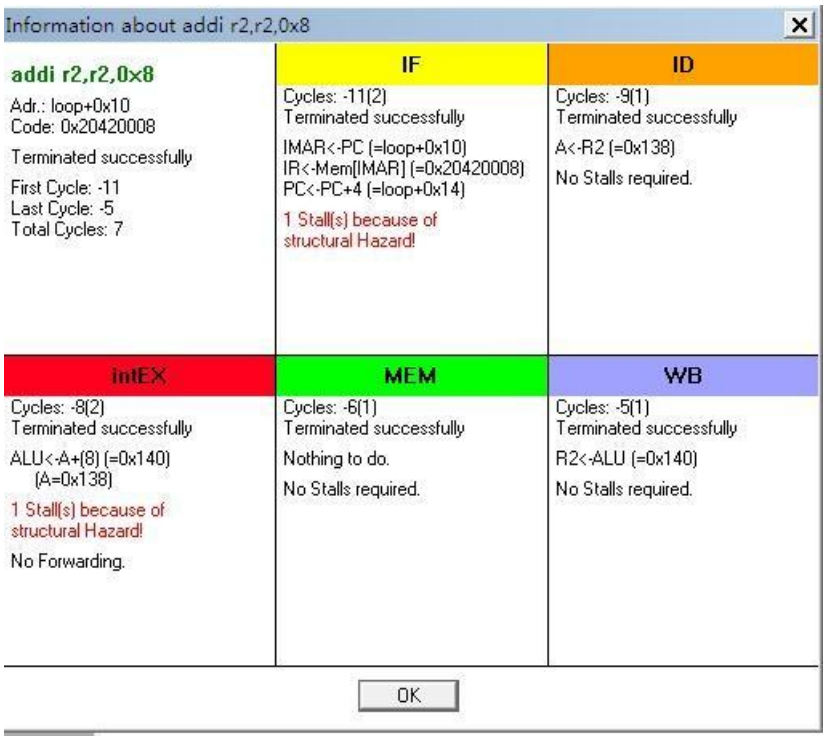


（3） 继续运行，该条指令存在两处结构相关，如下图所示

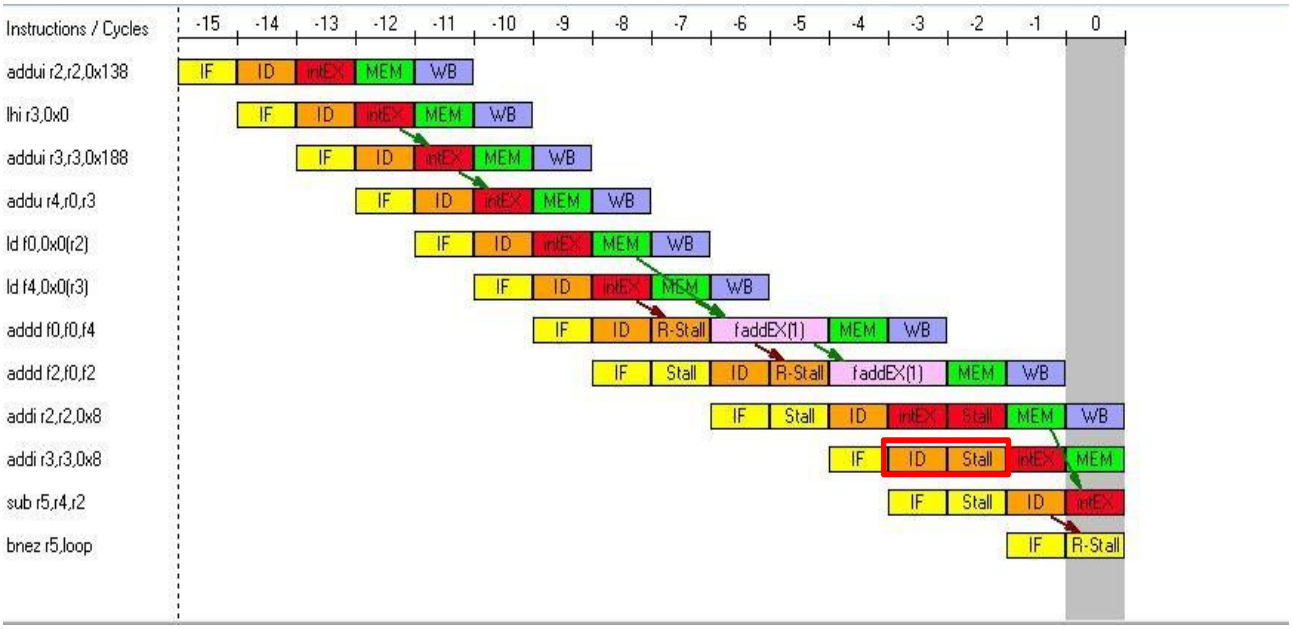


其中译码器处的结构相关产生原因与第一个结构相关类似，都是因为紧邻的上一条指令无法正常进入执行周期，从而延长占用译码器的时间，导致本条指令无法译码，从而产生结构相关。

而执行部件的结构相关是因为指令 addf f2，f0，f2 为双精度浮点数加法指令，而浮点数加法指令的执行周期占用 2 个时钟周期，导致后面的指令必须等待一个时钟周期，即插入一个 R-stall。同样地，双击指令，如下图所示：

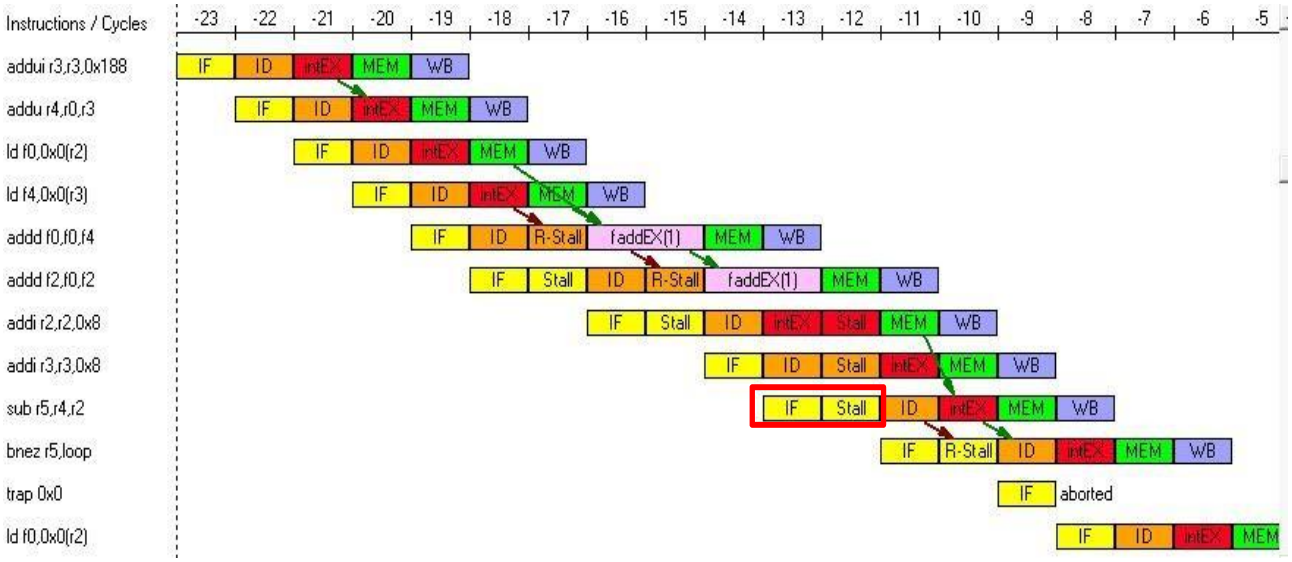


（4）单步追踪，执行到 addi r3，r3，0x8 指令。

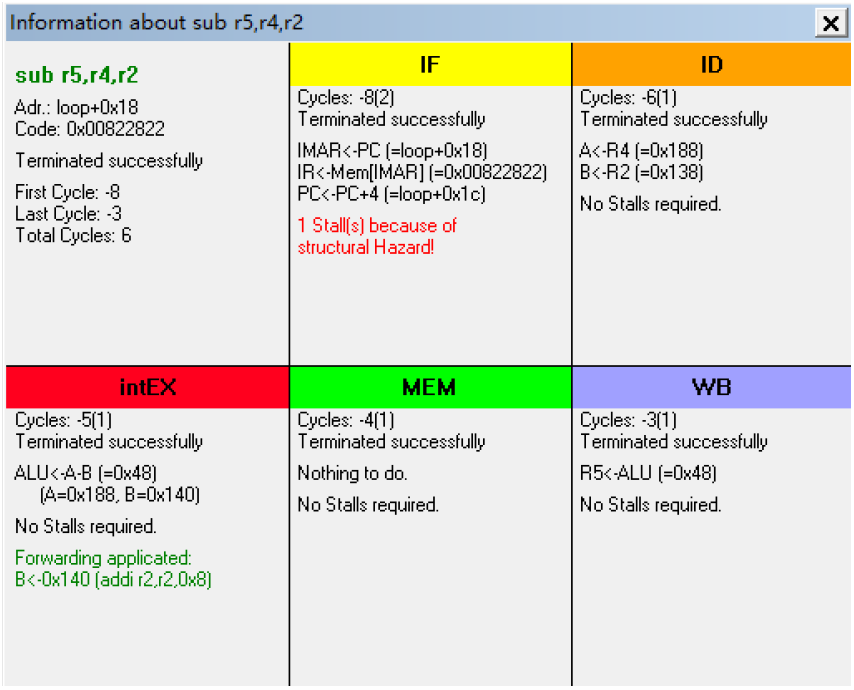


该指令也存在结构相关，这是因为上一条指令无法正常进入执行阶段，从而导致上一条指令占用执行器时间延长，使得该指令无法正常进入执行阶段。

（5）继续执行，直到执行到 sub r5，r4，r2。此时 Clock Cycle Diagram 窗口如下所示：



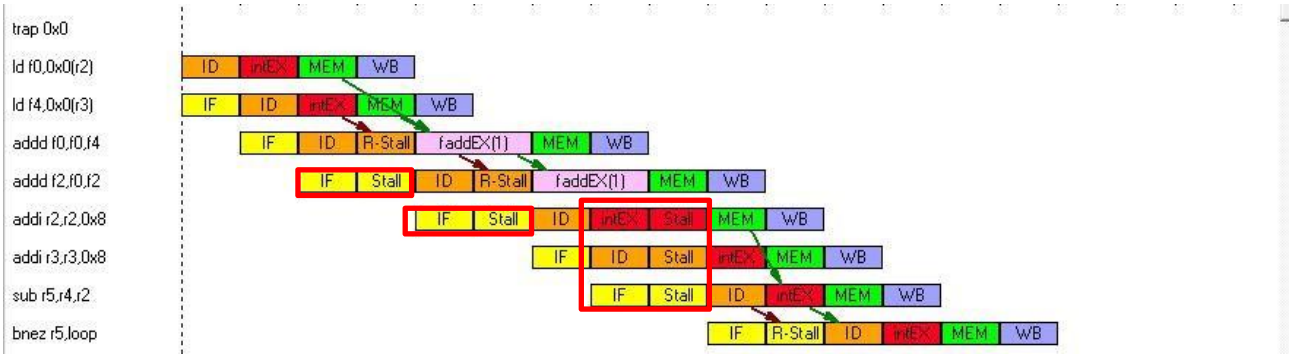
可以看到由于上一条指令被阻塞在译码阶段，因此，当取出指令 sub r5，r4，r2 之后无法译码，因而产生译码器结构相关。此时，双击指令，有如下的提示：



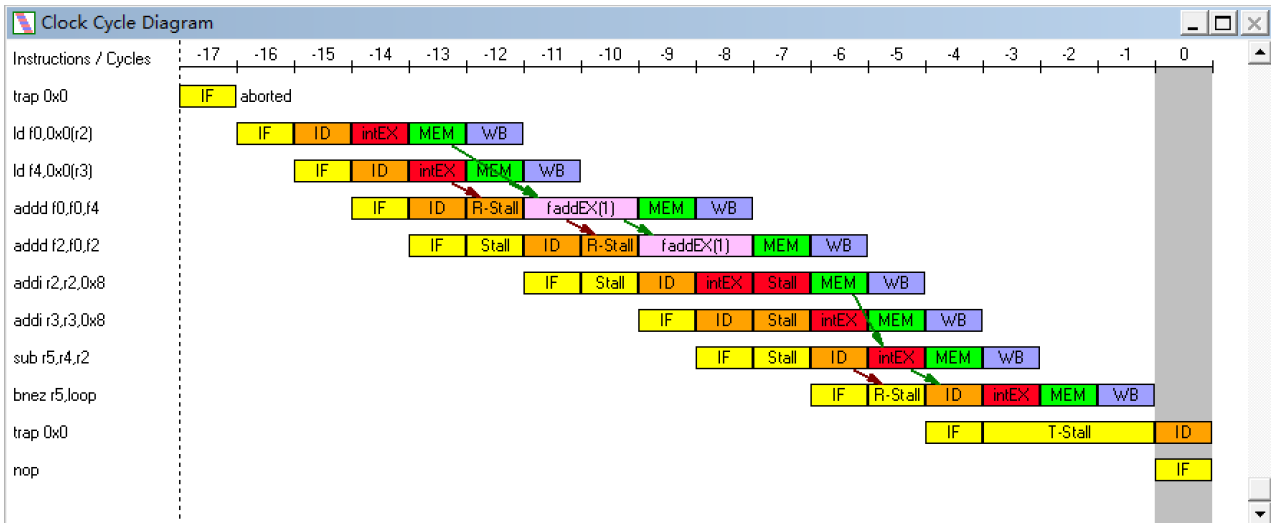
（6） 重复上述过程：分析汇编代码可知，程序会不断重复上述循环，直到次数达到 10 次后退出。而每一次过程中，如下指令中会出现结构相关：

结构相关的指令	导致结构相关的部件
addd f2，f0，f2	faddEX[1]
addi r2，r2，0x8	ID、faddEX[1]
addi r3，r3，0x8	intEX
sub r5，r4，r2	ID

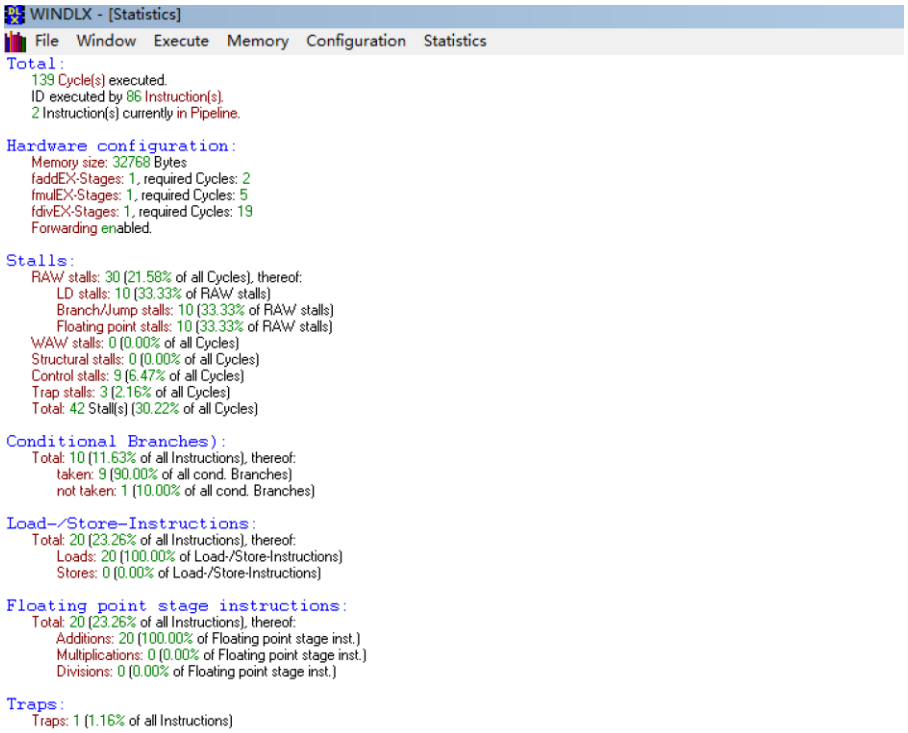
由于每一次循环都会产生五次结构相关，且后三次结构相关都在同一周期内发生，如下图所示：



（7） 按下 F5 执行至程序结束，此时各个执行部件的指令流水如下所示：



查看 Statistics 窗口内的统计信息，如下图所示：



所以 10 次循环一共 30 个 Stalls，而总的时钟周期数为 139，占比为21.58%。

结构相关会导致流水线的执行效率下降，但这不代表我们要想办法消除所有结构相关，这是因为可能通过增加硬件资源的成本大于流水线效率下降带来的损失。

发生结构相关时，必然会导致流水线效率的降低，如果处理不当甚至会发生错误。解决的方法主要包括：

(1) 增加气泡

(2)增加硬件资源，如设置独立的指令存储器和数据存储器

(3)流水化功能单位

(4)暂停流水线

结论分析与体会：

结论分析：

关于流水线相关的分类

分析：一般来说，流水线中的相关主要分为如下三种类型：

(1) 结构相关：当硬件资源满足不了指令重叠执行的要求，而发生资源冲突时，就发生了结构相关。
(2) 数据相关：当一条指令需要用到前面指令的执行结果，而这些指令均在流水线中重叠执行时，就可能引起数据相关。
(3) 控制相关：当流水线遇到分支指令和其它能够改变 PC 值的指令时，就会发生控制相关。
一旦流水线中出现相关，必然会给指令在流水线中的顺利执行带来许多问题，如果不能很好地解决相关问题，轻则影响流水线的性能，重则导致错误的执行结果。消除相关的基本方法是让流水线暂停执行某些指令，而继续执行其它一些指令。

体会：
本次实验主要涉及流水线中结构相关的基本概念，需要在跟踪指令的过程中观察各个部件的执行情况。当多条指令在流水线中重叠执行并争用同一硬件资源时，就会发生资源冲突，这种现象称为结构相关。为了确保所有指令组合在流水线中顺利执行且不产生结构相关，通常需要引入流水化的功能单元或资源复用机制。本实验还要求分析因结构相关而导致的流水线停顿周期数。由于一个周期内可能发生多个结构相关事件，因此结构相关的次数不一定与停顿的周期数相等。通过本次实验，不仅加深了对结构相关产生原因的理解，也认识到它在流水线中可能引发的一系列连锁问题。