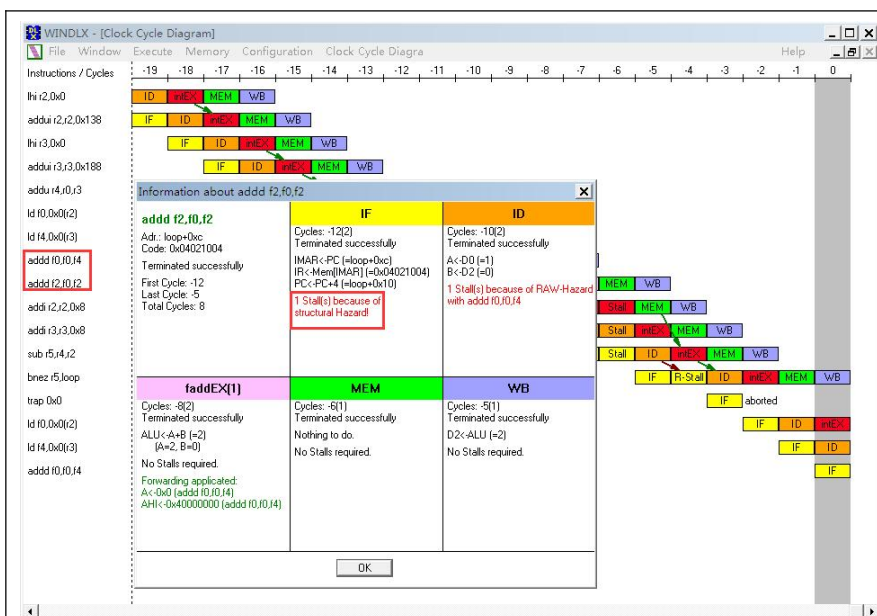
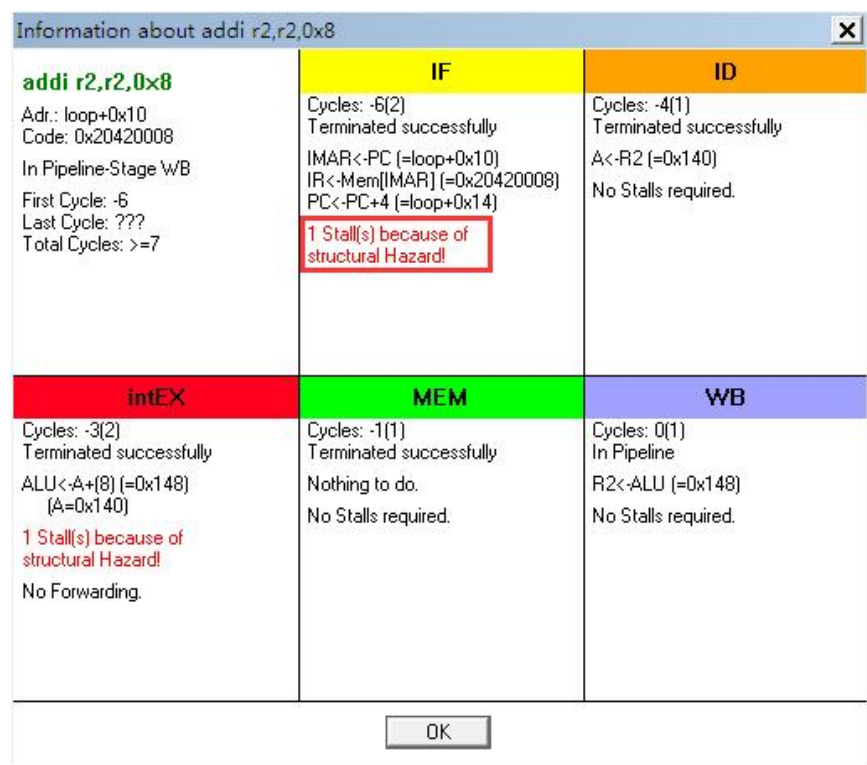


学号：202000130143	姓名： 郑凯饶	班级： 计科 1 班
实验题目：实验四：结构相关		
实验学时：2	实验日期：5.6	
<p>实验目的：</p> <p>通过本实验，加深对结构相关的理解，了解结构相关对 CPU 性能的影响。</p>		
<p>硬件环境：</p> <p>Dell Latitude 5411</p> <p>Intel (R) Core (TM) i5-10400H CPU @ 2.60GHz (8GPUs), ~2.6GHz</p>		
<p>软件环境：</p> <p>VMware Workstation 16 Player</p> <p>Windows 7</p>		
<p>实验步骤与内容：</p> <p>1. 阅读汇编代码，程序意图：将两个浮点数数组对应位置逐个相加。</p> <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <pre> LHI    R2, (A&gt;&gt;16)&amp;0xFFFF ; 将 16 位立即数加载到寄存器的高 16 位 ADDUI  R2, R2, A&amp;0xFFFF LHI    R3, (B&gt;&gt;16)&amp;0xFFFF ADDUI  R3, R3, B&amp;0xFFFF ; 可以理解为将一个 32 位的地址数加载到寄存器中 ADDU   R4, R0, R3  loop: LD      F0, 0(R2) LD      F4, 0(R3) ADDD    F0, F0, F4 ADDD    F2, F0, F2 ;; &lt;- A stall is found (an example of how to answer your questions) ADDI    R2, R2, #8 ADDI    R3, R3, #8 SUB     R5, R4, R2 BNEZ    R5, loop ; 将浮点数数组对应位置逐个相加 TRAP    #0      ;; Exit &lt;- this is a comment !!  A:      .double 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 B:      .double 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 </pre> </div>		
<p>2. F7 单步执行，发现结构相关：</p>		



由于浮点数加法指令的执行阶段占用 2 个时间周期，导致后面的指令必须等待一个周期，插入一个 R-stall。在指令详情界面可以看到在指令 `add f2, f0, f2` 的 IF 取指阶段侦测到了结构相关的风险，在 ID 阶段停留了 1 个 Stall。两条指令同样有数据相关，因为前者计算出 `f0`，后者使用 `f0`，因此通过定向技术，将结果从执行器件定向至执行器件。



这是之前浮点数加法指令发生结构相关导致的连锁反应，前一条指令停在了 ID 部件，这条指令只能在 IF 部件等待。后面因为 MEM 结构相关，导致该指令在 intEx 部件停留了一个

Stall。然后后面对 r3 操作的指令又不得不在 ID 部件等待。

3. 找出所有结构相关的指令以及导致结构相关的部件：

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, MEM
Addi r3, r3, 0x8	IntEX
Sub r5, r4, r2	ID

4. 进入统计视图，查看由结构相关引起的暂停时钟周期数，每次循环 addd f2, f0, f2 和前面一条指令产生数据相关导致 1 个 Stall，以及两条浮点数加法指令导致各一个 Stall，因此总共 10 次循环，总共 30 个 Stalls，而由结构相关引起的暂停时钟周期数为 20。

暂停时钟周期数占总执行周期数的百分比： $20/139=14.4\%$

```
Total:
  139 Cycle(s) executed.
  ID executed by 86 Instruction(s).
  2 Instruction(s) currently in Pipeline.

Hardware configuration:
  Memory size: 32768 Bytes
  faddEX-Stages: 1, required Cycles: 2
  fmulEX-Stages: 1, required Cycles: 5
  fdivEX-Stages: 1, required Cycles: 19
  Forwarding enabled.

Stalls:
  RAW stalls: 30 (21.58% of all Cycles), thereof:
    LD stalls: 10 (33.33% of RAW stalls)
    Branch/Jump stalls: 10 (33.33% of RAW stalls)
    Floating point stalls: 10 (33.33% of RAW stalls)
  WAW stalls: 0 (0.00% of all Cycles)
  Structural stalls: 0 (0.00% of all Cycles)
  Control stalls: 9 (6.47% of all Cycles)
  Trap stalls: 3 (2.16% of all Cycles)
  Total: 42 Stall(s) (30.22% of all Cycles)

Conditional Branches):
  Total: 10 (11.63% of all Instructions), thereof:
    taken: 9 (90.00% of all cond. Branches)
    not taken: 1 (10.00% of all cond. Branches)

Load-/Store-Instructions:
  Total: 20 (23.26% of all Instructions), thereof:
    Loads: 20 (100.00% of Load-/Store-Instructions)
    Stores: 0 (0.00% of Load-/Store-Instructions)

Floating point stage instructions:
  Total: 20 (23.26% of all Instructions), thereof:
    Additions: 20 (100.00% of Floating point stage inst.)
    Multiplications: 0 (0.00% of Floating point stage inst.)
    Divisions: 0 (0.00% of Floating point stage inst.)

Traps:
  Traps: 1 (1.16% of all Instructions)
```

5. 结构相关会导致流水线执行效率下降，但我们允许一定的结构相关存在，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。

解决结构相关的方法：①添加气泡 Stalls②增加硬件，如设置独立的指令存储器和数据存储器。

结论分析与体会：

本次实验详细研究了程序的结构相关，了解结构相关发生的场景以及原因，并进一步思考了结构相关的解决方法。更多地，体会到冲突在流水线中导致的一系列连锁反应。