

上机试验 2：指令动态调度性能分析

1、实验目的：

- (1) 加深对指令级并行性及其开发的理解；
- (2) 加深对 Tomasulo 算法的理解；
- (3) 掌握 Tomasulo 算法在指令流出、执行、写结果各阶段对浮点操作指令以及 load 和 store 指令进行什么处理；
- (4) 掌握采用了 Tomasulo 算法的浮点处理部件的结构；
- (5) 掌握保留站的结构；
- (6) 给定被执行代码片段，对于具体某个时钟周期，能够写出保留站、指令状态表以及浮点寄存器状态表内容的变化情况。

2、实验平台：

- (1) 自己设计一个 Tomasulo 算法模拟器（简称：模拟器 A）；
- (2) 使用开源 Tomasulo 算法模拟器（简称：模拟器 B）；
- (3) 使用教材附带的 Tomasulo 算法模拟器（简称：模拟器 C）；

3、实验内容和步骤：

- (1) 针对模拟器 A 和模拟器 B，应该至少实现以下功能：
 - A. 能够模拟 Tomasulo 算法的执行过程，以课堂讲授为参考；
 - B. 支持图形交互或者命令交互；
 - C. 支持单步执行（一次一个时钟周期）、一次多个周期，执行到程序结束；
 - D. 支持保留站、指令状态表、寄存器状态表等内容的查看（支持对已执行过的时钟周期的上述内容回看）；
 - E. 提供程序执行后的性能统计分析；
 - F. 按照 MIPS 语法，至少支持 load、store、add.d、sub.d、mul.d、div.d 操作；
 - G. 要执行的程序可以直接输入，或通过文件载入的方式运行；
 - H. 可以使用你喜欢的任何语言来实现；
 - I. 可参考模拟器 C 的功能；
- (2) 使用在 (1) 中 F 提到的指令，设计至少 3 种不同的代码组合，实现以下功能的演示：
 - A. 没有任何冲突的指令调度场景；
 - B. 有至少一次的 RAW 冲突；
 - C. 有至少一次的 WAR 冲突；
 - D. 使用课堂讲授的内容，对上述每一种场景进行分析

4、实验结果提交：

- (1) 模拟器 A 和模拟器 B 需提交完整的源代码、编译过程；
- (2) 完整的实验报告至少包含以下内容：
 - A. 模拟器 A、B 的设计思想、特色；
 - B. 模拟器的测试代码组合；
 - C. 测试代码在模拟器的执行过程，以图文并茂的方式呈现；
 - D. 实验感悟

5、实验成绩评定：

- A. 本次实验按 100 分计算；
- B. 模拟器 A 最高分为 100 分；
- C. 模拟器 B 最高分为 90 分；
- D. 模拟器 C 最高分为 80 分