

计算机体系结构

实验二 实现 Tomasulo 算法模拟器

昂伟 PB11011058

1 设计思想与实验分析

前端

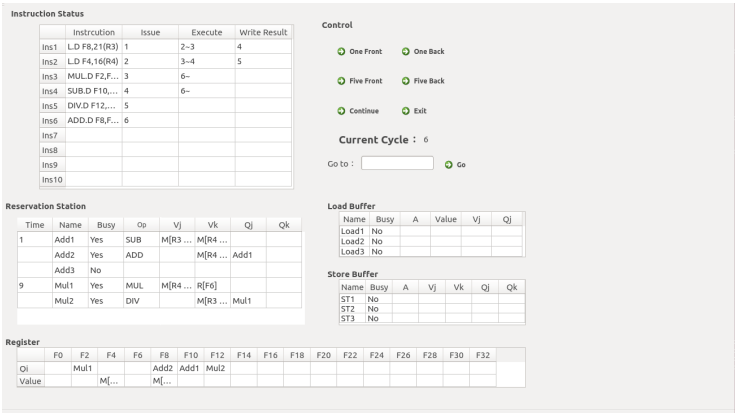
mainwindow.h, *ui_mainwindow.h*, *mainwindow.cpp* 代码主要实现 *Tomasulo* 算法前端图形化界面. 前端主要与用户交互, 包括指令队列的读取, *Tomasulo* 算法配置信息的设置, 并将 *Tomasulo* 算法后端运行的结果返回显示出来.

参数设置

The screenshot displays the graphical user interface of the Tomasulo algorithm simulator. It includes several key components:

- Instruction Queue:** A table with columns for instruction type (LD, LD, MUL.D, SUB.D, DIV.D, ADD.D, NOP) and their respective operands (F8, F4, F2, F10, F12, F8, F10, F4, Null).
- Setting:** A table for configuring operation cycles: Add (2), Sub (2), Mul (10), Div (40), Load (2), and Store (2). It also features 'Reset' and 'Execute' buttons.
- Reservation Station:** A table with columns for Time, Name, Busy, Op, Vj, Vk, Qj, and Qk, showing the status of reservation stations (Add1, Add2, Add3, Mul1, Mul2).
- Load Buffer:** A table with columns for Name, Busy, A, Value, Vj, and Qj, showing the status of load buffers (Load1, Load2, Load3).
- Store Buffer:** A table with columns for Name, Busy, A, Vj, Vk, Qj, and Qk, showing the status of store buffers (ST1, ST2, ST3).
- Register:** A table with columns for Register (F0-F32) and Value, showing the current state of the processor registers.

执行指令

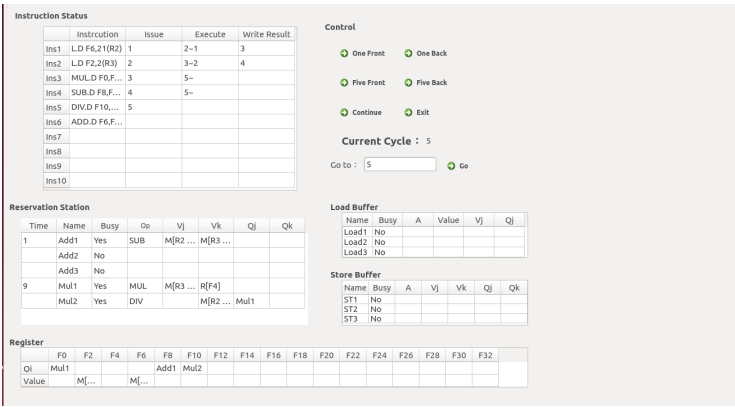


后端

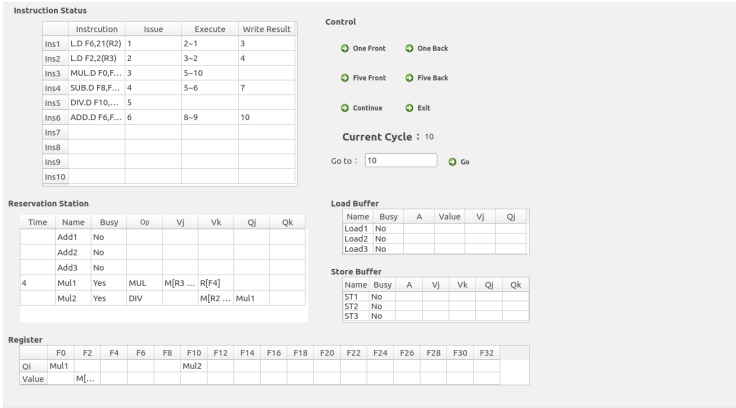
`cdb.h`, `virtualregister.h`, `tomasulo.h` `tomasulo.cpp` 代码组成 *Tomasulo* 算法的后端. 后端主要负责 *Tomasulo* 算法的具体实现. 其中 `virtualregister.h` 中包含了所有实现 *Tomasulo* 算法所需要的数据结构, 如 *reservation station*, *register status*, *load buffer* 等.`tomasulo.cpp` 是 *Tomasulo* 算法的具体实现细节, 包括指令的发射 (`issue()`), 执行 (`execute()`), 结果的写回 (`writeResult()`) 等函数的实现. 另外 `cdb.h` 中实现了 *common data bus* 结构, 可以控制 *Tomasulo* 算法中一个时钟周期内写结果的个数.

2 问题解答

周期五保留站状态



周期十保留站等相关器件状态



3 总结

通过实现 *Tomasulo* 算法, 对该算法的具体细节有了更深入的掌握, 同时对其提高指令级并行的原理和效果有了直观的了解和感受, 对系统内部指令执行的具体细节有了更深入的理解和掌握. 感受到提高指令并行能力对机器性能的影响.