设计实现 Tomasulo 调度算法

组员：计 02 宋文杰
计 02 孙伟伦
计 02 李凯威

【 文档目录 】

【 实验要求 】

设计实现 Tomasulo 算法，显示指令执行过程中的数据流向以及寄存器值变化情况，指示出各时钟周期各浮点部件的工作状态。

【 开发环境 】

程序在 Window 8 64bit 下使用 Visual Studio 2012 编写调试完成。

框架：.Net 4.0

编程语言：C#

GUI 绘制：Windows Presentation Foundation 4.0

【 运行环境 】

运行环境	最低配置	推荐配置
CPU	Intel Xeon E1230-V2 3.3Ghz 或 AMD 同性能处理器	Intel Haswell I7 4770K 3.5Ghz 或 AMD 同性能处理器
内存	24g	32g
显卡	NVIDIA GTX 660Ti (2048MB) 或 ATI 同性能显卡	NVIDIA GTX Titan (6144MB) 或 ATI 同性能显卡
操作系统	Windows 7 64 位/Win8 64 位	

【 实验原理 】

Tomasulo 算法：

Tomasulo 算法又称公共数据总线法（或令牌法），采用乱序流动方式执行指令，来提高流水线的吞吐率和效率，并通过分散控制的办法处理数据相关。

IBM360/91 处理机实现基本原理：

IBM360/91 处理机的浮点处理部件中，有一个浮点加法器和一个浮点乘/除法器。加法器为两段流水线，输入端有三个保存站 A1、A2、A3，乘/除法器为六段流水线，输入端有两个保存站 M1，M2。保存站采用随机方式工作，由保存站中的控制部件

控制。当任意一个保存站中的两个源操作数到齐后，如果对应的操作部件空闲，可以把两个操作数立即送到浮点操作部件中执行。

IBM360/91 处理机的浮点处理部件采用先行控制方式。浮点先行操作站中存放的是经过指令分析部件预处理之后的"寄存器－寄存器"型指令，这类指令中的源操作数可能来自浮点通用计数器，也可能来自浮点先行读数站，运算结果送到公共数据总线，送入浮点通用寄存器、浮点加法器的保存站或浮点乘/除法的保存站等;最终运算结果一般送到浮点后行写数站，由浮点后行写数站负责写到主存储器中。

本实验实现的模拟机实现原理：

在我们所实现的浮点处理模拟机，浮点处理部件有三个浮点加法器和两个浮点乘/除法器。(这两种处理部件的个数可以由用户需求设置)。

首先，先往指令队列输入指令（模拟先行指令缓冲站），然后指令队列察看是否有空闲的功能单元可用。如果有，发射，否则，等待直到有需要的执行功能单元空闲。

运行状态纪录每条指令的执行状态，记录三种状态：发射周期，浮点运算完周期和写回结果周期。

指令发射时，如果时 Load 指令，则送到 Load 缓冲栈；如果时 Store,，则送到 Load 缓冲栈；其他浮点运算送到相应的保存站 RS。当等待的数据有效时，各执行单元的计数器启动，每个一个时钟周期 Time 减 1。当某个部件的 Time 减为 0 时，表示执行完毕，则置执行完毕标志为当前的时钟，同时进入写会结果阶段。

在结果写回阶段，功能部件把数据广播到总线上（用 BroadCast 方法模拟)。每个功能部件都同时检查总线是否等待的数据出现，如果检查到想要的结果，计 Vj or Vk 的数据为 AVAILABLE 状态。当 Vj 和 Vk 同时为 AVAILABLE(或 STORE 的 Qi 为 AVAILABLE),该功能部件就可以开始执行。

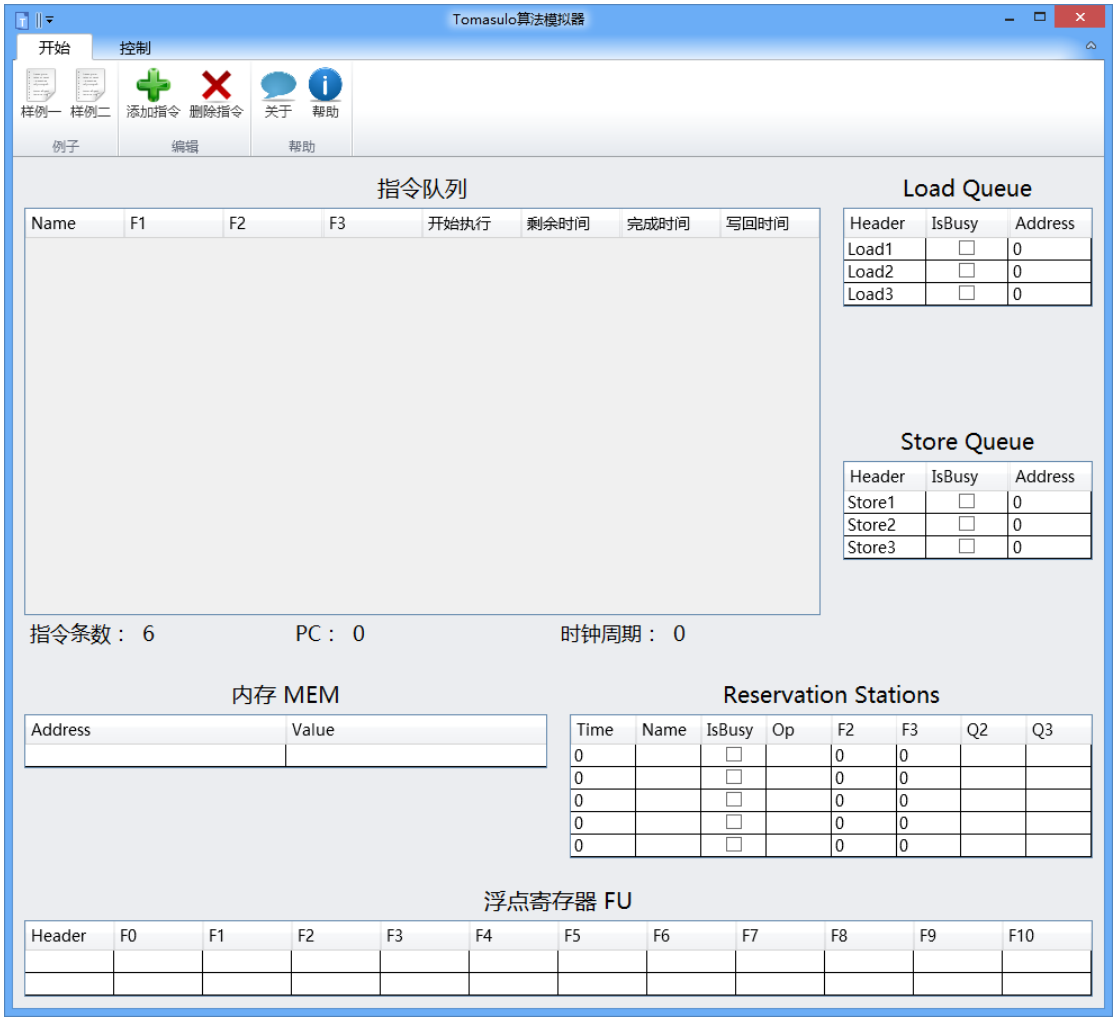
在该模拟机上,内存采用数组模拟,有两组寄存器,浮点寄存器和整型寄存器,具体运行功能如下详述!

【 程序功能 】

- 可支持任意输入指令
- 自动演示功能，可动态调整时钟周期大小
- 有带值计算和表达式计算两种运行模式
- 支持对内存，寄存器的赋值操作
- 能指示出时钟周期各浮点部件的工作状态
- 能显示指令执行过程的数据流
- 界面十分友好，易操作性强
- 良好程序数据结构，扩展性强

【 程序运行说明 】

程序运行在 Windows 操作系统下，运行后显示如下主窗口：



各部分的内容在图上一目了然。分别表示了指令，状态，内存以及寄存器等得内容，下面根据该模拟程序的具体操作来说明各个部分的运行方式。

首先是寄存器和内存的赋值

直接编辑内存表格内容即可，编辑内存地址，自动显示对应的值，编辑值，自动修改对应的内存，用过 Visual Studio 的话，就和 VS 调试的时候 Watch 窗口一样。

然后是程序指令的输入

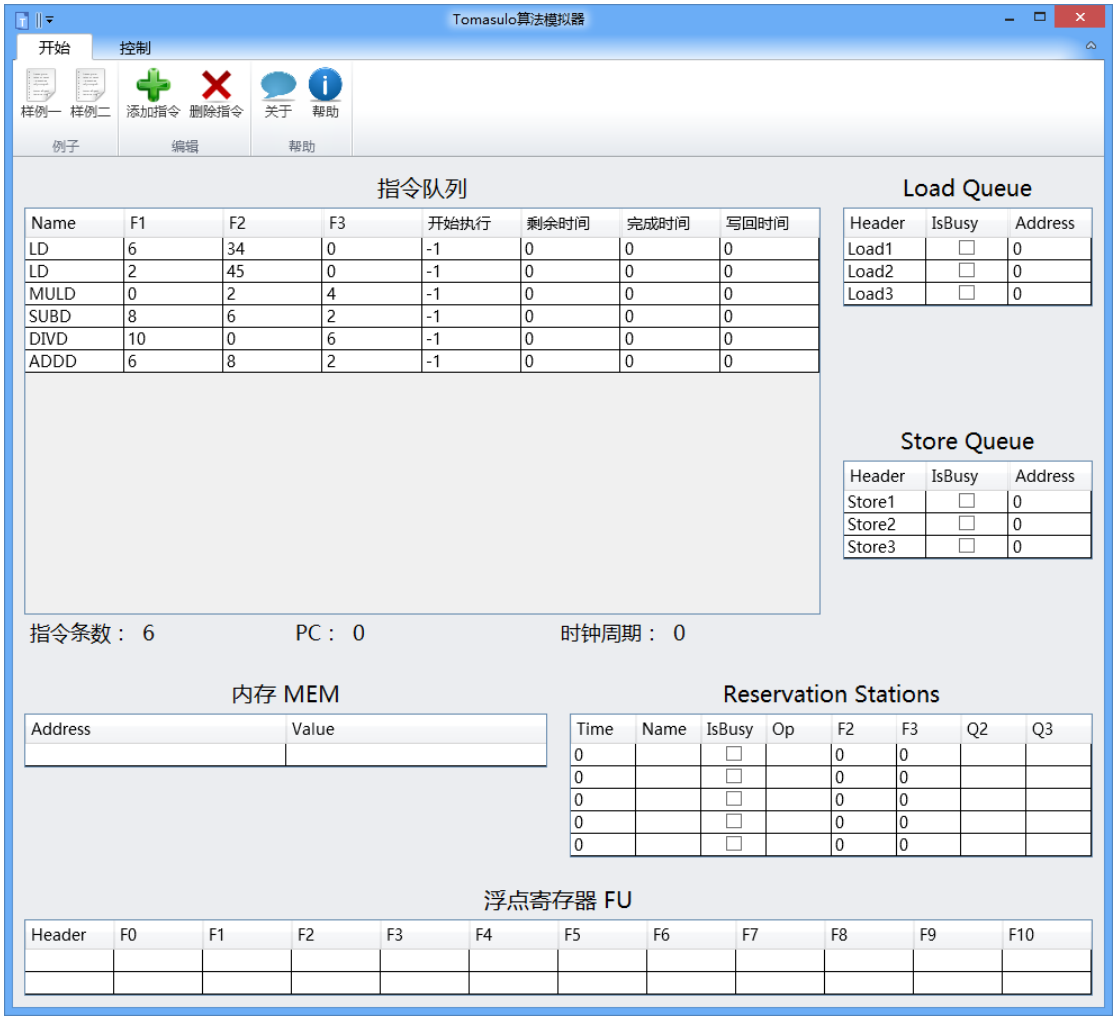
这个是添加指令

这个是删除指令

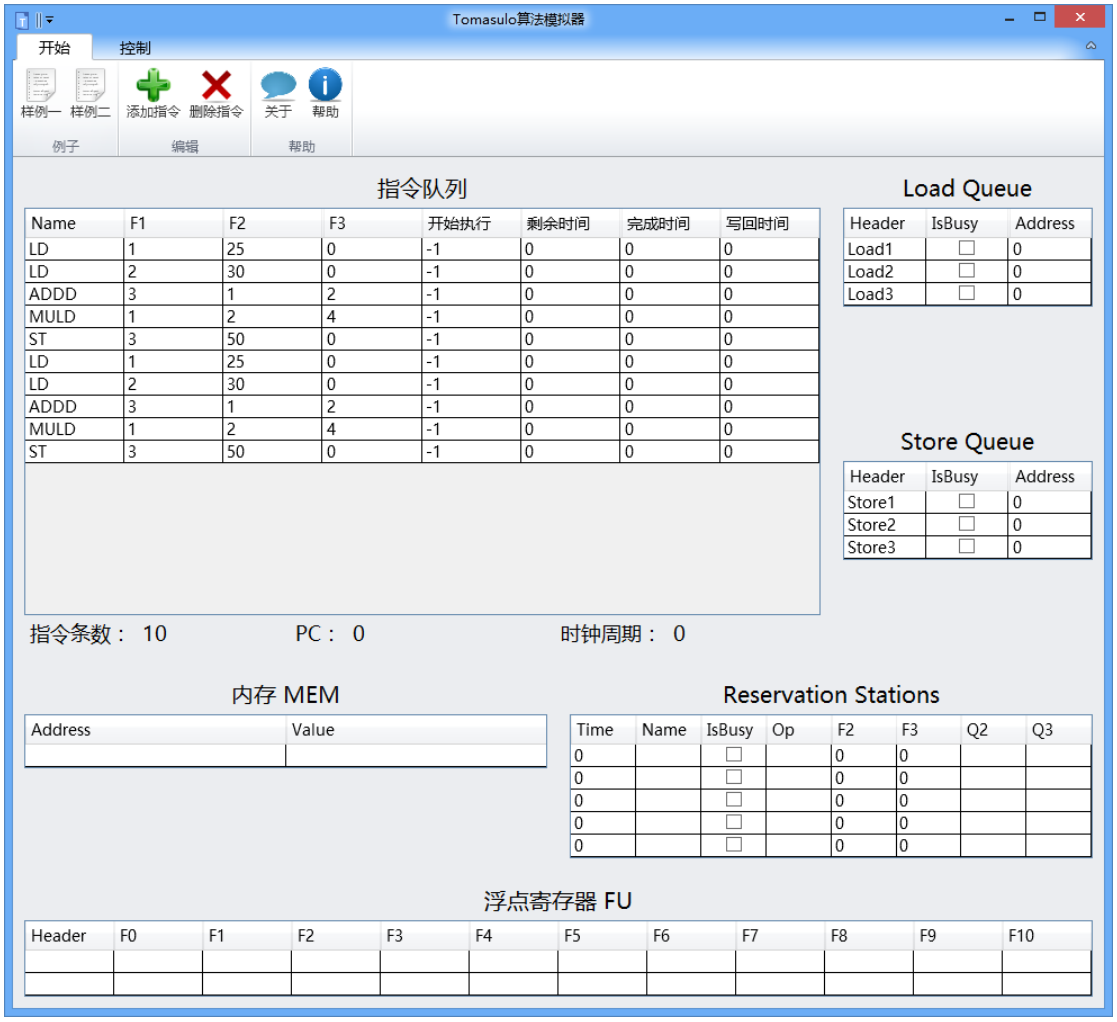
然后直接编辑指令表格即可。

有两个样例，分别单击样例 1 和样例 2:

这是样例一：



这是样例二：



接下来便是指令的运行了

程序运行有两种方法,一种是自动演示,另一种是手动操作,下面分别说明.

自动演示功能:

单击 Ribbon 工具栏的“控制”按钮

单击自动执行，就进入自动执行模式。

正在自动运行时，可以单击暂停，转为手动控制：

手动操作功能:

（这部分比较适合一步步演示和进行讲解！）

单击 Ribbon 菜单中的“单步执行”

之后，每次单击下一步都会进入下一个时钟周期

停止功能:

单击停止按钮，将会清楚当前程序状态，下次再按自动执行时，将会重新开始。

【 实验总结 】

本实验花了很多时间做，前后花了整整一个下午+一个晚上，但学到了很多东西，非常值得. 本程序功能已经比较完备，但有些地方还有待改善.

程序测试力度可能不够，还可能有 BUG 存在，可在日后不断修改. 总的来说，通过这个实验是我们更加深刻地了解 Tomasulo 算法和设计的基本原理，并且通过模拟实现，体会更加深入. 相对来说，这样学习效果要比看书好的多！

【 附 录 】

联系方式:
宋文杰
tel: 18667170650