**-**

**Tomasulo实验报告**

# 组员：

# 王龙涛 李昊阳 张鹿颂

## 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和寄存器值，内存值的变化情况，指示出各时钟周期各浮点部件的工作状态。

## 开发环境

IDE：Eclipse

开发语言：Java

运行环境：Windows10/Ubuntu

## 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。该算法的核心思想是记录和检测指令相关，操作数一旦就绪就立即执行，从而把RAW冲突的可能性减小到最小。另外，还通过寄存器换名解决了WAR和WAW冲突。

### 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。

首先，先往指令队列输入指令（模拟先行指令缓冲站），然后指令队列查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。

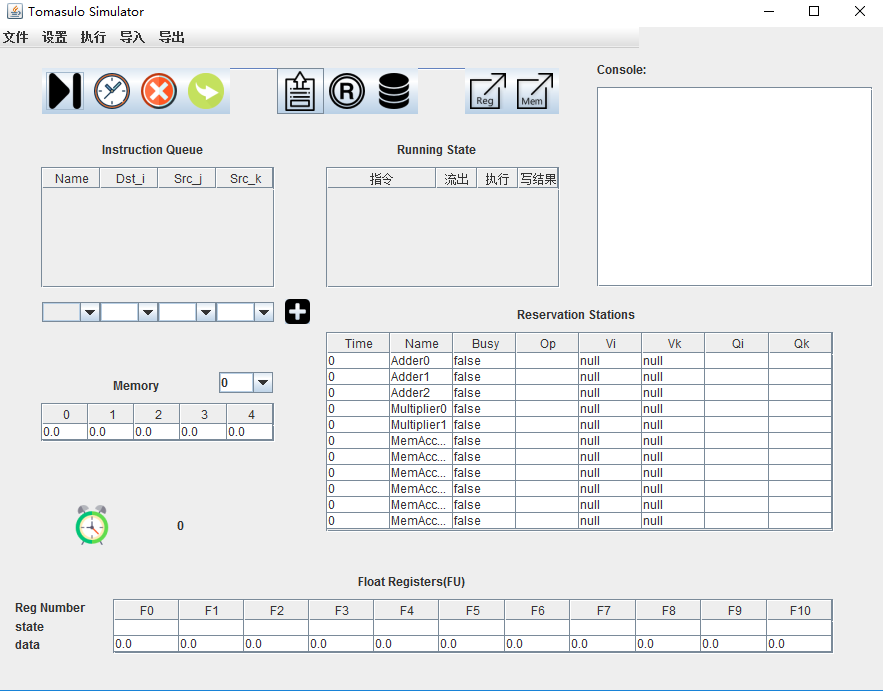
在结果写回阶段，功能部件把数据广播到总线上（用BroadCast方法模拟）。每个功能部件都同时检查总线是否等待的数据出现，如果检查到想要的结果，计Vj or Vk的数据为AVAILABLE状态。当Vj 和Vk同时为AVAILABLE(或STORE的Qi为AVAILABLE),该功能部件就可以开始执行。

## 实现细节

## 程序功能

* 寄存器、内存的文件导入导出功能
* 指令队列的文件导入功能
* 实时查看、修改寄存器、内存的值
* 实时显示保留栈的信息
* 实时显示指令队列，指令运行状态的信息
* 动态设置运行的时钟周期
* 单步运行
* 运行时终止
* 控制台显示调试信息

## 使用说明



模拟器的主体界面如上图所示，其中菜单栏包括：文件（导入指令）、设置（设置连续执行的周期数），执行（执行n个周期和单步执行），导入（导入寄存器和内存的值），导出（导出寄存器和内存的值）。

为了方便这些功能也可以通过点击下图所示的按钮来触发：



**（一）指令导入**

1. **指令的文件导入功能**

点击“文件”->“导入指令”，或者直接点击按钮，在弹出的对话框中选择保存有指令的文件即可将指令导入指令队列。其中，指令的格式为：

ADD F0，F1，F2

SUB F0，F1，F2

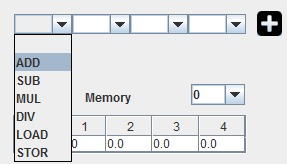
MUL F0，F1，F2

DIV F0，F1，F2

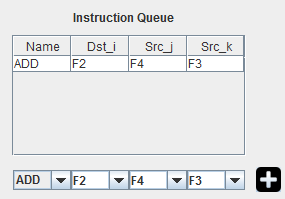
LOAD F0 1 //将地址为1的内存加载到F0寄存器中

STOR F0 1 //将F0写入地址为1的内存中

1. **指令的单条输入**



在指令队列下方有“单条指令添加”的按钮，首先在下拉框中选择或者直接设置操作码/操作数，然后点击右边的“加号”添加按钮，即可以把当前指令加入指令队列。



**（二） 指令执行**

**1. 执行n个周期**

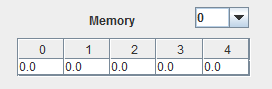
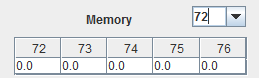
首先点击按钮，设置一次执行的周期数（如果不设置默认是100）。然后点击 开始执行。

1. **单步执行（执行1个周期）**

点击按钮，执行一个时钟周期，即单步运行。

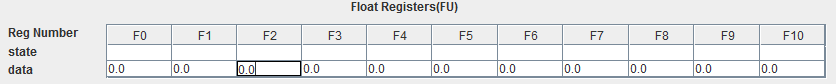
**（三）查看/修改寄存器和内存地址**

1. 内存空间一共有4096个，为了更好地查看内存地址，每次只显示连续5个地址空间。如果要查看地址为a的值，可以在右边的下拉框中选择a， 之后会显示地址为[a, a+5]的值。

如果要修改内存的值可以直接双击表格，输入修改后的值，按回车保存输入的值。

2. 寄存器的值显示在下方的表格中，如果要修改寄存器的值可以直接双击表格，然后输入修改后的值，回车确认。



**（四）导入/导出寄存器/内存的值**

**1. 导出寄存器/内存的值**

点击 按钮，在弹出的文件夹选择框中选择一个用来保存寄存器值的文件夹，点击“打开”后会将寄存器的值写入所选目录的reg.txt中。

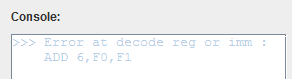
点击按钮可以实现导出内存值的功能，内存值保存在mem.txt中。

1. **导入寄存器/内存的值**

点击，在文件选择对话框中选择之前导出的保存有寄存器值的reg.txt文件，即可导入寄存器的值。点击，可实现导入内存值的功能。

**（五）控制台输出调试信息**

为了给使用者更好地反馈信息，我们提供了控制台功能，使用者可以通过控制台查看相关提示信息。比如，如果输入了一条错误的指令ADD 6，F0，F1，那么该指令并不会加入到指令队列中，而且控制台中会给出错误提示信息。当然，如果成功执行了某个功能，也会给出相应的提示信息。



## 结果展示

## 实验感想

**【附录】**

1. 指令格式问题：store指令的格式为STOR src dst，在前端界面的指令队列表格中，为了方便起见统一将dst列设定在了src列左边。
2. 运行过程如出现任何问题可以请联系我们：

王龙涛：[cunxinshuihua@gmail.com](mailto:cunxinshuihua@gmail.com)

李昊阳：[li-hy14@mails.tsinghua.edu.cn](mailto:li-hy14@mails.tsinghua.edu.cn)