|  |
| --- |
| 清华大学计算机科学与技术系 |
| MIPS-lite Simulator |
| 《计算机系统结构》课程实验报告 |
|  |
| kde9  kfirst 孔祥欣[[1]](#footnote-1)  deepsolo 胡玮玮[[2]](#footnote-2)  edwardtoday 卿培[[3]](#footnote-3) |
| **2009/6/2** |

目录

[实验目的 3](#_Toc231748521)

[实验内容 4](#_Toc231748522)

[实验环境 6](#_Toc231748523)

[实验环境准备 7](#_Toc231748524)

[实验流程及分工 8](#_Toc231748525)

[模拟器设计 9](#_Toc231748526)

[模拟器使用说明 10](#_Toc231748527)

[模拟器功能测试 11](#_Toc231748528)

[实验总结 15](#_Toc231748529)

# 实验目的

1. 深入了解指令系统以及指令流水线的原理和运作过程
2. 编程实现MIPS-lite模拟器，实现forwarding、stall等消除流水线冲突的方法

# 实验内容

根据系统结构实验的要求，使用java语言实现一个基于MIPS指令集的测试系统，系统的主要组成部分包括：

1. CPU模拟组件。CPU的主要要求如下：
   1. 指令系统

实现的CPU应基于MIPS指令系统，至少实现指定的40条指令，具体的指令编码和格式见“指令系统kde9.xls”。

* 1. 流水线

实现的CPU应当具有5段流水结构，即取指(IF)、译码(ID)、执行(EXE)、存储器访问(MEM)、写回(WB)五个阶段。

IF阶段是按照指令地址(PC)从存储器中取出指令码；ID是根据指令产生一系列控制信号，包括何种运算、运算值是否写回、操作数是什么等等，该阶段是整个CPU的最关键阶段；EXE阶段是ALU运算阶段，根据操作数和运算操作码对数据进行加、减移位、比大小等操作；MEM是从存储器中读取数据或者从向存储器中写数据的阶段；WB是根据写回信号将ALU运算结果或者从存储器中取的的数据写回寄存器的操作。

* 1. 数据旁路

实现的CPU应当实现数据旁路技术，以便提高流水线效率，减少流水线暂停。

1. Memory模拟组件。Memory的主要要求如下：
   1. 大小限制

要求至少具有1kB的存储空间。

* 1. 软件模拟运行效率和开销

该Memory实现应当具有较快的查找速度，并尽量减少系统资源的占用。

1. Cache模拟组件。Cache的主要要求如下：
   1. 大小限制

实现大小为512字节的指令Cache和同等大小的数据Cache，

* 1. 软件模拟运行效率和开销

Cache的软件模拟要求与Memory相同。

* 1. 关联模式与替换规则

该Cache的组大小为4字节，组间为全相连映射结构，Cache替换原则为最久未使用原则。

1. MIPS文本指令编译器。编译器的主要要求如下：
   1. 接收指定规则的文本输入，将其编译为机器可执行的二进制代码。
   2. 具有一定的错误提示功能。

# 实验环境

1. 编程语言：JAVA
2. 操作系统

MS Windows Vista Home Premium

MS Windows XP Professional SP2

Mac OS X Snow Leopardm, build 10A354

1. 编程环境：

JDK 1.5.0.0

MyEclipse 7.0

Eclipse 6.5

Eclipse Ganymade on Mac OS X

1. SVN

Google Code

TortoiseSVN 1.5

Subclipse 1.6 for Eclipse on Mac OS X

# 实验环境准备

我们首先在Google Code上建立了我们的项目，URL为：

<http://code.google.com/p/ca-simulator/>

然后进行本地IDE的配置，顺利进行了初始工程的建立。

# 实验流程及分工

|  |  |
| --- | --- |
| 日期 | 工作 |
| 5.27 | 了解实验内容，明确实验要求 |
| 5.28 | 实验总体规划及Google Code项目建立 |
| 5.29 | 设计实现Memory；设计CPU |
| 5.30 | 实现CPU |
| 5.31 | 设计实现模拟器UI及其他功能 |
| 6.1 | 调试模拟器功能及排错 |
| 6.2 | 撰写实验文档；实验总结 |

表 1 实验流程

|  |  |
| --- | --- |
| 姓名 | 分工 |
| 孔祥欣 | 流水线设计、CPU实现 |
| 胡玮玮 | 流水线设计、CPU设计、使用说明撰写 |
| 卿培 | 流水线设计、模拟器调试、实验报告撰写 |

表 2 实验分工

# 模拟器设计

参见MIPS-lite\_Simulator\_DesignDoc\_kde9

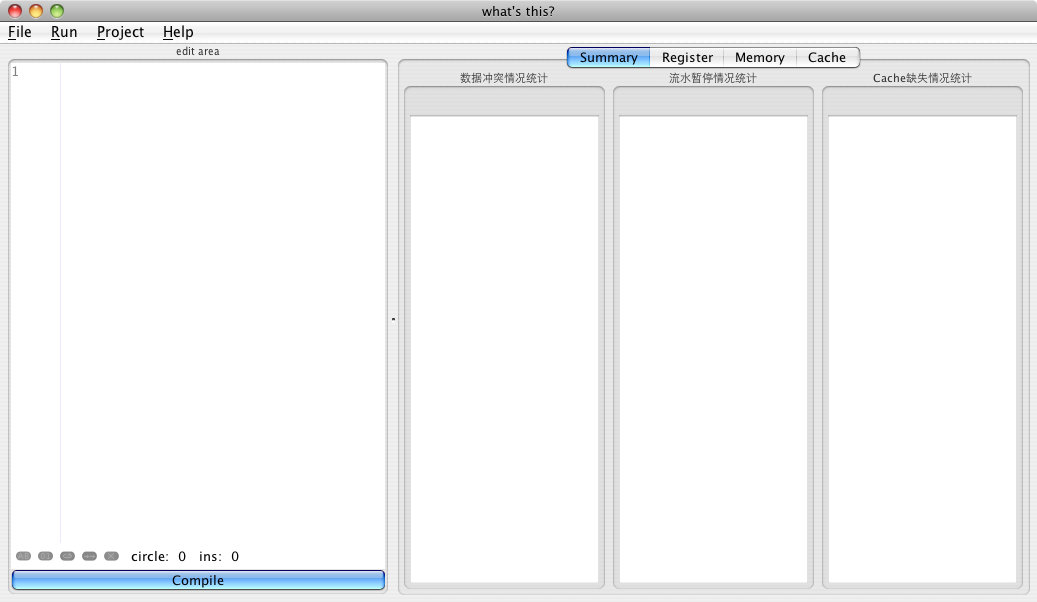
# 模拟器使用说明

参见MIPS-lite\_Simulator\_UserManual\_kde9

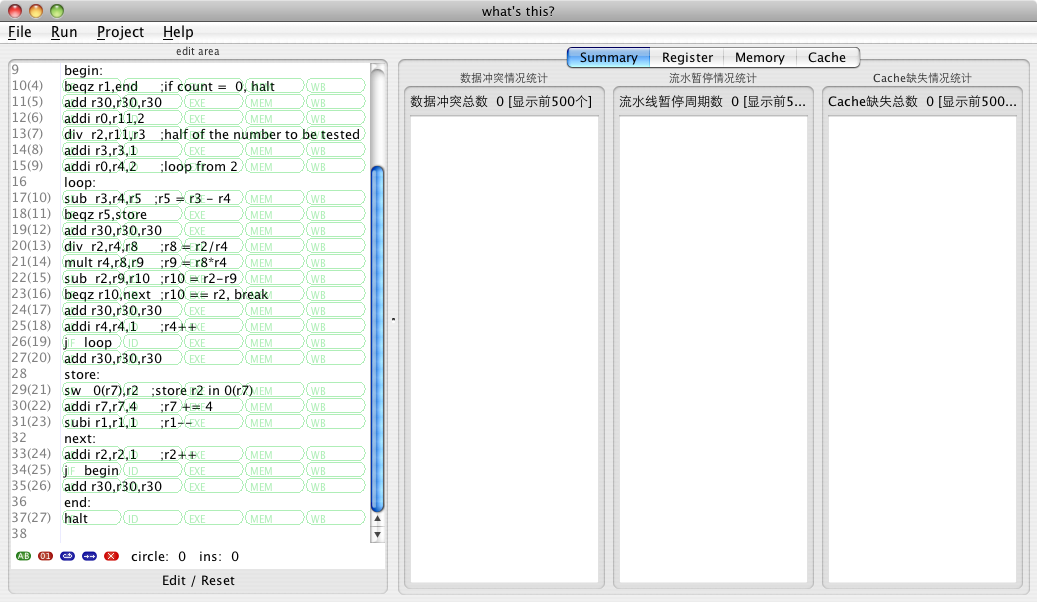
# 模拟器功能测试

我们用样例程序对模拟器进行了测试：

首先打开模拟器：

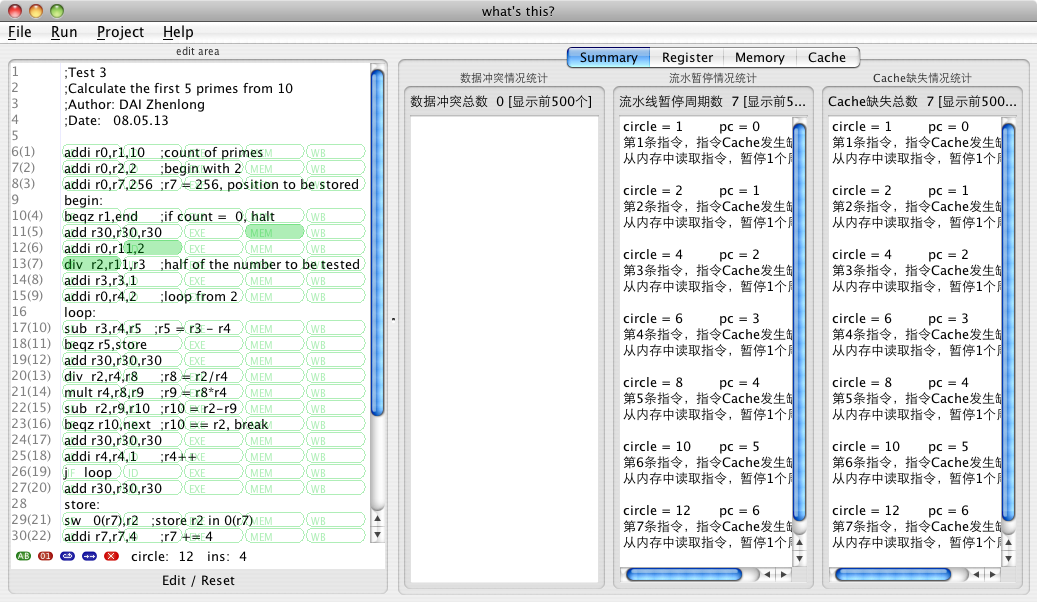


打开样例程序并编译：

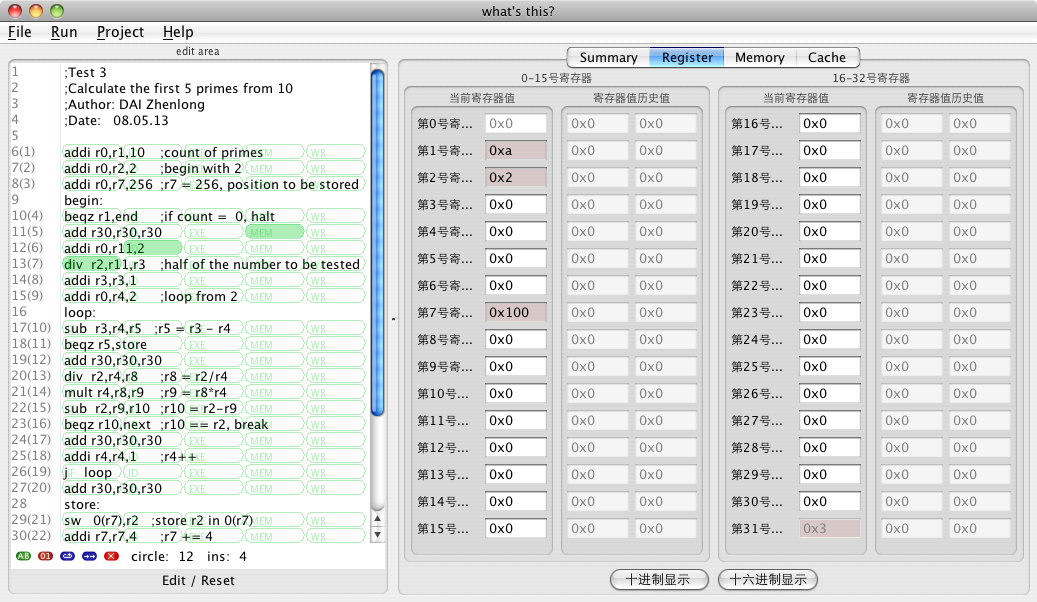


测试单步执行正确性：

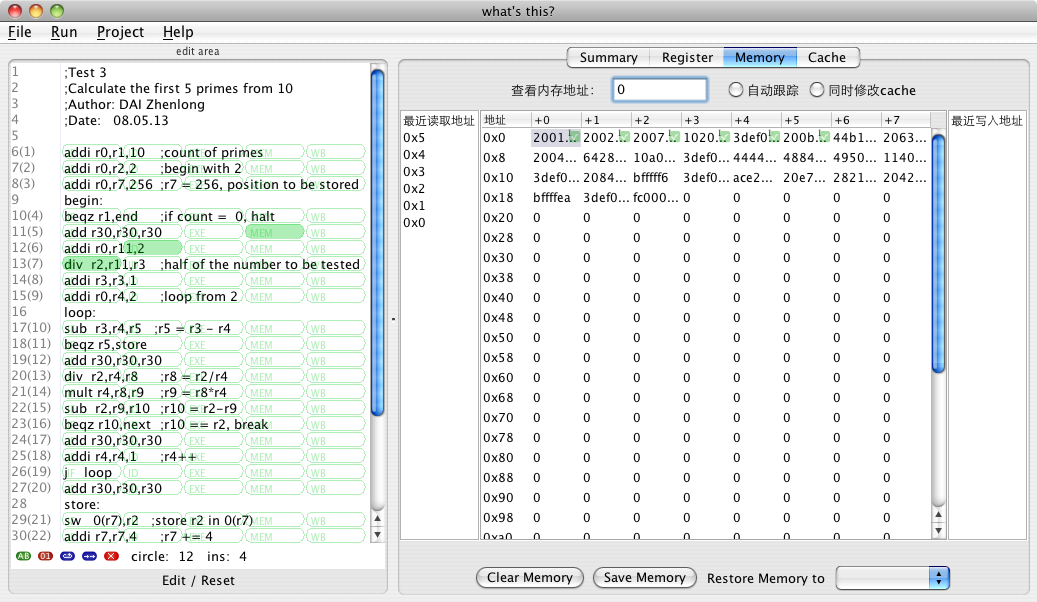




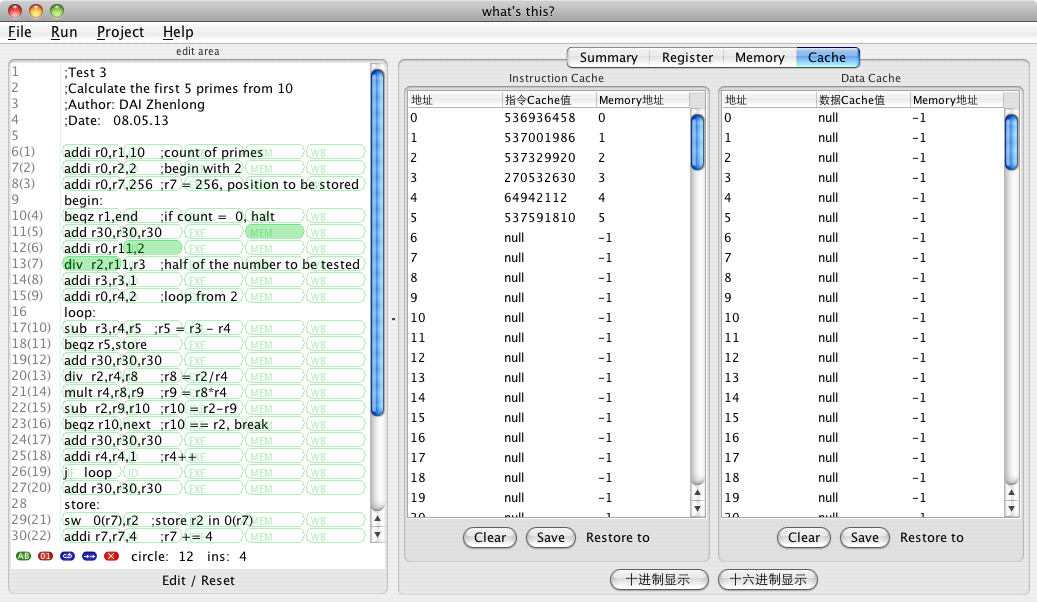
检查寄存器值：



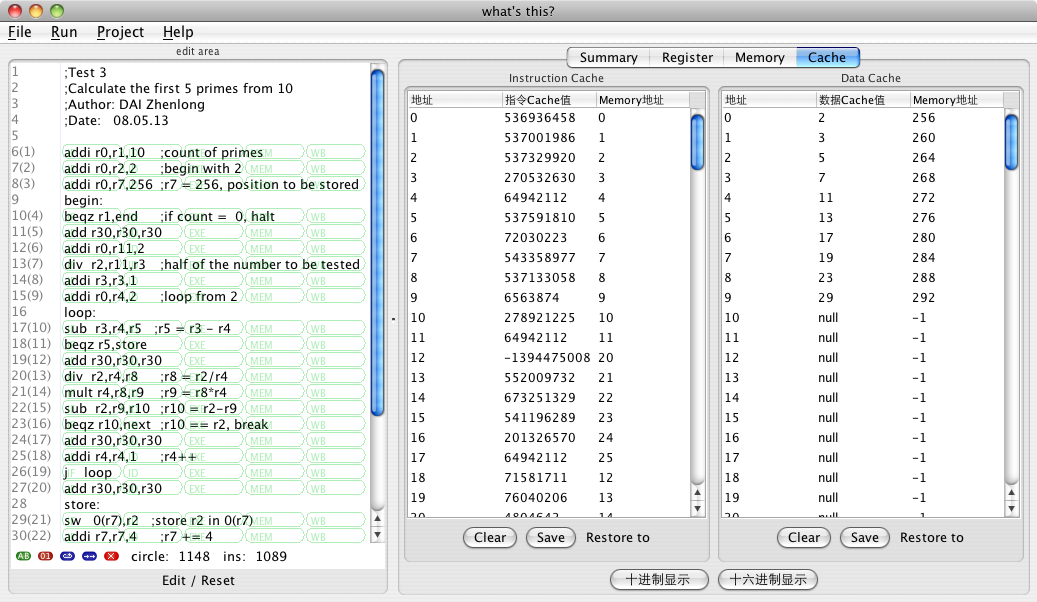
检查内存数据：



检查cache数据：



检查程序运行结果：



# 实验总结

通过这个实验对流水线的工作流程和对冲突的解决方法（data forwarding, stall）有了更深入的理解。同时通过实现这个简单的模拟器，对计算机的内部各功能部件之间的数据交换及信号传递有了更好的了解。

在流水线设计中，参考了《计算机组成原理》课程实验的设计思路，例如流水线的数据传递的参数设置等。

在UI设计中，借鉴了《软件工程》课程实验的经验，使用开源库来尽可能使界面简洁美观、便于使用。

本次实验是继《软件工程》课程实验PhoneMe之后，kde9工作小组第二次联合开发。通过SVN工具的使用，更加方便地进行代码的维护与更新。于此同时，我们也通过面对面交流直接进行沟通，以追求更好的交流效果，并借此更快地解决问题。

1. 2006011299, CS62, Mobile: 13401086576, Email: kxx006@gmail.com [↑](#footnote-ref-1)
2. 2006011293, CS62, Mobile: 13810313760, Email: huww06@gmail.com [↑](#footnote-ref-2)
3. 2006011291, CS62, Mobile: 15901033612, Email: edwardtoday@gmail.com [↑](#footnote-ref-3)