

MIPS-lite Simulator

《计算机系统结构》课程实验报告

kde9

kfirst 孔祥欣¹

deepsolo 胡玮玮²

edwardtoday 卿培³

2009/6/2

¹ 2006011299, CS62, Mobile: 13401086576, Email: kxx006@gmail.com

² 2006011293, CS62, Mobile: 13810313760, Email: huww06@gmail.com

³ 2006011291, CS62, Mobile: 15901033612, Email: edwardtoday@gmail.com

目录

实验目的	3
实验内容	4
实验环境	6
实验环境准备	7
实验流程及分工	8
模拟器设计	9
模拟器使用说明	10
模拟器功能测试	11
实验总结	15

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：

a) 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”。

b) 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数据写回寄存器的操作。

c) 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。

2. Memory 模拟组件。Memory 的主要要求如下：

a) 大小限制

要求至少具有 1kB 的存储空间。

b) 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。

3. Cache 模拟组件。Cache 的主要要求如下：

a) 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，

b) 软件模拟运行效率和开销

Cache 的软件模拟要求与 Memory 相同。

c) 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。

4. MIPS 文本指令编译器。编译器的主要要求如下：

- a) 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。
- b) 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
3. 编程环境:
 - JDK 1.5.0.0
 - MyEclipse 7.0
 - Eclipse 6.5
 - Eclipse Ganymede on Mac OS X
4. 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

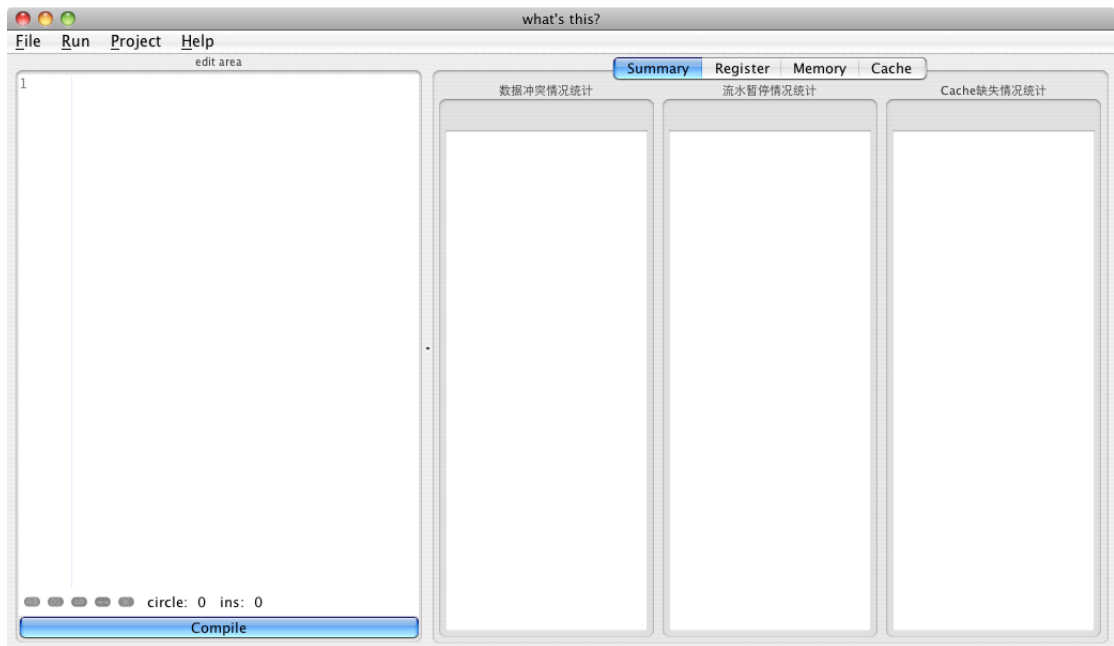
模拟器使用说明

参见 MIPS-lite_Simulator_UserManual_kde9

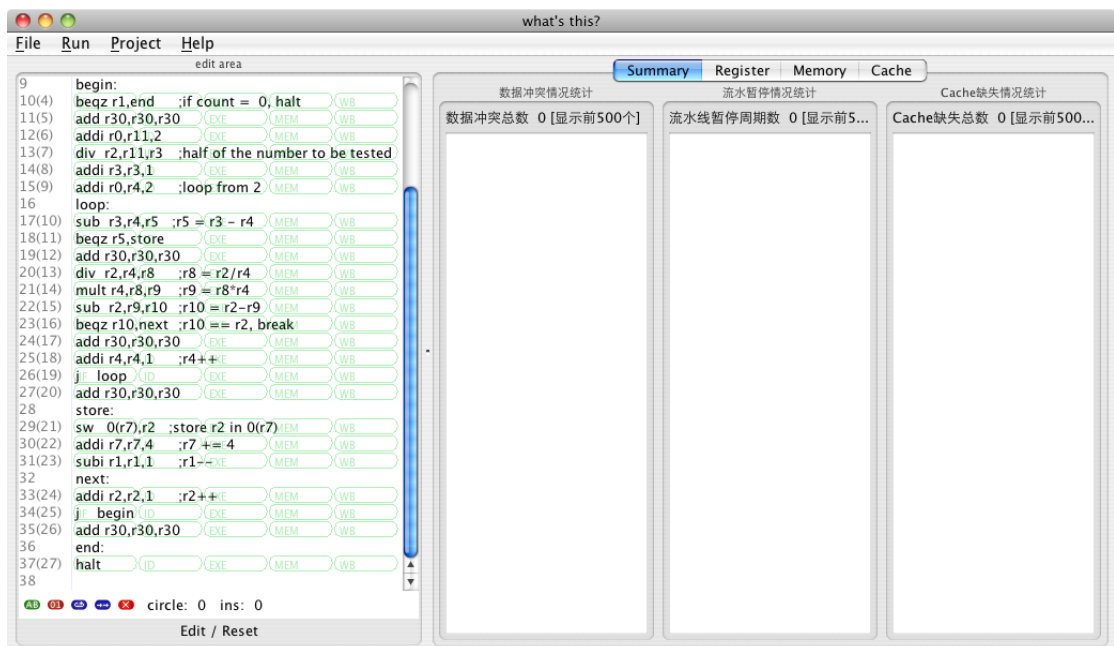
模拟器功能测试

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:

what's this?

File Run Project Help

edit area

```
1 ;Test 3
2 ;Calculate the first 5 primes from 10
3 ;Author: DAI Zhenlong
4 ;Date: 08.05.13
5
6(1) addi r0,r1,10 ;count of primes MEM WB
7(2) addi r0,r2,2 ;begin with 2 MEM WB
8(3) addi r0,r7,256 ;r7 = 256, position to be stored
9 begin:
10(4) beqz r1,end ;if count = 0, halt WB
11(5) add r30,r30,r30 EXE MEM WB
12(6) addi r0,r1,2 EXE MEM WB
13(7) div r2,r1,r3 ;half of the number to be tested
14(8) addi r3,r3,1 EXE MEM WB
15(9) addi r0,r4,2 ;loop from 2 MEM WB
16 loop:
17(10) sub r3,r4,r5 ;r5 = r3 - r4 MEM WB
18(11) beqz r5,store EXE MEM WB
19(12) add r30,r30,r30 EXE MEM WB
20(13) div r2,r4,r8 ;r8 = r2/r4 MEM WB
21(14) mult r4,r8,r9 ;r9 = r8*r4 MEM WB
22(15) sub r2,r9,r10 ;r10 = r2-r9 WB
23(16) beqz r10,next ;r10 == r2, break! WB
24(17) add r30,r30,r30 EXE MEM WB
25(18) addi r4,r4,1 ;r4++ EXE MEM WB
26(19) j loop ID EXE MEM WB
27(20) add r30,r30,r30 EXE MEM WB
28 store:
29(21) sw 0(r7),r2 ;store r2 in 0(r7) MEM WB
30(22) addi r7,r7,4 ;r7 += 4 MEM WB
```

circle: 2 ins: 0

Edit / Reset

Summary Register Memory Cache

数据冲突情况统计 数据冲突总数 0 [显示前500个]

流水线暂停周期数 2 [显示前500个]

Cache缺失情况统计 Cache缺失总数 2 [显示前500个]

circle = 1 pc = 0
第1条指令, 指令Cache发生缺失, 从内存中读取指令, 暂停1个周期

circle = 2 pc = 1
第2条指令, 指令Cache发生缺失, 从内存中读取指令, 暂停1个周期

what's this?

File Run Project Help

edit area

```
1 ;Test 3
2 ;Calculate the first 5 primes from 10
3 ;Author: DAI Zhenlong
4 ;Date: 08.05.13
5
6(1) addi r0,r1,10 ;count of primes MEM WB
7(2) addi r0,r2,2 ;begin with 2 MEM WB
8(3) addi r0,r7,256 ;r7 = 256, position to be stored
9 begin:
10(4) beqz r1,end ;if count = 0, halt WB
11(5) add r30,r30,r30 EXE MEM WB
12(6) addi r0,r1,2 EXE MEM WB
13(7) div r2,r1,r3 ;half of the number to be tested
14(8) addi r3,r3,1 EXE MEM WB
15(9) addi r0,r4,2 ;loop from 2 MEM WB
16 loop:
17(10) sub r3,r4,r5 ;r5 = r3 - r4 MEM WB
18(11) beqz r5,store EXE MEM WB
19(12) add r30,r30,r30 EXE MEM WB
20(13) div r2,r4,r8 ;r8 = r2/r4 MEM WB
21(14) mult r4,r8,r9 ;r9 = r8*r4 MEM WB
22(15) sub r2,r9,r10 ;r10 = r2-r9 WB
23(16) beqz r10,next ;r10 == r2, break! WB
24(17) add r30,r30,r30 EXE MEM WB
25(18) addi r4,r4,1 ;r4++ EXE MEM WB
26(19) j loop ID EXE MEM WB
27(20) add r30,r30,r30 EXE MEM WB
28 store:
29(21) sw 0(r7),r2 ;store r2 in 0(r7) MEM WB
30(22) addi r7,r7,4 ;r7 += 4 MEM WB
```

circle: 12 ins: 4

Edit / Reset

Summary Register Memory Cache

数据冲突情况统计 数据冲突总数 0 [显示前500个]

流水线暂停周期数 7 [显示前500个]

Cache缺失情况统计 Cache缺失总数 7 [显示前500个]

circle = 1 pc = 0
第1条指令, 指令Cache发生缺失, 从内存中读取指令, 暂停1个周期

circle = 2 pc = 1
第2条指令, 指令Cache发生缺失, 从内存中读取指令, 暂停1个周期

circle = 4 pc = 2
第3条指令, 指令Cache发生缺失, 从内存中读取指令, 暂停1个周期

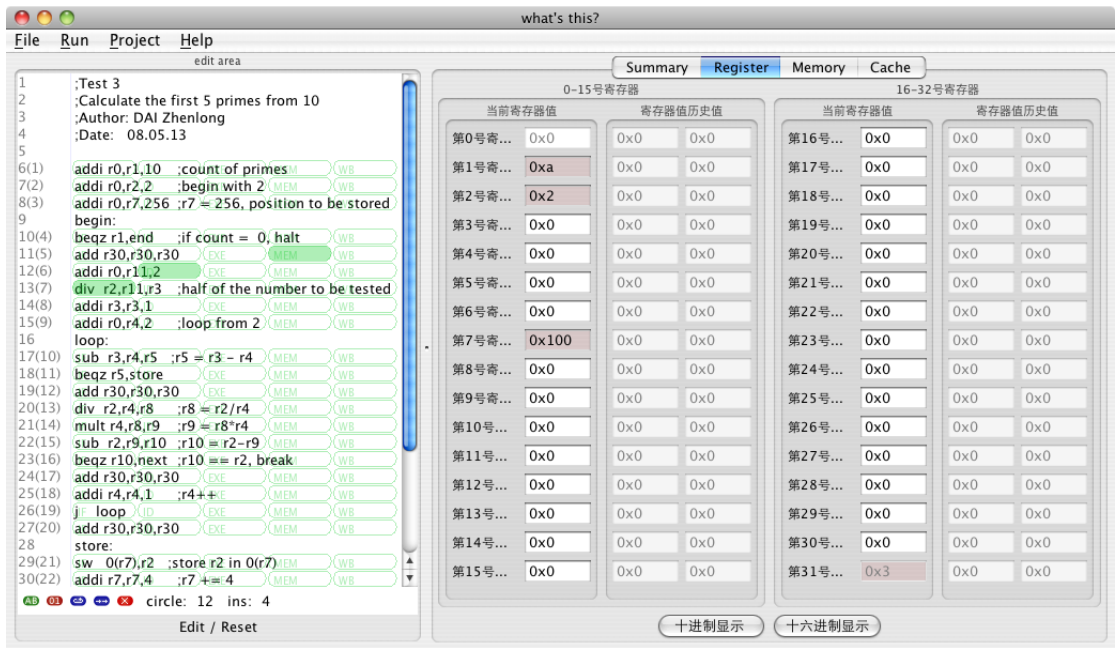
circle = 6 pc = 3
第4条指令, 指令Cache发生缺失, 从内存中读取指令, 暂停1个周期

circle = 8 pc = 4
第5条指令, 指令Cache发生缺失, 从内存中读取指令, 暂停1个周期

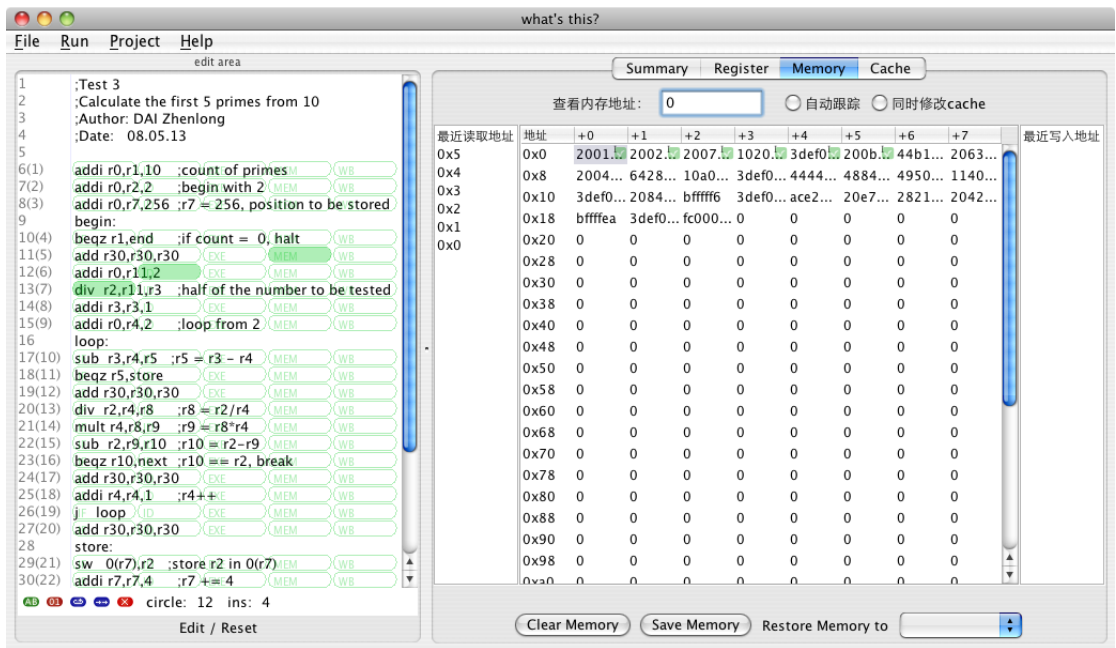
circle = 10 pc = 5
第6条指令, 指令Cache发生缺失, 从内存中读取指令, 暂停1个周期

circle = 12 pc = 6
第7条指令, 指令Cache发生缺失, 从内存中读取指令, 暂停1个周期

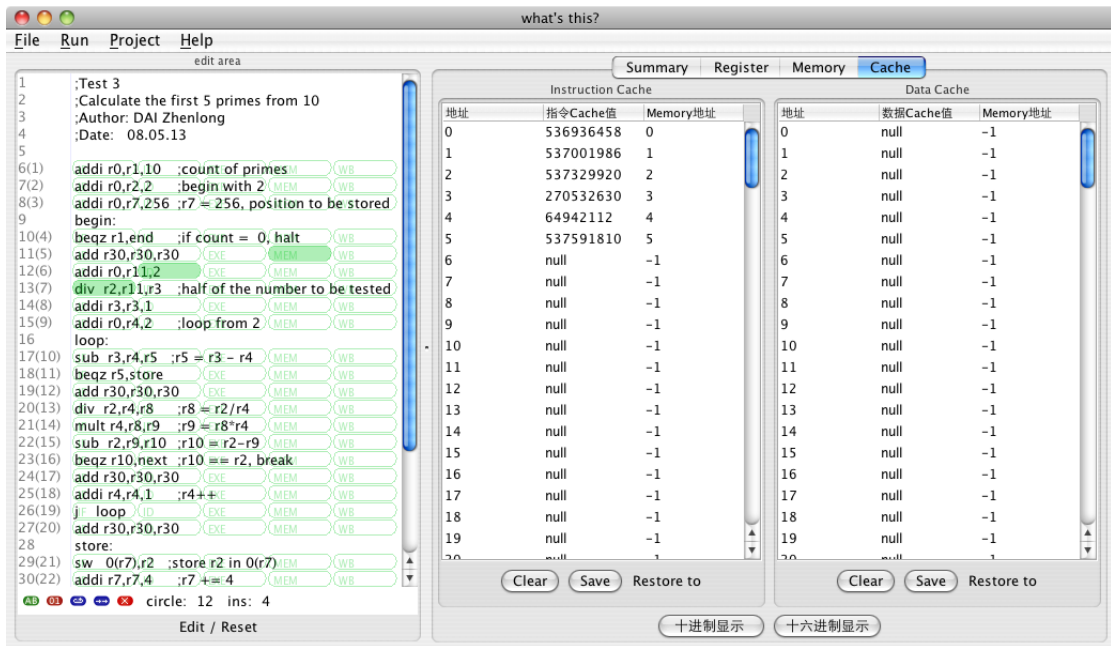
检查寄存器值:



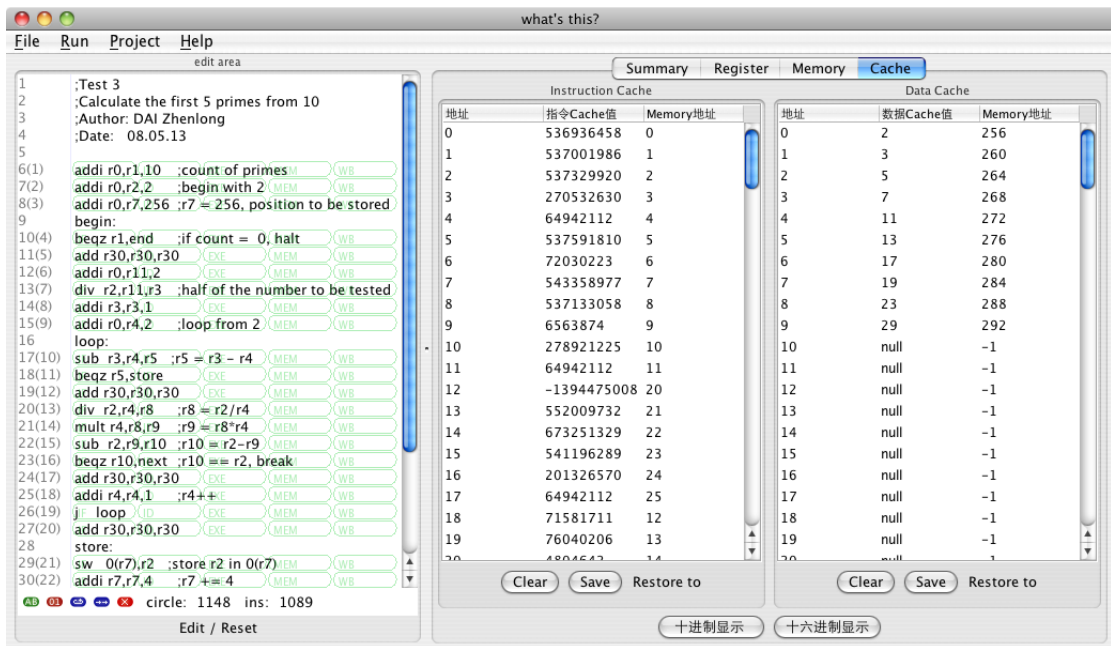
检查内存数据:



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。