

实验报告

计试 81 白思雨 2186123935

1、实验目的：

- (1) 加深对计算机流水线基本概念的理解；
- (2) 理解 MIPS 结构如何用 5 段流水线来实现，理解各段的功能和基本操作；
- (3) 加深对数据冲突、结构冲突、控制冲突的理解，并能够分析这些冲突对 CPU 性能的影响；
- (4) 进一步理解解决数据冲突的方法，掌握如何应用定性技术来减少数据冲突引起的流水线停顿。

2、实验平台：

自己设计一个流水线模拟器（简称：模拟器 A）

3、实验内容，步骤和结果：

(1) 模拟器 A 的设计思想、特色：

本模拟器参考借鉴了各个开源的 C++ 项目代码：借鉴了 MIPS Simulator 作为主框架，确保主程序的健全性及功能丰富性，**具体实现上进行了更多的兼容指令，并且优化了交互界面，使得交互更加清晰易懂且符合课程实验要求**；借鉴 Crystal Text Buffer 文本编辑器，实现 MIPS 代码编辑器、高亮、报错提示等功能；借鉴 WorldCom Grid Control 表格控件，实现统计运行数据、寄存器、内存的编辑展示。

特色：兼容大量常用指令，如：lw、sw、add、sub、mul、div、and、or、xor、addi、subi、muli、divi、andi、ori、xori、beqz、bnez、j、jr、trap、nop，共计 24 种指令；交互式界面，易于观察，操作简单便捷；可手动设定乘除法延迟时间、内存、寄存器等值，操作性强；流水线示意图中的步骤采用不同颜色加以区分，方便使用者学习理解；软件占用空间大小仅为 2.89MB，易于存储、拷贝携带。

(2) 模拟器内代码测试：

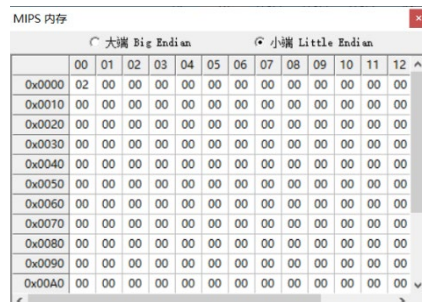
① 没有任何冲突的流水线场景：

从内存中读取数据并做加法，最后写回内存中

start:

初始内存：

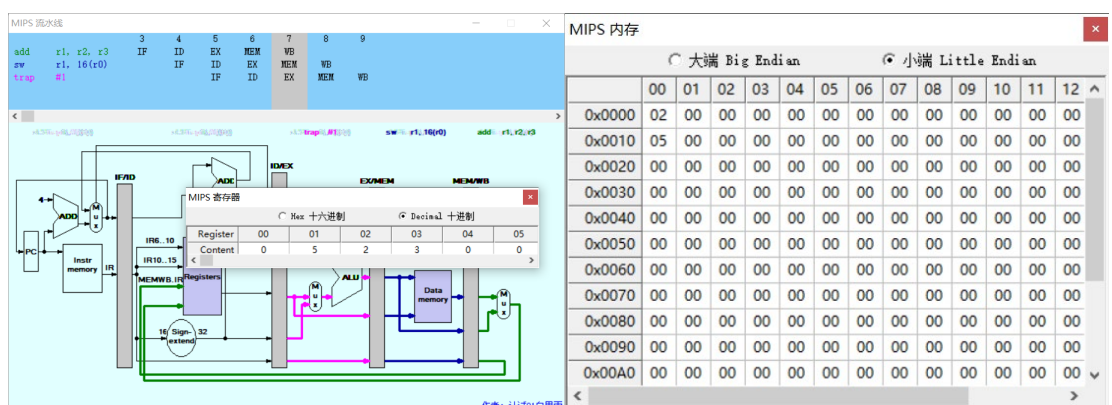
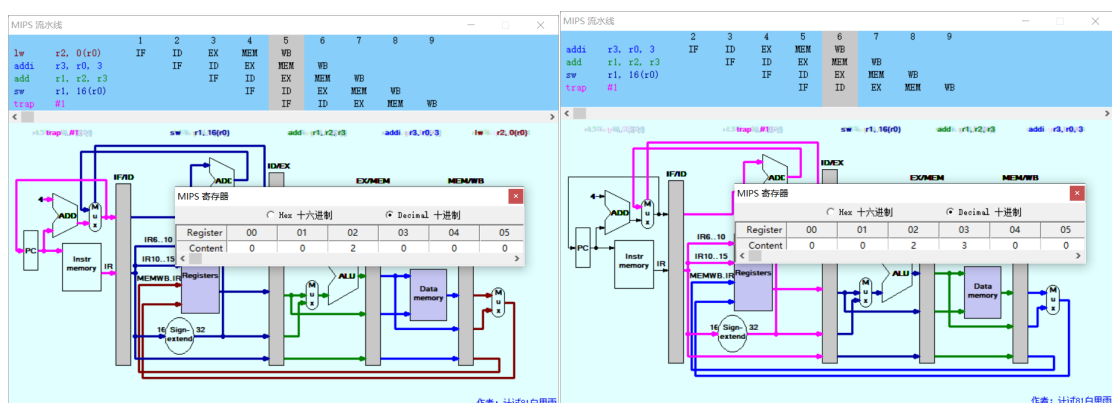
```
lw      r2, 0(r0)
addi    r3, r0, 3
add     r1, r2, r3
sw      r1, 16(r0)
trap    #1
```



	00	01	02	03	04	05	06	07	08	09	10	11	12
0x0000	02	00	00	00	00	00	00	00	00	00	00	00	00
0x0010	00	00	00	00	00	00	00	00	00	00	00	00	00
0x0020	00	00	00	00	00	00	00	00	00	00	00	00	00
0x0030	00	00	00	00	00	00	00	00	00	00	00	00	00
0x0040	00	00	00	00	00	00	00	00	00	00	00	00	00
0x0050	00	00	00	00	00	00	00	00	00	00	00	00	00
0x0060	00	00	00	00	00	00	00	00	00	00	00	00	00
0x0070	00	00	00	00	00	00	00	00	00	00	00	00	00
0x0080	00	00	00	00	00	00	00	00	00	00	00	00	00
0x0090	00	00	00	00	00	00	00	00	00	00	00	00	00
0x00A0	00	00	00	00	00	00	00	00	00	00	00	00	00

第一步 lw r2, 0(r0) 需要到 WB 写回阶段才能将数据写入寄存器中，而第二步

add r3, r0, 3 与之无关。故在第三步时不会产生 RAW 冲突，无需 stall 阻塞。



② 有至少一次的 RAW 冲突：

从内存中读取数据并做加法，最后写回内存中
start:

初始内存：

```
addi    r3, r0, 3
lw      r2, 0(r0)
add     r1, r2, r3
sw      r1, 16(r0)
trap    #1
```

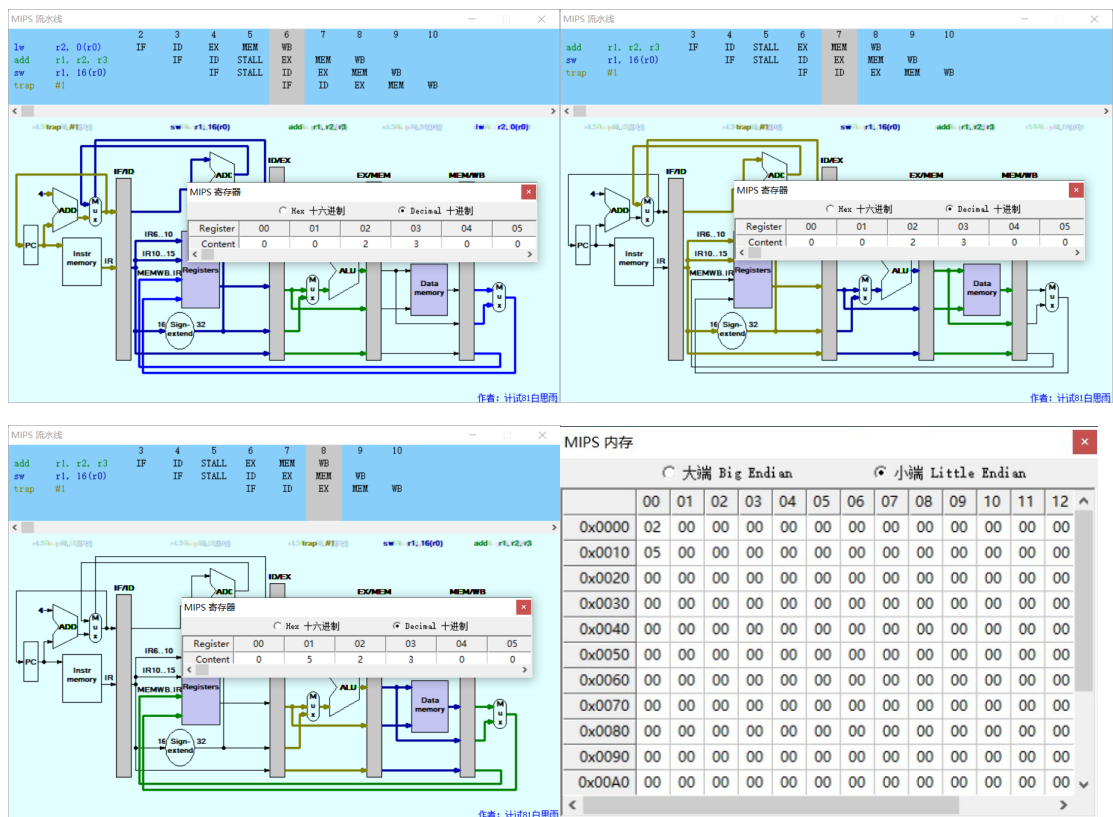
MIPS 内存

大端 Big Endian

小端 Little Endian

	00	01	02	03	04	05	06	07	08	09	10	11	12	^
0x0000	02	00	00	00	00	00	00	00	00	00	00	00	00	
0x0010	00	00	00	00	00	00	00	00	00	00	00	00	00	
0x0020	00	00	00	00	00	00	00	00	00	00	00	00	00	
0x0030	00	00	00	00	00	00	00	00	00	00	00	00	00	
0x0040	00	00	00	00	00	00	00	00	00	00	00	00	00	
0x0050	00	00	00	00	00	00	00	00	00	00	00	00	00	
0x0060	00	00	00	00	00	00	00	00	00	00	00	00	00	
0x0070	00	00	00	00	00	00	00	00	00	00	00	00	00	
0x0080	00	00	00	00	00	00	00	00	00	00	00	00	00	
0x0090	00	00	00	00	00	00	00	00	00	00	00	00	00	
0x00A0	00	00	00	00	00	00	00	00	00	00	00	00	00	^

第二步 `lw r2, 0(r0)` 需要到 WB 写回阶段才能将数据写入寄存器中。所以第三步 `add r1, r2, r3` 必须等待前面完成后才能继续进行，流水线出现 stall 阻塞，即数据冒险 RAW 冲突。



③ 有至少一次的分支跳转：

从 1 到 10 累加，将结果存到 r2 寄存器中。

代码：

start:

addi r1, r0, 1

addi r2, r0, 0

loop: add r2, r2, r1

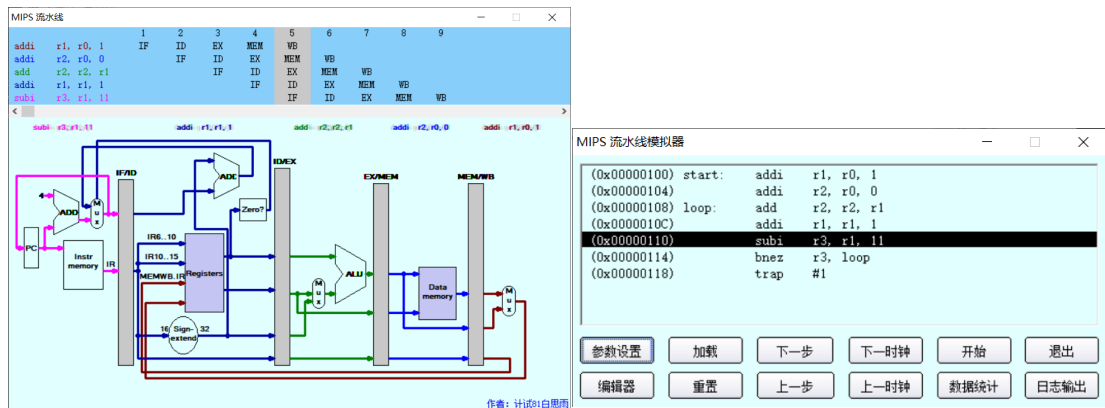
addi r1, r1, 1

subi r3, r1, 11

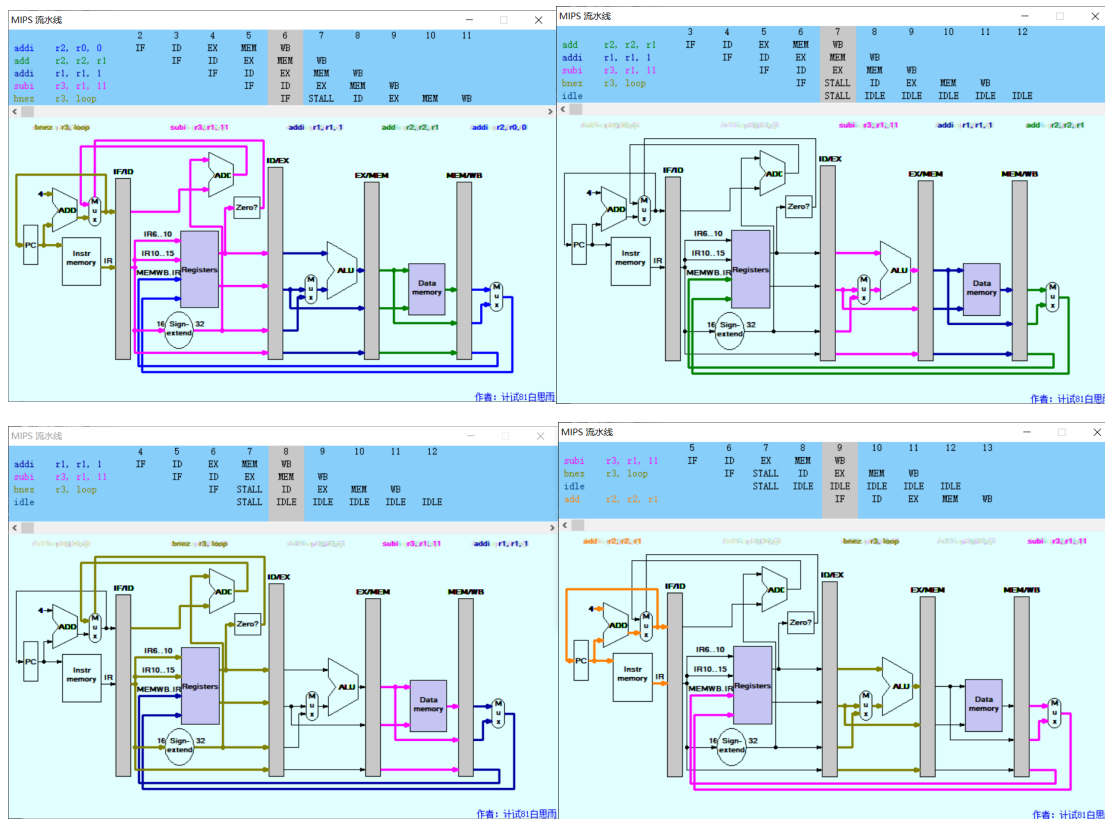
bnez r3, loop

trap #1

前面是正常的加减法，无冲突。



在 BNEZ 跳转时，必须等到 MEM 阶段才能确定是否执行分支，出现分支冒险。



最后程序运行结果可在寄存器内查看，从 1 到 10 累加结果为 55，与 r2 寄存器内的值一致。

MIPS 寄存器						
Hex 十六进制						
Register	00	01	02	03	04	05
Content	0	11	55	0	0	0

4.源码

见附件。

5.实验感悟

在最初着手做这个实验，搜寻网上资料时，发现大部分 5 段流水线是以 ModelSim 波形仿真的形式实现，更加偏向于硬件原理，且十分不便于操作观察。而主推交互式界面的模拟器少之极少，像 QtSpim 这类开源模拟器较为庞大，所有页面均为英文，且没有流水线示意图，相当不利于初学者学习；MIPSsim 在网上的报告教程多如牛毛，并不能较好地锻炼我们的编程能力与创新思维。

通过本次实验，引用开源项目，自主编写 MIPS 模拟器程序，来学习理解 5 段流水线与冲突影响，极大地锻炼了我们计算机体系结构初学者的动手编程能力，也让我们更好地学习理解了 MIPS 5 段流水线的原理以及实现过程，受益匪浅！

十分感谢任课老师对本次实验的精心设计和安排！