

流水线及流水线中的冲突

计算机体系结构实验二



目录

[1． 实验目的 1](#_Toc515348020)

[2． 实验平台 1](#_Toc515348021)

[3． 实验内容和步骤 1](#_Toc515348022)

[4． 实验总结 8](#_Toc515348023)

2017-5-24

[裴子祥 计科七班 学号2015211921]

[指导老师：黄智濒]

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）加深对数据冲突和资源冲突的理解，理解这两类冲突对 CPU 性能的影响。

（4） 进一步理解解决数据冲突的方法，掌握如何应用定向技术来减少数据冲突引起的停顿。

1. 实验平台

实验平台采用指令级和流水线操作级模拟器 MIPSsim。

1. 实验内容和步骤

首先要阅读 MIPSsim 模拟器的使用方法（见附录），然后了解 MIPSsim 的指令系统和汇编语言。

（1）启动 MIPSsim。

（2）进一步理解流水线窗口中各段的功能，掌握各流水寄存器的含义。（鼠标双击各段，即 可看到各流水寄存器的内容）

**各段含义：**

**取指令周期（IF）**

**指令译码/读寄存器周期（ID）**

**执行/有效地址计算周期（EX）**

**存储器访问／分支完成周期（MEM）**

**写回周期（WB）**

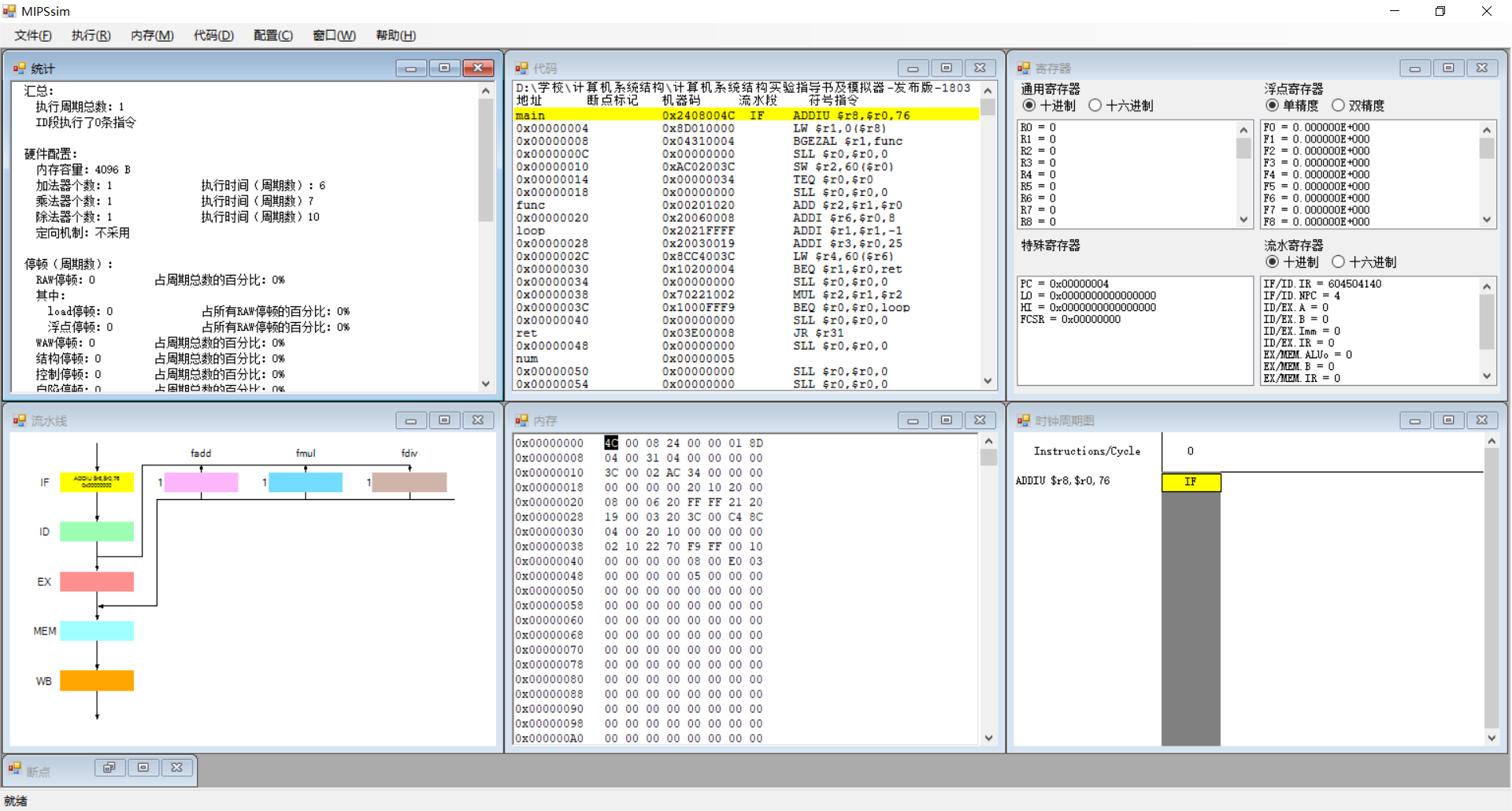
（3）载入一个样例程序（在本模拟器所在文件夹下的“样例程序”文件夹中），然后分别以 单步执行一个周期、执行多个周期、连续执行、设置断点等方式运行程序，观察程序的执行 情况，观察 CPU 中寄存器和存储器内容的变化，特别是流水寄存器内容的变化。

（4）选择配置菜单中的“流水方式”选项，使模拟器工作于流水方式下。

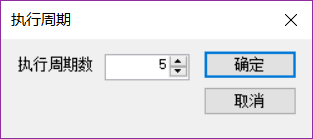
（5）观察程序在流水方式下的执行情况。

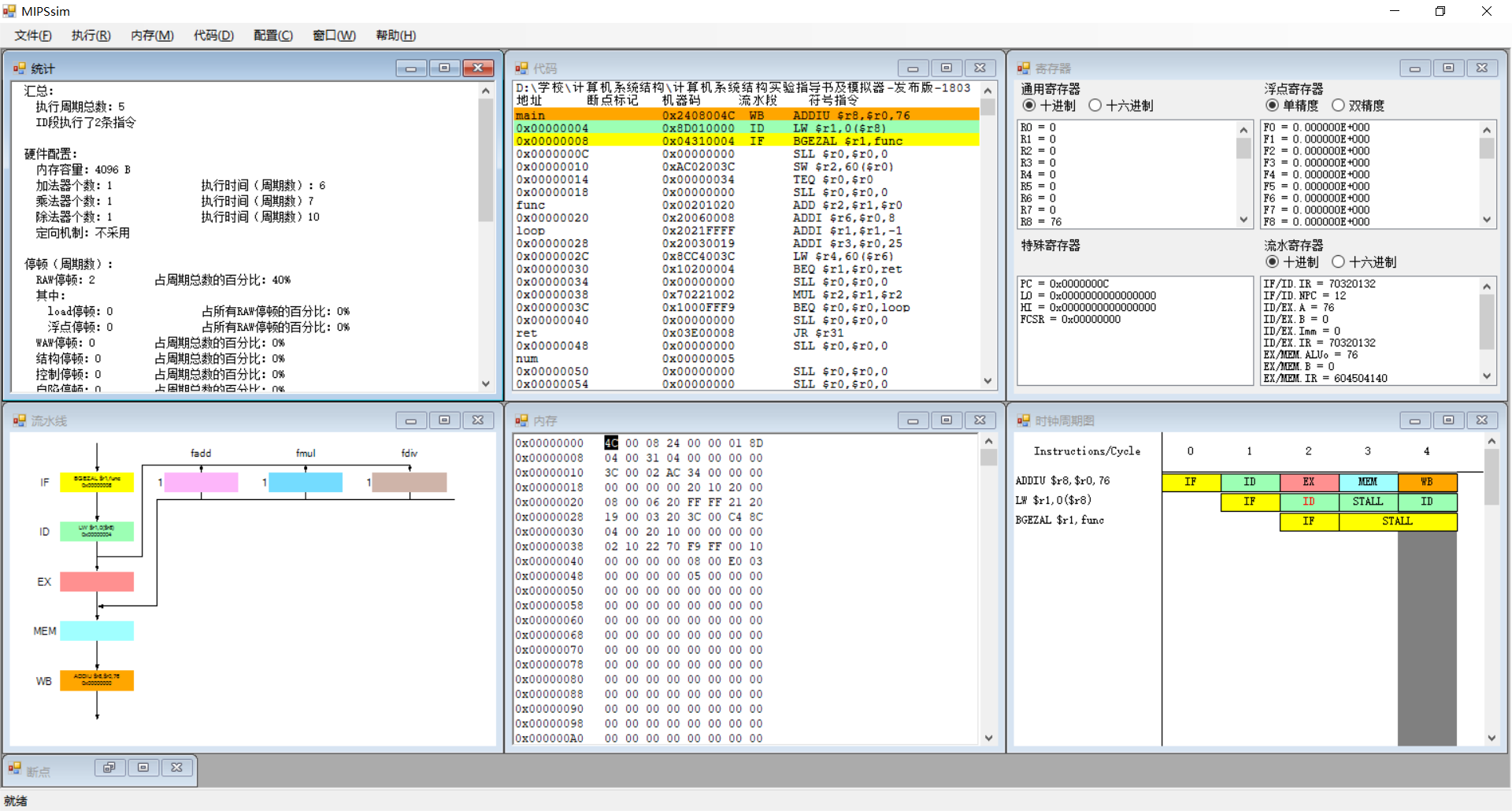
载入pipeline.s

1. 单步执行一个周期

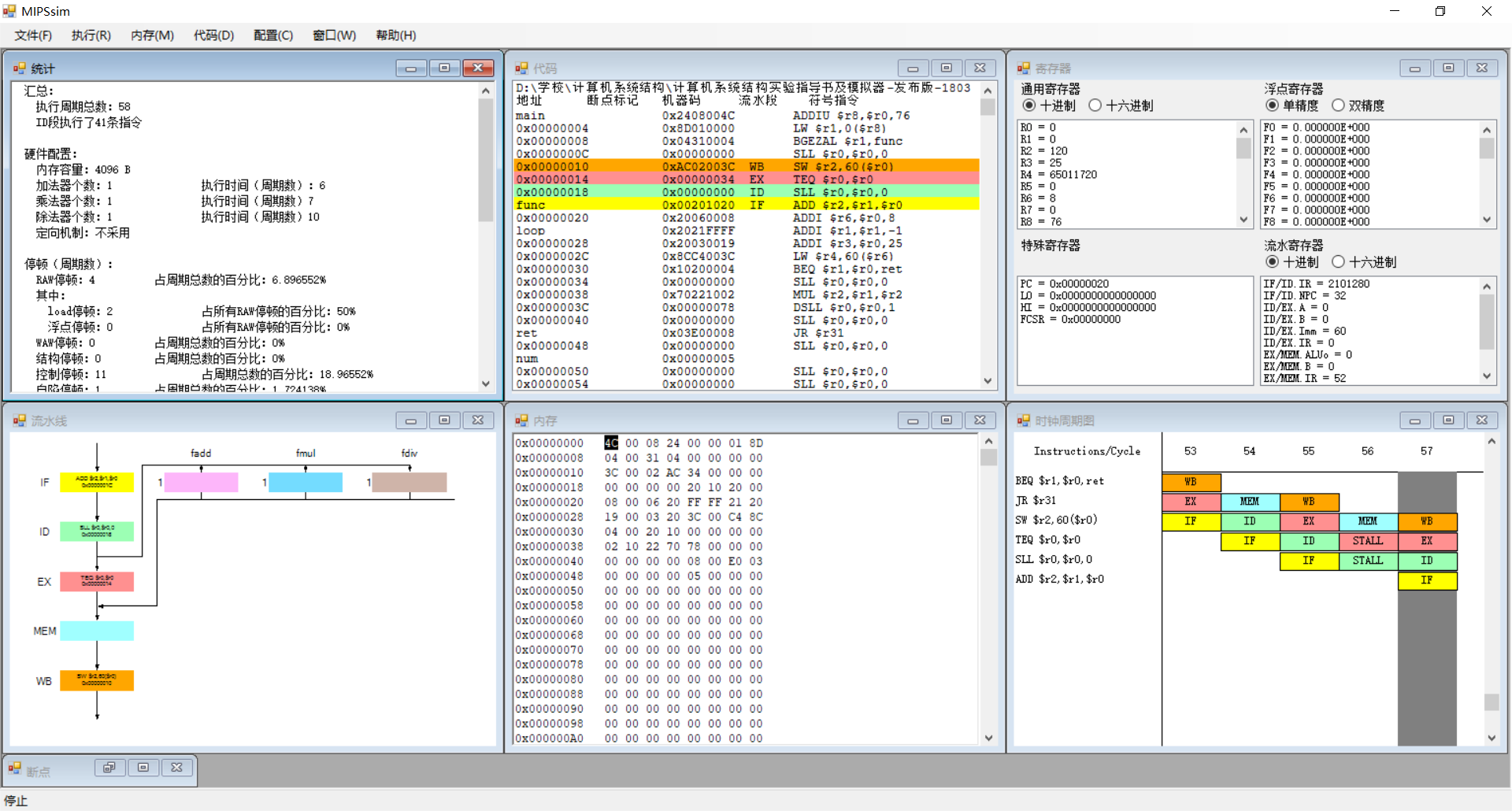


1. 执行多个周期，如5个周期

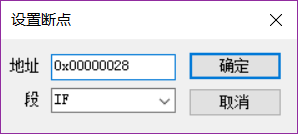


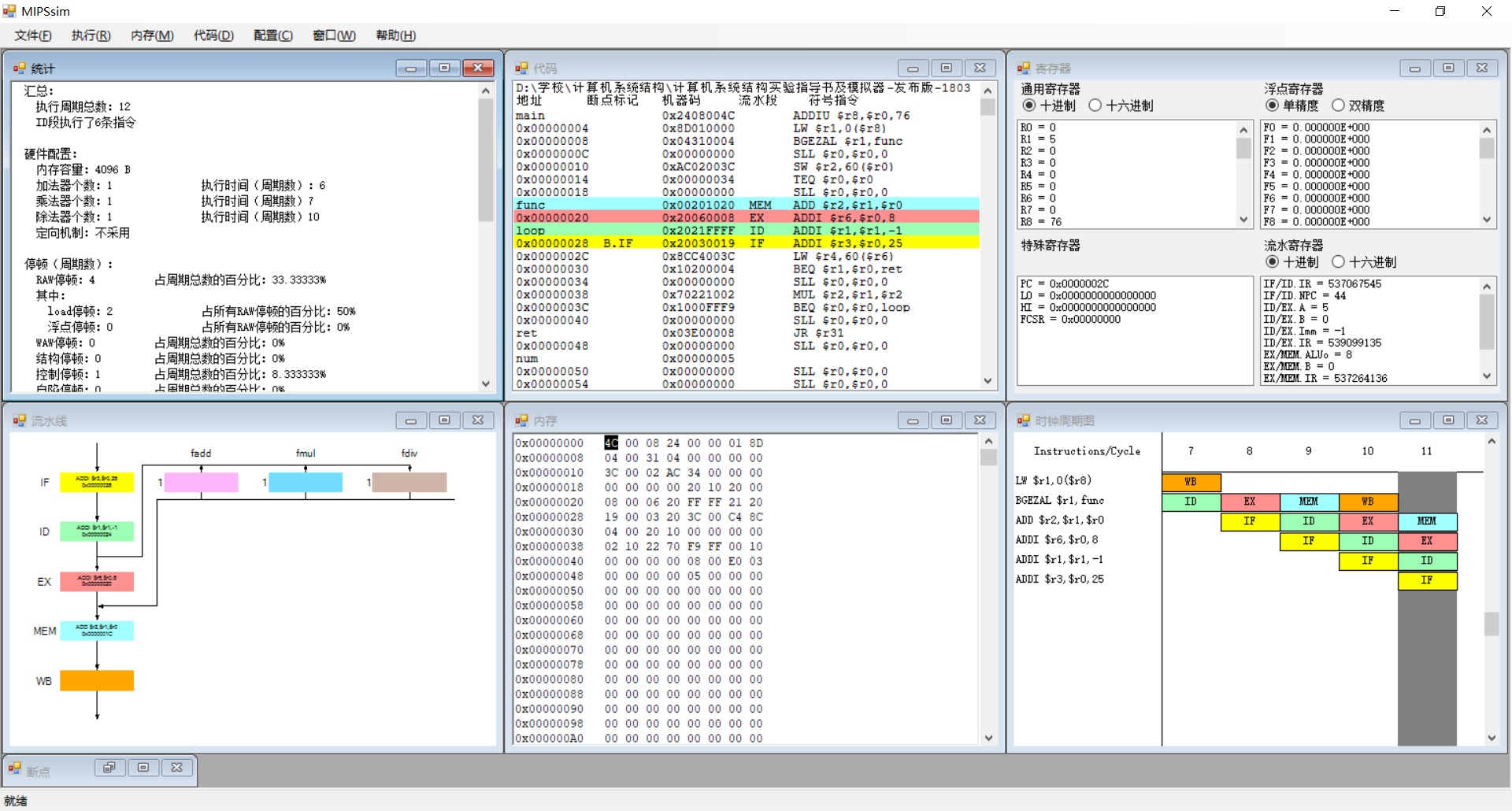


1. 连续执行



1. 设置断点





（6）观察和分析结构冲突对 CPU 性能的影响

步骤如下：

1. 加载 structure\_hz.s（在模拟器所在文件夹下的“样例程序”文件夹中）

.text

main**:**

ADD.D **$**f2,**$**f0,**$**f1

ADD.D **$**f3,**$**f0,**$**f1

ADD.D **$**f4,**$**f0,**$**f1

ADD.D **$**f5,**$**f0,**$**f1

ADD.D **$**f6,**$**f0,**$**f1

ADD.D **$**f7,**$**f0,**$**f1

ADD.D **$**f8,**$**f0,**$**f1

ADD.D **$**f9,**$**f0,**$**f1

TEQ **$**r0,**$**r0

1. 执行该程序，找出存在结构冲突的指令对以及导致结构冲突的部件

**存在结构冲突的指令：多条ADD**

**导致结构冲突的部件：加法器**

1. 记录由结构冲突引起的停顿周期数，计算停顿周期数占总执行周期数的百分比

**执行周期总数：52**

**结构停顿：35**

**占周期总数的百分比：67.30769%**

1. 把浮点加法器的个数改为 4 个



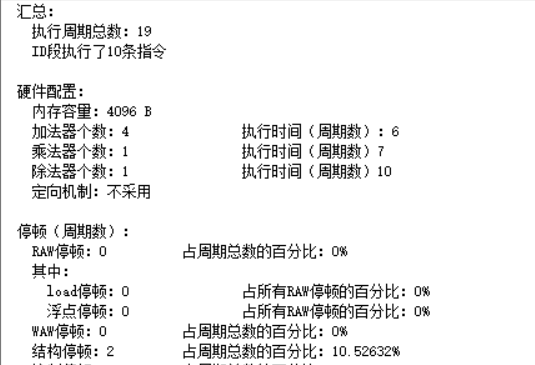
1. 再重复 a-c 的步骤

**导致冲突的部件仍是加法器**

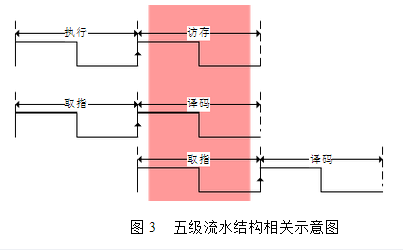
**执行周期总数：19**

**结构停顿：2**

**占周期总数的百分比：10.52632%**



1. 分析结构冲突对 CPU 性能的影响，讨论解决结构冲突的方法



**影响：结构冲突会降低CPU的利用率，吞吐率，流水线可能会出现停顿，从而降低流水线的效率和实际的加速比。**

**解决方法：**

1. **插入暂停周期，取后一条指令操作。**
2. **设置相互独立的指令存储器和数据存储器或设置相互独立的指令Cache和数据Cache。**

**3. 增加冲突部件，本实验中可以增加加法器的数量。**

（7）观察数据冲突并用定向技术来减少停顿。

步骤如下：

1. 全部复位。
2. 加载 data\_hz.s（在模拟器所在文件夹下的“样例程序”文件夹中）。

.text

main**:**

ADDIU **$**r2,**$**r0,A

ADDIU **$**r3,**$**r0,B

loop**:**

LW **$**r1,0**($**r2**)**

ADD **$**r1,**$**r1,**$**r3

SW **$**r1,0**($**r2**)**

LW **$**r5,0**($**r1**)**

ADDI **$**r5,**$**r5,10

ADDI **$**r2,**$**r2,4

SUB **$**r4,**$**r3,**$**r2

BGTZ **$**r4,loop

SLL **$**r0,**$**r0,0

TEQ **$**r0,**$**r0

SLL **$**r0,**$**r0,0

SLL **$**r0,**$**r0,0

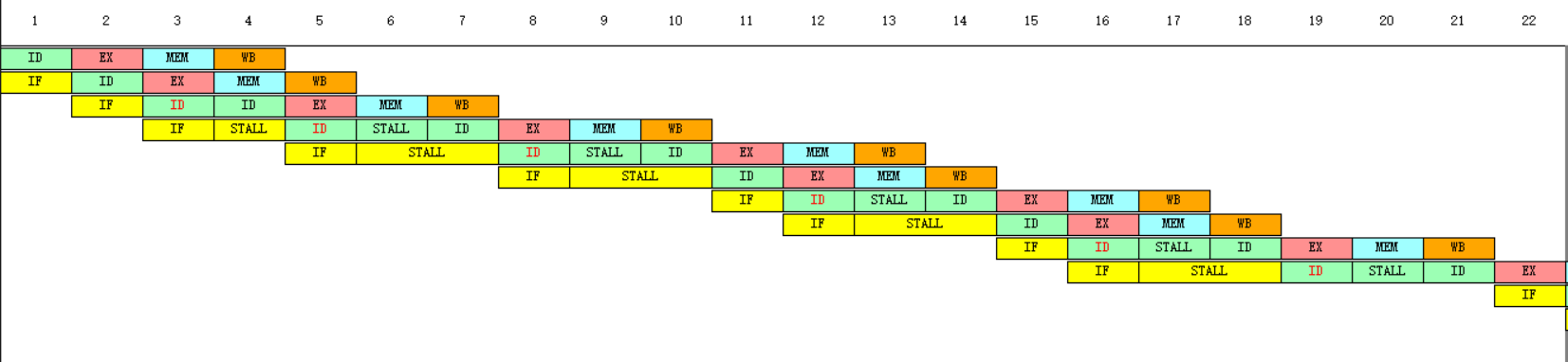
.data

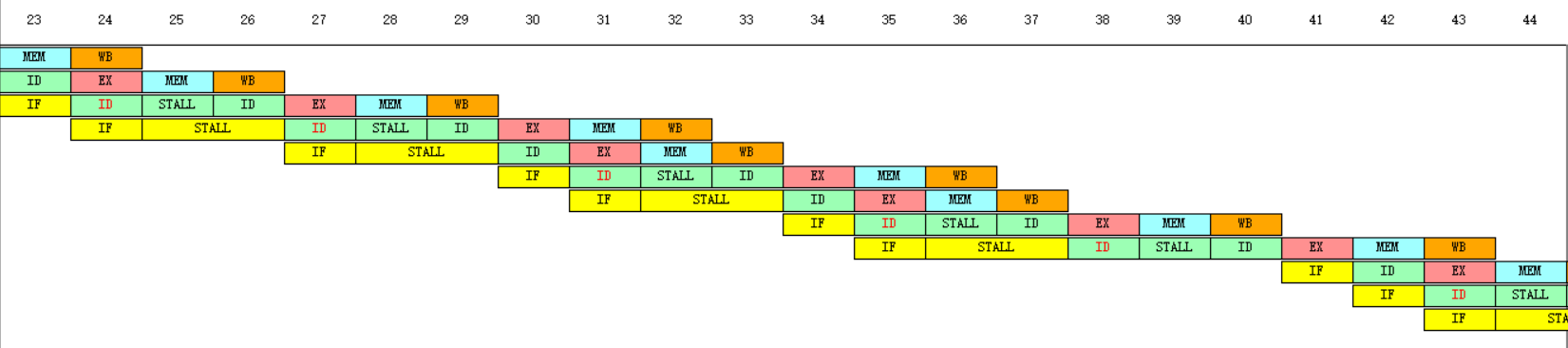
A**:** .word 0, 4, 8

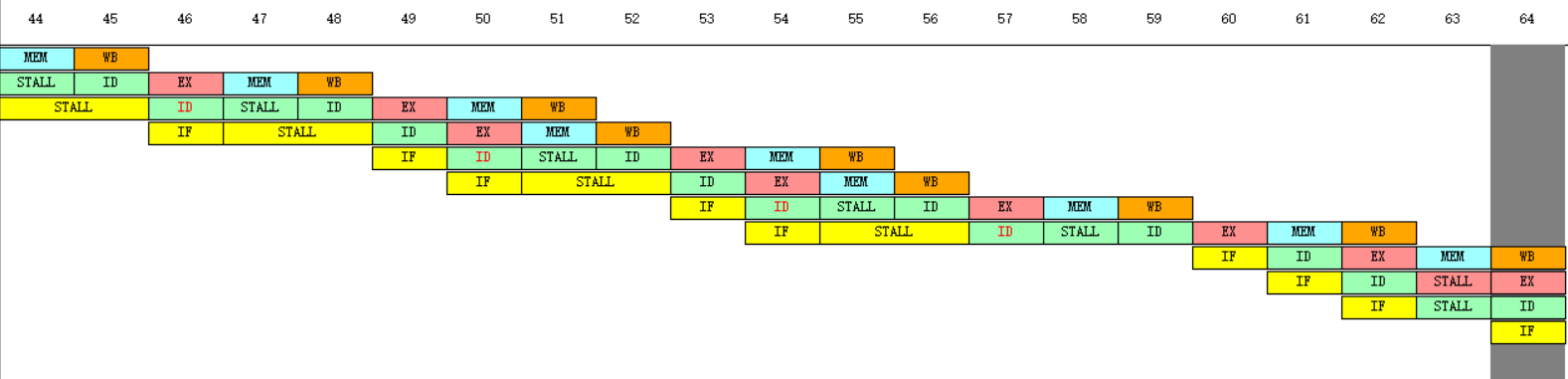
B**:** .word 2, 1, 0

1. 关闭定向功能（在“配置”菜单下选择取消“定向”）。
2. 用单步执行一个周期的方式执行该程序，观察时钟周期图，列出什么时刻发生了 RAW 冲突。

**在第 3、5、8、12、16、19、24、27、31、35、38、43、46、50、54、57周期发生了RAW冲突。**

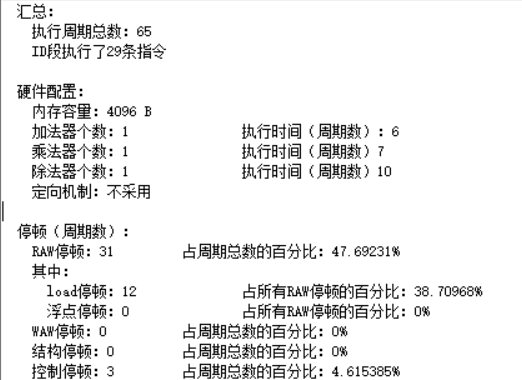






1. 记录数据冲突引起的停顿周期数以及程序执行的总时钟周期数，计算停顿时钟周期 数占总执行周期数的百分比。

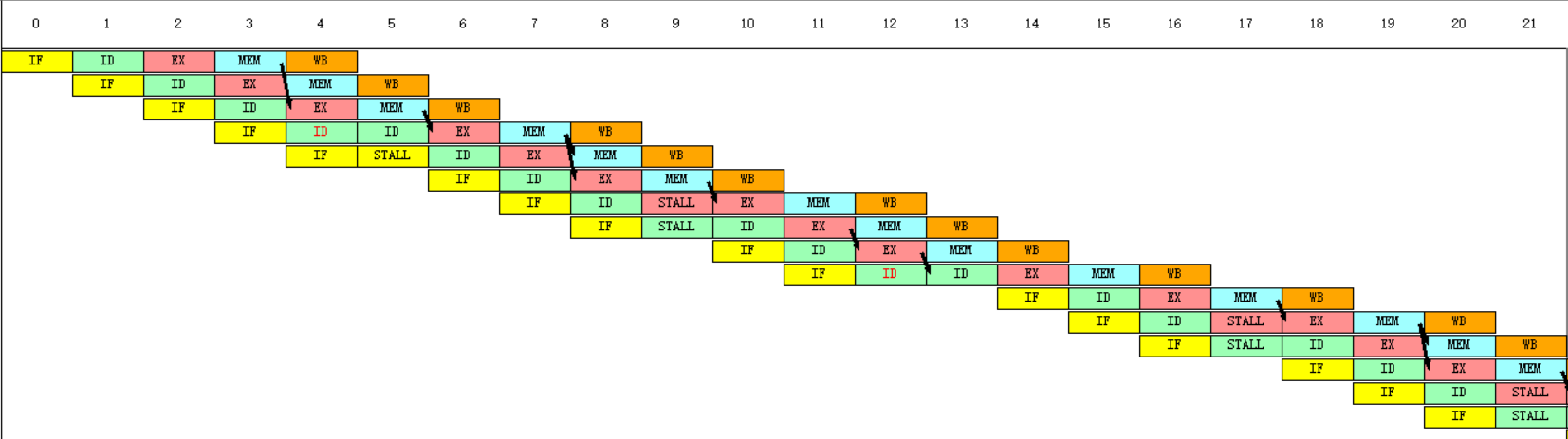
**执行周期总数：65，RAW停顿：31，占周期总数的百分比：47.69231%**

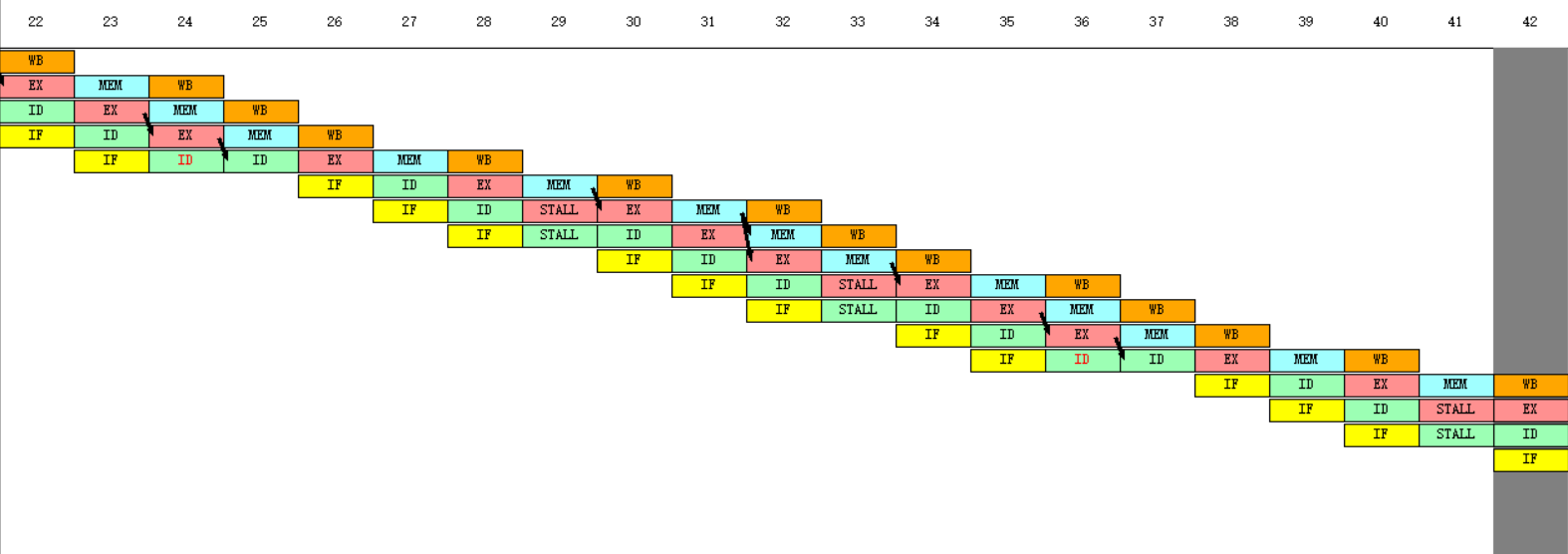


1. 复位 CPU。
2. 打开定向功能。
3. 用单步执行一个周期的方式执行该程序，查看时钟周期图，列出什么时刻发生了 RAW 冲突，并与步骤 c）的结果比较。

**在第 4、12、24、36时钟周期发生RAW冲突**

是时钟周期图如下:



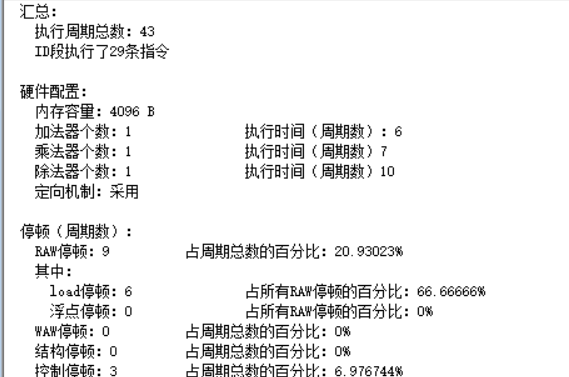


1. 记录数据冲突引起的停顿周期数以及程序执行的总周期数。计算采用定向以后性能比原来提高多少。

**执行周期总数：43**

**RAW停顿：9 ，占周期总数的百分比：20.93023%**

**65/43 = 1.51 提高了1.51倍**



1. 实验总结

此次实验，通过MIPSsim对指令集和流水线操作模拟，加深了对计算机流水线基本概念的理解。 并且通过5段流水线的模拟运行，加深对数据冲突和资源冲突的理解，以及分别对 CPU 性能的影响。

**结构冲突(资源冲突)** :流水线中多条指令在同一时钟周期内争用同一功能部件的现象。即因硬件资源满足不了指令重叠执行的要求而发生的冲突。常见的导致结构相关的原因：功能部件不是完全流水或者资源不够用。例如访存冲突。结构冲突解决方法：可以插入暂停周期，停顿一拍再运行；或者设置相互独立的指令存储器和数据存储器，设置相互独立的指令/数据cache。

**数据冲突：**当相关的指令靠得足够近时，它们在流水线中的重叠执行或者重新排序会改变指令读/写操作数的顺序，从而导致的程序执行逻辑上错误的现象称为数据相关冲突。数据冲突的解决方法：采用定向功能对RAW（先写后读）冲突所产生得影响，后推法以及定向技术对冲突有所改进。

**五段流水操作：**

