北京郵電大學

实验报告

题目：流水线及流水线中的冲突

班　　级：2020211310

学　　号：2020211616

姓　　名：付容天

学　　院：计算机学院（国家示范性软件学院）

2023 年 4 月 18 日

一、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) 加深对数据冲突的资源冲突的理解，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。

二、实验平台

实验平台采用指令级和流水线操作级模拟器 MIPSsim。

三、实验内容

首先要阅读附录中的 MIPSsim 模拟器的使用方法，然后了解 MIPSsim 的指令系统和汇编语言，并理解流水线窗口中各段的功能。在 MIPSsim 模拟器中载入样例程序 structure_hz.s，在模拟器中进行流水执行，查看并分析每条指令执行后相关寄存器的值，分析结构冲突对 CPU 性能的影响。

四、实验步骤及实验分析

在实验中，我首先启动了 MIPSsim，然后配置为流水方式，并载入了样例程序 structure_hz.s，可以发现存在冲突的部件为加法部件，存在冲突代码段为：

地址	断点标记	机器码	流水段	符号指令
main		0x46210080		ADD.D $f2,$f0,$f1
0x00000004		0x462100C0		ADD.D $f3,$f0,$f1
0x00000008		0x46210100		ADD.D $f4,$f0,$f1
0x0000000C		0x46210140		ADD.D $f5,$f0,$f1
0x00000010		0x46210180		ADD.D $f6,$f0,$f1
0x00000014		0x462101C0		ADD.D $f7,$f0,$f1
0x00000018		0x46210200		ADD.D $f8,$f0,$f1
0x0000001C		0x46210240		ADD.D $f9,$f0,$f1

图 1：存在冲突的代码段

观察可知，任何两条加法指令之间都存在冲突，执行得到如下图所示的结果：

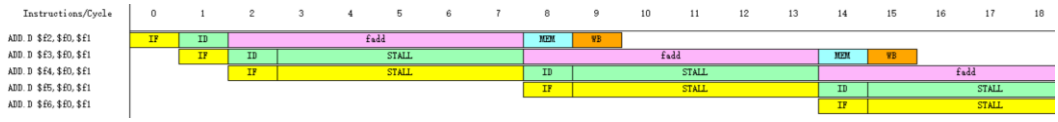

图 2：加法部件冲突下的执行结果

```
汇总：
  执行周期总数：52
  ID段执行了10条指令

硬件配置：
  内存容量：4096 B
  加法器个数：1                执行时间（周期数）：6
  乘法器个数：1                执行时间（周期数）7
  除法器个数：1                执行时间（周期数）10
  定向机制：不采用

停顿（周期数）：
  RAW停顿：0          占周期总数的百分比：0%
  其中：
    load停顿：0              占所有RAW停顿的百分比：0%
    浮点停顿：0              占所有RAW停顿的百分比：0%
  WAW停顿：0          占周期总数的百分比：0%
  结构停顿：35                占周期总数的百分比：67.30769%
  控制停顿：0          占周期总数的百分比：0%
  自陷停顿：6          占周期总数的百分比：11.53846%
  停顿周期总数：41     占周期总数的百分比：78.84615%
```

图 3：执行信息统计

可以看到，总执行周期为 52，结构冲突导致的停顿周期数为 35，约占总执行周期数的 67.3%。

现在我们将浮点加法器的数量改为 4 个，重复上述步骤，得到如下结果：

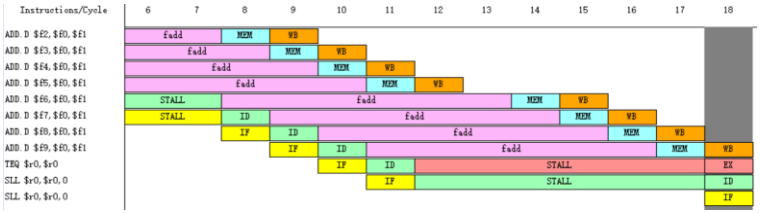

图 4：增加浮点加法器数量后的执行结果

```
汇总：
  执行周期总数：19
  ID段执行了10条指令

硬件配置：
  内存容量：4096 B
  加法器个数：4                执行时间（周期数）：6
  乘法器个数：1                执行时间（周期数）7
  除法器个数：1                执行时间（周期数）10
  定向机制：不采用

停顿（周期数）：
  RAW停顿：0          占周期总数的百分比：0%
  其中：
    load停顿：0              占所有RAW停顿的百分比：0%
    浮点停顿：0              占所有RAW停顿的百分比：0%
  WAW停顿：0          占周期总数的百分比：0%
  结构停顿：2         占周期总数的百分比：10.52632%
  控制停顿：0          占周期总数的百分比：0%
  自陷停顿：6          占周期总数的百分比：31.57895%
  停顿周期总数：8      占周期总数的百分比：42.10526%
```

图 5：执行信息统计

可以看到，总执行周期为 19，结构冲突导致的停顿周期数下降为 2，约占总执行周期数的 10.5%。

现在来分析结构冲突对 CPU 性能的影响：**结构冲突降低 CPU 性能、增加指令执行时间，可以通过增加功能部件的数量来解决结构冲突。**

下面我们观察数据冲突并用定向技术来减少停顿，首先关闭定向功能，然后载入示例程序中的 data_hz.s 文件。通过单步执行一个周期，我们可以发现最早在第 4 个周期发生了 RAW 冲突，然后发生 RAW 冲突的周期为：5、6、8、9、10、12、13、16、17、19、20、24、25、27、28、31、32、35、36、38、39 周期。

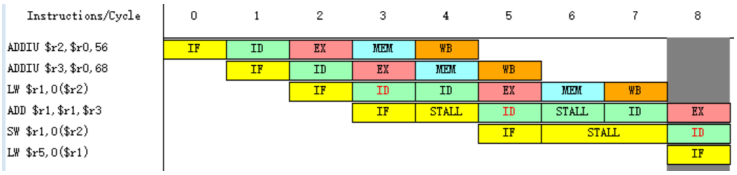

图 6：发生 RAW 冲突的示意图

程序执行完毕，部分统计信息如下：

```
汇总：
  执行周期总数：65
  ID段执行了29条指令

硬件配置：
  内存容量：4096 B
  加法器个数：1                执行时间（周期数）：6
  乘法器个数：1                执行时间（周期数）7
  除法器个数：1                执行时间（周期数）10
  定向机制：不采用

停顿（周期数）：
  RAW停顿：31          占周期总数的百分比：47.69231%
  其中：
    load停顿：12              占所有RAW停顿的百分比：38.70968%
    浮点停顿：0               占所有RAW停顿的百分比：0%
  WAW停顿：0           占周期总数的百分比：0%
  结构停顿：0          占周期总数的百分比：0%
  控制停顿：3          占周期总数的百分比：4.615385%
  自陷停顿：1          占周期总数的百分比：1.538462%
  停顿周期总数：35     占周期总数的百分比：53.84615%
```

图 7：执行信息统计

可以看到，总执行周期为 65，RAW 停顿为 31 个周期，约占周期总数的 38.7%。

接下来复位 CPU，打开定向功能，单步执行一个周期，执行结果如下所示：

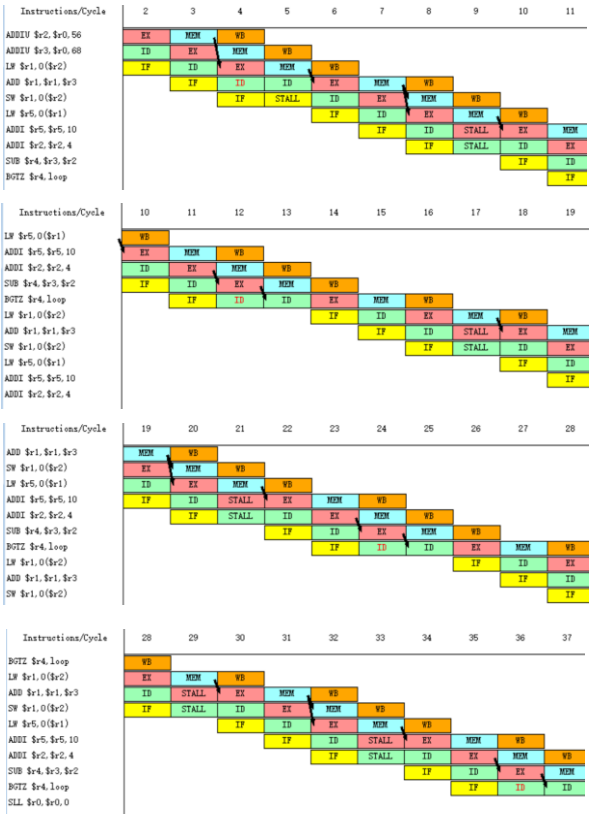

图 8：开启定向功能的执行结果

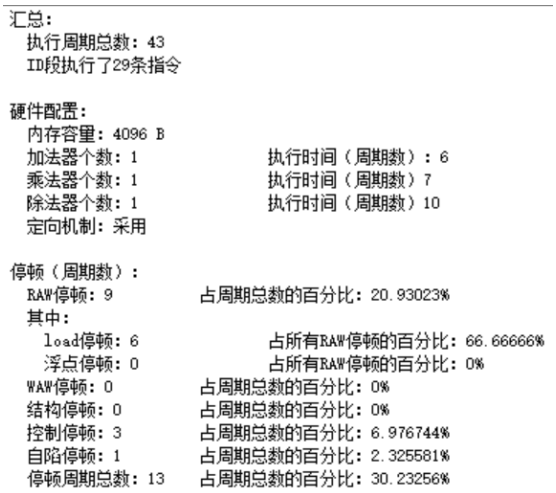

图 9: 执行信息统计

其中发生 RAW 冲突的周期为 4、9、12、17、21、24、29、33、36，总执行周期为 43，RAW 停顿周期为 9，约占总周期数的 20.9%。并且，采用定向技术后，性能提高了 65÷43=1.512 倍。

五、实验结果分析与总结

在本次实验中，我学习了 MIPSsim 模拟器的基本使用方法，圆满完成了实验二既定的实验任务。我熟悉了流水线及流水线冲突机制，并加深了对冲突处理方法的理解。我详细分析了实验二中程序的执行行为，收获颇丰！