

## Windlx模拟器与流水线相关

A.用WinDLX模拟器执行下列三个程序：

- 求阶乘程序fact.s
- 求最大公约数程序gcm.s
- 求素数程序prim.s

分别以步进、连续、设置断点的方式运行程序，观察程序在流水线中的执行情况，观察CPU中寄存器和存储器的内容。

注意：**fact.s**中调用了**input.s**中的输入子程序。**load**程序时，要两个程序一起装入（都**select**后再点击**load**）。**gcm.s**也是如此。

### Q：阐述程序的作用和运行原理

#### • gcm.s

这个程序从标准输入读入两个整数，求它们的**Greatest common measure**，然后将结果写到标准输出。该程序中调用了**input.s**中的输入子程序。

#### • fact.s

求阶乘程序。这个程序说明浮点指令的使用。

该程序从标准输入读入一个整数，求其阶乘，然后将结果输出。调用了输入子程序。

#### • prim.s

求素数程序。这个程序计算若干个整数的素数。

### Q：跟踪CPU中寄存器和存储器内容，把造成其内容发生变化的指令、操作涵义、变化情况（以截图体现）体现出来。

寄存器和存储内容变化情况：

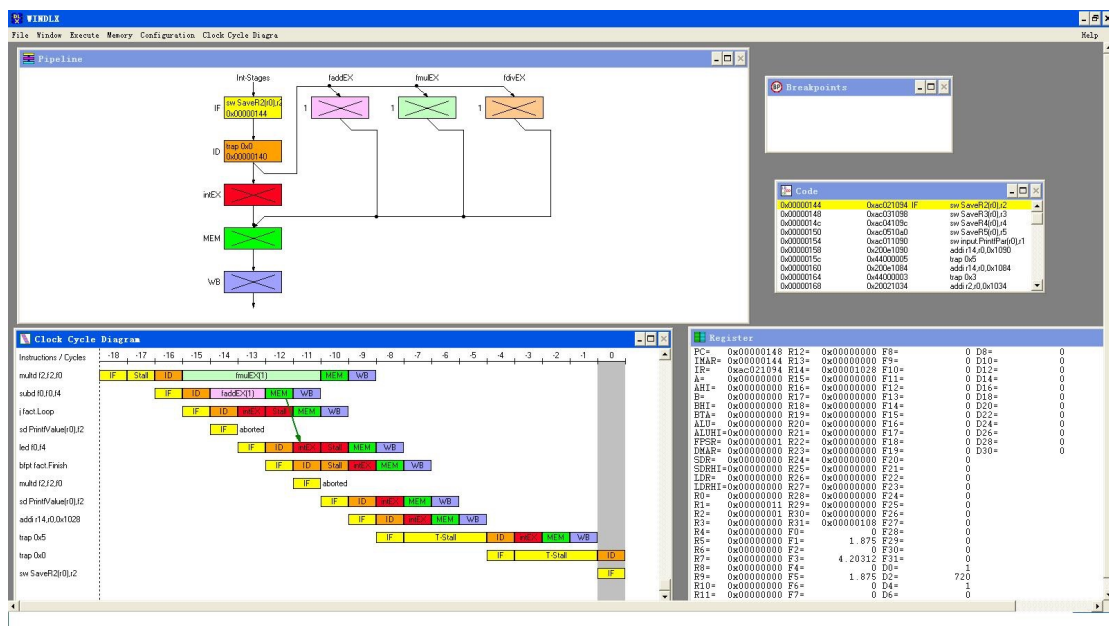


图1—连续执行fact.s

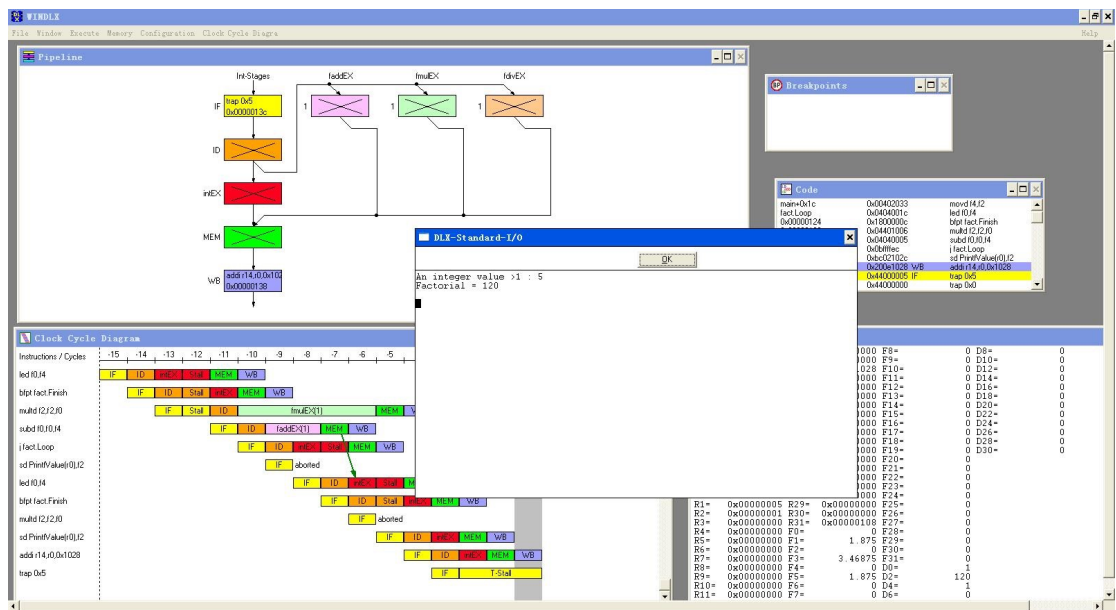


图2—步进执行fact.s

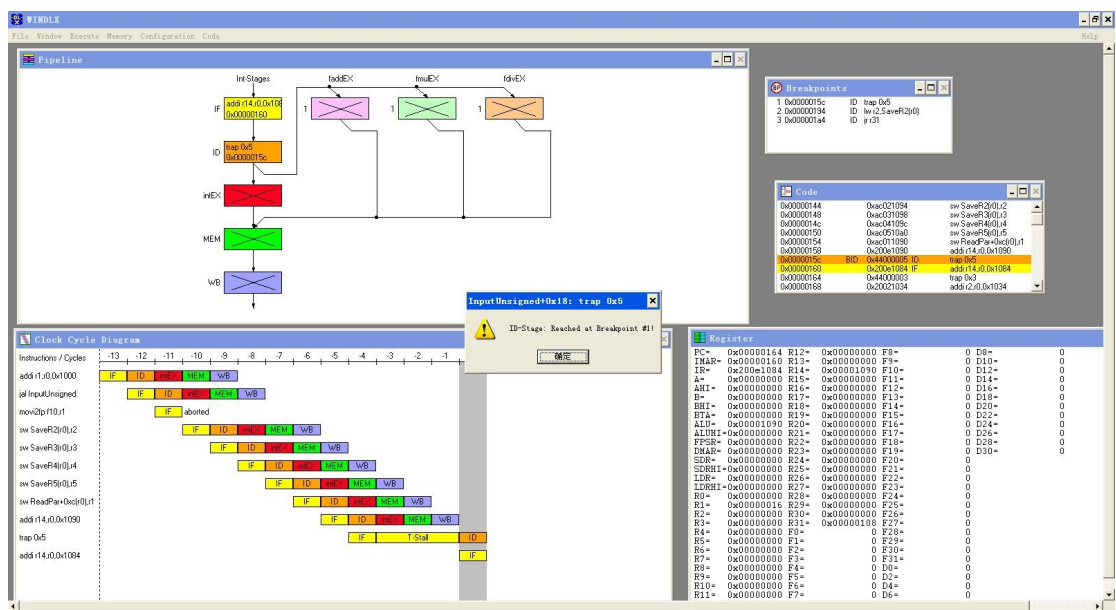


图3—断点执行fact.s（设置了三个断点，这是第一个）

对于fact.s：除去写寄存器造成流水线停顿之外，其他指令都会造成pc或寄存器或alu的变化。执行计算阶乘的运算是通过loop来进行跳转的，需要从标准输入输入一个数，才开始计算。

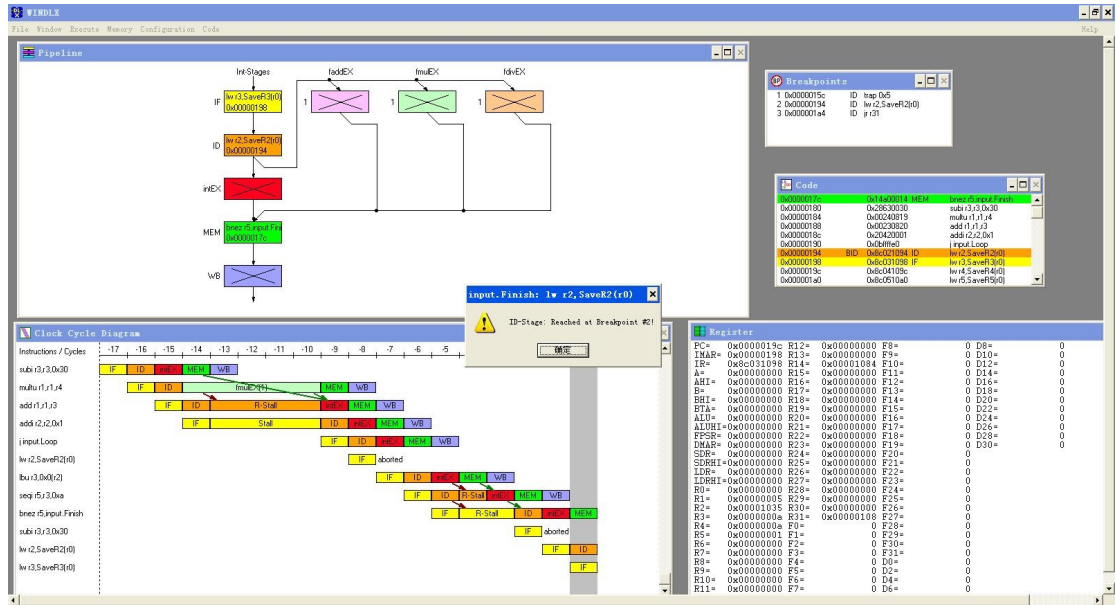


图4—断点执行fact.s（设置了三个断点，这是第二个）

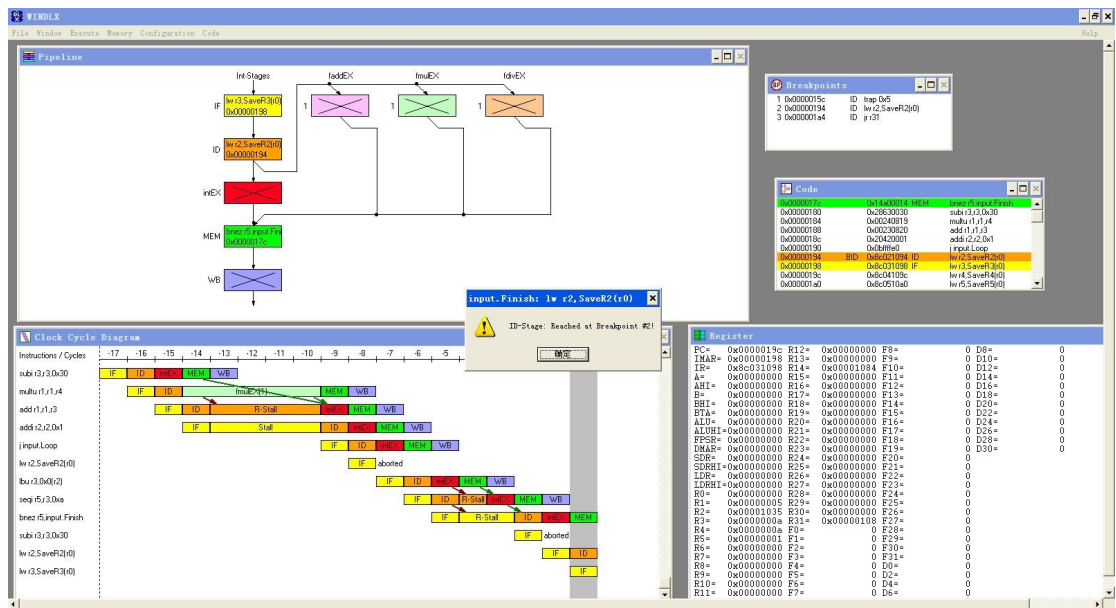


图5—断点执行fact.s（设置了三个断点，这是第三个）

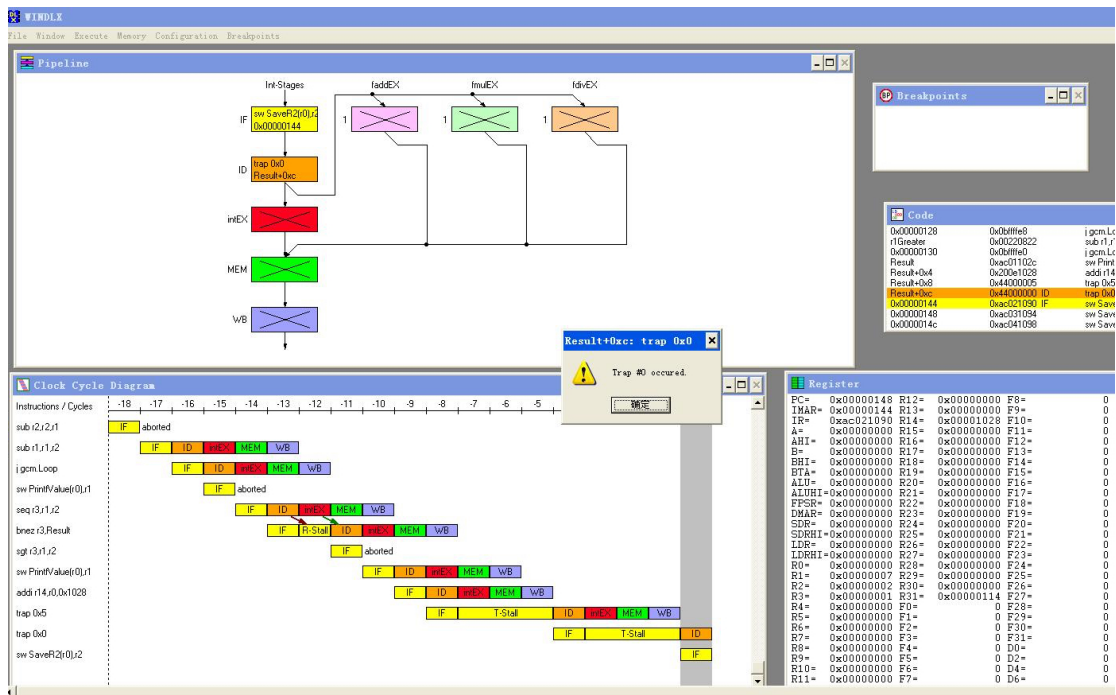


图6—连续执行gcm.s

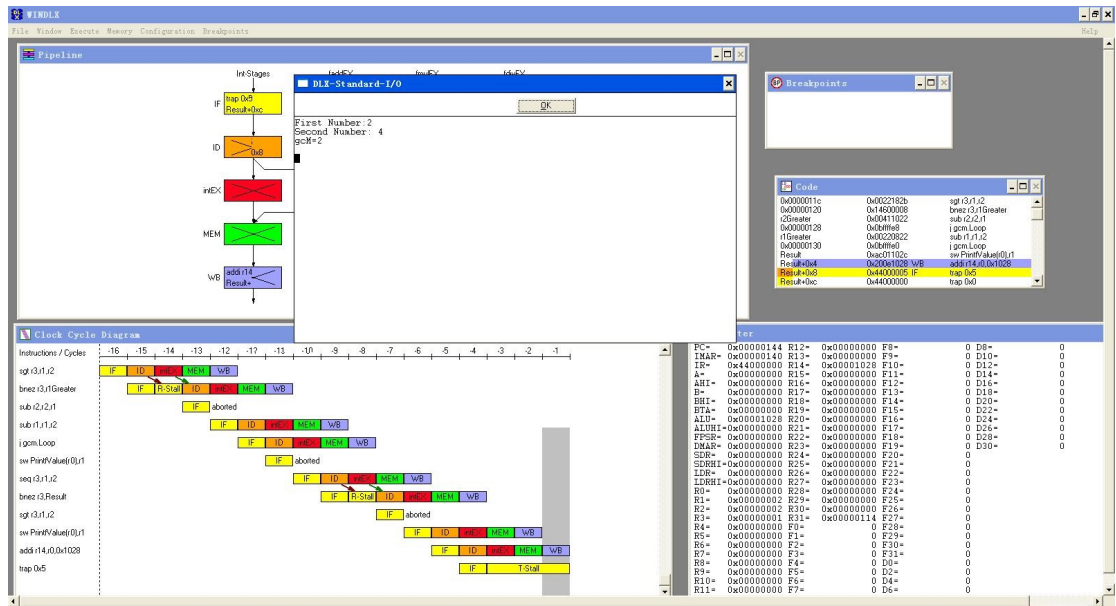


图7—步进执行gcm.s

对于gcm.s: 与fact.s同理，除去写寄存器造成停顿之外之外，都会造成变化。gcm需要从标准输入两个数，且停顿时间较长。

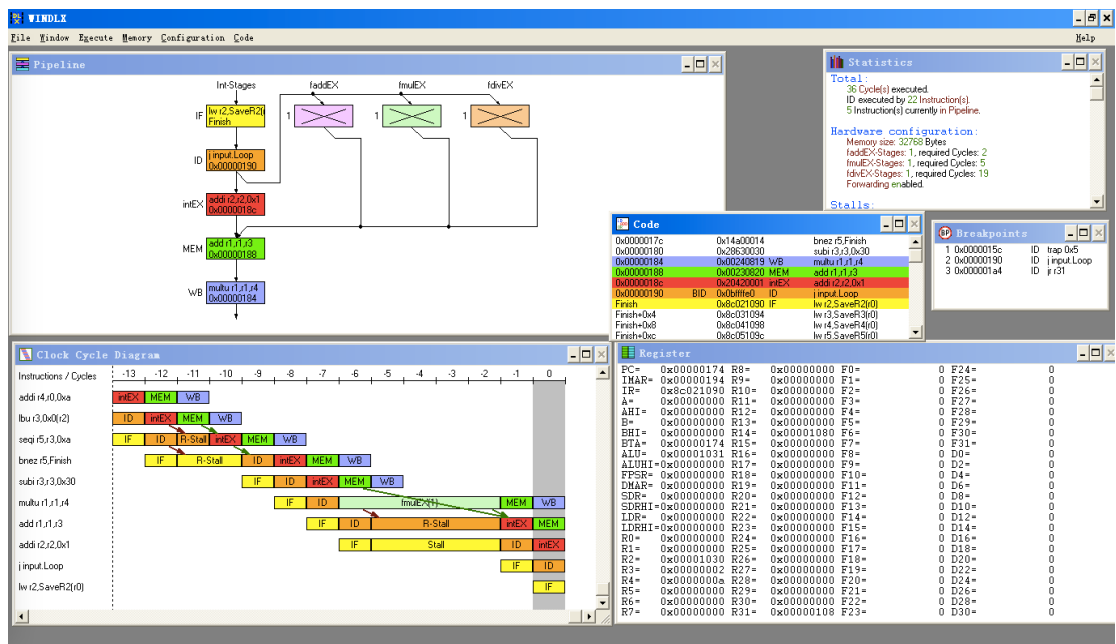


图8—断点执行gcm.s（第二个断点后）

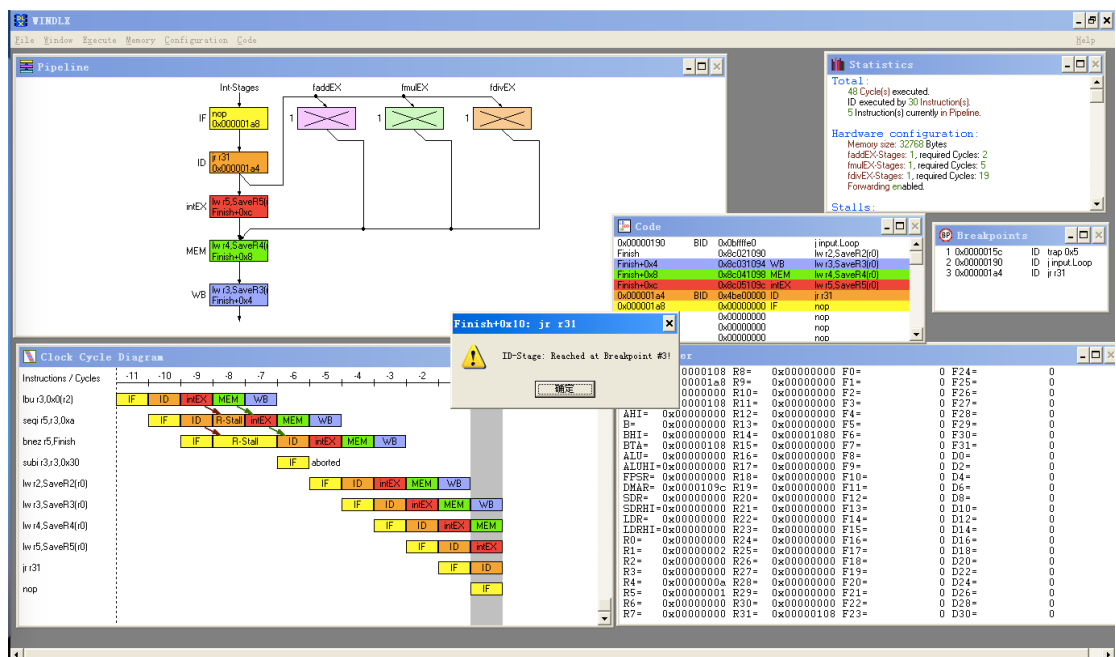


图9—断点执行gcm.s（第三个断点）

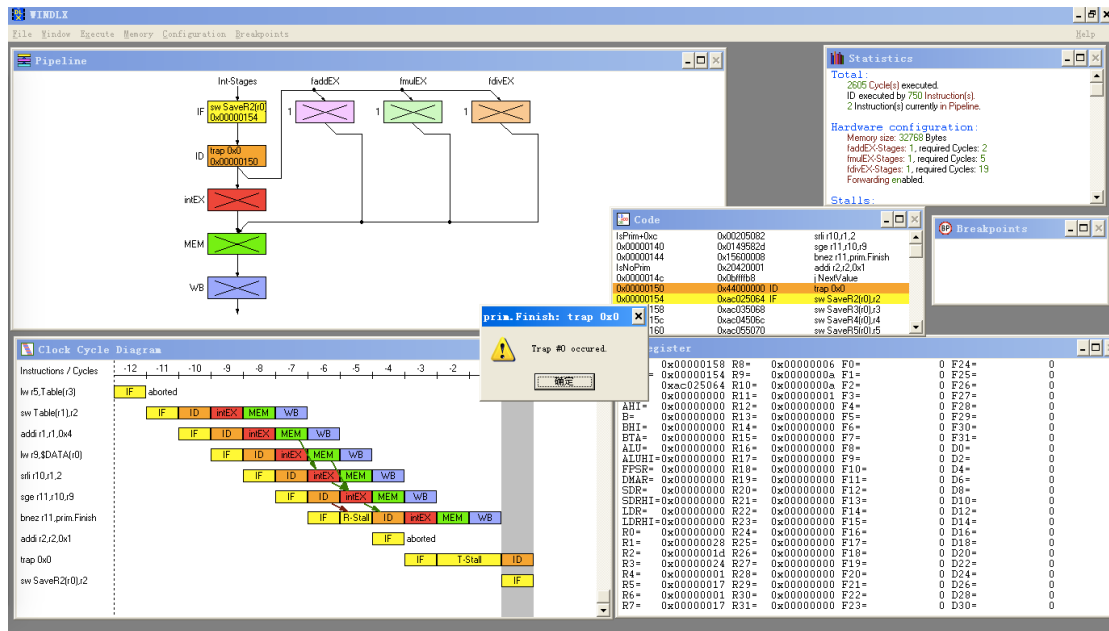


图9—连续执行prim.s

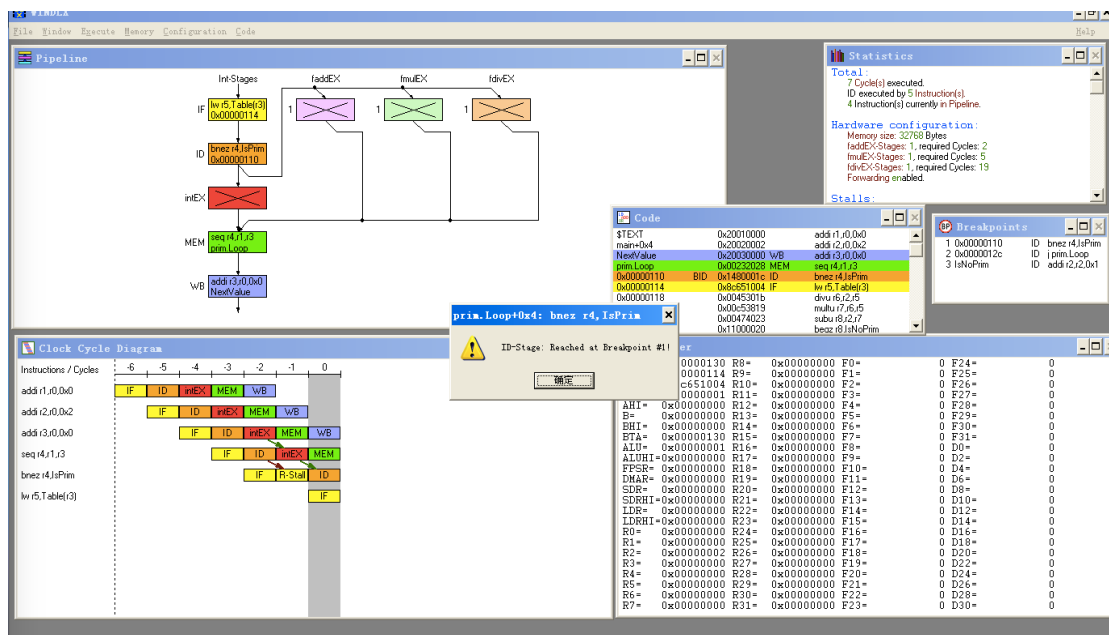


图10—步进执行prim.s

对于prim.s：造成变化的指令有写寄存器指令。



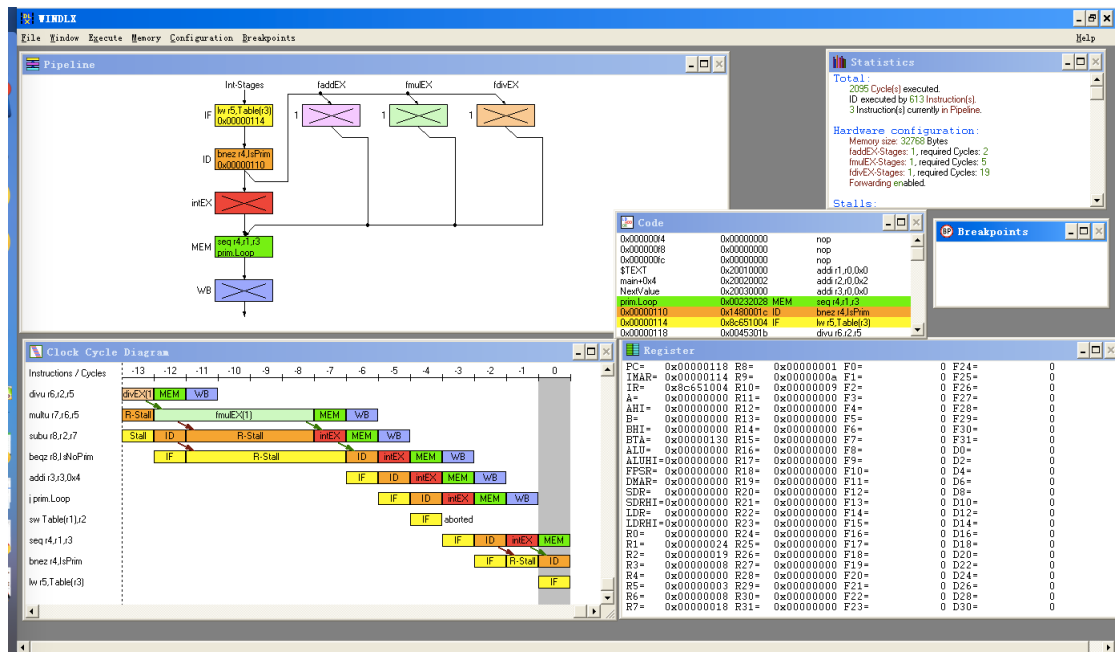


图11—断点执行prim.s（三个断点中的第一个）

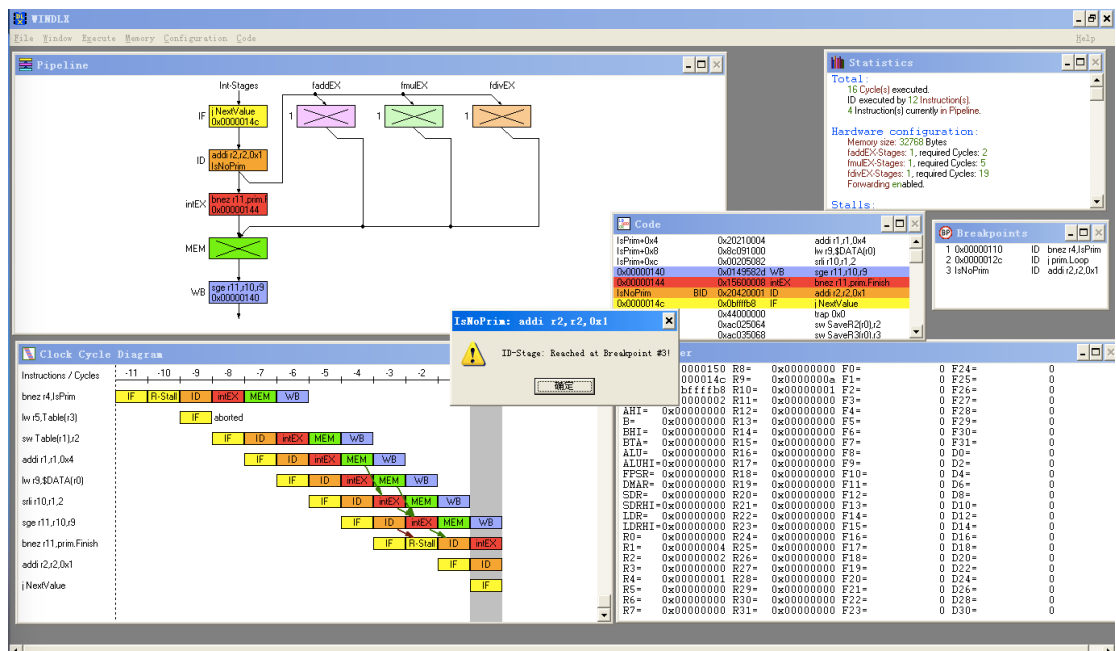


图12—断点执行prim.s（三个断点中的第三个）

**B. 用WinDLX运行程序structure\_d.s，通过模拟：**

- 找出存在结构相关的指令对以及导致结构相关的部件；
- 记录由结构相关引起的暂停时钟周期数，计算暂停时钟周期数占总执行周期数的百分比；
- 论述结构相关对CPU性能的影响，讨论解决结构相关的方法。

**Q：分析并找出全部结构相关的指令、原因、流水线停顿截图、停顿时钟周期数（循环的只写一轮）**

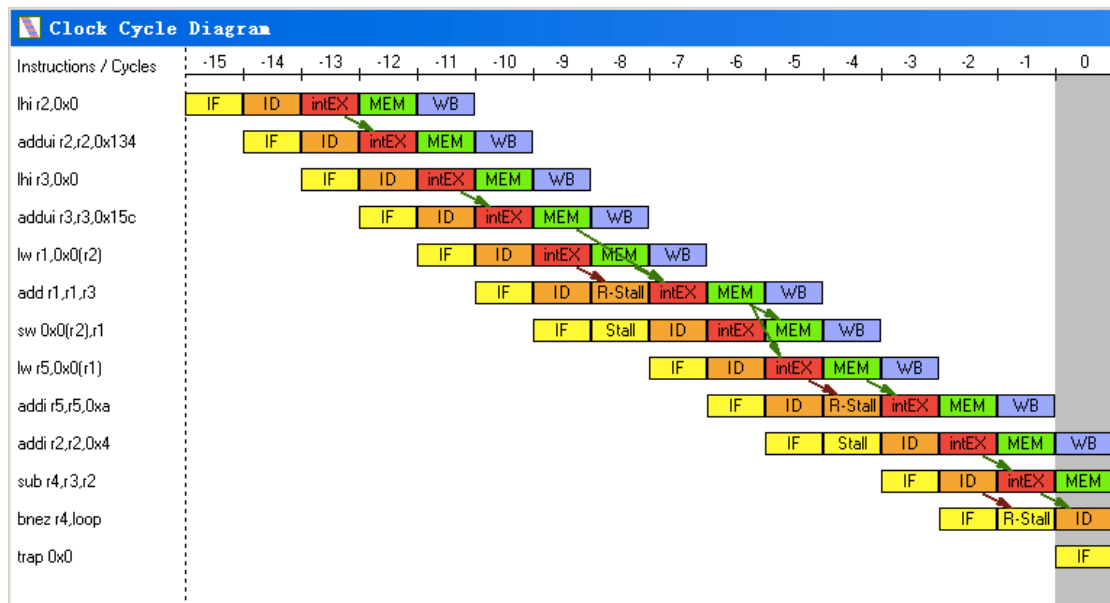


图13--一轮流水线

解：如图所示，stall为流水线停顿。

有两种情况：

一、浮点数寄存器r3。add r1,r1,r3 指令为指令在译码阶段ID停滞1周期，r3此时还未被回写存储器。

二、ALU。addi r2,r2,0x4 在执行阶段intEX停滞1周期，此时前面的ALU操作还未完成。

一共有4次停顿，如图所示：



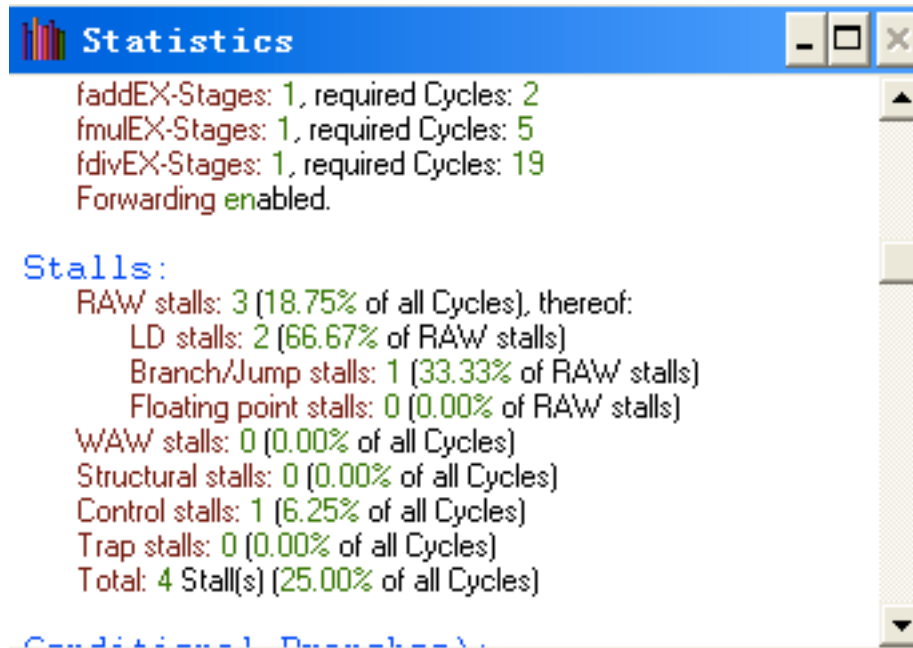


图14—流水线停顿数据

**Q：分析并找出导致结构相关的部件**

解：ALU、r3

**Q：记录由结构相关引起的暂停时钟周期数、总执行周期数、计算暂停时钟周期数占总执行周期数的百分比**

解：由上图可得，在一次循环内，由结构引起了2次时钟周期停顿，总执行周期为16，所占百分比为12.50%。

**Q：论述结构相关对CPU性能的影响，讨论解决结构相关的方法**

解：结构相关会造成CPU性能降低，在合理的指令调度范围内，需要尽量避免执行重复的指令，此外，可以使用寄存器换名的方法。

C. 在不采用定向技术的情况下（去掉Configuration菜单中Enable Forwarding选项前的勾选符），用WinDLX运行程序data\_d.s。记录数据相关引起的暂停时钟周期数以及程序执行的总时钟周期数，计算暂停时钟周期数占总执行周期数的百分比。

- Q: （1）分析并写出全部数据相关的指令、原因、流水线停顿截图、停顿时钟周期数（循环的只写一轮）
- （2）数据相关引起的暂停时钟周期总数、程序执行的总时钟周期数、计算暂停时钟周期数占总执行周期数的百分比

解：如图所示为停顿原因及流水线状态图。

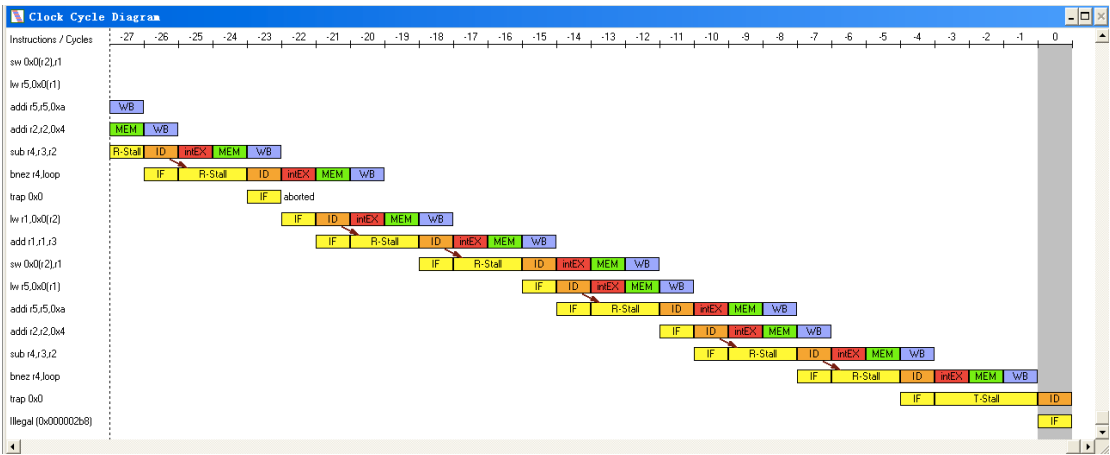


图15—流水线

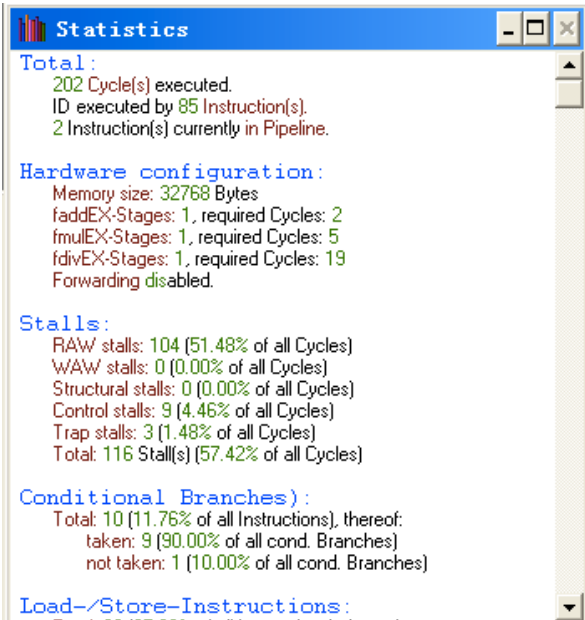


图16—停顿周期数

一共有116次流水线停顿，其中与数据相关有104次，总流水线周期为202，数据相关停顿所占百分比为51.49%

D. 在采用定向技术的情况下（勾选`Enable Forwarding`），用WinDLX再次运行程序 `data_d.s`。重复上述2中的工作，并计算采用定向技术后性能提高的倍数。

Q: （1）分析并写出此时全部数据相关的指令、原因、流水线停顿截图、暂停时钟周期数（循环的只写一轮）  
（2）数据相关引起的暂停时钟周期数、程序执行的总时钟周期数、计算采用定向技术后性能提高的倍数

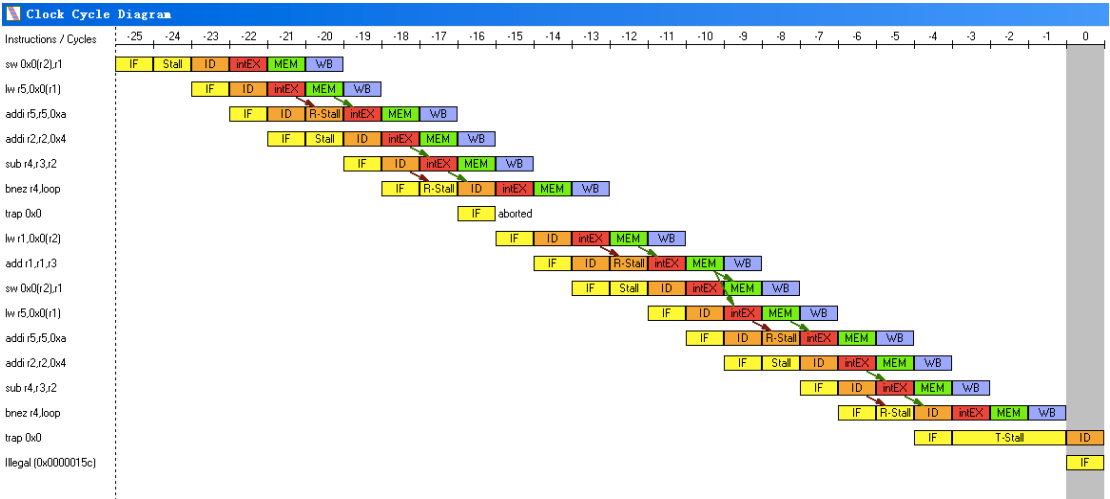


图17—流水线

解：（1）如图所示，`stall`为流水线停顿。  
一、浮点数寄存器r3。`add r1,r1,r3` 指令为指令在译码阶段ID停滞1周期，r3此时还未完成计算。  
二、浮点数寄存器r5。`addi r5,r5,0xa` 在执行阶段intEX停滞1周期，还未完成数据计算。  
三、浮点数寄存器r4。`bnez r4,loop` ,此时r4的数值还未被计算出来。  
一共有42次停顿，如图所示：

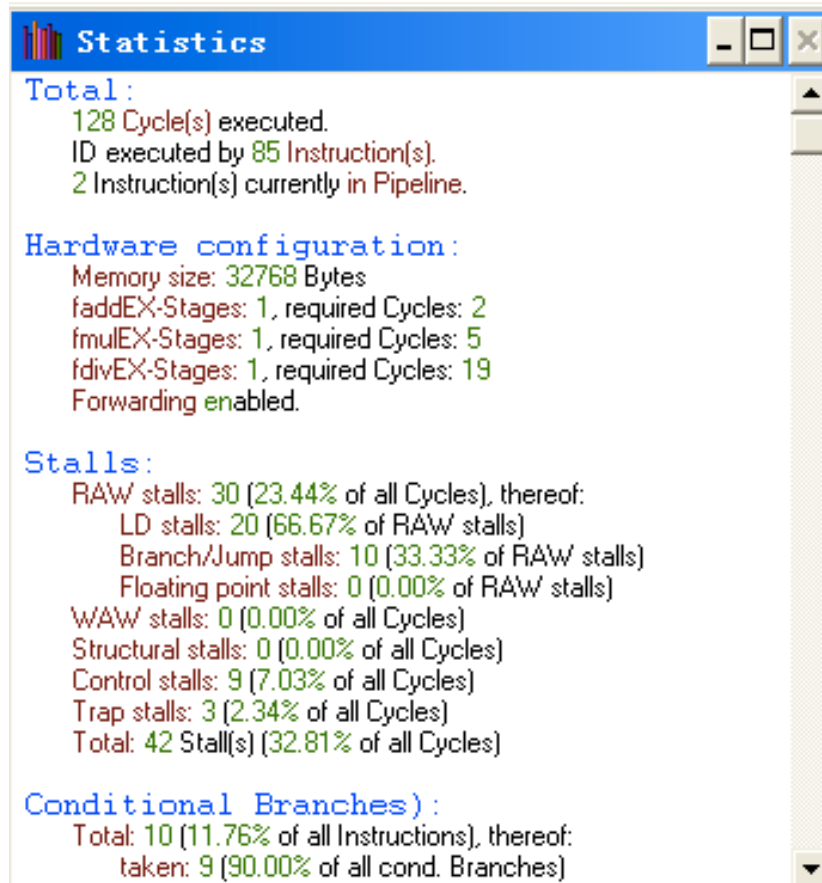


图18 — 停顿相关数据

(2) 由上图可得，在一次循环内，由数据引起了30次时钟周期停顿，总执行周期为128个时钟周期，所占百分比为23.44%。

采用定向技术提高的倍数为： $128/202 = 63.37\%$