计算机科学与技术学院

《现代计算机体系结构》课程实验报告

学 号

班 级

专 业

学生姓名

2020年12月28日

**实 验 报 告**

|  |  |  |
| --- | --- | --- |
| **学生姓名** | **学 号** | **时间：**    **地点：** |
| **实验课程名称：** | | |
| **一、实验名称：**已知密度和深度，求浮力 | | |
| **二、实验原理：**  1、WinDLX平台与流水线  WinDLX模拟器是一个图形化、交互式的DLX流水线模拟器，它采取伪汇编形式编码，模拟流水线的工作方式，能够演示DLX流水线是如何工作的。流水线的指令执行分为5个阶段：取指、译码、执行、访存、写回。  WinDLX模拟器还提供了对流水线操作的统计功能，便于对流水线进行性能分析。  2、流水线中的相关及解决办法  （1）结构相关：当某一条机器指令需要访问物理器件时，该器件可能正在被占用，例如连续的两条加法指令都需要用到浮点加法器，就产生结构相关，可以通过增加加法器的方式解决结构相关；  （2）数据相关：当某一条指令需要访问某个寄存器时，此时这个寄存器正被另一条指令所使用，从而产生数据相关，可以通过重定向技术解决数据相关；  （3）控制相关：当程序执行到某个循环语句时，顺序执行的下一条语句将被跳继续执行循环体的内容，从而产生控制相关，可以通过循环展开解决控制相关。 | | |
| **三、实验目的：**  1、加深对流水线理论知识的理解；  2、掌握对流水线性能分析的方法，了解影响流水线效率的因素；  3、熟悉在WinDLX体系结构下的汇编代码编写和优化；  4、了解相关的类型及各类相关的解决办法；  5、培养运用所学知识解决实际问题的能力。 | | |
| 1. **实验内容：**   1、根据WinDLX模拟器伪汇编指令规则编写已知密度和深度，求浮力的程序  2、根据公式F=ρgh来进行计算  3、观察程序中出现的数据、控制、结构相关，指出程序中出现上述现象的指令组合，并提出解决相关的办法；  4、分别考察各类解决的相关办法，分析解决相关后性能的变化。  注意： 除解决结构相关，其他情况下加、乘、除运算器都只有一个。 | | |
| **五、实验器材（设备、元器件）：**  电脑一台  VMware Workstation  虚拟机（WindowsXP 32位操作系统）  WinDLX模拟器 | | |
| **六、实验步骤及操作：**  1、初始化WinDLX模拟器  （1）为WinDLX创建目录，C:\WinDLX。将WinDLX和homework2.s和InputDouble.s放在这个目录中。  （2）初始化WinDLX模拟器：点击File 菜单中的 Reset all 菜单项，弹出一个“Reset DLX”对话框，点击窗口中的“确认”按钮即可。如图1所示。    图1 初始化模拟器界面  2、将程序装入WinDLX平台  点击File 菜单中的 Load Code or Data 菜单项，依次双homework2.s和InputDouble.s。点击load，将两个程序装入。如图2所示。    图2 程序装入界面  3、运行程序并观察  进行单步调试，在WinDLX模拟器的6个子窗口观察程序的执行情况。观察程序运行的总时钟周期，产生的相关种类以及每种相关的数量。  图3 运行结果截图  4、解决数据相关  勾选Enable Forwading，采用重定向技术添加专用数据通路减少数据相关，观察数据相关的数量变化。  5、解决结构相关  实验中没有结构相关。  6、解决控制相关  将homework2.s和InputDouble.s中，multd f4,f4,f8和multd f4,f2,f4选择推迟，即将第二天指令在不影响程序正常允许的情况下，进行一个推迟，先插入别的语句进行执行，然后再执行第二条指令 | | |
| **七、实验数据及结果分析：**  1、程序装入顺序对运行结果的影响  先装入homework2.s和InputDouble.s时，程序能够正确执行；当先装入InputDouble.s再装入homework2.时，因为InputDouble.s的地址高，而程序顺序执行到InputDouble.s时无法正确的输出，不会出现结果。  2、主要代码及说明  .data  Prompt: .asciiz "please input p:\n\n"  Prompt2: .asciiz " please input h: "  Prompt4: .asciiz "\n\nresult: "  PrintfFormat: .asciiz "%lf\n\n"  Prompt1: .asciiz "p: "  Prompt3: .asciiz "g=: "  .align 4  PrintfPar: .word PrintfFormat  PrintfValue: .space 8  PrintfPar2: .space 4  .text  .global main  main:  addi r1,r0,Prompt ;输出语句Prompt  sw PrintfPar2,r1  addi r14,r0,PrintfPar2  trap 5  ;第一个操作数  addi r1,r0,Prompt1  jal InputDouble  movd f8,f2 ; 输出语句Prompt1，并且获取ρ的值  ;第二个操作数  addi r1,r0,Prompt2 ; 输出语句Prompt2，并且获取h的值  jal InputDouble  movd f4,f2  ;第三个操作数  addi r1,r0,Prompt3 ;获取g的值  jal InputDouble  movd f6,f2  ;运算环节  multd f4,f4,f8 ;f4、f8两数相乘结果存入f4 ρ\*h  multd f4,f4,f6 ;f4继续乘以g  multd f4,f4,f10 ;将ρ\*g\*h放入f4寄存器  ;格式化为标准输出  addi r1,r0,Prompt4 ;  sw PrintfPar2,r1 ;  addi r14,r0,PrintfPar2 ;  trap 5 ;  sd PrintfValue,f4 ;将得到的浮力进行保存和输出  addi r14,r0,PrintfPar ;  trap 5  3、程序分析及运行结果  （1）根据提示输入一个正数ρ。  测试用例：ρ=2.0，h=3，g=9.8    图4 运行结果截图  由图4s可知运行结果为58.8，计算结果正确。  （2）点击Statistics窗口，查看程序执行的时钟周期以及数据相关、结构相关、控制相关的发生次数。  程序执行共用329个时钟周期，数据相关发生91次，结构相关发生0次，控制相关发生16次，如图5所示。    图5 时钟周期和相关数据截图   1. 数据相关及解决   （1）数据相关产生的原因  addi r1,r0,Prompt  sw PromptPar,r1  sw指令要使用r1寄存器的数据，但是上一条指令刚刚执行完数据还没有更新，产生数据相关，如图6所示。    图6 数据相关截图  （2）数据相关的解决  采用重定向技术，勾选Configuration的Enable Forwading选项。在第一条指令结束后直接将寄存器r1的内容更新，消除数据相关，  查看运行结果，数据相关数量降低，数据相关个数为62，如图7所示。    图7 解决数据相关的数据截图   1. 控制相关及解决 2. 控制相关的产生原因   addi r1,r0,Prompt1  jal InputDouble  movd f8,f2  出现了跳转语句，产生控制相关，如图8  图8 控制相关的数据截图   1. 控制相关的解决   采用循环展开的方式将循环体的内容复制一次。可以降低控制相关的个数。如下：  将InputDouble的函数展开，而不是进行跳转  重新运行后控制相关数量减少为2，如图9所示。    图9 解决控制相关的数据截图   1. 结构相关及解决   没有结构  7、程序流程图    图10 程序流程图 | | |
| **八、实验结论：**  1、通过采用重定向技术减少了数据相关；  2、通过减少子函数的调用来减少控制相关；  3、执行程序的顺序会影响程序执行是否正确，必须先执行homework2.s，再执行Inputdouble.s；修改后的程序必须清空之前所有的操作之后再重新运行。 | | |
| 1. **总结及心得体会：** | | |
| **报告评分：**  **指导教师签字：** | | |