山东大学 计算机科学与技术 学院

计算机体系结构 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号： | 姓名： | | 班级： |
| 实验题目：实验三. 用 WinDLX 模拟器完成求素数程序 | | | |
| 实验学时：4 | | 实验日期： 2025.5.19 | |
| 实验目的：  通过实验，熟练掌握 WINDLX 的操作方法，特别注意在单步执行 WinDLX 程序中，流水线  中指令的节拍数。 | | | |
| 硬件环境：  CPU: Intel(R) Core(TM) i5-8400 CPU @ 2.80GHz 2.81 GHz  RAM: 24.0 GB | | | |
| 软件环境：  Windows 11 专业版  Ovtdm 0.9.0  WinDLX | | | |
| 实验步骤与内容：  （1） 用 WinDLX 模拟器执行求素数程序 prim.s。这个程序计算若干个整数的素数。  （2） 单步执行两轮程序，求出素数 2 和3。  （3） 在执行程序过程中，注意体验单步执行除法和乘法指令的节拍数，并和主菜单  configuration/floating point slages 中的各指令执行拍数进行比较。  单步运行到 divu 语句，可以看到下方占用了非常长的周期：    可以看到 fdiv 执行了19个时钟周期，和 Floating Point Stage 设置的数字一致：    运行计算出前两个素数：    可以看到 $DATA 中第二、三个位置存储了2、3，表明计算出了2、3两个素数。  main:  ; 初始化  addi r1, r0, 0 ; r1 = 表格索引（初始0）  addi r2, r0, 2 ; r2 = 当前待检测数值（初始2）  NextValue:  ; 检查r2是否能被已有素数整除  addi r3, r0, 0 ; r3 = 表格遍历索引  Loop:  seq r4, r1, r3 ; r4 = (r1 == r3) ? 1 : 0  bnez r4, IsPrim ; 如果索引已遍历完，说明是素数  lw r5, Table(r3) ; 读取当前素数  divu r6, r2, r5 ; r6 = r2 ÷ r5 的商  multu r7, r6, r5 ; r7 = 商 × r5（计算余数的乘积）  subu r8, r2, r7 ; r8 = r2 - r7（余数）  beqz r8, IsNoPrim ; 如果余数为0，不是素数  addi r3, r3, 4 ; 移动到下一个素数  j Loop  IsPrim:  ; 存储新素数  sw Table(r1), r2 ; 将r2存入表格  addi r1, r1, 4 ; 更新表格索引  ; 检查是否达到指定数量  lw r9, Count  srli r10, r1, 2 ; r10 = 已存储素数数量  sge r11, r10, r9 ; r11 = (r10 >= Count) ? 1 : 0  bnez r11, Finish  IsNoPrim:  addi r2, r2, 1 ; 检查下一个数值  j NextValue | | | |
| 结论分析与体会：  通过WinDLX模拟器单步执行求素数程序，直观观察到素数检测中除法指令divu占用19个时钟周期的长延迟特性，验证了程序正确性，成功储存素数。实验加深了对流水线指令节拍配置的理解，同时通过跟踪寄存器与内存状态变化，强化了对汇编程序执行流程及数据依赖关系的掌握。 | | | |