**成都信息工程大学计算机学院**

**课**

**程**

**实**

**验**

**报**

**告**

|  |  |
| --- | --- |
| **实验课程：** | **计算机体系结构** |
| **实验项目：** | **单功能流水线调度机构模拟** |
| **指导教师：** | **文武** |
| **学生姓名:** | **XXX** |
| **学生学号：** | **XXXXXXXXXX** |
| **班 级：** | **计科XXX** |
| **实验地点：** | **B213** |
| **实验时间：** | **2024年4 月20日 8:30 ~ 12:00** |
| **实验成绩：** |  |

**！！！报告抄袭和被抄袭者一律0分**

**一【上机实验内容】**

通过模拟单功能流水线调度过程，掌握流水线技术，学会计算流水线的吞吐率、 加速比、效率。

1 流水线的表示法有三种： 连接图、时空图、预约表。对于线性流水线，主要考虑前二种。

2 流水线的主要特点： 在流水线的每一个功能部件的后面都要有一个缓冲器，称为锁存器、闸门寄存器等， 它的作用是保存本流水段的执行结果。各流水段的时间应尽量相等，否则回引起阻塞、 断流等。只有连续提供同类任务才能充分发挥流水线的效率。在流水线的每一个流水线段中都要设置一个流水锁存器。流水线需要有“装入时间”和“排空时间”。只有流水线完 全充满时，整个流水线的效率才能得到充分发挥。

**二【上机实验中的其他它问题及心得】**

**在进行单功能流水线调度机构模拟的实验时，遇到了一系列挑战和问题，同时也能获得宝贵的实践经验和心得。**

**理解流水线概念：流水线(pipeline)的概念以及它是如何提高指令执行效率的需要深入理解。**

**资源分配：在单功能流水线中，合理安排每个阶段的资源是一个难点，如寄存器使用冲突等问题。**

**代码调试：在流水线环境下进行代码调试比传统的顺序执行环境更加复杂。**

**实验心得：**

**加深理解：通过动手实践，可以更深入地理解流水线的原理和它在提高CPU性能中的作用。**

**设计能力提升：提高了系统设计的能力，了解如何平衡速度与成本，理论与实践之间的关系。**

**软件与硬件交互：理解了软件是如何与硬件交互的，以及编译器是如何针对流水线结构进行优化的。**

**批判性思维：学会如何批判性地分析问题，并寻找解决方案，而不是仅仅接受结果。**

**持续学习：认识到学习是一个不断的过程，激发了对更深层次计算机体系结构知识的探索兴趣。**

**通过完成流水线调度机构的模拟实验，不仅能够巩固理论知识，还能在实践中体会到计算机体系结构的精妙之处，为日后深入学习打下坚实的基础。**

**三【源程序、测试数据及结果】**

（对程序添加注释）

|  |
| --- |
| Main.java  **package** HuffmanByzxt.test;  **public class** Main {  **private static final int *SPACE*** = 4;*//功能部件数目* **private static final int *NUM*** = 5;*//需要流水处理的浮点加指令数目* **private static final int *TIME*** = ***NUM*** + ***SPACE*** - 1;*//不同时间段各个部件内的指令* **private static final** String[] ***INSTRUCTIONS*** = {**"规格化"**, **"尾数加"**, **"对阶"**, **"求阶差"**};  **private static int**[][] *ts* = **new int**[***SPACE***][***TIME***];*//流水线时空图* **public static void** main(String[] args) {  System.***out***.println(**"开始流水线\n"**);  *pipeline*(*ts*);*//处理流水线  print*();*//打印流水线* System.***out***.println(**"\n结束流水线\n"**);  System.***out***.println(**"本流水线吞吐率是 "** + (**double**)***NUM***/***TIME*** + **"t"**);  System.***out***.println(**"本流水线的加速比是 "** + ((**double**)***NUM***\****SPACE***)/***TIME***);  System.***out***.println(**"本流水线效率是 "** + ((**double**)***NUM***\****SPACE***)/(***TIME***\****SPACE***));  }   **public static void** print() {  **for** (**int** i = 0; i < ***TIME***; ++i) {  System.***out***.println(**"完成时间片 "**+ (i+1) + **"之后"**);  **for** (**int** j = 0 ; j < ***SPACE***; ++j) {  **if** (i < ***NUM*** && *ts*[j][i] == 0) {  System.***out***.println();  } **else** {  **for** (**int** k = 0; k < i+1; ++k) {  **if** (*ts*[j][k] != 0) {  System.***out***.print(***INSTRUCTIONS***[j]+*ts*[j][k]);  }  System.***out***.print(**"\t"**);  }  System.***out***.println();  }  }  }  }   **public static void** pipeline(**int**[][] ts) {  **int** tempSpace = 0;*//记录处理的指令* **int** tempTime = 0;*//记录时间* **for** (**int** s = ***SPACE***-1; s >= 0; s--) {  tempSpace = 1;  **for** (**int** t = tempTime; t < ***TIME***; t++) {  ts[s][t] = tempSpace++;  }  tempTime++;  }  } } |



