**汇编与接口课程设计**

**实验报告**

**MIPS指令系统与汇编程序设计**

**学号 姓名**

**班级：XXX**

二零一七年九月十八号

**MIPS指令系统与汇编程序设计**

1. **实验目的**
2. 掌握MIPS体系结构与指令系统
3. 熟悉MIPS指令格式以及汇编指令助记符
4. 掌握MIPS汇编语言与机器语言之间的对应关系
5. 学习MIPS汇编程序设计
6. 了解使用MIPS汇编模拟器的方法
7. 完成简单算法汇编程序实现实验(排序等)
8. **实验内容**

将实验素材Fibonacci数列(fibonaccime.asm)在MARS中打开并单步执行，观察各个寄存器的值是否和预期的一致。并将指令编码记录下来，供《计算机组成原理课程设计》中的CPU测试使用。

设计汇编程序，实现将一串数列输入$2-$6，并排序，排序算法自定，如冒泡排序算法等。

1. **实验器材及环境**

操作系统：Windows 10

汇编程序运行环境：MARS

1. **实验原理**

MIPS 是世界上很流行的一种 RISC 处理器。MIPS的意思是“无内部互锁流水级的微处理器”(Microprocessor without interlocked piped stages)，其机制是尽量利用软件办法避免流水线中的数据相关问题。它最早是在80年代初期由斯坦福(Stanford)大学Hennessy教授领导的研究小组研制出来的。MIPS架构的产品多见于工作站领域，而现在则广泛应用在嵌入式系统中，龙芯TM2也采用MIPS架构。

MIPS指令集具有以下特点:

1、简单的 LOAD/STORE 结构。所有的计算类型的指令均从寄存器堆中读取数据并把结果写入寄存器堆中。只有LOAD和STORE指令访问存储器。

2、易于流水线CPU的设计。MIPS32指令集1的指令格式非常规整，所有的指令均为32位，而且指令操作码在固定的位置上。

3、易于编译器的开发。MIPS 指令的寻址方式非常简单，每条指令的操作也非常简单。

MIPS32TM的指令格式只有3种。R(register)类型的指令从寄存器堆中读取两个源操作数，计算结果写回寄存器堆。I(immediate)类型的指令使用一个16 位的立即数作为源操作数。J(jump)类型的指令使用一个26位立即数作为跳转的目标地址(target address)。

指令格式中的op(opcode)是指令操作码。rs(register source)是源操作数的寄存器号。rd(register destination)是目的寄存器号。rt(register target)即可作为源寄存器号，又可作为目的寄存器号，有具体的指令决定。func(function)可被认为是扩展的操作码。sa(shift amount)由移位指令使用，定义移位位数。immediate 是16位立即数，使用之前由指令进行0扩展或符号扩展。26位 target由jump指令使用，用于产生跳转的目标地址。

MIPS32有32个通用寄存器，编号从0到31，其中寄存器0的内容总是0，这些通用寄存器的成为寄存器堆(register file)。MIPS32TM还定义了32个浮点寄存器。另外还有一些通用寄存器，PC(program counter)就是其中的一个，CPU使用它从存储器中取指令。

1. **实验步骤**

## Fibonacci数列

将实验素材Fibonacci数列(fibonaccime.asm)在MARS中打开并单步执行，执行结束后各个寄存器的值如下表：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 寄存器号 | 内容 | 寄存器号 | 内容 | 寄存器号 | 内容 |
| $0 | 0x00000000 | $11 | 0x00000a18 | $22 | 0x00000000 |
| $1 | 0x10010000 | $12 | 0x00001055 | $23 | 0x00000000 |
| $2 | 0x0000000a | $13 | 0x00000014 | $24 | 0x00000000 |
| $3 | 0x00000000 | $14 | 0x00000000 | $25 | 0x00000000 |
| $4 | 0x10010054 | $15 | 0x00000000 | $26 | 0x00000000 |
| $5 | 0x00000014 | $16 | 0x00000000 | $27 | 0x00000000 |
| $6 | 0x00000000 | $17 | 0x00000000 | $28 | 0x10008000 |
| $7 | 0x00000000 | $18 | 0x00000000 | $29 | 0x7fffeffc |
| $8 | 0x10010050 | $19 | 0x00000000 | $30 | 0x00000000 |
| $9 | 0x00000000 | $20 | 0x00000000 | $31 | 0x00400054 |
| $10 | 0x10001a6d | $21 | 0x00000000 |  |  |

## 排序算法

用MIPS语言实现冒泡排序，我首先写了一份C语言的代码，然后根据C语言代码编写了MIPS的冒泡排序，具体代码可参见打包的源码。

我将用户输入的数据存放在了一个数组中，数组最大为80个字节，也就是最多可对20个数字进行排序，用户第一行需要输入排序的数字的数量N，接下去的N行输入需要排序的数字。程序最后会输出从小到大排序完成的数列。

1. **实验总结**

本次实验还是比较简单的，因为上个学期的编译原理课程已经接触过MIPS汇编语言，也用过MARS，所以本次试验没什么问题，很顺利就做完了。

个人感觉MIPS汇编语言和x86汇编语言最大的区别就是通用寄存器很多，通用寄存器用起来比x86方便许多，而且每条指令格式很简单，容易记。MIPS汇编语言的访存也比x86许多方便，只有两条指令能够进行访存。