

杭州电子科技大学

# 计算机组成原理（甲） 实 验 报 告

学 院	网络空间安全学院
专 业	网络工程
班 级	19272401
学 号	19061440
学生姓名	F001
教师姓名	袁理峰
完成日期	2020.12.18
成 绩	

## 实验六 MIPS 汇编器与模拟器实验

### 一、 实验目的

- (1) 学习 MIPS 指令系统，熟悉 MIPS 指令格式及其汇编指令助记符，掌握机器指令编码方法。
- (2) 学习 MIPS 汇编程序设计，学会使用 MIPS 汇编器将汇编语言程序翻译成二进制文件。
- (3) 了解使用 MIPS 教学系统模拟器运行程序的方法。

### 二、 实验原理

计算机硬件只能识别 0、1 代码，因为电子器件只能在电信号下才能工作，能让及其理解的最简单的信号就是“开”和“关”信号。计算机硬件设计者设计出的计算机，对计算机设计者的命令言听计从，计算机设计者的“言”就是机器指令，“命令”就是机器指令。换言之，机器指令就是能被计算机硬件识别并直接执行的 0、1 代码串。

使用 0、1 代码编写机器语言程序，不仅烦琐枯燥，而且要求程序员对机器指令格式和硬件了如指掌。为了将机器指令用人类容易理解的符号来表达，出现了汇编语言——用助记符来表示机器指令，这种指令又叫汇编指令。显而易见，汇编语言程序虽然易于程序员理解，但是计算机硬件并不能识别。因此，需要将汇编语言程序翻译成计算机语言程序，能完成这种翻译功能的程序叫汇编程序，又叫汇编器。汇编程序将用符号表示的汇编指令翻译成二进制的机器指令。

虽然汇编语言对程序员来说是一个巨大的进步，但是他仍然是面向机器的语言，程序员需要按照机器的思维进行变成。高级程序设计语言的出现突破了汇编语言的局限性。高级语言更接近人类的思维方式，他是面向用户的自然语言，使得程序员的编程效率更加高效，并且可以执行使得程序独立于开发环境。高级语言程序通过编译器翻译成为二进制的机器语言程序。

### 三、 实验环境

所用电脑的软硬件配置：自己的笔记本电脑、Windows10 操作系统

实验所用的软件：PCSpim

### 四、 主要操作步骤及实验结果记录（不能光截图，要有相应的文字说明）

（对实验过程中的主要操作步骤进行描述，并随时记录实验过程中观察到的结果，必要时可辅助截图）

任务一：使用记事本程序或者任何纯文本编辑器编辑 test.asm 文件，输入相关内容。

（编程五分钟，注释两小时==）



R0 (r0) = 00000000	R8 (t0) = 00007000	R16 (s0) = ffffffff0	R24 (t8) = 00000000
R1 (at) = ffffffff	R9 (t1) = ffff8fff	R17 (s1) = 00000010	R25 (t9) = 00000000
R2 (v0) = 00000001	R10 (t2) = ffff8ffe	R18 (s2) = 70000000	R26 (k0) = 00000000
R3 (v1) = 00000002	R11 (t3) = 00006ff2	R19 (s3) = 00020000	R27 (k1) = 00000000
R4 (a0) = 00000003	R12 (t4) = ffff900e	R20 (s4) = 80000000	R28 (gp) = 00000000
R5 (a1) = 00000005	R13 (t5) = ffff800e	R21 (s5) = 00000000	R29 (sp) = 7ffff664
R6 (a2) = 00000007	R14 (t6) = ffff9fff	R22 (s6) = 00000000	R30 (s8) = 00000000
R7 (a3) = 0000000e	R15 (t7) = 0000000f	R23 (s7) = 00000000	R31 (ra) = 00000000

这是全部的处理结果，当文件执行到以下这一行时出现了错误：

```
0x00400078] 0x0281a820 add $21, $20, $1 ; 22: add $21,$20,$1; #21=7FFF_FFFF
```

经分析得，r1 已经是 ffffffff 了，此时再进行加法操作，会溢出，因此报错，自动跳转到 kernel 中进行计算。

## 五、 实验分析总结及心得

（结合所学知识对实验过程中观察到的实验结果进行分析总结，以便加深对知识的理解，并总结通过实验学到的知识或技术）

我认为，之前学的 ISE 对于实际理解计算机组成原理的帮助不大。直到这一章的 PCSpim，才算是有意思了不少。汇编语言可以说是计算机的灵魂，它是人和机器沟通的最快的桥梁。因此，对于 PCSpim 的学习尤为必要。

在这章中，我查阅了相关博客，了解了每一条指令的原理以及应用方式。对于实验中产生的奇奇怪怪的现象，也进行了相应的错误分析与调试。我认为这样是最好的学习方式，我也希望有机会一直学习汇编语言。在实验结果上，两个实验都不算成功。第一个实验错误位置，但是表现在分支结构后的 move 语句出现了问题，查阅后是某个参数配置错误，但是目前尚未找到解决办法。第二个实验则是给定的程序文件存在问题。出现了溢出的错误。因此之后的程序直接被忽略。

R0 (r0) = 00000000	R8 (t0) = 00007000	R16 (s0) = ffffffff0	R24 (t8) = 00000000
R1 (at) = ffffffff	R9 (t1) = ffff8fff	R17 (s1) = 00000010	R25 (t9) = 00000000
R2 (v0) = 00000001	R10 (t2) = ffff8ffe	R18 (s2) = 70000000	R26 (k0) = 00000000
R3 (v1) = 00000002	R11 (t3) = 00006ff2	R19 (s3) = 00020000	R27 (k1) = 00000000
R4 (a0) = 00000003	R12 (t4) = ffff900e	R20 (s4) = 80000000	R28 (gp) = 00000000
R5 (a1) = 00000005	R13 (t5) = ffff800e	R21 (s5) = 00000000	R29 (sp) = 7ffff664
R6 (a2) = 00000007	R14 (t6) = ffff9fff	R22 (s6) = 00000000	R30 (s8) = 00000000
R7 (a3) = 0000000e	R15 (t7) = 0000000f	R23 (s7) = 00000000	R31 (ra) = 00000000

上述的全部内容均出现在这张图片里。

我将会继续关注指令以及相关汇编语言。