**上海大学 计算机学院**

**《计算机组成原理实验》报告十一**

**姓名 胡才郁 学号 20121034**

**时间 周四 9-11 机位 指导教师 刘学民**

**实验名称: 课堂练习(3)**

**一、实验目的**

1. 深化练习程序转移机制。

2. 深刻体会延迟程序的作用

**二、实验原理**

**2.1 程序转移机制：**

指令计数器用来存放当前正在执行的指令的地址，指令地址的形成有两种：一种是顺序执行的情况，通过指令计数器加“1”形成下一条指令地址；一种是改变执行顺序的情况，由转移类指令形成地址，送到指令计数器内，作为下一条指令的地址。

即对于程序转移，只有以下两种情况：

**对PC寄存器的自动加1功能实现程序顺序执行。**

**对PC寄存器的打入初值功能实现程序转移。**

在做综合实验时，可以用CP226计算机组成原理实验软件输入、修改程序，汇编成机器码并下载到实验仪上，由软件控制程序实现各种指令，故在完成本实验时，可以运用汇编语言程序解决一些问题，进行一些模拟，在手动模式下，也可以通过操纵PC及ST寄存器对程序转移的原理及细节进行验证。

**2.2 子程序调用**

一般使用堆栈来保存断点，在每一个程序运行时，操作系统都会为这个程序在主存之中开辟一段空间作为此进程的运行堆栈，使用堆栈指针SP指向当前运行堆栈的栈顶元素，进栈操作为先修改指针，后存入数据。

子程序调用时，会执行另一个程序，然后返回。所以在调用另一个程序时必须保存断点。

图示, 示意图

描述已自动生成

图1 堆栈寄存器中断点存储示意图

**2.3 斐波那契数列：**

斐波那契数列（Fibonacci sequence），又称黄金分割数列，因数学家莱昂纳多·斐波那契（Leonardo Fibonacci）以兔子繁殖为例子而引入，故又称为“兔子数列”，指的是这样一个数列：1、1、2、3、5、8、13、21、34、……这个数列从第3项开始，每一项都等于前两项之和。在数学上，斐波那契数列以如下被以递推的方法定义：F(0)=0，F(1)=1, F(n)=F(n - 1)+F(n - 2)（n ≥ 2，n ∈ N\*）

图示

描述已自动生成

图2 斐波那契数列为边的几何表示

**三、实验内容**

**1.实验任务一：斐波那契数列**

斐波那契数列，又称黄金分割数列，指的是这样一个数列：0、1、1、2、3、5、8、13、…..。这个数列从第3项开始，每一项都等于前两项之和。试将该数列各个值依次显示在OUT寄存器中，当某个数列值大于33H时，系统停机。请编程实现它。

1. **实验步骤**

①打开集成编程环境。连接实验设备，打开集成编程环境

②连接PC机与实验箱的通信口，选择“COM4，打开编写好的后缀为asm的程序，编译并下载源程序。

③运行程序，并观察实验箱中OUT寄存器的变化情况。

1. **实验现象**

观察运行结果，OUT寄存器中每1秒数字发生一次变化，当OUT寄存器中输出16进制数37，即十进制数55时，程序停止。

图片包含 游戏机, 电子, 电路

描述已自动生成

图3 程序结束时OUT寄存器的示数

1. **数据记录、分析与处理**

现分析该程序如何实现本实验任务。汇编程序与分析如下表：

表1 程序功能分析

|  |  |  |
| --- | --- | --- |
|  | **MOV A, #00H** | **赋初值** |
|  | MOV R0, #01H | 将立即数01H送至R0寄存器 |
| LOOP1: | **ADD A, R0** | **将前两项数值相加，并将相加之和送入OUT** |
|  | OUT | 将累加器 A 中内容送至 OUT |
|  | MOV 30H, A | 将累加器A中的内容送入30H内存单元 |
|  | SUB A, #33H | 累加器 A 中内容减去立即数33H |
|  | JC LOOP2 | 当运算产生进位标志，跳转LOOP2地址处 |
|  | JMP STOP | 无条件跳转至STOP地址处 |
| LOOP2: | **CALL DELAY** | **调用延迟程序** |
|  | **MOV A, R0** | **形成最新的前两项数值** |
|  | MOV 31H, A | 累加器A中内容送入31H内存单元 |
|  | MOV A, 30H | 将30H内存单元内容送至累加器 A |
|  | MOV R0, A | 将累加器A中内容送至R0寄存器 |
|  | MOV A, 31H | 将31H内存单元内容送至累加器 A |
|  | JMP LOOP1 | 无条件跳转至LOOP1地址处 |
| DELAY: | **MOV A, #11H** | **延迟1秒** |
| LOOP: | SUB A, #01H | 累加器 A 中内容减去立即数01H |
|  | JZ EXIT | 此处如果遇到零值，跳转EXIT地址处 |
|  | JMP LOOP | 无条件跳转至STOP地址处 |
| EXIT: | RET | 堆栈寄存器内容打入PC寄存器 |
| STOP: | **MOV A, #FFH** | **系统停机** |
|  | JMP STOP | 无条件跳转至STOP地址处 |
|  | END | 结束程序 |

①程序从上而下，首先为将斐波那契数列的前两项00H、01H送入A寄存器与R0寄存器，之后在LOOP1之中完成相加操作。由于A寄存器在之后的减法指令中会再次使用，需要保存两数相加之后得到的中间结果，因此MOV 30H, A 语句将两数相加之和暂存在30H内存单元。

②之后，如果两数之和与33H相减有借位的话，说明两数之和小于33H，此时程序转移到LOOP2中，LOOP2是调用延迟子程序，当1秒的延迟子程序运行完毕后，会自动返回到调用指令的下一条指令。

③当LOOP2中的延时子程序结束后，之后连续的5条MOV指令，通过将暂存的中间计算结果进行转移，形成最新的前两项斐波那契数列数值，并且把此两项数据，分别送入A寄存器与R0寄存器，接着无条件转移至LOOP1，在LOOP1中继续进行两数相加的操作，循环操作，直至两数相加和与直接数33H相减没有错位。

④程序运行阶段中A，R0中分别为斐波那契数列的连续两项，其中A为较小项，R0为较大项。当OUT寄存器输出16进制数37，即10进制数55时，此时累加器A中值为34时，进行SUB A, #33H 语句后A的值为01H，不产生进位项，因此不跳转值LOOP2，从而继续执行JMP STOP，跳转至结束语句，程序停机。

1. **实验结论**

最终实验现象与初始的预期相符。JC语句的功能位当运算产生进位标志时跳转指定地址。

**四、建议和体会**

本次实验中，学校课程为学生准备了丰富的预习资料，因此学生仅需要一步一步从无到有跟随教学内容前进即可完成预习工作与实验任务，本次实验教学效果良好，本人没有建议。本次实验我收获颇丰，体会良多。

如果没有疫情，本次课堂练习会在课堂上亲自动手实践完成，我也可以编程实验除“斐波那契数列”之外其他有趣的数列，更加深入的理解程序调用的原理。在今天，短暂的“自由”结束了，由于疫情的反复，学校重新恢复了封闭式管理，足不出户、足不出寝，希望疫情早日结束，恢复正常的课程之中。