上海大学　计算机学院

《计算机组成原理实验》报告九

姓名 XXXXXXXX　　**学号** XXXXXXXX

时间 周三 9-11　　**机位** 8　　**指导教师** 刘跃军

实验名称: 程序转移机制

一、实验目的

1、学习实现程序转移的硬件机制。
2、掌握堆栈寄存器的使用。

二、实验原理

1、程序转移

在任何一个程序段的内部，执行流程有顺序、分支、循环三种，而程序段之间又有相互调用（例如：调用子程序、中断服务、子程序返回、进程调度、任务切换 ••）看似很复杂，其实计算机硬件用非常简单的技术解决了这些问题。

分支和循环总是可以相互替代，所以也常说程序段内的执行流程其实仅有顺序和转移两种，而程序段之间的调用也只是把执行流程转移到了另外一个程序段上。所以，任何复杂的程序流程，在硬件实现机制上只有两种情况：顺序执行和转移。硬件实现这两种情况的技术很简单：

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。

当转移目标为本段内未执行过的指令时就形成分支；当转移目标是本段内执行过的指令时就形成循环；当转移目标为其他段的指令时就形成段间调用。可见：转移操作决定于“给 PC 赋值”，而转移类型决定于“所赋的值同当前指令的关系”。

2、实验箱系统的程序转移硬件机制

当 LDPC 有效（即：LDPC=0）时，如果此时 DBUS 上的值就是转移的目标地址，则此目标地址被打入 PC（即：PC 被赋新值），从而实现程序的转移。若 LDPC 为 0 是附带条件的，就形成“条件转移”。实验箱依靠“PC 打入电路”实现“有进位”时转移和“计算结果为零”时转移，以及无条件转移。

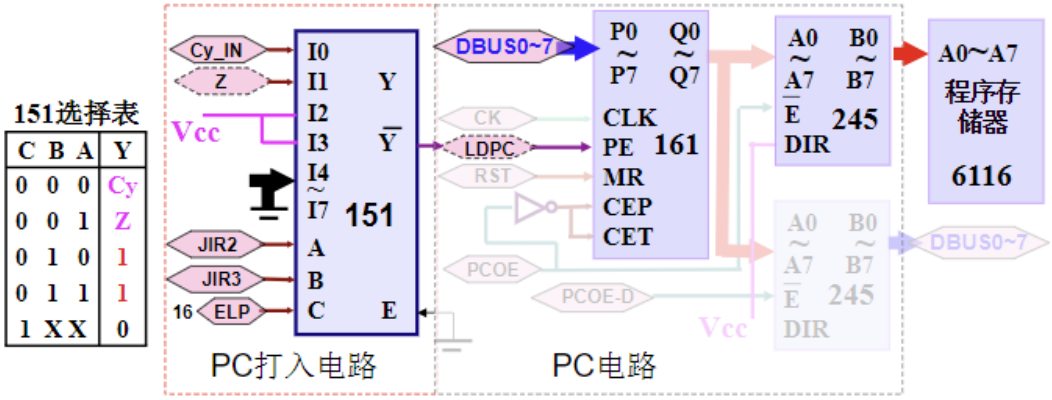

图 1. 实验箱程序转移硬件机制

3、子程序调用和保护断点

子程序的调用和返回是两次转移，特殊点在于：返回时转移的目标一定是调用时转移的出发点。为实现这个特点，在调用转移时必须把出发地址（即：断点地址）保存起来。这个“保存”还必须有以下两个要求：

（1）不被一般用户所知或改变。（2）返回转移时能方便地找到它。

第一个要求决定了它不能被保存在数据存储区或程序存储区，第二个要求决定了返回指令的目标地址获得方法与其它转移指令完全不同，返回指令的目标地址一定从一个特殊的“保存区”得到，指令本身不需要再带目标地址，而其他转移指令必须自带目标地址。再考虑到子程序调用的“可嵌套性”，这个“保护区”里的数据应该有“先入后出”特点，这与“货栈”中堆放的货物相似，故称其为“堆栈”。堆栈的容量决定了子程序的嵌套深度。

各系统实现堆栈的技术各不相同。实验箱系统用一个锁存器（574 芯片）构成堆栈寄存器（ST）。由于 574 芯片只能存放一个字节，所以，本系统的子程序调用深度只有 1 级，不能形成子程序嵌套。

4、ST 寄存器结构和子程序调用与返回控制信号

实验箱子程序调用和返回的结构由 PC 电路和 ST 电路组成。

（1）当调用子程序时，PC 的当前值（即：断点地址)经下面的 245 送上 DBUS，进入 ST 保存。然后给 PC 打入子程序入口地址，该子程序入口地址是由调用指令自身携带的目标地址，至此转子程序过程完成。

(2）当子程序返回（RET）时，返回指令开启 ST 的输出，并给出 PC 打入信号（无条件转移），于是 ST 保存的断点经由 DBUS 打入 PC，实现子程序返回。

三、实验内容

1．实验任务一（1.试用手动方式实现子程序调用转移过程。假设调用子程序指令的下一条指令存放在 11H 单元，子程序的入口地址为 22H；2.试用手动方式实现子程序返回转移过程。假设调用子程序指令的下一条指令存放在 11H 单元，子程序的入口地址为 22H。)（由于任务一与任务二可以一起进行实现并在验收环节一起验收，故放在一起来说）

(1) 实验步骤

先对两个实验分任务进行分析：

1、实现子程序调用转移过程。假设调用子程序指令的下一条指令存放在 11H 单元，子程序的入口地址为 22H

需要手动操控 PC 寄存器与 ST 寄存器实现程序转移。 具体的过程为，首先将数据准备好，将 11H 打入 PC 寄存器将其作为目前正在执行的指令的地址，（需要将 X2 X1 X0 置为低电平 0 来切换到 IN 进行手动输入，还需要令 ELP 为 0 以令 PC 寄存器接收输入的数据）。接着，将程序要跳转到的地址暂存于累加器 A（同样需要将 X2 X1 X0 置为低电平 0 来切换到 IN 进行手动输入，还需要令 AEN 为 0 来用寄存器 A 接收输入的数据）。

下面就可以进行程序转移，先将 PC 寄存器中的值暂存于 ST 寄存器。（将 X2 X1 X0 置为 0 1 1 来选择用 PC 寄存器输出数据，将 STEN 置为 0 打入 ST 寄存器。至此，完成将 PC 寄存器中此时的值保存在 ST 寄存器中。

最后，需要让 A 寄存器中值被传送到 PC 寄存器中。（将 X2 X1 X0 置为 0 1 1 来从 PC 寄存器输出数据，将 STEN 置为 0 来打入 ST 寄存器接收输入的数据。令 X2 X1 X0 置为 1 0 0 以选择 D 输出数据，这个数据就是 A 寄存器中的数据，还需要令 ELP 为 0 以令 PC 寄存器接收输入的数据。

至此，完成子程序调用转移的操作。

2、实现子程序返回转移过程。假设调用子程序指令的下一条指令存放在 11H 单元，子程序的入口地址为 22H

操作方式与分任务一类似，需要将 ST 寄存器中的值传送到 PC 寄存器中。（将 X2 X1 X0 置为 010，即选中 ST 寄存器作为输出，并打入置为低电平 0 的 ELP 的 PC 寄存器中，按下 STEP 键即可将 ST 寄存器中的值传送到 PC 寄存器中。

下面是具体的连线及操作过程：

将 X2、X1、X0 与 K2、K1、K0 相连，用于控制输入方式，将 AEN 与 K4 相连，用于控制 A 寄存器的写入，将 ELP 与 K7 相连，将 STEN 与 K8 相连。

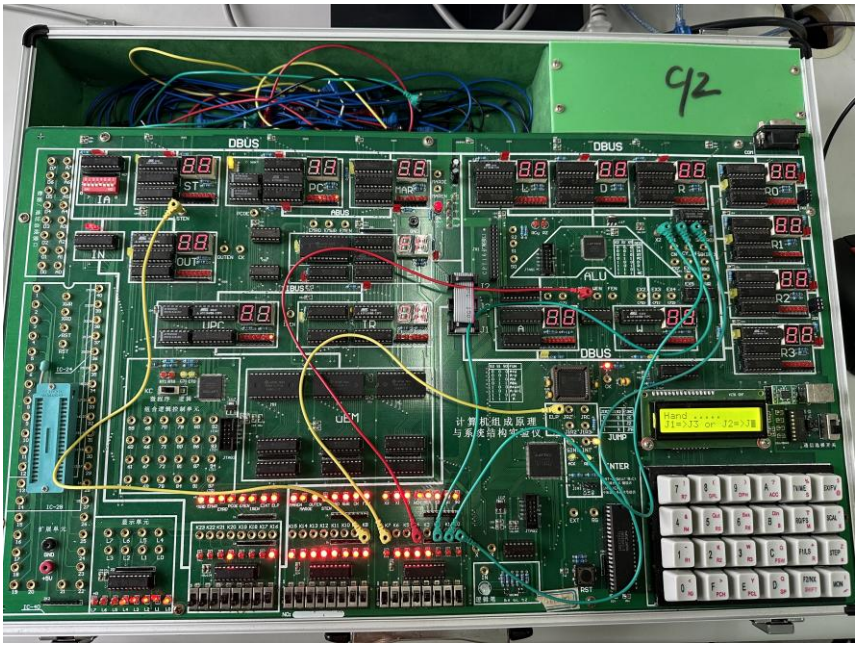

图 2.实验分任务一初始连线

操作如下：

(1)将 K23-K16 置为 00010001，X2 X1 X0 置为 0 0 0，STEN 置为 1，ELP 置为 0，AEN 置为 1，按下三次 TV/ME 按键，进入 HAND 手动模式，按下 STEP 键，可以看到 PC 寄存器显示屏上数字为 11H；

(2)再将 K23-K16 置为 00100010H，ELP 置为 1，AEN 置为 0，按下 STEP 键，可以看到 A 寄存器显示屏上数字为 22H；

(3)X2 X1 X0 置为 0 1 1，STEN 置为 0，ELP 置为 1，AEN 置为 1，按下 STEP 键，可以看到 ST 寄存器中数据为 11H；

(4)最后，将 K23~K16 置为 11111111H，STEN 置为 1，ELP 置为 0，AEN 置为 1，按下 STEP 键，可以看到 PC 寄存器中数据为 22H。

至此，完成了子程序调用转移过程。

~~至此，实验分任务一完成，下面进入实验分任务二~~：

(1)X2 X1 X0 置为 0 1 0，STEN 置为 1，ELP 置为 0，AEN 置为 1，按下 STEP 键，可以看到 PC 寄存器中数据为 11H，说明数据正确从 ST 寄存器中打入了 PC 寄存器中。

实现了子程序返回转移过程。

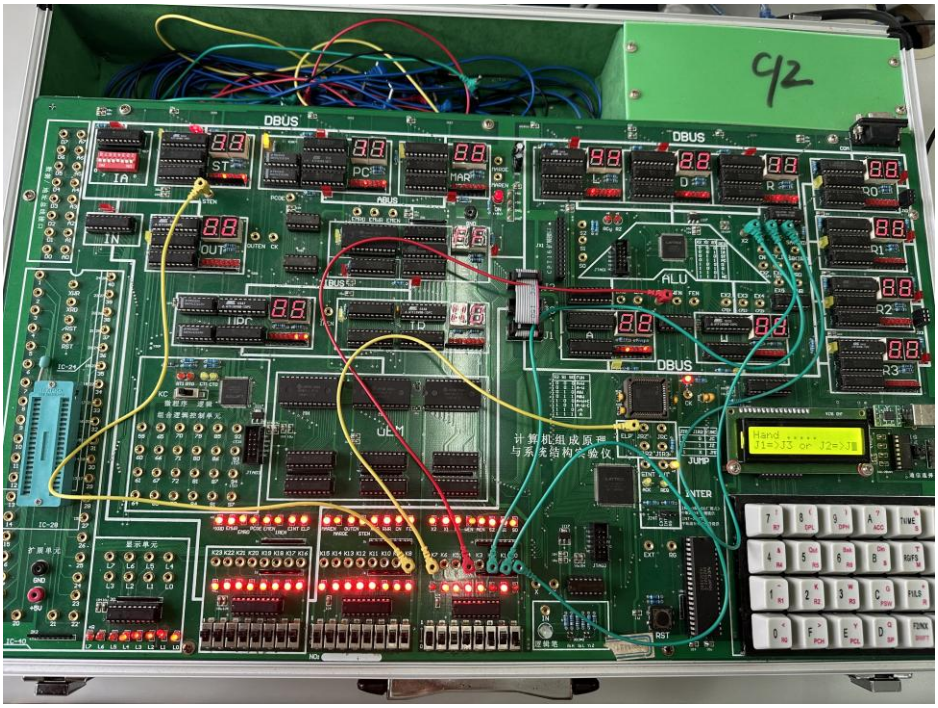

图 3 实验分任务二操作完后实验箱状态

(2) 实验现象

在各实验分任务操作过程中已经进行记录，在此不过多赘述。

本实验无数据分析与处理，数据记录同实验现象。

(3) 实验结论

可以认为，本次实验正确地完成了用手动方式实现子程序调用转移过程、手动方式实现子程序返回转移过程这两个分任务。

2. 实验任务二（3.编程实现 OUT 寄存器交替显示 11 和 55，交替频率为可以

清晰辨识，且不小于每秒一次。（实验箱的工作频率为：114.8Hz。））

先对任务进行分析，可以把任务分解为：

(1)将 A 寄存器中数据改为 11H，并送到 OUT 寄存器。(2)进入延时函数。(3)将 A 寄存器中数据改为 55H，并送到 OUT 寄存器。

其中(1)与(3)都十分好实现，只需要 MOV A,#11(55)H 与 OUT 即可实现。而因为指令的操作需要时间，需要我们可以通过让机器运行一定次数的运算后，跳出延时函数，显示另一个数据，即可以通过给寄存器 A 赋一个初值后，不断自减一再判断是否为 0，如果为 0 则跳出循环，显示下一个数据。因此，本任务汇编语言代码如下：

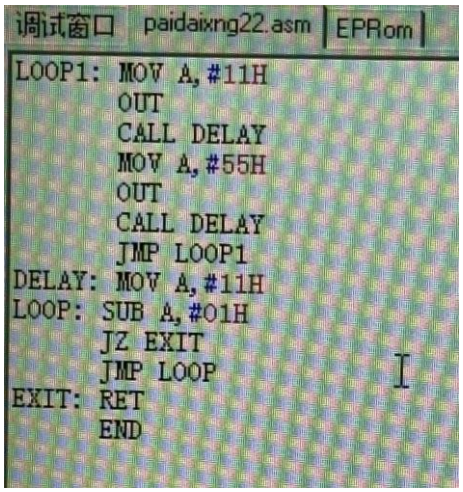

图 4.实验任务二汇编语言代码

用 CP226 文件打开含有上述代码的.asm 文件，确定正确打开后，点击快捷操作中的连接实验箱，确保与实验箱相连。点击编译写入，发现实验箱闪烁。按下分步操作的快捷操作，观察实验箱上 A 寄存器与 OUT 寄存器中的数值变化，可以发现 A 寄存器中的数值变化为：11、11、10、……、00、55、11、10、……、00(后面继续循环)，OUT 寄存器中的数值变化为：11 与 55 交替显示，且交替频率大于等于每秒一次，说明正确的编写了程序，实现了任务要求。

理论交替频率计算如下：

LOOP: SUB A, #01H　　三个时钟周期

JZ EXIT　　二个时钟周期

JMP LOOP　　二个时钟周期

以上为延时子函数的时钟周期分析，可以看到，整个子函数需要耗时七个时钟周期，用实验箱的工作频率除以运行一遍子函数需要花的时钟周期数，即可以得到至少需要的循环次数（即 114.8/7=16.4H，16H 在 10 进制下为 10，所以子延时函数需要运行>10 次，所以在我的操作中将 A 赋为 11，运行 11 次进入下个循环是满足了**交替频率不小于每秒一次**的要求的）

(4) 实验结论

在 OUT 寄存器显示屏上，可以看到 11 与 55 交替显示，且交替频率大于等于每秒一次，说明正确的编写了程序，实现了任务要求。

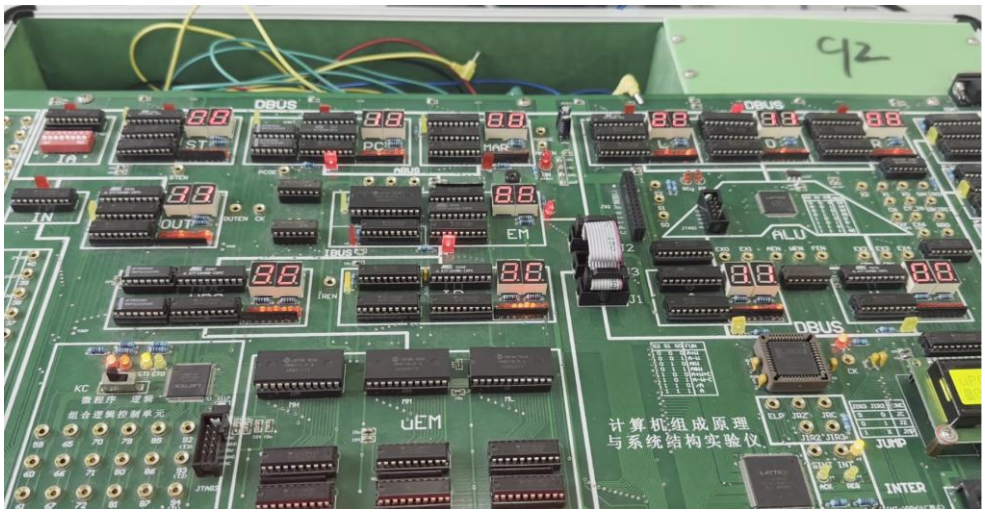

图 5.实验任务二显示 11

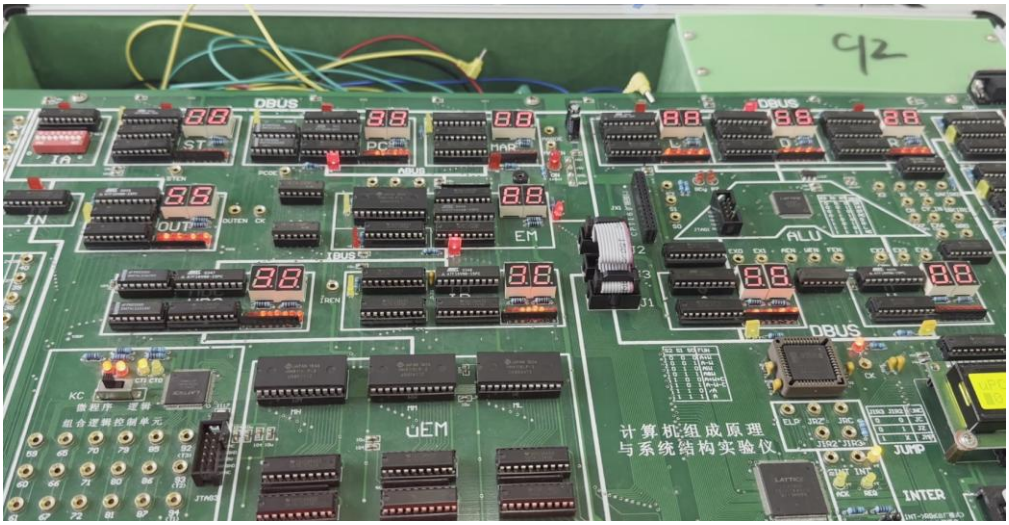

图 6.实验任务二显示 55

四、建议和体会

1、实验开始前，要先检查一下导线是否功能正常。
2、通过自己的尝试与摸索，自己学会了如何进行 CP226 软件上编写汇编语言与实验箱操作的结合。
3、依然还是老样子，提前预习的重要性，即在进行实验课前，需要对该节课实验的内容对应的理论知识（虽然好像与机组理论课关系不大）进行温故与复习并且对实验步骤进行大框架上的构想与思考，最好能够提前学会要用到的知识点。

（如果不复习、提前构想的话，仅靠课堂上的那两个小时可能不足以完成所有任务，或者说进展的比较慢，没有什么头绪）。

五、思考题

1、若要求 11 和 55 各显示 50 次后停机，应该如何修改程序?

答：可以在循环开始前，先给 A 赋值 50（16 进制下为 32H）(MOV A,#32H)，并将 A 打入寄存器 R0(MOV A,R0)，每次进入循环时将 R0 赋给 A(MOV R0,A)，A 自

减一(SUB A,#01H)，判断是否为0，如果为0则跳出循环(JZ EXIT)，直接结束。