

上海大学 计算机学院

《计算机组成原理实验》报告九

姓名 严昕宇

学号 20121802

时间 2022. 05. 03

机位 19

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。

可知：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 有效时（即 LDPC=0），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。其机制结构如图 1 所示。

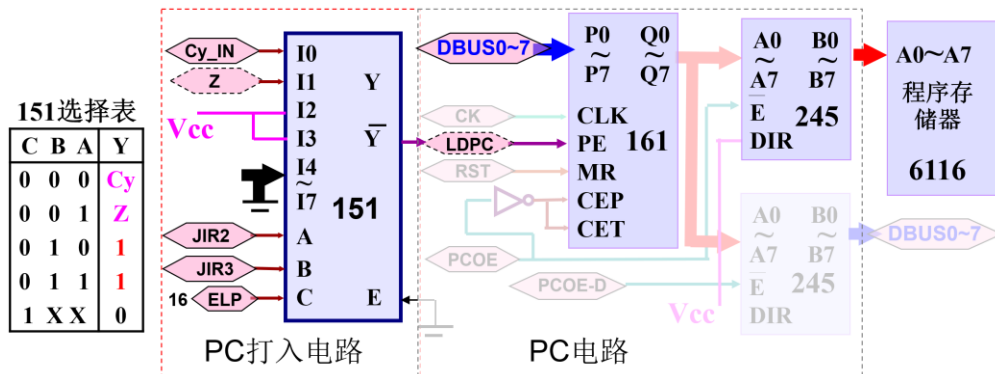


图 1. 程序转移硬件机制结构示意图

AEN	ELP	STEN	X2	X1	X0
K6	K5	K4	K3	K2	K1
1	0	1	0	0	0

- ③ 按下 STEP 键，将 11H 写入程序计数器 PC。
- ④ 设置 K23~K16 为 00100010 (22H)，并按照下表给出对应的信号：

AEN	ELP	STEN	X2	X1	X0
K6	K5	K4	K3	K2	K1
0	1	1	0	0	0

- ⑤ 按下 STEP 键，22H 写入 A 寄存器。
- ⑥ 按照下表给出对应的信号：

AEN	ELP	STEN	X2	X1	X0
K6	K5	K4	K3	K2	K1
1	1	0	0	1	1

- ⑦ 按下 STEP 键，将当前 PC 值打入 ST 堆栈寄存器。
- ⑧ 设置 K23~K16 为 11111111 (FFH)，按照下表给出对应的信号：

AEN	ELP	STEN	X2	X1	X0
K6	K5	K4	K3	K2	K1
1	0	1	1	0	0

- ⑨ 按下 STEP 键，使 A 寄存器内容——子程序入口地址输出，即 A 寄存器内容打入 PC。
- ⑩ 记录实验现象并分析，继续完成实验任务二。

(2) 实验现象

首先 PC 寄存器被打入 11H 数据，A 寄存器被打入 22H 数据。再将当前 PC 寄存器中的值打入 ST 寄存器，则 ST 寄存器显示 11H。最后将 A 寄存器内容打入 PC 寄存器，PC 显示 22H。

(3) 数据记录、分析与处理

根据上述实验步骤及实验现象可知，如果要想实现实验任务一，就需要手动操控 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，这需要令 X2X1X0 均为 0 以选择输入门输入在电键 K23-K16 上的 11H 数据，还需要令 ELP 为 0 以令 PC 寄存器接收输入的数据。

其次，需要转备好程序要跳转到的地址，这里将程序要跳转到的地址暂存于累加器 A，这需要令 X2X1X0 均为 0 以选择输入门输入在电键 K23-K16 上的 22H 数据，还需要令 AEN 为 0 以令累加器 A 接收输入的数据。

准备工作完成后便可以进行实验任务了。该实验任务要求实现手动进行程序转移，所以需要将 PC 寄存器中的值暂存于 ST 寄存器。这是因为程序要进行跳转时需要知道运行完子程序的返回地址，对于一般的计算机而言，这一返回地址通过入栈的方式得以保存，所以用 ST 寄存器暂存 PC 寄存器此时的值。这需要令 X2X1X0 置为 011 以选择 PC 寄存器输出数据，还需要令 STEN 为 0 以令 ST 寄存器接收输入的数据。至此，已完成将 PC 寄存器中此时的值保存的工作。

接下来，需要跳转到子程序的地址进行执行，因为假定将子程序的入口地址暂存于 A 寄存器，所以需要让 A 寄存器中值被传送到 PC 寄存器中。这需要令 X2X1X0 置为 011 以选择 PC 寄存器输出数据，还需要令 STEN 为 0 以令 ST 寄存器接收输入的数据。至此，已完成将 PC 寄存器中此时的值保存的工作。令 X2X1X0 置为 100 以选择直通门 D 输出数据，而通过直通门 D 输出的数据就是 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，已完成将子程序地址打入 PC 寄存器的操作。

(4) 实验结论

成功在手动模式下，实现子程序调用转移过程。
完成实验任务一的目的。

2. 实验任务二（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）

(1) 实验步骤

- ① 接着实验任务一的结果。ST 寄存器中存放着子程序返回主程序的地址 11H，PC 寄存器中存放着子程序的入口地址 22H。确认无误后，将实验箱调整到手动模式，按照下表连线，并给出对应的信号：

AEN	ELP	STEN	X2	X1	X0
K6	K5	K4	K3	K2	K1
1	0	1	0	1	0

- ② 按下 STEP 键，11H 从 ST 寄存器进入 PC 寄存器。

(2) 实验现象

进行步骤①后，程序计数器 PC 被打入 11H 数据。

(3) 数据记录、分析与处理

根据上述实验步骤及实验现象可知，如果要想实现实验任务二，需要手动操控 PC 寄存器与 ST 寄存器实现程序返回。

若想要完成这一操作，需要将 ST 寄存器中的值传送到 PC 寄存器中。因为此时 ST 寄存器中存放着程序在运行完子程序之后要返回的地址，所以只需要让 ST 寄存器中的值传送回 PC 即可实现程序返回。这需要令 X2X1X0 为 010，即选中 ST 寄存器作为数据的输出端，并令 ELP 为 0 以使 PC 寄存器接收传送过来的值，再按下 STEP 键即可将 ST 寄存器中的值传送到 PC 寄存器中。

(4) 实验结论

成功在手动模式下，实现了子程序返回转移过程。
完成实验任务二的目的。

3. 实验任务三（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]）

(1) 实验步骤

- ① 打开实验箱电源，检验实验箱的基本功能是否正常。确认无误后，启动 CP226 软件。
- ② 连接 PC 机与实验箱的通信口 COM4。
- ③ 在 CP226 汇编语言程序集成开发环境下编写如下汇编程序：

```
ORG 00H
LOOP1: MOV A, #11H
      OUT
      CALL DELAY
      MOV A, #55H
      OUT
      CALL DELAY
      JUMP LOOP1
DELAY: MOV A, #01H
LOOP:  SUB A, #01H
      JZ EXIT
      JMP LOOP
EXIT:  RET
      END
```

- ④ 保存文件，文件扩展名为.asm（这一步一定要添加，软件不会自己带上后缀）。
- ⑤ 编译并下载源程序至实验箱，调试并运行程序，观察并记录实验结果。

(2) 实验现象

随着每一步运行，OUT 寄存器交替显示 11H 和 55H，交替频率大约为每秒一次。

(3) 数据记录、分析与处理

汇编程序的具体分析如下所示

ORG 00H	通知汇编软件下面的程序从 00H 始存放
	LOOP1: 主程序
LOOP1: MOV A, #11H	将立即数 11H 送至寄存器 A
OUT	将寄存器 A 中内容送至 OUT
CALL DELAY	调用 DELAY 子程序
MOV A, #55H	将立即数 55H 送至寄存器 A
OUT	将寄存器 A 中内容送至 OUT
CALL DELAY	调用 DELAY 子程序
JUMP LOOP1	无条件跳转 LOOP1

DELAY、LOOP、EXIT:		
延时 1 秒的子程序		
DELAY: MOV A, #11H		将立即数 11H 送至寄存器 A 十进制 17, 即 $114.8/7 \approx 17$
LOOP: SUB A, #01H		从寄存器 A 中内容减去立即数 01H 三个时钟周期
JZ EXIT		根据 PSWR 中零标志位 ZF 决定是否 二个时钟周期
JMP LOOP		跳转到 EXIT (若零标志位置 1 则跳转) 无条件跳转 LOOP 二个时钟周期
EXIT: RET		子程序返回
END		汇编程序终止 (如果不加程序不会停止)

(4) 实验结论

成功利用 CP226 软件编写汇编程序完成实验任务三，实现了数值的交替显示。

完成实验任务三的目的。

四、建议和体会

体会

通过本次实验,我对计算机内存系统有了更加深入的了解。在前两个实验中,是在手动模式下完成任务的。通过连线,选择对应的信号,设置对应的数值,到最后按 STEP 键执行,这样完成任务的方式将一个看似很简单的操作拆分成一个又一个的细节,落实到具体每步选择哪个寄存器,对选中的寄存器该给出什么信号等等,使我对内存系统的运作模式有了非常清晰地认识。至于第三个实验,我们是通过编写汇编程序实现的。平时经常接触高级语言的我们,通过使用汇编语言,可以深入理解计算机的工作、组成方式,使得程序中的变量在我们眼里不再只是他们在高级语言中对应的变量名,而是在寄存器中被实实在在存储的一个数据。编写汇编程序首先需要熟悉模型机对应的指令集,然后即可进行程序设计。

由于疫情的影响,本次实验是居家完成。通过观看线上的实验讲解视频与学习实验内容 PPT,并未能亲身操作实验箱和 CP226 软件来进行实验,这会一定程度上影响学习效果,因此需要课前高质量的预习与课后的总结以加深理解。并且如果有机会,在返回学校进入实验室后,应该重新实验以验证自己实验报告中的内容。

五、思考题

若要求 11H 和 55H 各显示 50 次后停机，应该如何修改程序？

答：要求 11H 和 55H 各显示 50 次后停机，则需要一个寄存器用于存储循环次数，每一次循环中减一，具体汇编代码如下：

ORG 00H	通知汇编软件下面的程序从 00H 始存放
MOV A, #32H	将立即数 32H 送至寄存器 A (十进制的 10 等于十六进制的 32H)
LOOP1: MOV A, #11H	将立即数 11H 送至寄存器 A
OUT	将寄存器 A 中内容送至 OUT
CALL DELAY	调用 DELAY 子程序
MOV A, #55H	将立即数 55H 送至寄存器 A
OUT	将寄存器 A 中内容送至 OUT
CALL DELAY	调用 DELAY 子程序
MOV A, R1	将寄存器 R1 的值送到寄存器 A 中
SUB A, #01H	从寄存器 A 中内容减去立即数 01H
JZ EXIT1	根据 PSWR 中零标志位 ZF 决定是否 跳转到 EXIT1 (若零标志位置 1 则跳转)
MOV R1, A	将寄存器 A 的值送到寄存器 R1 中
JUMP LOOP1	无条件跳转 LOOP1
DELAY: MOV A, #11H	将立即数 11H 送至寄存器 A
LOOP: SUB A, #01H	从寄存器 A 中内容减去立即数 01H
JZ EXIT	根据 PSWR 中零标志位 ZF 决定是否 跳转到 EXIT (若零标志位置 1 则跳转)
JMP LOOP	无条件跳转 LOOP
EXIT: RET	子程序返回
EXIT1: END	汇编程序终止 (如果不加程序不会停止)