

由于 Mar 没有自动+1 功能，因此数据存储是随机的。读取存储器数据时，6116 的输出值将同时通过 245 总线和 IBus。为了避免影响指令中微指令的执行，必须确保 μ PC 使能控制 Iren 行无效。

3. 模型机的内存结构

在模型机中，程序和数据共享一个公共的物理内存（分区使用）。MAROE 和 PCOE 不能同时有效，否则 MAR 和 PC 输出的“和”值将出现在 ABUS 上。

三、实验内容

1. 实验任务一：手动方式把立即数 33H 写入内存 D1H 单元。

(1) 实验步骤

手动方式把立即数 33H 写入内存 D1H 单元。

1. 从 MAROE, MAREN, EMEN, EMRD, EMWR, STEN, X2, X1, X0 接口处引出导线连接，打开装置，按三下 TV-ME，将实验箱调至手动模式

2. 数据输入 0xd1，将 MAREN 置 0，X2, X1, X0 置 0，按下 STEP 键

3. 数据输入 0x33，将 MAROE 置 0，X2, X1, X0, EMEN, EMWR 置 0，按下 STEP 键

(2) 实验现象

根据视频中老师的操作，在输入地址后，在 MAR 中可看到 D1，写入到 D1 单元时，在存储器可看到数据 0x33

(3) 数据记录、分析与处理

将 D1 写入存储器中，先送入地址，后送入数据。最终讲存储器的数据单元送至 D1 存储单元处。

(4) 实验结论

理解了写入内存的基本操作，实现了目的一

2. 实验任务二：手动方式把 D1H 单元的内容读出，再送入 E1H 单元。

(1) 实验步骤

手动方式把 D1H 单元的内容读出，再送入 E1H 单元。

在实验一的基础上完成以下操作：

1. MAROE, EMEN, EMRD, STEN 置为 0，按下 STEP 键

2. MAREN, X2, X1, X0 置为 0，数据输入 0xE1，按下 STEP 键

3. MAROE, X2, X0, EMWR, EMEN 置为 0，按下 STEP 键

(2) 实验现象

将数据从 d1 单元读出后，暂存在 ST 堆栈寄存器中，输入 E1 地址后，在 MAR

中可看到 E1，最终将 33 写入存储器中。

(3) 数据记录、分析与处理

读取到的数据先暂存在 ST 堆栈寄存器中，然后送入地址，最后写入存储器，分三步进行

(4) 实验结论

成功实现了写入内存与读取存储器中数据的基本操作，完成接线控制的作用与选择。

3. 实验任务三：在 CP226 汇编语言程序集成开发环境下编写程序，调试和单微步（跟踪）运行，完成下面任务，观察数据走向及寄存器的输入输出状态。将初始值 03H 赋予累加器 A，每次减 1，到 0 时，OUT 寄存器输出 FFH。

(1) 实验步骤

打开 CP226，按照实验要求，编写汇编程序：

```
MOV A #03H

LOOP: SUB A, #01H

      JZ LOOP1

      JMP LOOP

LOOP1: MOV A, #FFH

      OUT

      END
```

保存文件，文件扩展名为.asm

编译下载源程序至实验箱，调试并单步运行程序，观察并记录实验结果。

(2) 实验现象

在多步调试结束以后，OUT 寄存器数码管显示 FFH。

(3) 数据记录、分析与处理

汇编程序的具体分析如下所示：

MOV A #03H	； 将立即数 03H 送至累加器 A
LOOP: SUB A, #01H	； 累加器 A 中内容减去 01H
JZ LOOP1	； 此处如果遇到零值，就跳转 LOOP1
JMP LOOP	； 无条件跳转 LOOP
LOOP1: MOV A, #FFH	； 将 FF 送至累加器 A

OUT ：将累加器 A 中内容送至 OUT

END ：汇编程序终止（如果不加程序不会停止）

此处选取了一个循环周期，对于循环过程中各个寄存器数值变化进行了分析，10 个 CK 脉冲的寄存器记录表如下：

CK	PC	UPC	UEM	A	W	OUT
1	01	00	CBFFFF	00	00	00
2	01	7C	C7FFF7	00	00	00
3	02	7D	CBFFFF	03	00	00
4	03	3C	C7FDEF	03	00	00
5	04	3D	FFFE91	03	01	00
6	04	3E	CBFFFF	02	01	00
7	05	A4	C6FFFF	02	01	00
8	06	A5	CBFFFF	02	01	00
9	07	AC	C6FFFF	02	01	00
10	02	AD	CBFFFF	02	01	00

分析：以下对一次循环之中的变化做了分析

CK=1 时，IR（见寄存器记录表）中存储当前执行的指令为 00，PC 存储（下一条指令的地址），微程序 um 中存储的微指令为 CBFFFF（取指指令），微地址为 00；

CK=2 时，um 中存储的微指令为 C7FFF7（EMRD、PCOE、EMEN、AEN=0，PC 值送 ABUS，DBUS 上立即数 03H 写入 A 寄存器），微地址为 7C，

CK=3 时，um 中存储的微指令为 CBFFFF（取指指令），微地址为 7D，完成“MOV A，#03H”；

CK=4 时，um 中存储的微指令为 C7FDEF（EMRD、PCOE、EMEN、WEN=0，DBUS 上数据 01H 写入 W 寄存器），微地址为 3C，

CK=5 时，um 中存储的微指令为 FFFE91（FEN、AEN=0，S2S1S0=001，A—W 减法操作，X2X1X0=100，直通门 01H 输出写入 A 寄存器）微地址为 3D。

CK=6 时，um 中存储的微指令为 CBFFFF（取指指令），微地址为 7D，完成指令“SUB A，#01H”；

CK=7 时，um 中存储的微指令为 C6FFFF，判断是否遇到零值。

CK=8 时，um 中存储的微指令为 CBFFFF（取指指令），微地址为 7D，完成“JZ LOOP1”，由于未遇到零值，所以不跳转；

CK=9 时，um 中存储的微指令为 C6FFFF，无条件跳转 LOOP，跳转到 PC=2 处。

CK=10 时, um 中存储的微指令为 CBFFFF (取指指令), 微地址为 AD, 完成“JMP LOOP”。

至此, 一次循环完成。

(4) 实验结论

实验 3 成功利用 CP226 软件编写汇编程序完成实验任务三, 观察周期下各种数值的变化类型, 最终实验现象与初始的预期相符。

四、建议和体会

由于疫情的原因, 无法亲自动手完成实验从本次实验开始, 在这样的情况下如果没有做好提前的预习工作那么便很难搞清楚原理。提前做好实验的预习, 认真观看学习通上的操作视频十分的重要。可以在课前建立一个对本实验清晰的认知。

在这次实验中, 我们主要学习了内存访问机制, 丰富了知识面, 通过理解代码和数据的分区存放原理和技术, 收获颇丰。了解并验证了数据分区处理的过程, 期待接下来的课程。

建议方面:

本次实验学习效果良好, 基本流程也与上学期的实验差距不大, 尽管疫情使得无法动手实验, 但只要认真预习, 观看视频, 也能学到原理。

在实验操作时, 由于摄像角度问题, 数码管读数难以看清, 建议录制时更换角度。

PS: 实验视频之中第三个实验任务之中, 立即数为 05H, 而视频中的操作均为 03H, 本实验报告按照 03H (即视频中的版本) 撰写。

五、思考题

既然有 ORG 伪指令, 为什么集成开发环境下载到实验箱的目标程序的第一条

执行语句最好存放在 0 号地址？

在计算机中，程序是顺序执行的。因此，在执行程序时，它会受到目标程序的第一个执行语句的地址之前的地址中存储的指令语句的影响，这可能导致程序无法正确运行。

而如果第一条执行语句设置存放在 0 号地址的话，可以避免之前地址下留有的语句影响实验结果，能够很好的解决这个问题并确保实验结果准确。