

《计算机组成原理与系统结构 A》实验报告

机箱实验 5-6-7

学院：计算机学院

学号：22122861

姓名：邱姜铭

评分内容	评分
内容完整，有实验目的、步骤、分析、总结	
内容质量，表述清楚正确，条理清晰，分析总结到位	
格式规范，标题、段落、公式、图表、代码符合专业文献出版要求	
综合得分	

数据传送实验

1. 目的

1. 理解自然语言形式命令的人工译码过程。
2. 学习系统部件和数据总线间传送数据的操作。

2. 步骤

实验任务：

- 1 将 57H 写入 A 寄存器。
- 2 将 68H 写入 W 寄存器。
- 3 将 12H 写入 R0 寄存器，
- 4*并将 23 写入 R2，并从寄存器 OUT 输出。

2.1 实验环境（软硬件配置）

实验箱一台

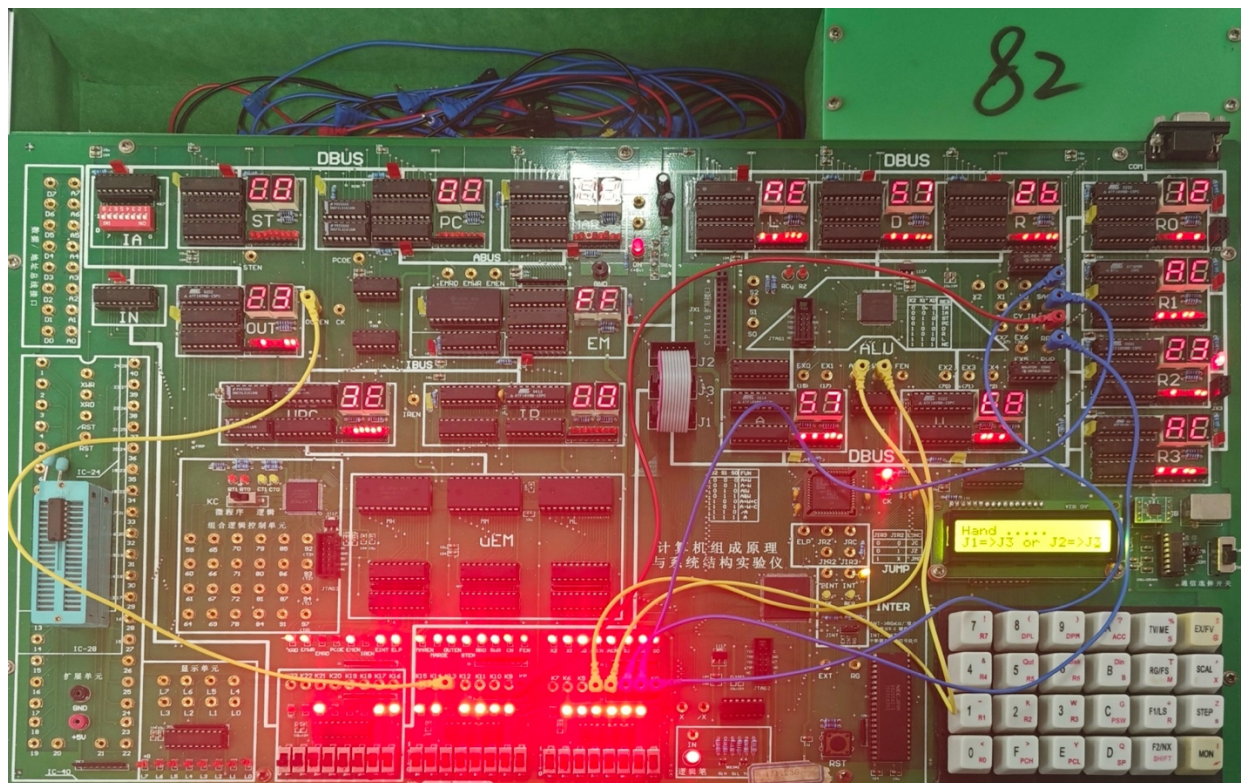
2.2 接线部分

1. 关闭电源
2. 用 8 位扁平线把 J3 和 J1 连接
3. 将 K0 与 SA，K1 与 SB，K2 与 RWR，K3 与 WEN，K4 与 AEN，K5 与 RRD，K6 与 OUTEN 连接
4. K0~K23 全部放在 1 位

2.3 实验操作

1. 注视仪器，打开电源，手不要远离电源开关，随时准备关闭电源，注意各数码管、发光管的稳定性，静待 10 秒，确信仪器稳定、无焦糊味。
2. 设置实验箱进入手动模式。
3. 设置 $K6K5K4K3K2K1K0=1101111$ 。
4. 设置 $K23\sim K16=0101\ 0111$ 。
5. 注视 A 及 DBUS 的发光管，按下 STEP 键，CK 灯灭、A 旁的灯亮。
6. 放开 STEP 键，CK 灯亮、A 寄存器显示 57，实验任务 1 完成。
7. 设置 $K6K5K4K3K2K1K0=1110111$ 。
8. 设置 $K23\sim K16=0110\ 1000$ 。
9. 注视 W 及 DBUS 的发光管，按下 STEP 键，CK 灯灭、W 旁的灯亮。
10. 放开 STEP 键，CK 灯亮、W 寄存器显示 68，实验任务 2 完成。
11. 设置 $K6K5K4K3K2K1K0=1111000$ 。
12. 设置 $K23\sim K16=0001\ 0010$ 。
13. 注视 R0 及 DBUS 的发光管，按下 STEP 键，CK 灯灭、R0 旁的灯亮。
14. 放开 STEP 键，CK 灯亮、R0 寄存器显示 12，实验任务 3 完成。
15. 设置 $K6K5K4K3K2K1K0=1111010$ 。
16. 设置 $K23\sim K16=0010\ 0011$ 。
17. 注视 R2 及 DBUS 的发光管，按下 STEP 键，CK 灯灭、R2 旁的灯亮。
18. 放开 STEP 键，CK 灯亮、R2 寄存器显示 23。
19. 设置 $K6K5K4K3K2K1K0=0011110$ 。
20. 注视 OUT 及 DBUS 的发光管，按下 STEP 键，CK 灯灭、OUT 旁的灯亮。
21. 放开 STEP 键，CK 灯亮、OUT 寄存器显示 23，实验任务 4 完成。
22. 关闭实验箱电源。

3. 实验结果



4. 总结

1. 在这个实验中，人理解（分析）命令（自然语言形式）的含义，然后产生控制总线上的信号，所以在这里人是“指令解码器或控制器”。
2. 在本接线形式下，完成这个操作的“机器命令”就是 $K2K1K0=010$ 。若一个计算机控制总线有 16 条，标记为 $C15 \sim C0$ ，而 $K2K1K0$ 正好是控制总线的 $C2C1C0$ ，则这个操作的机器命令就是：

1111 1111 1010。

3. 在本实验中，一个 **CK 周期**，或说一组控制总线信号，可以完成操作，称为其为基本操作，（在微程序结构中称为微指令）。
4. 可见(微)指令的**具体编码**是由控制总线的布线决定的。
5. 显然：如果对一个器件输入这个操作的（某种形式的）（微）命令，这个器件能输出对应的 01 编码（控制总线上的信号），这个器件就是这个机器的这条（微）命令的译码器或称控制器。若一个器件能对一台计算机的所有基本操作产生正确的控制总线信号，这个器件就是这个型号计算机的指令译码器——控制器。目前常用的指令译码器有寄存器方式和组合逻辑方式两种。

微指令系统实验

1. 目的

1. 读出系统已有的微指令，并理解其含义。
2. 设计并实现微指令系统。

2. 步骤

实验任务：

1. 观察微指令寄存器地址为 00H 和 11H 单元的内容
2. 编制一条微指令实现“A 非”运算后左移一位的值送 OUT；把这条微指令放入微程序寄存器的 02H 单元；验证它的功能是否实现。
- 3*. 编制一条微指令实现 $A+W$ 后左移一位的值送 R1；把这条微指令放入微程序寄存器的 05H 单元；验证它的功能是否实现。

2.1 实验环境（软硬件配置）

实验箱一台

2.2 任务 1

操作步骤：

在小键盘的 μEM 状态，输入微程序存储器地址 00H 或 11H，可以从液晶屏上读到该地址中微指令的十六进制数形式，然后将它转换成二进制码，对照 24 条控制线的排列次序，分析其功能。

2.3 任务 2

操作步骤:

第一步: 将 J1 与 J2 相连;

第二步: 根据题意可知: $S_2 S_1 S_0=110$, $X_2 X_1 X_0=110$, $OUTEN=0$

$C_{23} \sim C_0=1111, 1111, 1101, 1111, 1101, 1110=FF, DF, DE H$

第三步: 进入微程序存储器模式(μEM 状态), 给 00H, 01H 地址单元都打入 $ff\ ff\ ffH$, 给 02H 地址单元打入 $FF, DF, DE H$ 。

第四步: 再进入“内部寄存器”方式, 将 μPC 、PC、A 寄存器分别设置成 00、00、11, 按动 STEP 键, 观测 A、L 和 OUT 结果。

2.4 任务 3

操作步骤:

第一步: 将 J1 与 J2 相连;

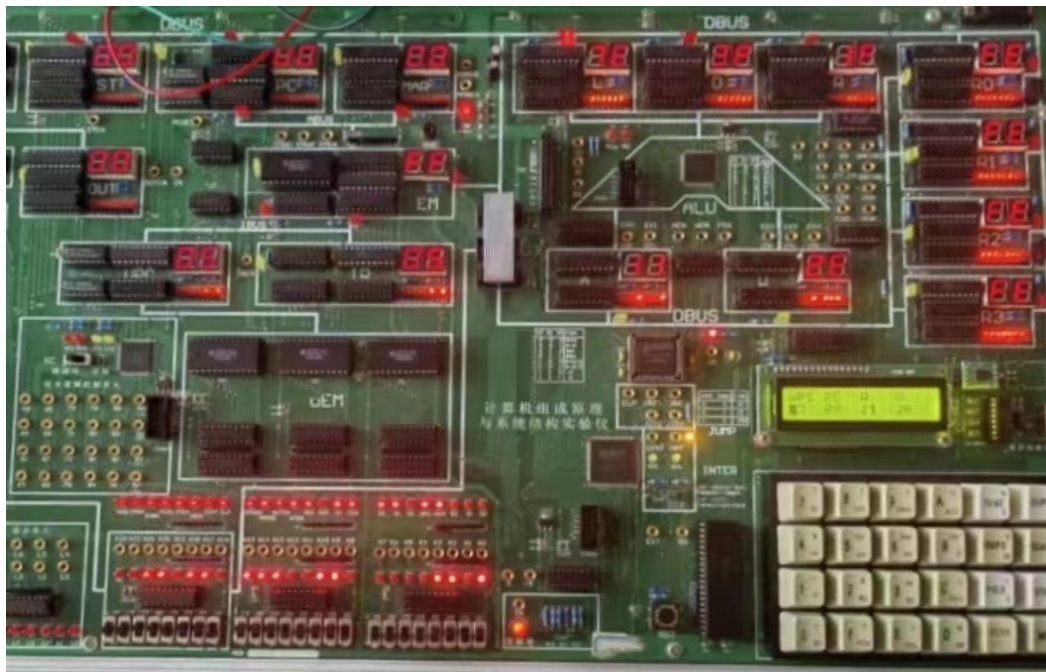
第二步: 根据题意可知: $S_2 S_1 S_0=000$, $X_2 X_1 X_0=110$, $RWR=0$

$C_{23} \sim C_0=1111, 1111, 1111, 1011, 1101, 1000=FF, FB, D8 H$

第三步: 进入微程序存储器模式(μEM 状态), 给 04H 地址单元打入 $ff\ ff\ ffH$, 给 05H 地址单元打入 $FF, FB, D8 H$ 。

第四步: 再进入“内部寄存器”方式, 将 μPC 、PC、A、W 寄存器分别设置成 05、00、11、22, 按动 STEP 键, 观测 A、W 和 R1 寄存器的结果。

3. 实验结果



4. 总结

通过这次实验，我们了解到译码器、微指令结构的基本工作原理：

译码器是一种多输入多输出的组合逻辑电路，负责将二进制代码翻译为特定的对象（如逻辑电平等），功能与编码器相反。译码器的工作原理是将输入的数字信号转换成输出信号，这个过程需要通过一定的逻辑电路来实现。译码器的输入端通常是数字信号，可以是二进制、八进制或十六进制等不同的进制形式，而输出端通常是模拟信号或者其他数字信号，可以是电压、电流、频率等不同的形式。

微指令是指在微程序控制的计算机中，同时发出的控制信号所执行的一组微操作。微指令是由同时发出的控制信号的有关信息汇集起来形成的。将一条指令分成若干条微指令，按次序执行就可以实现指令的功能。若干条微指令可以构成一个微程序，而一个微程序就对应了一条机器指令。一条机器指令的功能是若干条微指令组成的序列来实现的，即一条机器指令所完成的操作分成若干条微指令来完成，由微指令进行解释和执行，这个微指令序列通常叫做微程序。

而设计微指令的方法通常有以下几种：

直接表示法：特点是操作控制字段中的每一位代表一个微命令。优点是简单直观，其输出直接用于控制。

编码表示法：编码表示法是把一组相斥性的微命令信号组成一个小组（即一个字段），然后通过小组（字段）译码器对每一个微命令信号进行译码，译码输出作为操作控制信号。

混合表示法：这种方法是把直接表示法与字段编码法混合使用，以便能综合考虑指令字长、灵活性、执行微程序速度等方面的要求。

运算器实验

1. 目的

1. 学习数据处理部件的工作方式控制。
2. 学习机器语言程序的运行过程。

2. 步骤

实验任务：

1. 计算 $37H + 56H$ 后左移一位的值。
2. 把 $36H$ 取反后同 $54H$ 相与的值送入 R1 寄存器。
- 3*. 计算 $36H + 45H + 23H$ 的值，并送入 PC 寄存器。

2.1 实验环境（软硬件配置）

实验箱一台

2.2 任务 1

步 骤	功能
第一步：J1 与 J3 相连	
第二步：37H 送到 A 寄存器，56H 送到 W 寄存器；	
第三步：S2 S1 S0=000；	A+W=>D
第四步：L 寄存器即为所求	D << 1=>L

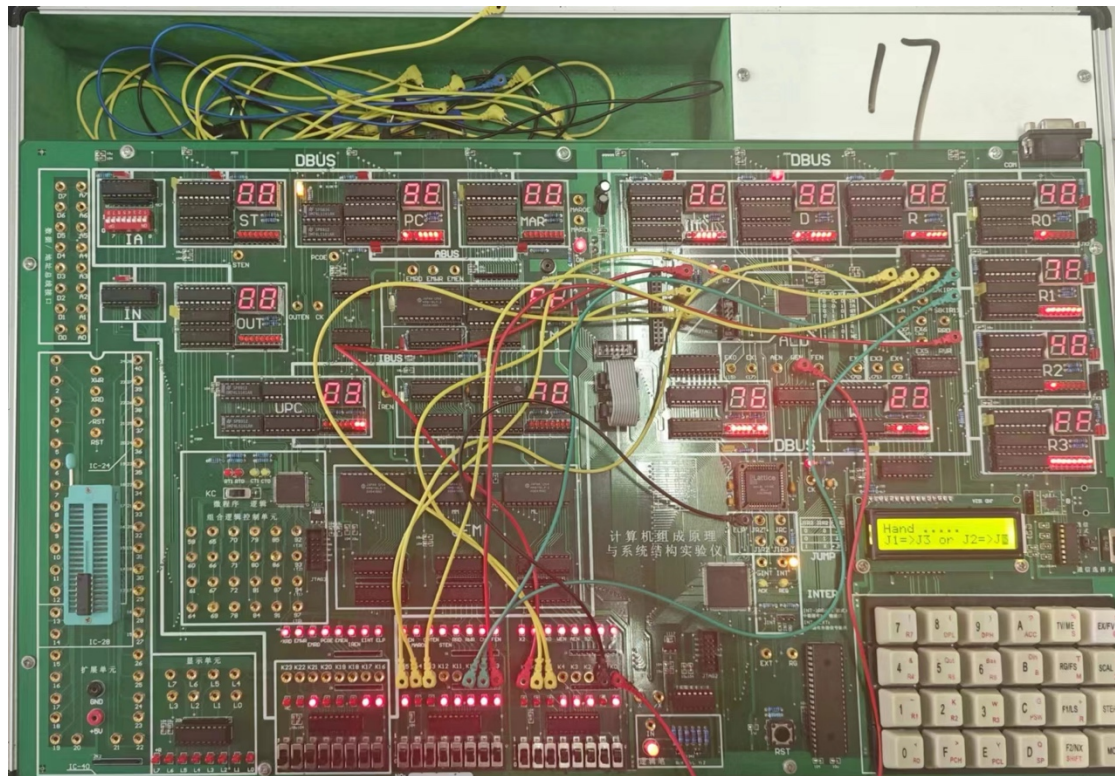
2.3 任务 2

步 骤	功能
第一步：J1 与 J3 相连	
第二步：36H 送到 A 寄存器，54H 送到 W 寄存器；	
第三步：J1 与 J2 相连，S2 S1 S0=110；	$\sim A \Rightarrow D$
第四步：X2 X1 X0=100，AEN=0；	D =>A
第五步：S2 S1 S0=011；	A & W=>D
第六步：X2 X1 X0=100，RWR=0，SB SA=01；	D=>R1

2.4 任务 3

步 骤	功能
第一步：J1 与 J3 相连	
第二步：36H 送到 A 寄存器，54H 送到 W 寄存器；	
第三步：J1 与 J2 相连，S2 S1 S0=110；	$\sim A \Rightarrow D$
第四步：X2 X1 X0=100，AEN=0；	D =>A
第五步：S2 S1 S0=011；	A & W=>D
第六步：X2 X1 X0=100，RWR=0，SB SA=01；	D=>R1

3. 实验结果



4. 总结

1. 复杂命令可以通过有序执行一串简单命令来完成。
2. 命令的顺序很重要，如④⑤操作的顺序不能颠倒。
3. 有些命令相互无关，如操作③和④，它们不仅可以颠倒次序，如果有两个处理器的话，还可以让二者在两个不同的处理器中同时进行——命令级并行，在现代的“多核“计算机中就是这样。
4. 改造控制总线，可能使多步操作合成一步。例如第⑤步用到控制总线 $k4k3k2=101$ ；第⑥步用到控制总线 $k8k7k6=100$ ，两步用到的控制总线没有重复，于是，可以在第⑤步发出： $k8k7k6 \quad k4k3k2=100 \ 101$ ，则两步顺序操作在一步内完成。这些方法和技术在实际系统中都有应用。