

---

《计算机组成与设计》  
实验指导书

(第三版)

姜立秋 肖大薇 于晓海 编著

2015年6月

## 前 言

《计算机组成与设计》课程主要讲述计算机系统各部件的组成以及设计原理，包括 CPU 组成、ALU 运算器原理、存储器与控制器原理等，最终达到从系统的、整机的角度，完整的、准确的理解和掌握计算机系统的理论和原理的目的，为计算机专业技术的学习打下良好的基础。

《计算机组成与设计》是计算机科学与技术专业的一门专业基础课。本课程实践性强，因此，该实验课程必须有相应的实践开发平台支持，这里选用的是 EL-JY-II 计算机组成原理实验系统。该系统是一个 16 位的计算机模型实验系统，简单实用，且运算器、数据通路、控制器、控制台各部分划分清晰。通过实验旨在提高学生的实践能力、提高学生对计算机整体和各组成部分的深入理解、提高学生对计算机系统的综合分析与设计能力、增强学生理论联系实际的能力、增强学生的工程观念和创新意识，从而培养高素质的应用型人才。

本课程主要讲授内容如下：

- ◆ 运算器实验
- ◆ 存储器读写实验
- ◆ 总线控制实验
- ◆ 微程序控制器原理与设计实验
- ◆ 简单模型机组成原理实验
- ◆ 带移位运算的模型机组成原理实验
- ◆ 复杂模型机组成原理实验
- ◆ 具有简单中断处理功能的模型机实验

# 计算机组成原理实验系统介绍

EL-JY-II 型计算机组成原理实验系统是为计算机组成原理课的教学实验而研制的，涵盖了目前流行教材的主要内容，能完成主要的基本部件实验和整机实验，可供大学本科、专科、成人高校以及各类中等专业学校学习《计算机组成原理》、《微机原理》和《计算机组成和结构》等课程提供基本的实验条件，同时也可供计算机其它课程的教学和培训使用。

## 1、系统组成

本系统由两大部分组成：

### A、基板：

本部分是 8 位机和 16 位机的公共部分，包括以下几个部分：

- 1) 数据输入和输出电路
- 2) 显示及监控电路
- 3) 脉冲源及时序电路
- 4) 数据和地址总线
- 5) 8255 扩展实验电路
- 6) 单片机控制电路和键盘操作部分
- 7) 与 PC 机通讯的接口电路
- 8) 主存储器电路
- 9) 微代码输入及显示电路
- 10) 电源电路
- 11) CPLD 实验板（选件）
- 12) 自由实验区（面包板）

### B、CPU 板：

本 CPU 板为 16 位机，其数据总线为 16 位，地址总线为 8 位，包括以下几个部分：

- 1) 运算器电路
- 2) 微程序控制器电路
- 3) 寄存器堆电路
- 4) 程序计数器电路
- 5) 指令寄存器电路
- 6) 指令译码电路
- 7) 地址寄存器电路
- 8) 数据和控制总线电路

其中，运算器电路中的累加器电路由 74LS181 及其外围电路组成，此外所有的其它电路都由 ALTERA 公司的 FPGA— EP1K10 实现。板上的 JTAG 口、芯片 EPC2LC20、跳线 J1—J6 用于配置 EP1K10。当跳线 J1—J6 均跳至 EPC2 OFF 时，可通过 JTAG 口直接配置 EP1K10，但断电后需重新配置。当跳线 J1—J6 均跳至 EPC2 ON 时，通过 EPC2LC20 来配置 EP1K10。系统出厂时，已将配置文

件烧录进 EPC2LC20。由于 EPC2LC20 为非易失性器件，故每次上电时可自动配置 EP1K10，无需重新烧录。

C、底板使用说明：

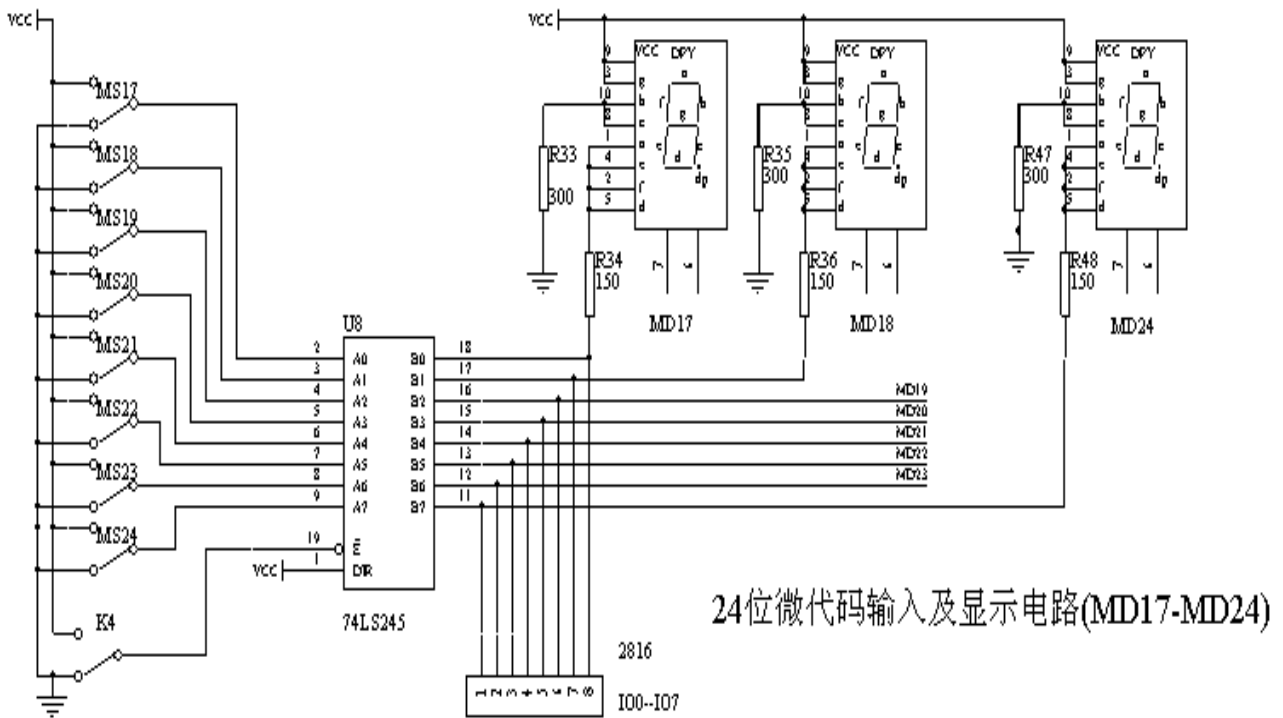
底板的系统布局如下图所示：

- 1) 控制开关电路用于开关方式下各种控制信号的输入，每个开关对应一个 LED 指示灯。当 LED 点亮时，表示相对应的开关输出为高电平，反之则为低电平。
- 2) 键盘及监控显示灯用于键盘方式下的实验，其用法见各实验说明（注：当开关 K4 为“ON”时键盘被封锁）。



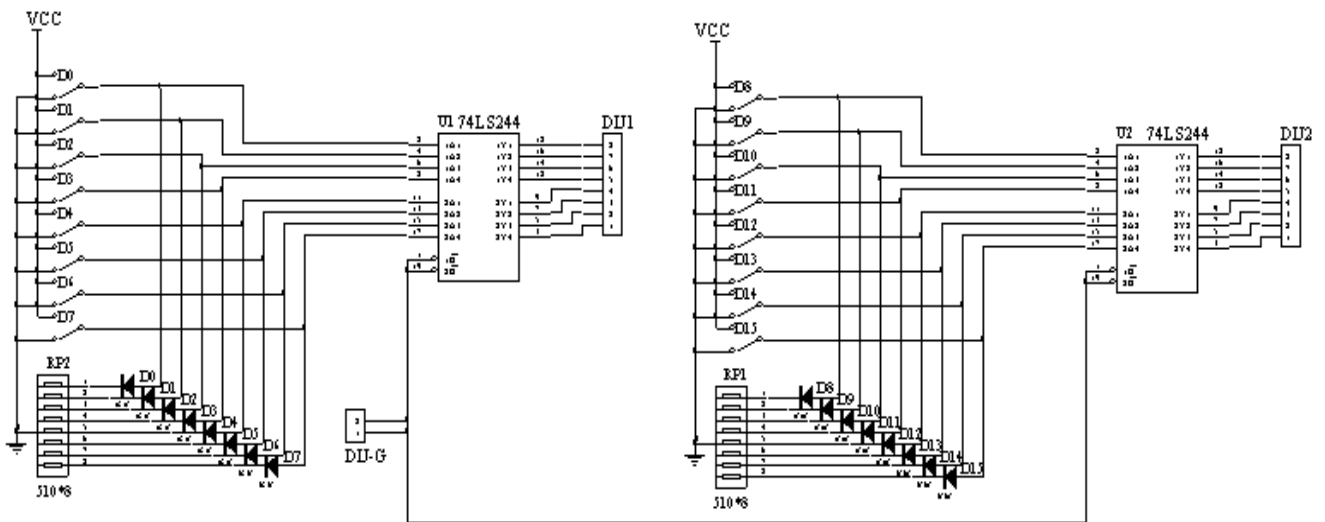
图1 基板布局

- 3) 24 位微代码输入及显示电路用于读写微程序，其原理如下图所示（仅以 MD17-MD24 为例，MD1-MD16 电路与此相同）。



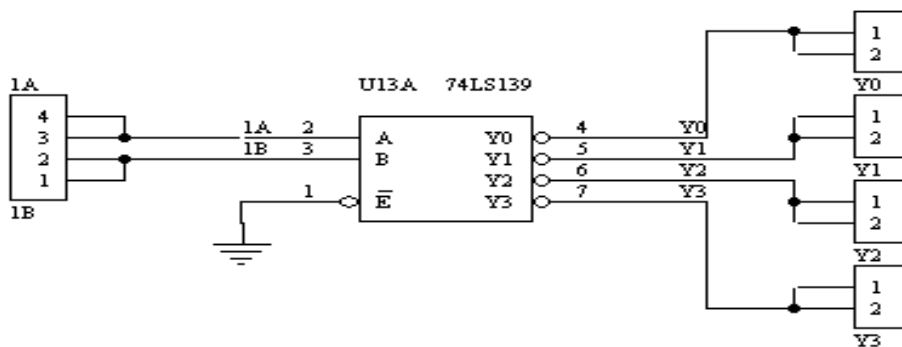
当 K4 为“OFF”（VCC）时，24 位开关无效，24 个数码管的显示由 2816 的数据口决定，用于键盘方式读写微代码和开关方式读微代码。当 K4 为“ON”（GND）时，24 位开关有效，24 个数码管显示每一位开关的状态（“0”或“1”），用于开关方式写微代码。

- 4) 脉冲源及时序电路用于开关方式下产生时序信号；F、F/2、F/4、F/8 分别为固定时钟频率输出端，其频率分别为 1M、500K、250K、125K。Fin 为时钟输入，可接至 F、F/2、F/4、F/8 中的任何一个输出；按下“单脉冲”键时，T<sub>+</sub>、T<sub>-</sub>端分别产生一个正脉冲、一个负脉冲；按下“单步”键时，T<sub>1</sub>、T<sub>2</sub>、T<sub>3</sub>、T<sub>4</sub> 端依次产生一个正脉冲，用于程序的单步运行；按下“启动”键时，T<sub>1</sub>、T<sub>2</sub>、T<sub>3</sub>、T<sub>4</sub> 端依次产生连续的正脉冲，用于程序的全速运行；按下“停止”键时，T<sub>1</sub>、T<sub>2</sub>、T<sub>3</sub>、T<sub>4</sub> 端不产生脉冲，用于中止程序运行。
- 5) 16 位数据输入电路如下图所示：DIJ2 为高 8 位数据，DIJ1 为低 8 数据，当 DIJ—G 为低电平时，DIJ2、DIJ1 输出 16 位开关量数据，否则为高阻态。



16 位输出显示电路由四个数码管和四片可编程逻辑芯片 GAL16V8 组成。GAL16V8 为显示提供译码和驱动，当 W/R、D-G 均为低电平时，将 D15—D0 的数据送至数码管显示。

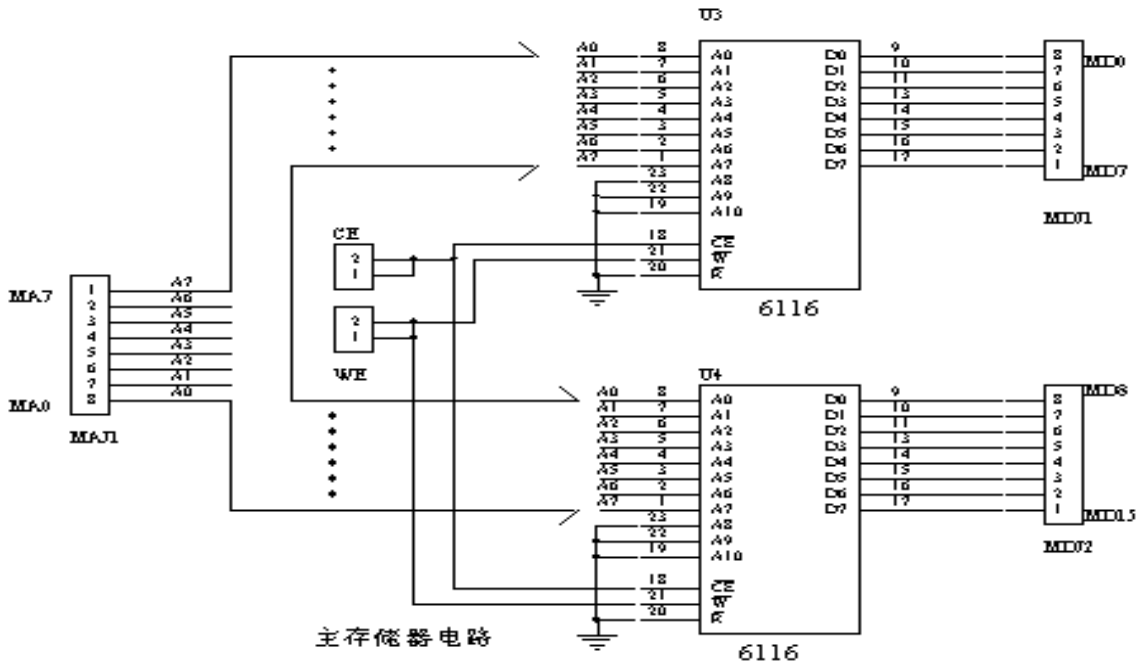
6) I/O 控制电路由一片 74LS139 构成，用于为外部器件提供选通信号。其原理和逻辑关系如下图所示：



输 入		输 出			
1A	1B	Y0	Y1	Y2	Y3
0	0	0	1	1	1
1	0	1	0	1	1
0	1	1	1	0	1
1	1	1	1	1	0

7) 显示灯电路：该电路有四个绿色 LED 指示灯。当输入为高电平时，点亮相应位置的 LED 灯。

8) 主存储器电路其原理如下图所示：



9) 8255 接口电路的数据、地址、控制线和 PA 口以及 PB 口的第四位均通过单排插针引出。底板上的数据总线 BD15—BD0（三组接口相同，可互换）和 CPU 板上的数据线相连，地址总线 AD7—AD0（三组接口相同，可互换）和 CPU 板上的地址线相连。

## 2、参考内容

1. 运算器实验
2. 存储器读写实验
3. 总线控制实验
4. 微程序控制器原理与设计实验
5. 微程序设计实验
6. 简单模型机组成原理实验
7. 带移位运算的模型机组成原理实验
8. 复杂模型机组成原理实验

# 实验一 运算器实验

## 一、实验目的

1. 掌握运算器的组成及工作原理；
2. 了解4位函数发生器74LS181的组合功能，熟悉运算器执行算术操作和逻辑操作的具体实现过程；
3. 验证带进位控制的74LS181的功能。

## 二、预习要求

1. 复习本次实验所用的各种数字集成电路的性能及工作原理；
2. 预习实验步骤，了解实验中要求的注意之处。

## 三、实验设备

EL-JY-II型计算机组成原理实验系统一套，排线若干。

## 四、电路组成

本模块由算术逻辑单元ALU 74LS181 (U7、U8、U9、U10)、暂存器74LS273 (U3、U4、U5、U6)、三态门74LS244 (U11、U12) 和控制电路(集成于EP1K10内部)等组成。电路图见图1-1(a)、1-1(b)。

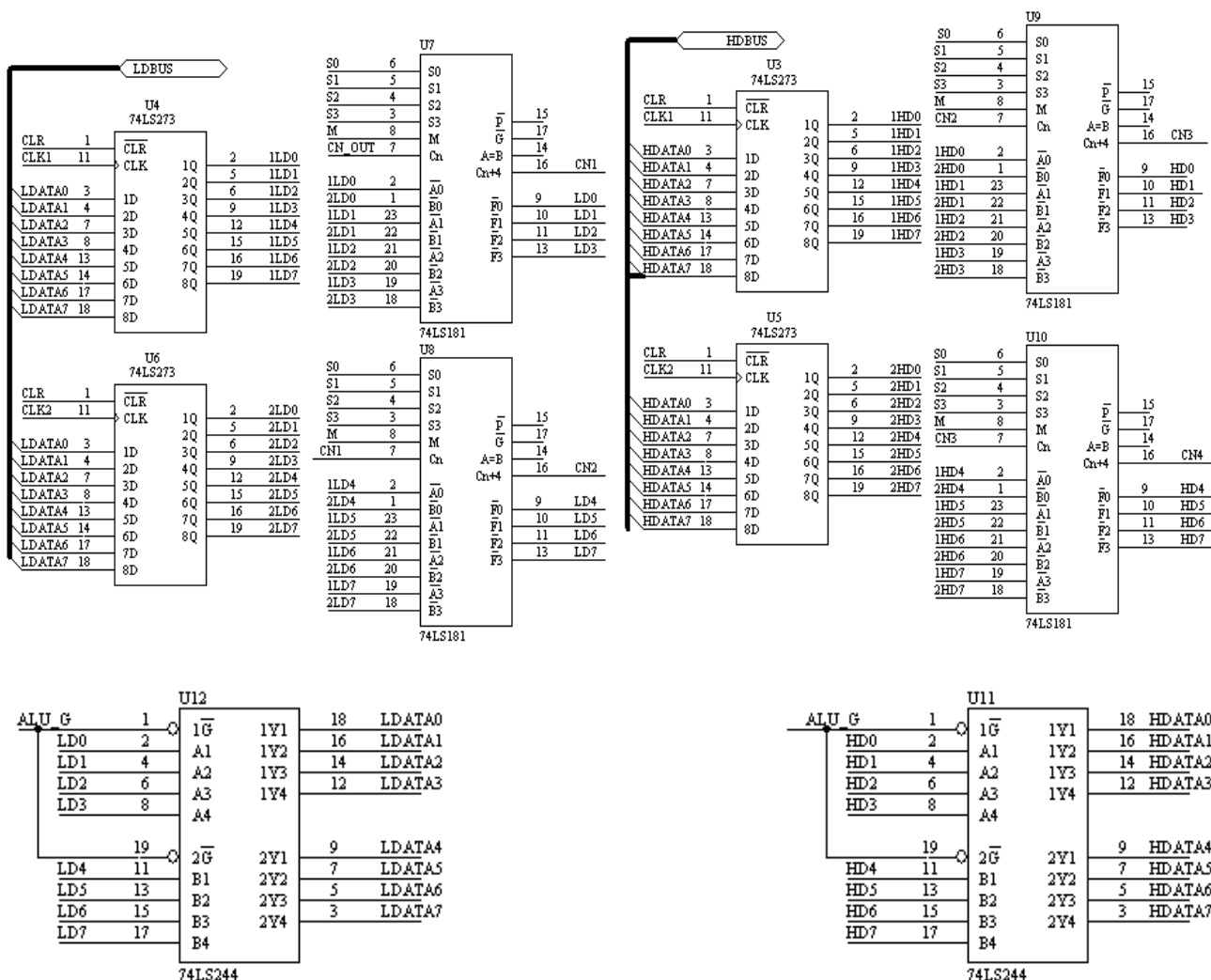




图 1-1 (a) ALU 电路

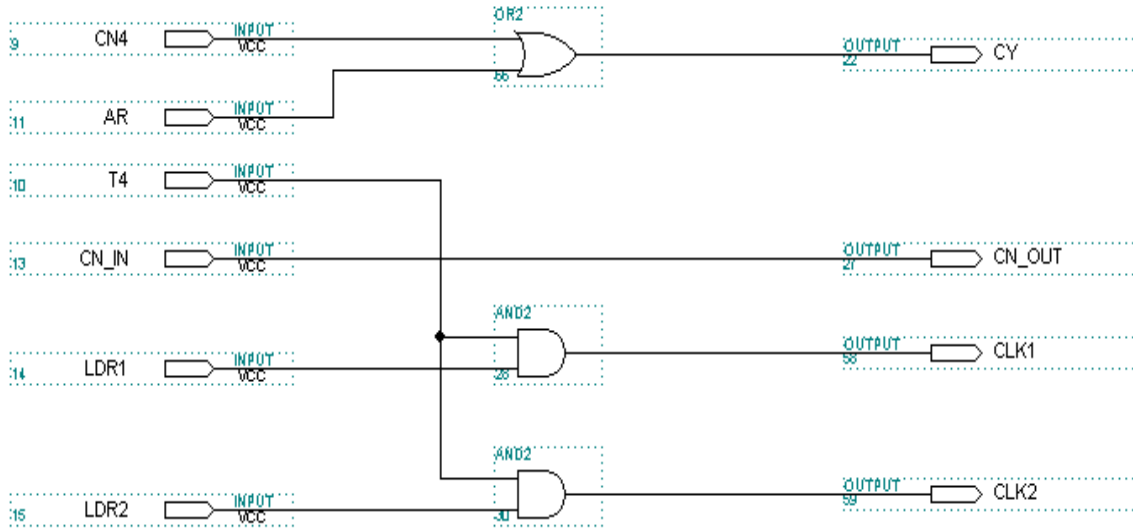


图 1-1 (b) ALU 控制电路

算术逻辑单元 ALU 是由四片 74LS181 构成。74LS181 的功能控制条件由 S3、S2、S1、S0、M、Cn 决定。高电平方式的 74LS181 的管脚分配、输出端功能符号和功能详见图 1-2(a)、图 1-2(b)和表 1-1。

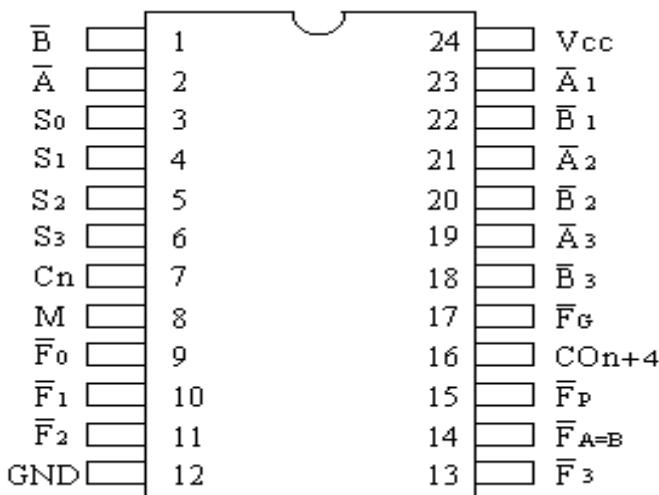


图 1-2 (a) 74LS181 管脚分配

$A_0 \sim A_3$	运算数输入端(低电平有效)
$B_0 \sim B_3$	运算数输入端(低电平有效)
Cn	进位输入端
$C_{O_{n+4}}$	进位输出端
$F_0 \sim F_3$	运算输出端(低电平有效)
$F_{A=B}$	比较输出端
$F_G$	进位产生输出端(低电平有效)
$F_P$	进位传输输出端(低电平有效)
M	工作方式控制
$S_0 \sim S_3$	功能选择

图 1-2 (b) 74LS181 输出端功能符号

74LS181 功能表见表 1-1，其中符号“+”表示逻辑“或”运算，符号“\*”表示逻辑“与”运算，符号“/”表示逻辑“非”运算，符号“加”表示算术加运算，符号“减”表示算术减运算。

表 1-1 74LS181 功能表

选择				M=1 逻辑操作	M=0		算术操作	
S3	S2	S1	S0		Cn=1 (无进位)	Cn=0 (有进位)		
0	0	0	0	$F=\bar{A}$	$F=A$		$F=A$ 加 1	
0	0	0	1	$F=\bar{(A+B)}$	$F=A+B$		$F=(A+B)$ 加 1	
0	0	1	0	$F=\bar{A*B}$	$F=A+B$		$F=(A+B)$ 加 1	
0	0	1	1	$F=0$	$F=-1$		$F=0$	
0	1	0	0	$F=\bar{(A*B)}$	$F=A$ 加 $A*B$		$F=A$ 加 $A*B$ 加 1	
0	1	0	1	$F=\bar{B}$	$F=(A+B)$ 加 $A*B$		$F=(A+B)$ 加 $A*B$ 加 1	
0	1	1	0	$F=\bar{(A*B+A*/B)}$	$F=A$ 减 $B$ 减 1		$F=A$ 减 $B$	
0	1	1	1	$F=A*/B$	$F=A*/B$ 减 1		$F=A*/B$	
1	0	0	0	$F=\bar{A+B}$	$F=A$ 加 $A*B$		$F=A$ 加 $A*B$ 加 1	
1	0	0	1	$F=\bar{/(A*B+A*/B)}$	$F=A$ 加 $B$		$F=A$ 加 $B$ 加 1	
1	0	1	0	$F=B$	$F=(A+B)$ 加 $A*B$		$F=(A+B)$ 加 $A*B$ 加 1	
1	0	1	1	$F=A*B$	$F=A*B$ 减 1		$F=A*B$	
1	1	0	0	$F=1$	$F=A$ 加 $A$		$F=A$ 加 $A$ 加 1	
1	1	0	1	$F=A+B$	$F=(A+B)$ 加 $A$		$F=(A+B)$ 加 $A$ 加 1	
1	1	1	0	$F=A+B$	$F=(A+B)$ 加 $A$		$F=(A+B)$ 加 $A$ 加 1	
1	1	1	1	$F=A$	$F=A$ 减 1		$F=A$	

四片 74LS273 构成两个 16 位数据暂存器，运算器的输出采用三态门 74LS244。它们的管脚分配和引出端功能符号详见图 1-3 (a)、图 1-3 (b) 和图 1-4 (a) 和图 1-4 (b)。

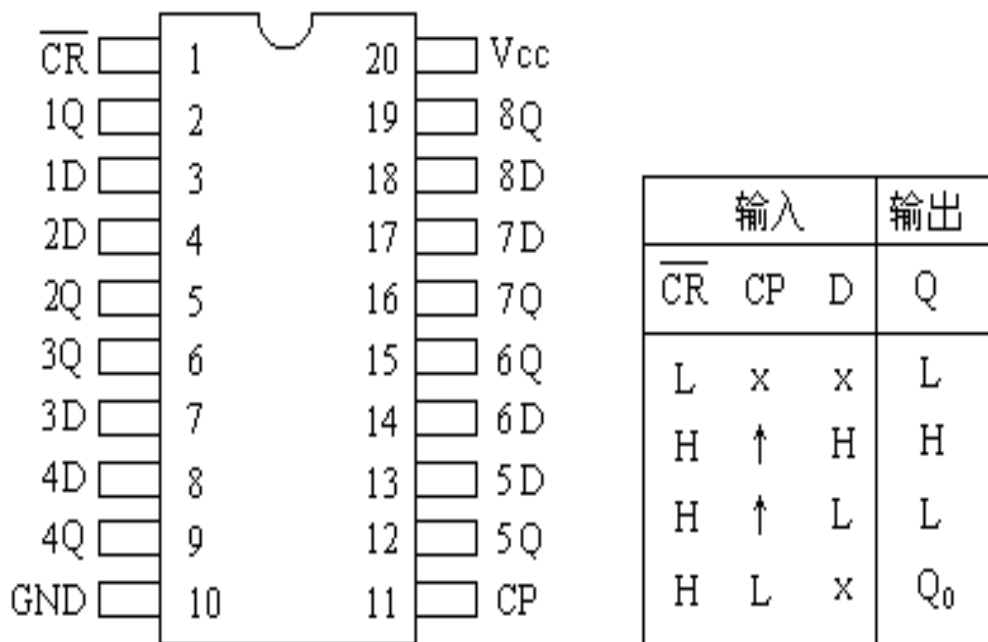


图 1-3 (a) 74LS273 管脚分配

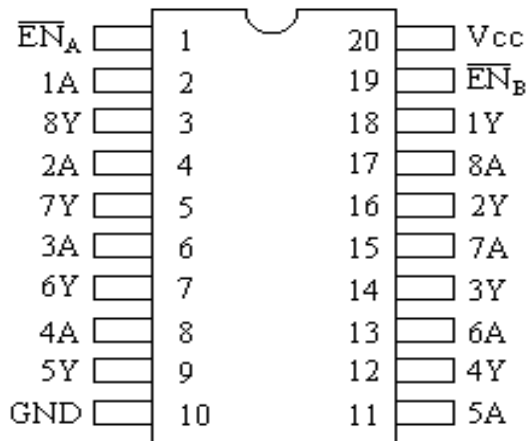


图 1-3 (b) 74LS273 功能表

输入		输出	
EN	A	Y	
L	L	L	H - 高电平
L	H	H	L - 低电平
H	X	Z	X - 任意
			Z - 高阻

图 1-4 (a) 74LS244 管脚分配

图 1-4 (b) 74LS244 功能

## 五、工作原理

运算器的结构见图 1-5:

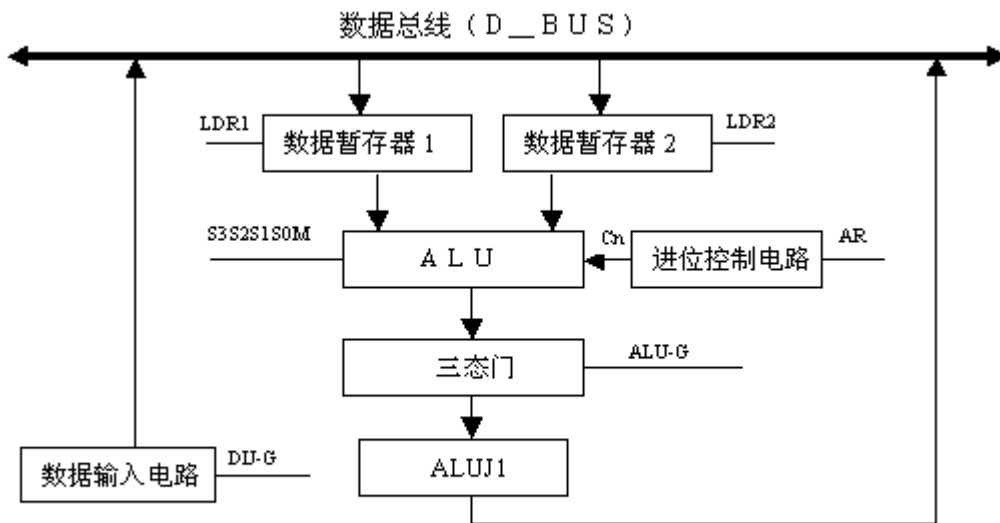


图 1-5 运算器的结构框图

算术逻辑单元 ALU 是运算器的核心。集成电路 74LS181 是 4 位运算器，四片 74LS181 以并 / 串形式构成 16 位运算器。它可以对两个 16 位二进制数进行多种算术或逻辑运算，74LS181 有高电平和低电平两种工作方式，高电平方式采用原码输入输出，低电平方式采用反码输入输出，这里采用高电平方式。

三态门 74LS244 作为输出缓冲器由 ALU-G 信号控制，ALU-G 为“0”时，三态门开通，此时其输出等于其输入；ALU-G 为“1”时，三态门关闭，此时其输出呈高阻。

四片 74LS273 作为两个 16 数据暂存器，其控制信号分别为 LDR1 和 LDR2，当 LDR1 和 LDR2 为高电平有效时，在 T4 脉冲的前沿，总线上的数据被送入暂存器保存。

## 六、实验内容

验证 74LS181 运算器的逻辑运算功能和算术运算功能。

## 七、实验步骤

### 单片机键盘操作方式实验

注：在进行单片机键盘控制实验时，必须把开关 K4 置于“OFF”状态，否则系统处于自锁状态，无法进行实验。

#### 1、实验连线（键盘实验）

实验连线如图 1-6 所示。

（连线时应按如下方法：对于横排座，应使排线插头上的箭头面向自己插在横排座上；对于竖排座，应使排线插头上的箭头面向左边插在竖排座上。注意：F4 只用一个排线插头孔）

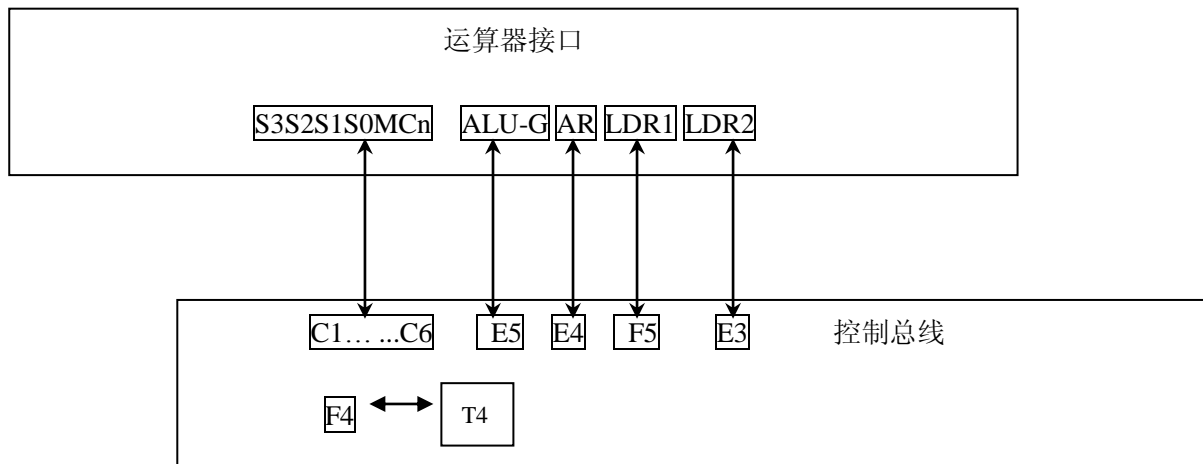


图 1-6 键盘实验连线图

#### 2、实验过程

- (1) 拨动清零开关 CLR，使其指示灯灭。再拨动 CLR，使其指示灯亮。
- (2) 在监控滚动显示【CLASS SELECT】时按【实验选择】键，显示【ES--\_\_】输入 01 或 1，按【确认】键，监控显示为【ES01】，表示准备进入实验一程序，也可按【取消】键来取消上一步操作，重新输入。
- (3) 再按【确认】键，进入实验一程序，监控显示【InSt--】，提示输入运算指令，输入两位十六进制数（参考表 1-2 和表 1-1），选择执行哪种运算操作，按【确认】键。
- (4) 监控显示【Lo=0】，此处 Lo 相当于表 1-1 中的 M，默认为“0”，进行算术运算，也可以输入“1”，进行逻辑运算。按【确认】，显示【Cn=0】，默认为“0”，由表 1-1 可见，此时进行带进位运算，也可输入“1”，不带进位运算（注：如前面选择为逻辑运算，则 Cn 不起作用）。按【确认】，显示【Ar=1】，使用默认值“1”，关闭进位输出。也可输入“0”，打开进位输出，按【确认】。

(5) 监控显示【DATA】，提示输入第一个数据，输入十六进制数【1234H】，按【确认】，显示【DATA】，提示输入第二个数据，输入十六进制数【5678H】，按【确认】键，监控显示【FINISH】，表示运算结束，可从数据总线显示灯观察运算结果，CY 指示灯显示进位输出的结果。按【确认】后监控显示【ES01】，可执行下一运算操作。

表 1-2 运算指令关系对照表

运算指令 ( S3 S2 S1 S0)	输入数据 (十六进制)
0 0 0 0	00 或 0
0 0 0 1	01 或 1
0 0 1 0	02 或 2
0 0 1 1	03 或 3
0 1 0 0	04 或 4
0 1 0 1	05 或 5
0 1 1 0	06 或 6
0 1 1 1	07 或 7
1 0 0 0	08 或 8
1 0 0 1	09 或 9
1 0 1 0	0A 或 A
1 0 1 1	0B 或 B
1 1 0 0	0C 或 C
1 1 0 1	0D 或 D
1 1 1 0	0E 或 E
1 1 1 1	0F 或 F

在给定 LT1=1234H、LT2=5678H 的情况下，改变运算器的功能设置，观察运算器的输出，填入表 1-3 中，并和理论值进行比较和验证：

表 1-3 练习表

LT1	LT2	S3S2S1S0	M=0 (算术运算)		M=1 (逻辑运算)
			Cn=1 (无进位)	Cn= 0 (有进位)	
		00 或 0	F=	F=	F=
		01 或 1	F=	F=	F=
		02 或 2	F=	F=	F=
		03 或 3	F=	F=	F=
		04 或 4	F=	F=	F=
		05 或 5	F=	F=	F=
		06 或 6	F=	F=	F=
		07 或 7	F=	F=	F=
		08 或 8	F=	F=	F=
		09 或 9	F=	F=	F=
		0A 或 A	F=	F=	F=
		0B 或 B	F=	F=	F=
		0C 或 C	F=	F=	F=
		0D 或 D	F=	F=	F=
		0E 或 E	F=	F=	F=
		0F 或 F	F=	F=	F=

## 实验二 存储器读写实验

### 一、实验目的

1. 掌握半导体静态随机存储器 RAM 的特性和使用方法。
2. 掌握地址和数据在计算机总线的传送关系。
3. 了解运算器和存储器如何协同工作。

### 二、预习要求

预习半导体静态随机存储器 6116 的功能。

### 三、实验设备

EL-JY-II 型计算机组成原理实验系统一套，排线若干。

### 四、电路组成

电路图见图 2-1，6116 的管脚分配和功能见图 2-2 (a)、2-2 (b)：

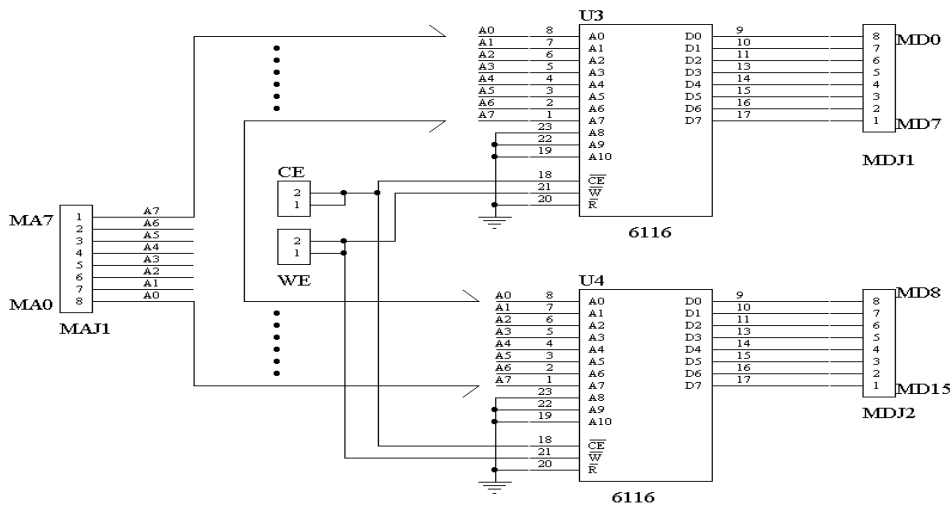


图 2-1 存储器电路

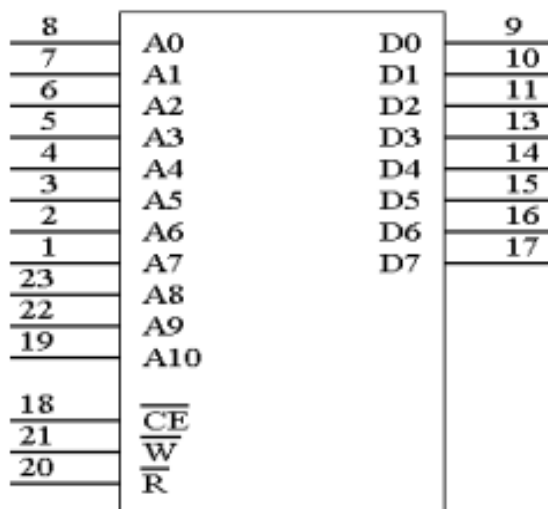


图 2-2 (a) 6116 管脚分配

CE	W	R	输入/输出
H	X	X	不选择
L	H	L	读
L	$\overline{L}$	H	写
L	$\overline{L}$	$\overline{L}$	写

图 2-2 (b) 6116 功能

## 五、工作原理

实验中的静态存储器由 2 片 6116 (2K×8) 构成, 其数据线 D0~D15 接到数据总线, 地址线 A0~A7 由地址锁存器 74LS273(集成于 EP1K10 内)给出。黄色地址显示灯 A7-A0 与地址总线相连, 显示地址总线的内容。绿色数据显示灯与数据总线相连, 显示数据总线的内容。

因地址寄存器为 8 位, 接入 6116 的地址 A7-A0, 而高三位 A8-A10 接地, 所以其实际容量为  $2^8=256$  字节。6116 有三个控制线, /CE (片选)、/R (读)、/W (写)。其写时间与 T3 脉冲宽度一致。

当 LARI 为高时, T3 的上升沿将数据总线的低八位打入地址寄存器。当 WEI 为高时, T3 的上升沿使 6116 进入写状态。

## 六、实验内容

学习静态 RAM 的存储方式, 往 RAM 的任意地址里存放数据, 然后读出并检查结果是否正确。

**注: 6116 为静态随机存储器, 如果掉电, 所存的数据全部丢失!**

## 七、实验步骤

### 单片机键盘操作方式实验

**注: 在进行单片机键盘控制实验时, 必须把 K4 开关置于“OFF”状态, 否则系统处于自锁状态, 无法进行实验。**

### 1. 实验连线

实验连线图如图 2-3 所示。

连线时应按如下方法: 对于横排座, 应使排线插头上的箭头面向自己插在横排座上; 对于竖排座, 应使排线插头上的箭头面向左边插在竖排座上。(注意: F3 只用一个排线插头孔)

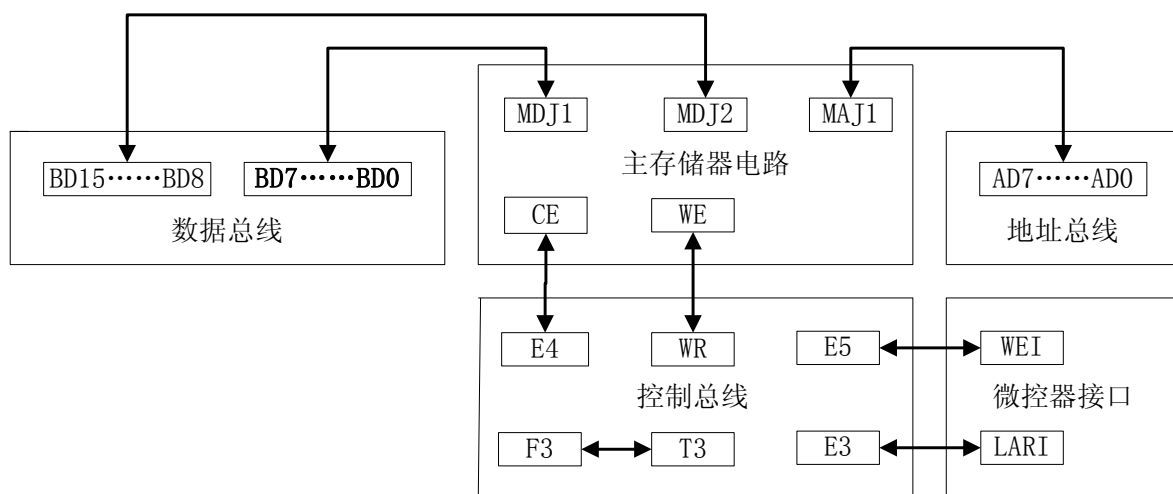


图 2-3 键盘实验接线图

### 2. 写数据

- (1) 拨动清零开关 CLR, 使其指示灯显示状态为亮—灭—亮。
- (2) 在监控指示灯滚动显示【CLASS SELECT】时按【实验选择】键, 显示【ES--\_ \_】输入 03 或 3, 按【确认】键, 监控指示灯显示为【ES03】, 表示准备进入实验三程序, 也可按【取消】键来取消上一

步操作，重新输入。再按【确认】键，进入实验三程序。

- (3) 监控指示灯显示为【CtL= - -】，输入 1,表示准备对 RAM 进行写数据，在输入过程中，可按【取消】键进行输入修改，按【确认】键。
- (4) 监控指示灯显示【Addr- -】，提示输入 2 位 16 进制数地址，输入“00”按【确认】键，监控指示灯显示【dAtA】，提示输入写入存储器该地址的数据（4 位 16 进制数），输入“3344”按【确认】键，监控指示灯显示【PULSE】，提示输入单步，按【单步】键，完成对 RAM 一条数据的输入，数据总线显示灯（绿色）显示“0011001101000100”，即数据“3344”，地址显示灯显示“0000 0000”，即地址“00”。
- (5) 监控指示灯重新显示【Addr- -】，提示输入第二条数据的 2 位十六进制的地址。重复上述步骤，按表 2-1 输入 RAM 地址及相应的数据。

表 2-1 数据表

地址（十六进制）	数据（十六进制）
00	3333
71	3434
42	3535
5A	5555
A3	6666
CF	ABAB
F8	7777
E6	9D9D

### 3. 读数据及校验数据

- (1) 按【取消】键退出到监控指示灯显示为【ES03】，或按【RST】退到步骤 2 初始状态进行实验选择。
- (2) 拨动清零开关 CLR，使其指示灯显示状态为亮一灭一亮。在监控指示灯显示【ES03】状态下，按【确认】键。
- (3) 监控指示灯显示为【CtL= - -】，输入 2,表示准备对 RAM 进行读数据，按【确认】键。
- (4) 监控指示灯显示【Addr- -】，提示输入 2 位 16 进制数地址，输入“00”，按【确认】键，监控指示灯显示【PULSE】，提示输入单步，按【单步】键，完成对 RAM 一条数据的读出，数据总线显示灯（绿色）显示“0011001101000100”，即数据“3344”，地址显示灯显示“0000 0000”，即地址“00”。
- (5) 监控指示灯重新显示【Addr- -】，重复上述步骤读出表 2—1 的所有数据，注意观察数据总线显示灯和地址显示灯之间的对应关系，检查读出的数据是否正确。

**注：6116 为静态随机存储器，如果掉电，所存的数据全部丢失！**



## 实验三 总线控制实验

### 一、实验目的：

- 1、了解总线的概念及其特性。
- 2、掌握总线的传输控制特性。

### 二、实验设备：

EL-JY-II 型计算机组成原理实验系统一套，排线若干。

### 三、实验说明

#### 1、总线的基本概念

总线是多个系统部件之间进行数据传送的公共通路，是构成计算机系统的骨架。借助总线连接，计算机在系统各部件之间实现传送地址、数据和控制信息的操作。因此，所谓总线就是指能为多个功能部件服务的一组公用信息线。

#### 2、实验原理说明

在本实验中，挂接在数据总线上的有输入设备、输出设备、存储器和加法器。为了使它们的输出互不干扰，就需要这些设备都有三态输出控制，且任意两个输出控制信号不能同时有效。其结构如下图所示：

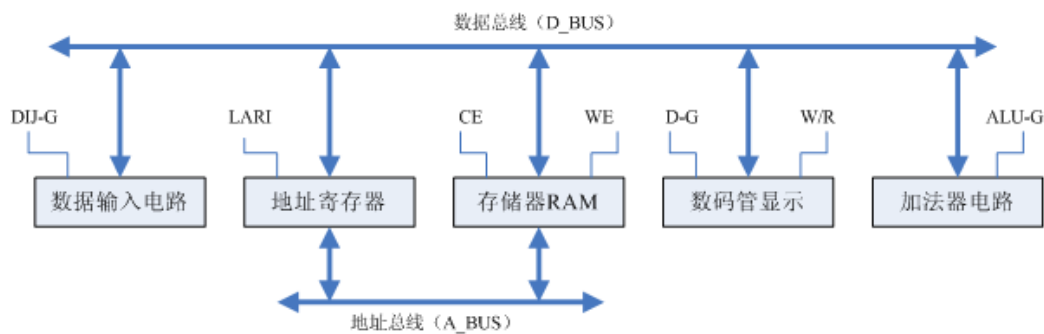


图 3-1 总线结构图

其中，数据输入电路和加法器电路结构见图 1-5，地址寄存器和存储器电路见图 2-1、2-3。数码管显示电路用可编程逻辑芯片 ATF16V8B 进行译码和驱动，D-G 为使能信号，W/R 为写信号。当 D-G 为低电平时，W/R 的下降沿将数据线上的数据打入显示缓冲区，并译码显示。

#### 本实验的流程为：

- (1) 输入设备将一个数打入 LT1 寄存器。
- (2) 输入设备将一个数打入 LT2 寄存器。
- (3) LT1 与 LT2 寄存器中的数相加。
- (4) 输入设备将另一个数打入地址寄存器。
- (5) 将两数之和写入当前地址的存储器中。
- (6) 将当前地址的存储器中的数用数码管显示出来。

### 四、实验连线

本实验采用开关方式，连线见下图。（连线时应按如下方法：对于横排座，应使排线插头上的箭头面

向自己插在横排座上；对于竖排座，应使排线插头上的箭头面向左边插在竖排座上)

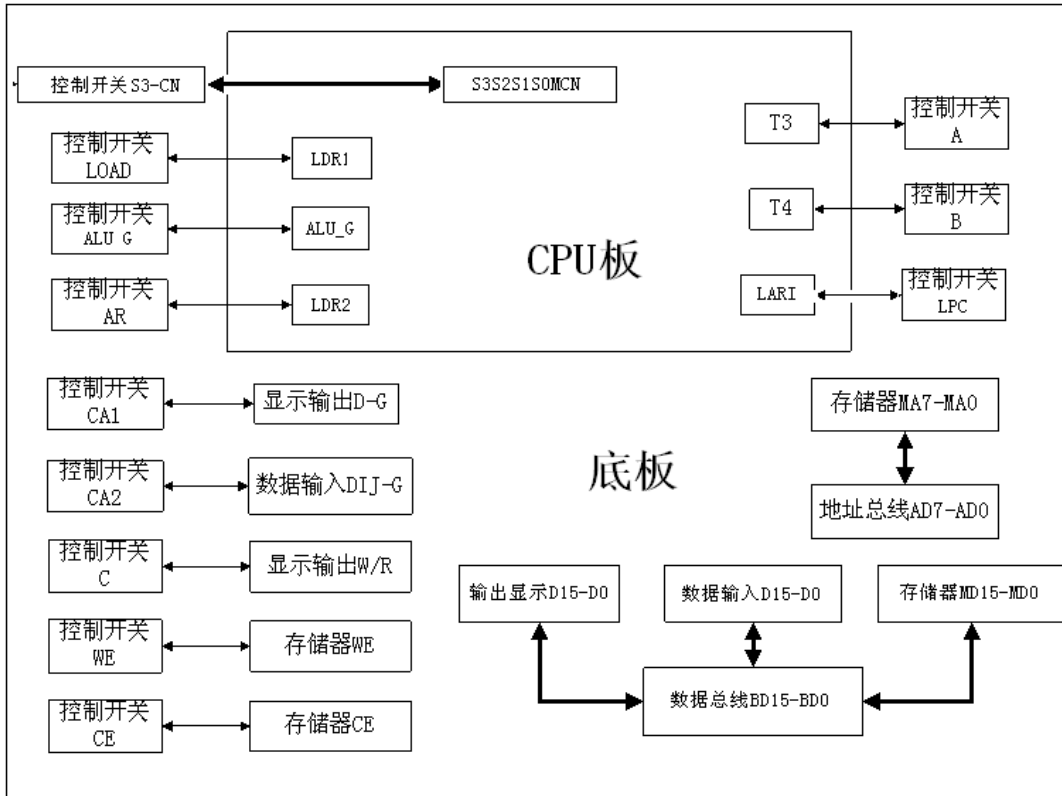


图 3-2 总线控制实验接线图

## 五、实验步骤

- 1、按照上图所示将所有连线接好。
- 2、总线初始化。关闭所有三态门置控制开关 ALU\_G=1(加法器控制信号), CA1=1(显示输出), CA2=1(数据输入), CE=1(存储器片选)。其它控制信号为 LOAD=0, AR=0, LPC=0, C=1, WE=1, A=1, B=1。
- 3、将 D15—D0 拨至 “0001001000110100”，置 CA2=0, LOAD=1, 然后置 LOAD=0, 将 “1234H” 打入 LT1 寄存器。
- 4、将 D15—D0 拨至 “0101011001111000”，置 AR=1, 然后置 AR=0, 将 “5678H” 打入 LT2 寄存器。
- 5、将 S3S2S1S0MCN 拨至 “100101”，计算两数之和。
- 6、将 D7—D0 拨至 “00000001”，置 LPC=1, 然后置 LPC=0, 将 “01H” 打入地址寄存器。
- 7、置 CA2=1, ALU-G=0, WE=0, CE=0, 将上述计算结果写入当前地址的存储器中。然后置 CE=1, WE=1。
- 8、置 ALU-G=1, CE=0, CA1=0, C=0, 将当前地址的存储器中的数输出至数码管, 然后置 C=1, CE=1, CA1=1。

## 六、实验结果

照以上 8 步操作完成后，输出显示电路 LED 上显示 “68AC”。

## 实验四 微程序控制器原理实验

### 一、实验目的

掌握微程序控制器的组成及工作过程；

### 二、预习要求

1. 复习微程序控制器工作原理；
2. 预习本电路中所用到的各种芯片的技术资料。

### 三、实验设备

EL-JY-II 型计算机组成原理实验系统一台，连接线若干。

### 四、电路组成

微程序控制器的原理图见图 4-1(a)、4-1(b)、4-1(c)。

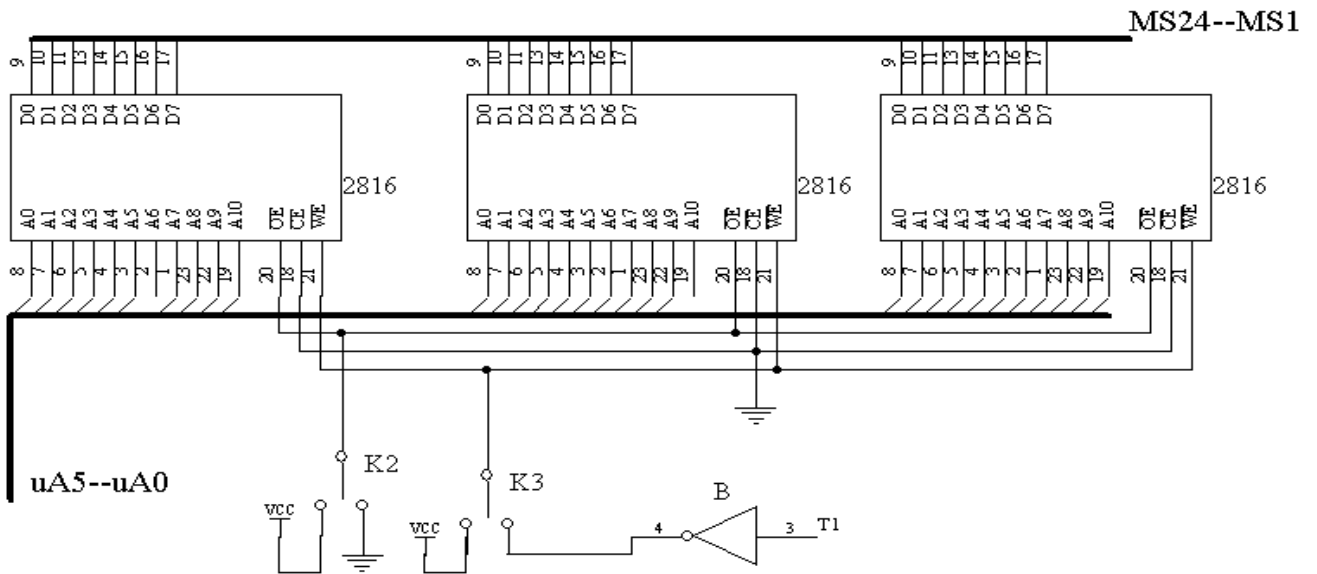


图 4-1 (a) 控制存储器电路

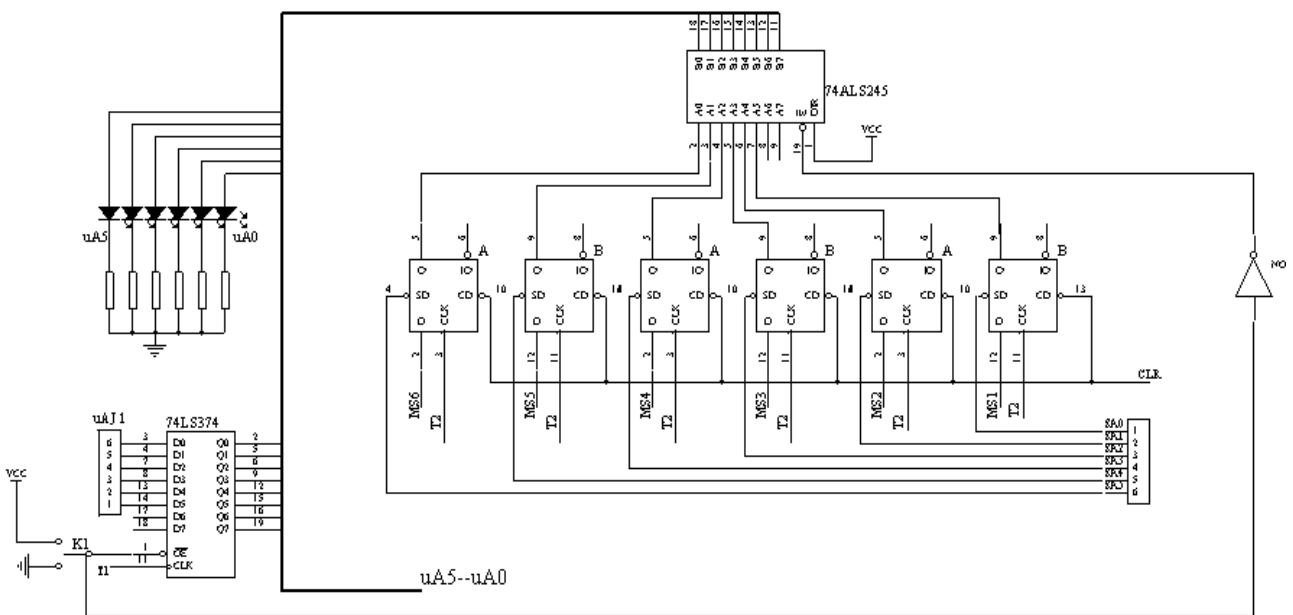


图 4-1 (b) 微地址形成电路

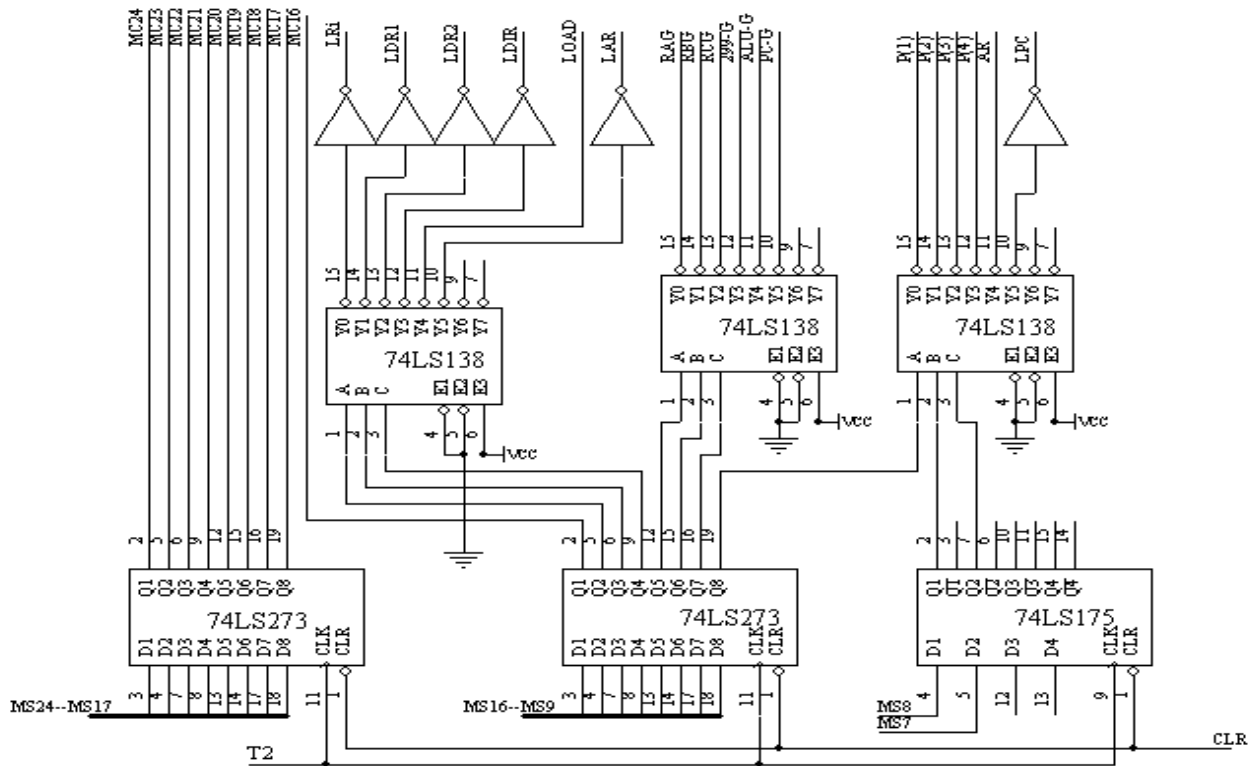


图 4-1 (c) 微指令译码电路

以上电路除一片三态输出 8D 触发器 74LS374、三片 EEPROM2816 和一片三态门 74LS245，其余逻辑控制电路均集成于 EP1K10 内部。28C16、74LS374、74LS245 芯片的技术资料分别见图 4-2~图 4-4。

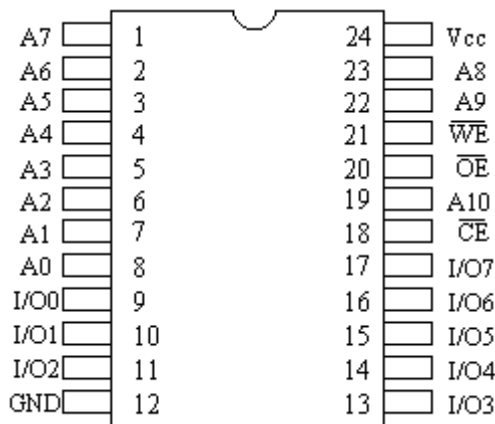


图 4-2 (a) 28C16 引脚

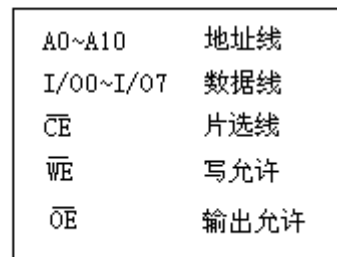


图 4-2 (b) 28C16 引脚说明

工作方式	/CE	/OE	/WE	输入/输出
读	L	L	H	数据输出
后备	H	×	×	高阻
字节写	L	H	L	数据输入
字节擦除	L	12V	L	高阻
写禁止	×	×	H	高阻
写禁止	×	L	×	高阻
输出禁止	×	H	×	高阻

图 4-2 (c) 28C16 工作方式选择

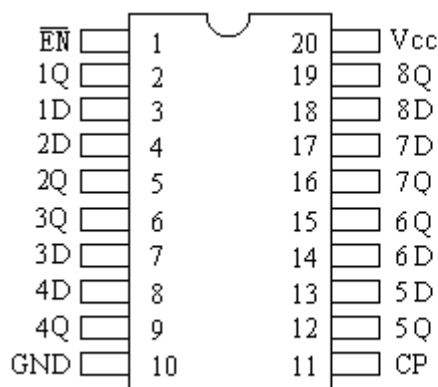


图 4-3 (a) 74LS374 引脚

输入			输出
EN	CP	D	Q
L	↑	H	H
L	↑	L	L
L	L	x	Q <sub>0</sub>
H	x	x	Z

图 4-3 (b) 74LS374 功能

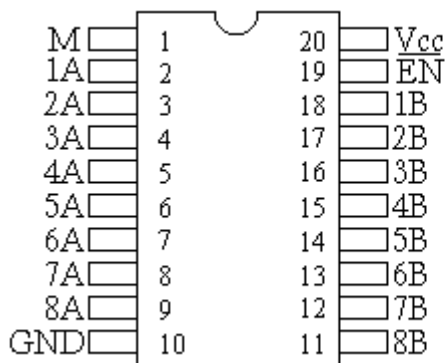


图 4-4 (a) 74LS245 引脚

输入		功能
EN	M	
L	L	A数据至B总线
L	H	B数据至A总线
H	x	隔离

图 4-4 (b) 74LS245 功能

## 五、工作原理:

### 1、写入微指令

在写入状态下, 图 4-1 (a) 中 K2 须为高电平状态, K3 须接至脉冲/T1 端, 否则无法写入。MS1—MS24 为 24 位写入微代码, 在键盘方式时由键盘输入, 在开关方式时由 24 位微代码开关提供。uA5—uA0 为写入微地址, 在键盘方式时由键盘输入, 在开关方式时由微地址开关提供。K1 须接低电平使 74LS374 有效, 在脉冲 T1 时刻, uAJ1 的数据被锁存形成微地址 (如图 4-1 (b) 所示), 同时写脉冲将 24 位微代码写入当前微地址中 (如图 4-1 (a) 所示)。

### 2、读出微指令

在写入状态下, 图 4-1 (a) 中 K2 须为低电平状态, K3 须接至高电平。K1 须接低电平使 74LS374 有效, 在脉冲 T1 时刻, uAJ1 的数据被锁存形成微地址 uA5—uA0 (如图 4-1 (b) 所示), 同时将当前微地址的 24 位微代码由 MS1—MS24 输出。

### 3、运行微指令

在运行状态下, K2 接低电平, K3 接高电平。K1 接高电平。使控制存储器 2816 处于读出状态, 74LS374 无效因而微地址由微程序内部产生。在脉冲 T1 时刻, 当前地址的微代码由 MS1—MS24 输出; T2 时刻将 MS24—MS7 打入 18 位寄存器中, 然后译码输出各种控制信号 (如图 4-1 (c) 所示, 控制信号功能见实验五); 在同一时刻 MS6—MS1 被锁存, 然后在 T3 时刻, 由指令译码器输出的 SA5—SA0 将其

中某几个触发器的输出端强制置位，从而形成新的微地址  $uA5—uA0$ ，这就是将要运行的下一条微代码的地址。当下一个脉冲  $T1$  来到时，又重新进行上述操作。

#### 4、脉冲源和时序：

在开关方式下，用脉冲源和时序电路中“脉冲源输出”作为时钟信号， $f$  的频率为  $1MHz$ ， $f/2$  的频率为  $500KHz$ ， $f/4$  的频率为  $250KHz$ ， $f/8$  的频率为  $125KHz$ ，可根据实验自行选择一种频率的方波信号。每次实验时，只需将“脉冲源输出”的四个方波信号任选一种接至“信号输入”的“ $fin$ ”，时序电路即可产生 4 种相同频率的等间隔的时序信号  $T1~T4$ 。电路提供了四个按钮开关，以供对时序信号进行控制。工作时，如按一下“单步”按钮，机器处于单步运行状态，即此时只发送一个 CPU 周期的时序信号就停机，波形见图 4-8。利用单步运行方式，每次只读一条微指令，可以观察微指令的代码与当前微指令的执行结果。如按一下“启动”按钮，机器连续运行，时序电路连续产生如图 4-9 的波形。此时，按一下“停止”按钮，机器停机。

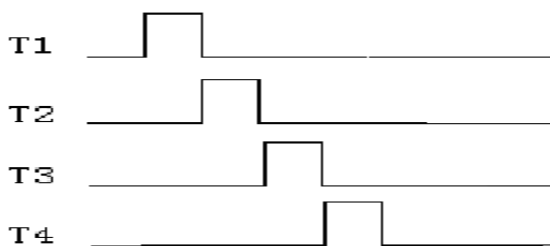


图 4-8 单步运行波形图

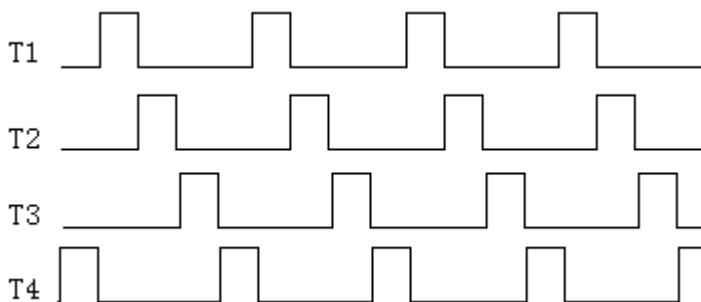


图 4-9 全速运行波形图

按动“单脉冲”按钮，“ $T+$ ”和“ $T-$ ”输出图 4-10 的波形：



图 4-10 单脉冲输出波形

各个实验电路所需的时序信号端均已分别连至“控制总线”的“ $T1$ 、 $T2$ 、 $T3$ 、 $T4$ ”，实验时只需将“脉冲源及时序电路”模块的“ $T1$ 、 $T2$ 、 $T3$ 、 $T4$ ”端与“控制总线”的“ $T1$ 、 $T2$ 、 $T3$ 、 $T4$ ”端相连，即可给电路提供时序信号。

对于键盘方式的实验，所需脉冲信号由系统监控产生（其波形与脉冲方式相同），并通过控制总线的  $F1—F4$  输出。实验时只需将“控制总线”的“ $F4F3F2F1$ ”与“ $T4T3T2T1$ ”相连，即可给电路提供时序信号。

#### 六、实验内容：

往 EEPROM 里任意写 24 位微代码，并读出验证其正确性。

## 七、实验步骤

### 单片机键盘操作方式实验

在进行单片机键盘控制实验时，必须把 K4 开关置于“OFF”状态，否则系统处于自锁状态，无法进行实验。

#### 2. 实验连线：

实验连线图如图 4-11 所示。

连线时应按如下方法：对于横排座，应使排线插头上的箭头面向自己插在横排座上；对于竖排座，应使排线插头上的箭头面向左边插在竖排座上。

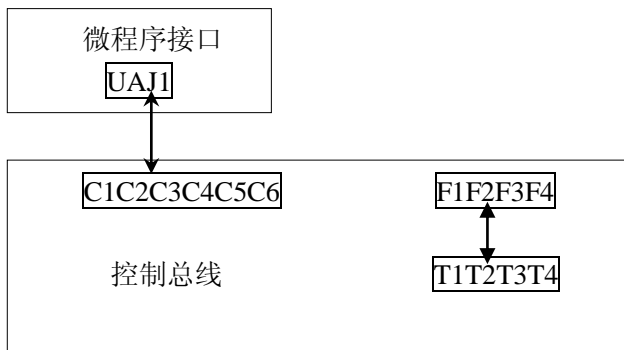


图 4-11 键盘实验接线图

#### 3. 写微代码：

将开关 K1K2K3K4 拨到写状态即 K1 off、K2 on、K3 off、K4 off，其中 K1、K2、K3 在微程序控制电路，K4 在 24 位微代码输入及显示电路上。

- 在监控指示灯滚动显示【CLASS SELEcT】状态下按【实验选择】键，显示【ES--\_ \_】输入 04 或 4，按【确认】键，显示为【ES04】，表示准备进入实验四程序，也可按【取消】键来取消上一步操作，重新输入。
- 再按下【确认】键，显示为【CtL1=\_】，表示对微代码进行操作。输入 1 显示【CtL1\_1】，表示写微代码，也可按【取消】键来取消上一步操作，重新输入。按【确认】。
- 监控显示【U-Addr】，此时输入【000000】6 位二进制数表示的微地址，然后按【确认】键，监控指示灯显示【U\_CodE】，这时输入微代码【000001】，该微代码是用 6 位十六进制数来表示前面的 24 位二进制数，注意输入微代码的顺序，先右后左，此过程中可按【取消】键来取消上一次输入，重新输入。按【确认】键则显示【PULSE】，按【单步】完成一条微代码的输入，重新显示【U-Addr】提示输入表 4-1 第二条微代码地址。
- 按照上面的方法输入表 4-1 微代码，观察微代码与微地址显示灯的对应关系（注意输入微代码的顺序是由右至左）。

表 4-1 实验四微代码表

微地址（二进制）	微代码（十六进制）
000000	000001
000001	000002

000010	000003
000011	015FC4
000100	012FC8
001000	018E09
001001	005B50
010000	005B55
010101	06F3D8
011000	FF73D9
011001	017E00

### 3、读微代码：

(1) 先将开关 K1K2K3K4 拨到读状态即 K1 off、K2 off、K3 on、K4 off，按【RESET】按钮复位，使监控指示灯滚动显示【CLASS SELEcT】状态。

(2) 按【实验选择】键，显示【ES--\_ \_】输入 04 或 4，按【确认】键，显示【ES04】。按【确认】键。

(3) 监控显示【CtL1=\_】时，输入 2，按【确认】显示【U\_Addr】，此时输入 6 位二进制微地址，进入读微代码状态。再按【确认】显示【PULSE】，此时按【单步】键，监控显示【U\_Addr】，微地址指示灯显示输入的微地址，微代码显示电路上显示该地址对应的微代码，至此完成一条微指令的读过程。观察黄色微地址显示灯和微代码的对应关系，对照表 4-1 表检查微代码是否有错误，如有错误，可按步骤 2 重写这条微代码。



# 实验五 微程序设计实验

## 一、实验目的

深入掌握微程序控制器的工作原理，学会设计简单的微程序。

## 二、预习要求

1. 复习微程序控制器工作原理；
2. 复习计算机微程序的有关知识。

## 三、实验设备

EL-JY-II 型计算机组成原理实验系统一台，连接线若干。

## 四、微程序的设计

### 1. 微指令格式

设计微指令编码格式的主要原则是使微指令字短、能表示可并行操作的微命令多、微程序编写方便。

微指令的最基本成份是控制场，其次是下地址场。控制场反映了可以同时执行的微操作，下地址场指明下一条要执行的微指令在控存的地址。微指令的编码格式通常指控制场的编码格式，以下几种编码格式较普遍。

#### 1) 最短编码格式

这是最简单的垂直编码格式，其特点是每条微指令只定义一个微操作命令。采用此格式的微指令字短、容易编写、规整直观，但微程序长度长，访问控存取微指令次数增多从而使指令执行速度慢。

#### 2) 全水平编码格式

这种格式又称直接编码法，其特点是控制场每一位直接表示一种微操作命令。若控制场长  $n$  位，则至多可表示  $n$  个不同的微操作命令。

采用此格式的微指令字长，但可实现多个允许的微操作并行执行，微程序长度短，指令执行速度快。

3) 分段编码格式是将控制场分成几段。若某段长  $i$  位，则经译码，该段可表示  $2^i$  个互斥的即不能同时有效的微操作命令。

采用这种格式的微指令长度较短，而可表示的微操作命令较多，但需译码器。

### 2. 微程序顺序控制方式的设计

微程序顺序控制方式指在一条指令对应的微程序执行过程中，下一条微指令地址的确定方法，又叫后继地址生成方式。下面是常见的两种。

#### 1) 计数增量方式

这种方式的特点是微程序控制部件中的微地址中的微地址产生线路主要是微地址计数器 MPC。MPC 的初值由微程序首址形成线路根据指令操作码编码形成。在微程序执行过程中该计数器增量计数，产生下一条微指令地址。这使得微指令格式中可以不设置“下地址场”。缩短了微指令长度，也使微程序控制部件结构较简单。但微程序必须存放在控存若干连续单元中。

#### 2) 断定方式

微指令中设有“下地址场”，它指出下条微指令的地址，这使一条指令的微程序中的微指令在控存中不一定要连续存放。在微程序执行过程中，微程序控制部件中的微地址形成电路直接接受微指令下地址

场信息来产生下条微指令地址，微程序的首址也由此微地址形成电路根据指令操作码产生。

### 3. 本系统的微指令格式

微程序设计的关键技术之一是处理好每条微指令的下地址，以保证程序正确高效地进行。本系统采用分段编码的指令格式，采用断定方式确定下一条微指令的地址。

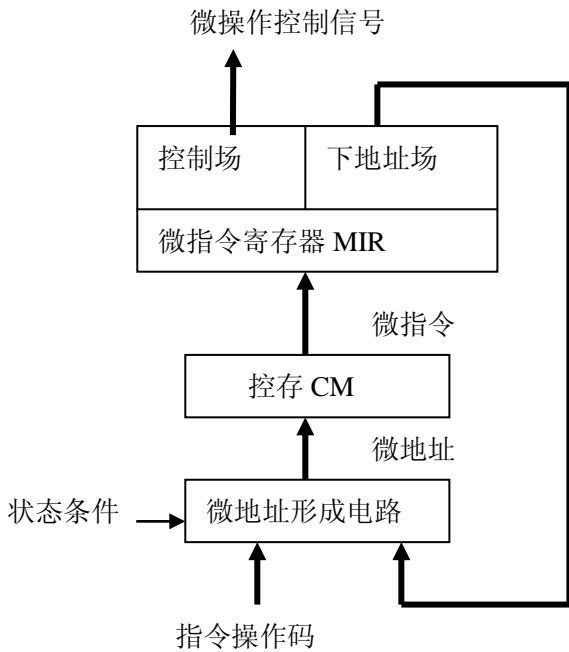


图 5-1 断定方式微程序控制部件示意图

每条微指令由 24 位组成，其控制位顺序如下：

24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
S3	S2	S1	S0	M	Cn	WE	1A	1B	F1	F2	F3	uA5	uA4	uA3	uA2	uA1	uA0						

微指令译码电路如下：

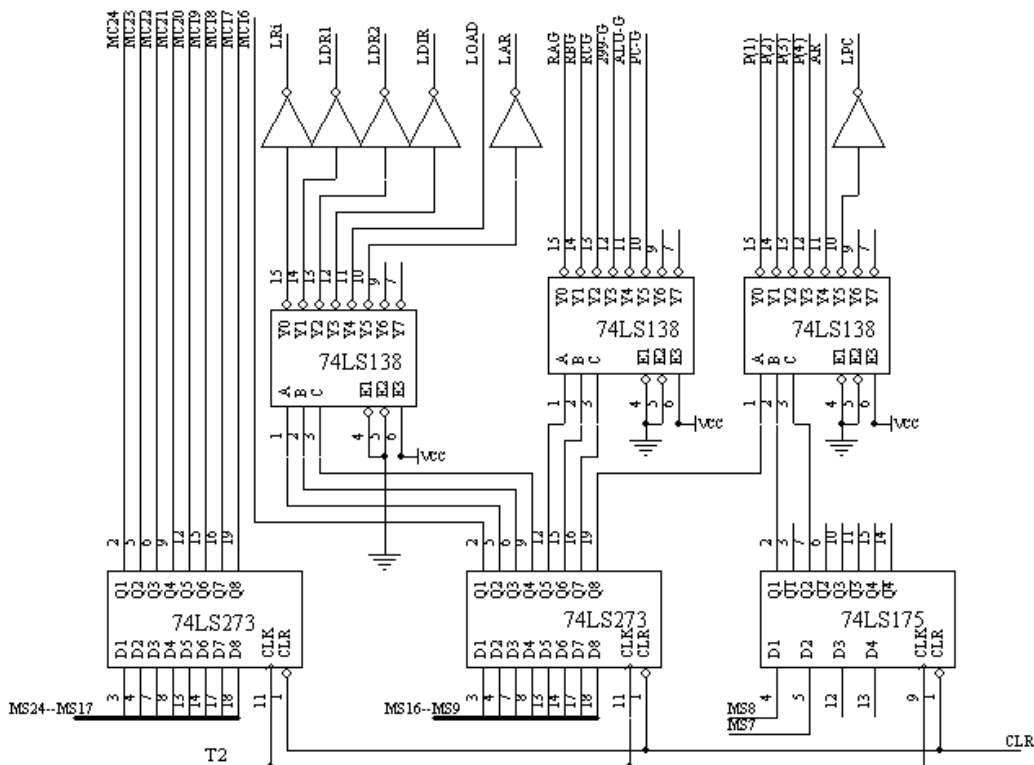


图 5—2 微指令译码电路

图中 MS24—MS16 对应于微指令的第 24—16 位，S3S2S1S0MCn 为运算器的方式控制，详见实验一和实验二；WE 为外部器件的读写信号，‘1’表示写，‘0’表示读；1A、1B 用于选通外部器件，通常接至底板 IO 控制电路的 1A1B 端，四个输出 Y0Y1Y2Y3 接外部器件的片选端。（注：再实验六及以后的实验中，Y3 被系统占用，用于输入中断，Y0Y1Y2 仍能被用户使用）

图 5—2 中 MS15—MS13 对应于微指令中的 F1，经锁存译码后产生 6 个输出信号：LRi、LDR1、LDR2、LDIR、LOAD、LAR。其中 LDR1、LDR2 为运算器的两个锁存控制（见实验一）；LDIR 为指令寄存器的锁存控制（见系统介绍中指令寄存器电路）；LRi 为寄存器堆的写控制，它与指令寄存器的第 0 位和第 1 位共同决定对哪个寄存器进行写操作（见系统介绍中寄存器堆电路和图 5-4）；LOAD 为程序计数器的置数控制，LAR 为地址寄存器的锁存控制（见系统介绍中程序计数器和地址寄存器电路）。以上 6 个输出信号均为 ‘1’ 有效。

图 5—2 中 MS12—MS10 对应于微指令中的 F2，经锁存译码后产生 6 个输出信号：RAG、RBG、RCG、299-G、ALU-G、PC-G。其中 RAG、RBG、RCG 分别为寄存器 Ax、Bx、Cx 的输出控制（见系统介绍中寄存器堆电路）；299-G 为移位寄存器的输出控制（见实验二）；ALU-G 为运算器的输出控制（见实验一）；PC-G 为程序计数器的输出控制（见系统介绍中程序计数器和地址寄存器电路）。以上信号均为 ‘0’ 有效。

图 5—2 中 MS9—MS9 对应于微指令中的 F3，经锁存译码后产生 6 个输出信号：P1、P2、P3、P4、AR、LPC。其中 P1、P2、P3、P4 位测试字，其功能是对机器指令进行译码，使微程序转入相应的微地址入口，从而实现微程序的顺序、分支和循环运行；AR 为运算器的进位输出控制（见实验一）；LPC 为程序计数器的时钟控制（见系统介绍中程序计数器电路）。以上信号均为 ‘1’ 有效。

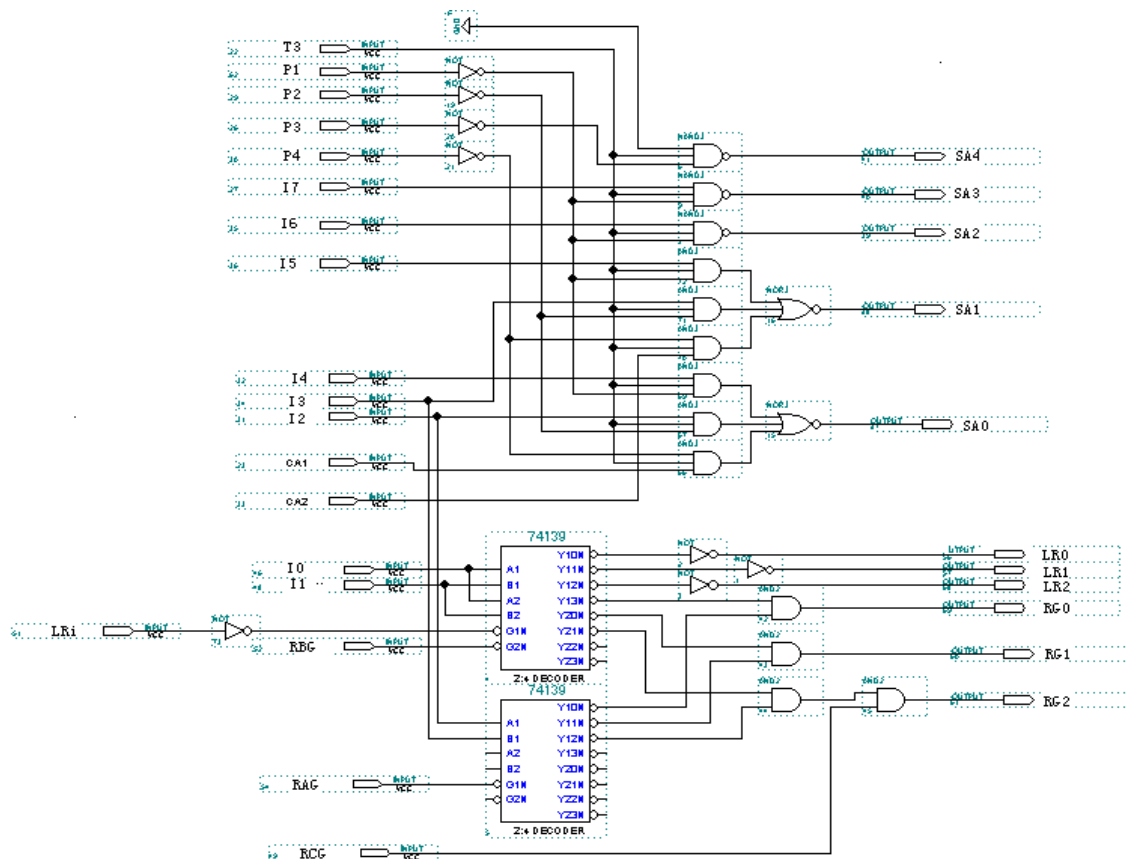


图 5-3 指令译码器电路

微指令中的 uA5-uA0 为 6 位的后续微地址。

F 1、F 2、F 3 三个字段的编码方案如表 5-1:

表 5-1

F1 字段		F2 字段		F3 字段	
15 14 13	选择	12 11 10	选择	9 8 7	选择
0 0 0	LDRi	0 0 0	RAG	0 0 0	P1
0 0 1	LOAD	0 0 1	ALU-G	0 0 1	AR
0 1 0	LDR2	0 1 0	RCG	0 1 0	P3
0 1 1	自定义	0 1 1	自定义	0 1 1	自定义
1 0 0	LDR1	1 0 0	RBG	1 0 0	P2
1 0 1	LAR	1 0 1	PC-G	1 0 1	LPC
1 1 0	LDIR	1 1 0	299-G	1 1 0	P4
1 1 1	无操作	1 1 1	无操作	1 1 1	无操作

### 五、实验内容:

编写几条可以连续运行的微代码，熟悉本实验系统的微代码设计方式。表 5-2 为几条简单的可以连续运行的二进制微代码表:

注意 UA5-----UA0 的编码规律，观察后续地址。

表 5-2 实验五微代码表

微地址 (二进制)	S3 S2 S1 S0	M CN WE 1A 1B	F1	F2	F3	UA5...UA0
000000	0 0 0 0	0 0 0 0 0 0	000	000	000	000001
000001	0 0 0 0	0 0 0 0 0 0	000	000	000	000010
000010	0 0 0 0	0 0 0 0 0 0	000	000	000	000011
000011	0 0 0 0	0 0 0 0 0 1	101	111	111	000100

000100	0 0 0 0 0 0 0 1 0	010	111	111	001000
001000	0 0 0 0 0 0 0 0 0	000	111	000	001001
001001	0 0 0 0 0 0 0 1 1	101	101	101	010000
010000	0 0 0 0 0 0 0 1 1	101	101	101	010101
010101	0 0 0 0 0 1 1 0 1	111	001	111	011000
011000	1 1 1 1 1 1 1 1 0	111	001	111	011001
011001	0 0 0 0 0 0 0 1 0	111	111	000	000000

以下举例说明微代码的含义：

- 1、微地址“000011”：读 Y1 设备上的数据，并将该数据打入地址寄存器。然后跳转至微地址“000100”。
- 2、微地址“000100”：读 Y1 设备上的数据，并将该数据打入运算暂存器 2，然后跳转至微地址“001000”。
- 3、微地址“011000”：运算暂存器 1 数据输出至数据总线，将该数据写入 Y1 设备，然后跳转至微地址“011001”。
- 4、微地址“011001”：读 Y1 设备上的数据，然后进行 P1 测试。由于未对指令寄存器操作，I7—I0 均为 0，强制置位无效，仍跳转至后续微地址“000000”。

## 六、实验步骤：

### I、单片机键盘操作方式实验

在进行单片机键盘控制实验时，必须把 K4 开关置于“OFF”状态，否则系统处于自锁状态，无法进行实验。

#### 1.实验连线

实验连线图如图 5-4 所示。

连线时应按如下方法：对于横排座，应使排线插头上的箭头面向自己插在横排座上；对于竖排座，应使排线插头上的箭头面向左边插在竖排座上。

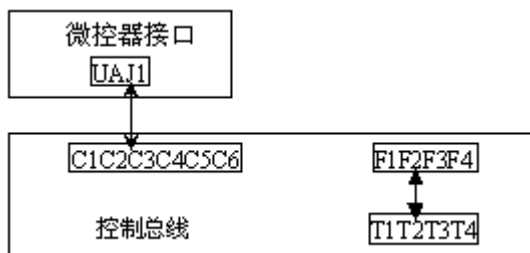


图 5—4 键盘实验连线图

#### 2.写微代码

(1) 将开关 K1K2K3K4 拨到写状态即 K1 off、K2 on、K3 off、K4 off，其中 K1、K2、K3 在微程序控制电路，K4 在 24 位微代码输入及显示电路上。在监控指示灯滚动显示【CLASS SELEcT】状态下按【实验选择】键，显示【ES--\_ \_】输入 05 或 5，按【确认】键，显示为【ES05】，再按下【确认】键。

(2) 监控显示为【CtL1=\_】，表示对微代码进行操作。输入 1 显示【CtL1\_1】，表示写微代码，按【确认】。

(3) 监控显示【U-Addr】，此时输入【000000】6 位二进制数表示的微地址，然后按【确认】键，监控指示灯显示【U\_CodE】，这时输入微代码【000001】，该微代码是用 6 位十六进制数来表示前面的 24 位

二进制数，注意输入微代码的顺序，先右后左，此过程中可按【取消】键来取消上一次输入，重新输入。按【确认】键则显示【PULSE】，按【单步】完成一条微代码的输入，重新显示【U-Addr】提示输入表 5-3 第二条微代码地址。

(4) 按照上面的方法输入表 5-3 微代码,观察微代码与微地址显示灯的对应关系（注意输入微代码的顺序是由右至左）。

表 5-3 实验五微代码表

微地址（二进制）	微代码（十六进制）
000000	000001
000001	000002
000010	000003
000011	015FC4
000100	012FC8
001000	018E09
001001	005B50
010000	005B55
010101	06F3D8
011000	FF73D9
011001	017E00

### 3.读微代码:

(1) 先将开关 K1K2K3K4 拨到读状态即 K1 off、K2 off、K3 on、K4 off，按【RESET】按钮对单片机复位，使监控指示灯滚动显示【CLASS SELEct】状态。

(2) 按【实验选择】键，显示【ES--\_ \_】输入 05 或 5，按【确认】键，显示【ES05】。按【确认】键。

(3) 监控显示【CtL1=\_】时，输入 2，按【确认】显示【U\_Addr】，此时输入 6 位二进制微地址，进入读微代码状态。再按【确认】显示【PULSE】，此时按【单步】键，监控显示【U\_Addr】，微地址指示灯显示输入的微地址，微代码显示电路上显示该地址对应的微代码，至此完成一条微指令的读过程。对照表 5-3 表检查微代码是否有错误，如有错误，可按步骤 2 写微代码重新输入这条微代码。

### 4.微代码的运行:

(1) 先将开关 K1K2K3K4 拨到运行状态即 K1 on、K2 off、K3 on、K4 off，按【RESET】按钮对单片机复位，使监控指示灯滚动显示【CLASS SELEct】状态。

(2) 按【实验选择】键，显示【ES--\_ \_】输入 05 或 5，按【确认】键，显示【ES05】。按【确认】键。

(3) 监控指示灯显示【CtL1=\_】，输入 3，显示【CtL1\_3】，表示进入运行微代码状态，拨动 CLR 清零开关（在控制开关电路上，注意对应的 JUI 应短接）对程序计数器清零，清零结果是地址指示灯（A7—A0）和微地址显示灯（uA5—uA0）全灭，清零步骤是使其电平高一低—高即 CLR 指示灯状态为亮—

灭一亮，使程序入口地址为 000000。

#### 1)、单步运行

在监控指示灯显示【CtL1\_3】状态下，确认清零后，按【确认】键，监控指示灯滚动显示【Run CodE】，此时可按【单步】键单步运行微代码，观察微地址显示灯，显示“000001”，再按【单步】，显示为“000010”，连续按【单步】，则可单步运行微代码，注意观察微地址显示灯和微代码的对应关系。

#### 2)、全速运行

在控指示灯滚动显示【Run CodE】状态下，按【全速】键，开始自动运行微代码，微地址显示灯显示从“000000”开始，到“000001”、“000010”、“000011”、“000100”、“001000”、“001001”、“010000”、“010101”、“011000”、“011001”再到“000000”，循环显示。

## 实验六 简单模型机组成原理实验

### 一、实验目的

1. 在掌握各部件功能的基础上，组成一个简单的计算机整机系统—模型机；
2. 了解微程序控制器是如何控制模型机运行的，掌握整机动态工作过程；
3. 定义五条机器指令，编写相应微程序并具体上机调试。

### 二、预习要求

1. 复习计算机组成的基本原理；
2. 预习本实验的相关知识和内容。

### 三、实验设备

EL-JY-II 型计算机组成原理实验系统一套，排线若干。

### 四、模型机结构

模型机结构见图 6-1：

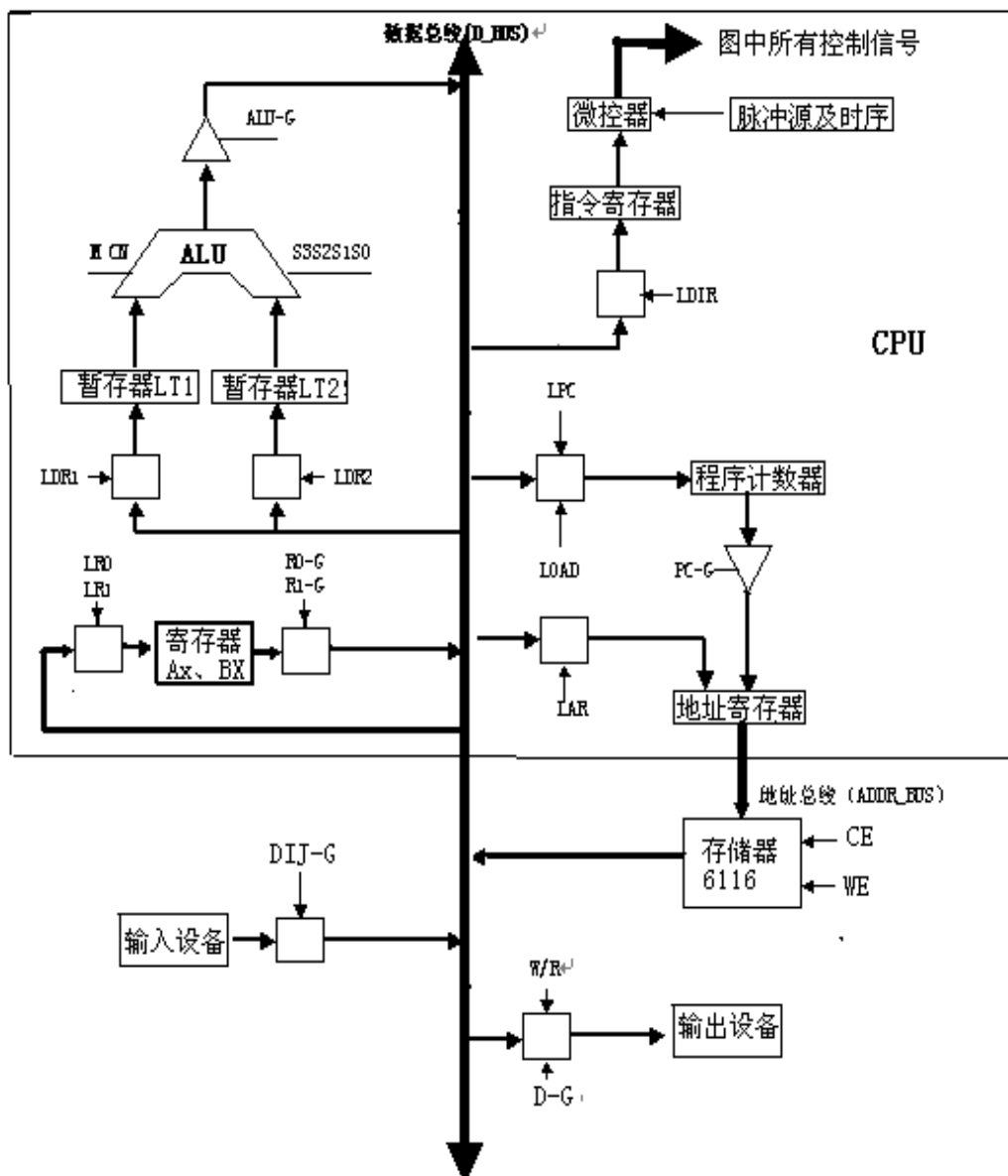




图 6-1 模型机结构框图

图中运算器 ALU 由 U7—U10 四片 74LS181 构成，暂存器 1 由 U3、U4 两片 74LS273 构成，暂存器 2 由 U5、U6 两片 74LS273 构成。微控器部分控存由 U13—U15 三片 2816 构成。除此之外，CPU 的其它部分都由 EP1K10 集成（其原理见系统介绍部分）。

存储器部分由两片 6116 构成 16 位存储器，地址总线只有低 8 位有效，因而其存储空间为 00H—FFH。

输出设备由底板上的四个 LED 数码管及其译码、驱动电路构成，当 D-G 和 W/R 均为低电平时将数据总线的数据送入数码管显示。在开关方式下，输入设备由 16 位电平开关及两个三态缓冲芯片 74LS244 构成，当 DIJ-G 为低电平时将 16 位开关状态送上数据总线。在键盘方式或联机方式下，数据可由键盘或上位机输入，然后由监控程序直接送上数据总线，因而外加的数据输入电路可以不用。

**注：本系统的数据总线为 16 位，指令、地址和程序计数器均为 8 位。当数据总线上的数据打入指令寄存器、地址寄存器和程序计数器时，只有低 8 位有效。**

## 五、工作原理

在本实验中我们将学习读、写机器指令和运行机器指令的完整过程。在机器指令的执行过程中，从 CPU 从内存取出一条机器指令到执行结束为一个指令周期，指令由微指令组成的序列来完成，一条机器指令对应一段微程序。另外，读、写机器指令也分别由相应的微程序段来完成。

为向 RAM 中装入程序和数据，检查写入是否正确，并能启动程序执行，必须设计三个控制操作微程序。

存储器读操作 (MRD)：拨动清零开关 CLR 对地址、指令寄存器清零后，指令译码输入 CA1、CA2 为“00”时，按“单步”键，可对 RAM 连续读操作。

存储器写操作 (MWE)：拨动清零开关 CLR 对地址、指令寄存器清零后，指令译码输入 CA1、CA2 为“10”时，按“单步”键，可对 RAM 连续写操作。

启动程序 (RUN)：拨动开关 CLR 对地址、指令寄存器清零后，指令译码输入 CA1、CA2 为“11”时，按“单步”键，即可转入到第 01 号“取指”微指令，启动程序运行。

**注：CA1、CA2 由控制总线的 E4、E5 给出。键盘操作方式时由监控程序直接对 E4、E5 赋值，无需接线。开关方式时可将 E4、E5 接至控制开关 CA1、CA2，由开关来控制。**

本系统设计的微指令字长 24 位，其控制位顺序如下：

24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
S3	S2	S1	S0	M	Cn	WE	1A	1B	F1			F2		F3		uA5	uA4	uA3	uA2	uA1	uA0		

F 1、F 2、F 3 三个字段的编码方案如表 6-1。

表 6-1 编码表

F1 字段		F2 字段		F3 字段	
15 14 13	选择	12 11 10	选择	9 8 7	选择
0 0 0	LDRi	0 0 0	RAG	0 0 0	P1
0 0 1	LOAD	0 0 1	ALU-G	0 0 1	AR
0 1 0	LDR2	0 1 0	RCG	0 1 0	P3
0 1 1	自定义	0 1 1	自定义	0 1 1	自定义
1 0 0	LDR1	1 0 0	RBG	1 0 0	P2
1 0 1	LAR	1 0 1	PC-G	1 0 1	LPC
1 1 0	LDIR	1 1 0	299-G	1 1 0	P4
1 1 1	无操作	1 1 1	无操作	1 1 1	无操作

系统涉及到的微程序流程图 6-2（图中各方框内为微指令所执行的操作，方框外的标号为该条微指令所处的八进制微地址）。控制操作为 P4 测试，它以 CA1、CA2 作为测试条件，出现了写机器指令、读机器指令和运行机器指令 3 路分支，占用 3 个固定微地址单元。当分支微地址单元固定后，剩下的其它地方就可以一条微指令占用控存一个微地址单元随意填写。

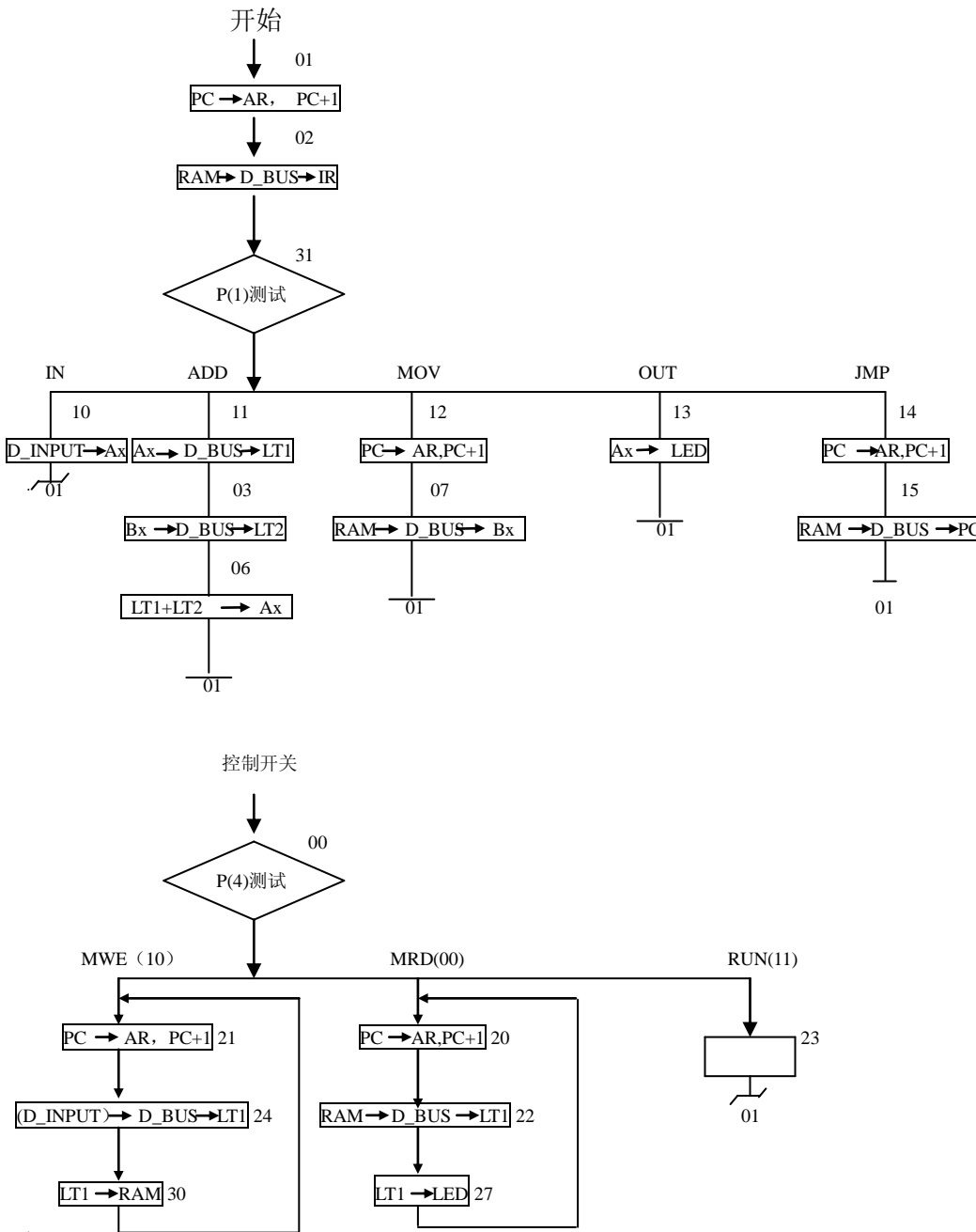


图 6-2 微程序流程图

机器指令的执行过程如下：首先将指令在外存储器的地址送上地址总线，然后将该地址上的指令传送至指令寄存器，这就是“取指”过程。之后必须对操作码进行 P1 测试，根据指令的译码将后续微地址中的某几位强制置位，使下一条微指令指向相应的微程序首地址，这就是“译码”过程。然后才顺序执行该段微程序，这是真正的指令执行过程。

在所有机器指令的执行过程中，“取指”和“译码”是必不可少的，而且微指令执行的操作也是相同

的，这些微指令称为公用微指令，对应于图 4-2 中 01、02、31 地址的微指令。31 地址为“译码”微指令，该微指令的操作为 P(1) 测试，测试结果出现多路分支。本实验用指令寄存器的前 4 位 (I7-I4) 作为测试条件，出现 5 路分支，占用 5 个固定微地址单元。

当全部微程序流程图设计完毕后，应将每条微指令代码化，表 6-2 即为将图 6-2 的微程序流程按微指令格式转化而成的“二进制微代码表”。

表 6-2 二进制编码表

微地址 (二进制)	S3 S2 S1 S0	M CN WE 1A 1B	F1	F2	F3	UA5...UA0
000000	0 0 0 0	0 0 0 0 0	111	111	110	010000
000001	0 0 0 0	0 0 0 0 0	101	101	101	000010
000010	0 0 0 0	0 0 0 1 0	110	111	111	011001
000011	0 0 0 0	0 0 0 0 0	010	100	111	000110
000110	1 0 0 1	0 1 0 0 0	000	001	111	000001
000111	0 0 0 0	0 0 0 1 0	000	111	111	000001
001000	0 0 0 0	0 0 0 1 1	000	111	000	000001
001001	0 0 0 0	0 0 0 0 0	100	000	111	000011
001010	0 0 0 0	0 0 0 0 0	101	101	101	000111
001011	0 0 0 0	0 0 1 0 1	111	000	111	000001
001100	0 0 0 0	0 0 0 0 0	101	101	101	001101
001101	0 0 0 0	0 0 0 1 0	001	111	101	000001
010000	0 0 0 0	0 0 0 0 0	101	101	101	010010
010001	0 0 0 0	0 0 0 0 0	101	101	101	010100
010010	0 0 0 0	0 0 0 1 0	100	111	111	010111
010011	0 0 0 0	0 0 0 0 0	111	111	111	000001
010100	0 0 0 0	0 0 0 1 1	100	111	111	011000
010111	0 0 0 0	0 1 1 0 1	111	001	111	010000
011000	1 1 1 1	1 1 1 1 0	111	001	111	010001
011001	0 0 0 0	0 0 0 1 0	110	111	000	001000

## 六、实验内容及参考代码

本实验采用五条机器指令，根据上面所说的工作原理，设计参考实验程序如下：

地址 (二进制)	机器指令 (二进制)	助记符	说 明
0000 0000	0000 0000	IN AX, KIN	数据输入电路 → AX
0000 0001	0010 0001	MOV Bx, 01H	0001H → Bx
0000 0010	0000 0001		
0000 0011	0001 0000	ADD Ax, Bx	Ax+Bx → Ax
0000 0100	0011 0000	OUT DISP, Ax	Ax → 输出显示电路
0000 0101	0100 0000	JMP 00H	00H → PC
0000 0110	0000 0000		

注：其中 MOV、JMP 为双字长 (32 位)，其余为单字长指令。对于双字长指令，第一字为操作码，第二字为操作数；对于单字长指令只有操作码，没有操作数。上述所有指令的操作码均为低 8 位有效，高

八位默认为 0。而操作数 8 位和 16 位均可。KIN 和 DISP 分别为本系统专用输入、输出设备。

## 七、实验步骤

### 单片机键盘操作方式实验

在进行单片机键盘控制实验时，必须把 K4 开关置于“OFF”状态，否则系统处于自锁状态，无法进行实验。

#### 1、实验连线

实验连线图如图 6-3 所示。

连线时应按如下方法：对于横排座，应使排线插头上的箭头面向自己插在横排座上；对于竖排座，应使排线插头上的箭头面向左边插在竖排座上。

#### 2、写微代码

- (1) 将开关 K1K2K3K4 拨到写状态即 K1 off、K2 on、K3 off、K4 off，其中 K1、K2、K3 在微程序控制电路，K4 在 24 位微代码输入及显示电路上。在监控指示灯滚动显示【CLASS SELEcT】状态下按【实验选择】键，显示【ES--\_ \_】输入 06 或 6，按【确认】键，显示为【ES06】，再按下【确认】键。
- (2) 监控显示为【CtL1=\_】，输入 1 显示【CtL1\_1】，按【确认】。
- (3) 监控显示【U-Addr】，此时输入【000000】6 位二进制数表示的微地址，然后按【确认】键，监控指示灯显示【U\_CodE】，显示这时输入微代码【007F90】，该微代码是用 6 位十六进制数来表示前面的 24 位二进制数，注意输入微代码的顺序，先右后左，按【确认】键则显示【PULSE】，按【单步】完成一条微代码的输入，重新显示【U-Addr】提示输入第二条微代码地址。
- (4) 按照上面的方法输入表 6-3 微代码，观察微代码与微地址显示灯的对应关系（注意输入微代码的顺序是由右至左）。

表 6-3 十六进制微代码表

微地址（八进制）	微地址（二进制）	微代码（十六进制）
00	000000	007F90
01	000001	005B42
02	000010	016FD9
03	000011	0029C6
06	000110	9403C1
07	000111	010FC1
10	001000	018E01
11	001001	0041C3
12	001010	005B47
13	001011	02F1C1
14	001100	005B4D
15	001101	011F41
20	010000	005B52
21	010001	005B54
22	010010	014FD7
23	010011	007FC1
24	010100	01CFD8
25	010101	06F3C1
26	010110	011F41
27	010111	06F3D0

30	011000	FF73D1
31	011001	016E08

### 3、读微代码及校验微代码

(1) 先将开关 K1K2K3K4 拨到读状态即 K1 off、K2 off、K3 on、K4 off，按【RESET】按钮对单片机复位，使监控指示灯滚动显示【CLASS SELEcT】状态。

(2) 按【实验选择】键，显示【ES--\_ \_】输入 06 或 6，按【确认】键，显示【ES06】。按【确认】键。

(3) 监控显示【CtL1=\_】时，输入 2，按【确认】显示【U\_Addr】，此时输入 6 位二进制微地址，进入读代码状态。再按【确认】显示【PULSE】，此时按【单步】键，微地址指示灯显示输入的微地址，同时微代码显示电路上显示该地址对应的微代码，至此完成一条微指令的读过程。

(4) 此时监控显示【U\_Addr】，按上述步骤对照表 6-3 表检查微代码是否有错误，如有错误，可按步骤 2 重新输入微代码。

### 4、写机器指令

(1) 先将 K1K2K3K4 拨到运行状态即 K1 on、K2 off、K3 on、K4 off，按【RESET】按钮对单片机复位，使监控指示灯滚动显示【CLASS SELEcT】状态。

(2) 按【实验选择】键，显示【ES--\_ \_】输入 06 或 6，按【确认】键，显示【ES06】，再按【确认】。

(3) 监控显示【CtL1=\_】，按【取消】键，监控指示灯显示【CtL2=\_】，输入 1 显示【CtL2\_1】表示进入对机器指令操作状态，此时拨动 CLR 清零开关（在控制开关电路上，注意对应的 JUI 应短接）对地址寄存器、指令寄存器清零，清零结果是微地址指示灯（uA5—uA0）和地址指示灯（A7—A0）全灭，清零步骤是使其电平高一低一高即 CLR 指示灯状态为亮—灭—亮。如不清零则会影响机器指令的输入!!! 确定清零后，按【确认】。

(4) 监控显示闪烁的【PULSE】，连续按【单步】键，当微地址显示灯显示“010100”，时按【确认】键，监控指示灯显示【data】，提示输入机器指令“00”或“0000”（两位或四位十六进制数），输入后按【确认】，显示【PULSE】，再按【单步】，微地址显示灯显示“011000”，数据总线显示灯显示“0000000000000000”，即输入的机器指令。

(5) 再连续按【单步】，当微地址显示灯再次显示“010100”时，按【确认】输入第二条机器指令。依此规律逐条输入表 4-4 的机器指令，输完后，可连续按【取消】或【RESET】退出写机器指令状态。**注意：**每当微地址显示灯显示“010100”时，地址指示灯自动加 1 显示。如输入指令为 8 位，则高 8 位自动变为 0。

表 6-4 机器指令表

地址（十六进制）	机器指令（十六进制）
00	0000
01	0021
02	0001
03	0010
04	0030
05	0040
06	0000

### 5、读机器指令

在监控指示灯显示【CtL2=\_】状态下，输入 2，显示【CtL2\_2】，表示进入读机器指令状态，按步骤 4 的方法拨动 CLR 开关对地址寄存器和指令寄存器进行清零，然后按【确认】键，显示【PULSE】，连续按【单步】键，微地址显示灯从“000000”开始，然后按“010000”、“010010”、“010111”方式循环显示。当微地址灯再次显示为“010000”时，输出显示数码管上显示写入的机器指令。读的过程注意微地址显示灯，地址显示灯和数据总线指示灯的对应关系。如果发现机器指令有误，则需重新输入机器指令。

注意：机器指令存放在 RAM 里，掉电丢失，故断电后需重新输入。

## 6、运行程序

在监控指示灯显示【CtL2=\_】状态下，输入 3，显示【CtL2\_3】，表示进入运行机器指令状态，按步骤 4 的方法拨动 CLR 开关对地址寄存器和指令寄存器进行清零，使程序入口地址为 00H，可以【单步】运行程序也可以【全速】运行，运行过程中提示输入相应的量，运行结束后从输出显示电路上观察结果。

## 7、实验结果说明

### 1) 单步运行结果

在监控指示灯显示【run CodE】状态下，连续按【单步】键，可以单步运行程序。当微地址显示灯显示“001000”时，监控显示【dAtA】，提示输入数据，即被加数，输入 1234，按【确认】，再连续按【单步】，在微地址灯显示“010101”时，按【单步】，此时可由输出显示电路的数码管观察结果为 1235，即  $1234 + 0001 = 1235$ ，同时数据显示灯显示“0001001000110101”。表示结果正确。

### 2) 全速运行结果

在监控指示灯显示【run CodE】状态下，按【全速】键，则开始自动运行程序，在监控指示灯显示【dAtA】时输入数据，按【确定】键，程序继续运行，此时可由输出显示电路的数码管显示加 1 运算结果。

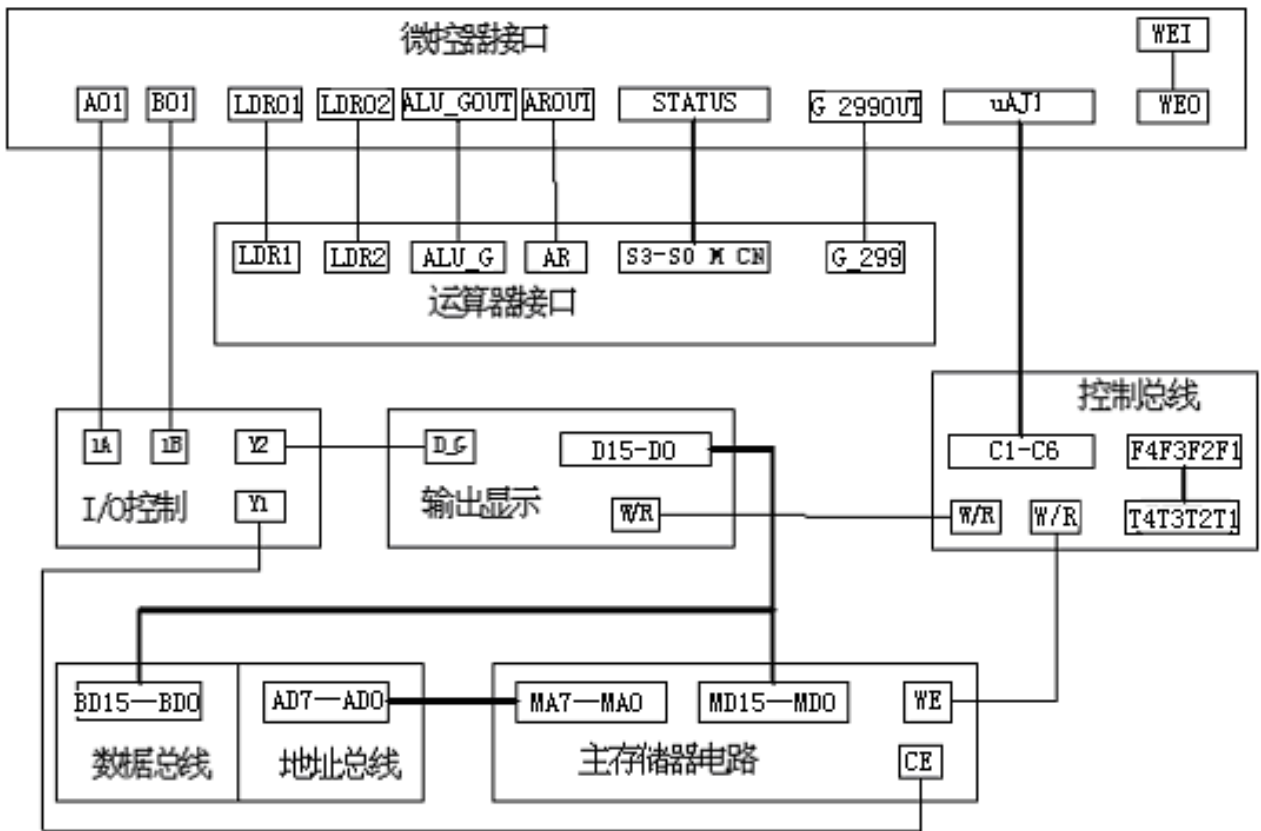


图 6-3 键盘方式连线图 (删掉 G-299 与 G-299OUT 的连接)

## 实验七 带移位运算的模型机组成原理实验

### 一、实验目的

在实验六的基础上进一步构造一台带移位功能的简单模型机。

### 二、预习要求

1 认真预习本实验的相关知识和内容。

### 三、实验设备

EL-JY-II 型计算机组成原理实验系统一套，排线若干。

### 四、模型机结构

模型机结构见图 7-1：

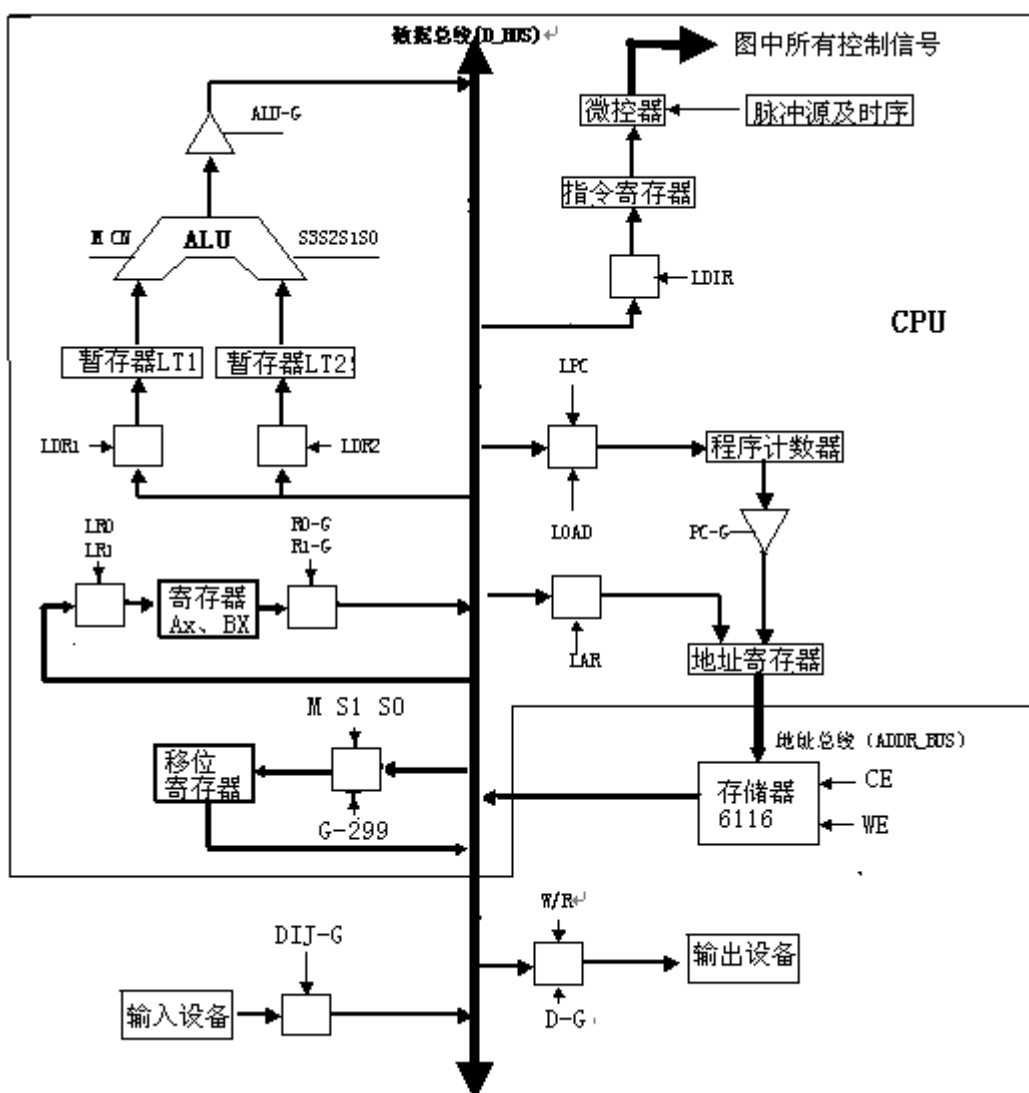


图 7-1 模型机结构框图

图中运算器 ALU 由 U7—U10 四片 74LS181 构成，暂存器 1 由 U3、U4 两片 74LS273 构成，暂存器 2 由 U5、U6 两片 74LS273 构成。微控器部分控存由 U13—U15 三片 2816 构成。除此之外，CPU 的其它部分都由 EP1K10 集成（其原理见系统介绍部分）。

存储器部分由两片 6116 构成 16 位存储器，地址总线只有低八位有效，因而其存储空间为 00H—FFH。



输出设备由底板上的四个 LED 数码管及其译码、驱动电路构成，当 D-G 和 W/R 均为低电平时将数据总线的的数据送入数码管显示。在开关方式下，输入设备由 16 位电平开关及两个三态缓冲芯片 74LS244 构成，当 DIJ-G 为低电平时将 16 位开关状态送上数据总线。在键盘方式或联机方式下，数据可由键盘或上位机输入，然后由监控程序直接送上数据总线，因而外加的数据输入电路可以不用。

**注：本系统的数据总线为 16 位，指令、地址和程序计数器均为 8 位。当数据总线上的数据打入指令寄存器、地址寄存器和程序计数器时，只有低 8 位有效。**

## 五、工作原理

设计三个控制操作微程序：

存储器读操作 (MRD)：拨动清零开关 CLR 对地址、指令寄存器清零后，指令译码输入 CA1、CA2 为“00”时，按“单步”键，可对 RAM 连续读操作。

存储器写操作 (MWE)：拨动清零开关 CLR 对地址、指令寄存器清零后，指令译码输入 CA1、CA2 为“10”时，按“单步”键，可对 RAM 连续写操作。

启动程序 (RUN)：拨动开关 CLR 对地址、指令寄存器清零后，指令译码输入 CA1、CA2 为“11”时，按“单步”键，即可转入到第 01 号“取指”微指令，启动程序运行。

**注：CA1、CA2 由控制总线的 E4、E5 给出。键盘操作方式时由监控程序直接对 E4、E5 赋值，无需接线。开关方式时可将 E4、E5 接至控制开关 CA1、CA2，由开关来控制。**

本系统设计的微指令字长共 24 位，其控制位顺序如下：

24	23	22	21	20	19	18	17	16	15 14 13	12 11 10	9 8 7	6	5	4	3	2	1
S3	S2	S1	S0	M	Cn	WE	1A	1B	F1	F2	F3	uA5	uA4	uA3	uA2	uA1	uA0

F 1、F 2、F 3 三个字段的编码方案如表 7-1：

**表 7-1 编码表**

F1 字段		F2 字段		F3 字段	
15 14 13	选择	12 11 10	选择	9 8 7	选择
0 0 0	LDRi	0 0 0	RAG	0 0 0	P1
0 0 1	LOAD	0 0 1	ALU-G	0 0 1	AR
0 1 0	LDR2	0 1 0	RCG	0 1 0	P3
0 1 1	自定义	0 1 1	自定义	0 1 1	自定义
1 0 0	LDR1	1 0 0	RBG	1 0 0	P2
1 0 1	LAR	1 0 1	PC-G	1 0 1	LPC
1 1 0	LDIR	1 1 0	299-G	1 1 0	P4
1 1 1	无操作	1 1 1	无操作	1 1 1	无操作

系统涉及到的微程序流程见图 7-2（图中各方框内为微指令所执行的操作，方框外的标号为该条微指令所处的八进制微地址）。控制操作为 P4 测试，它以 CA1、CA2 作为测试条件，出现了写机器指令、读机器指令和运行机器指令 3 路分支，占用 3 个固定微地址单元。当分支微地址单元固定后，剩下的其它地方就可以一条微指令占用控存一个微地址单元随意填写。

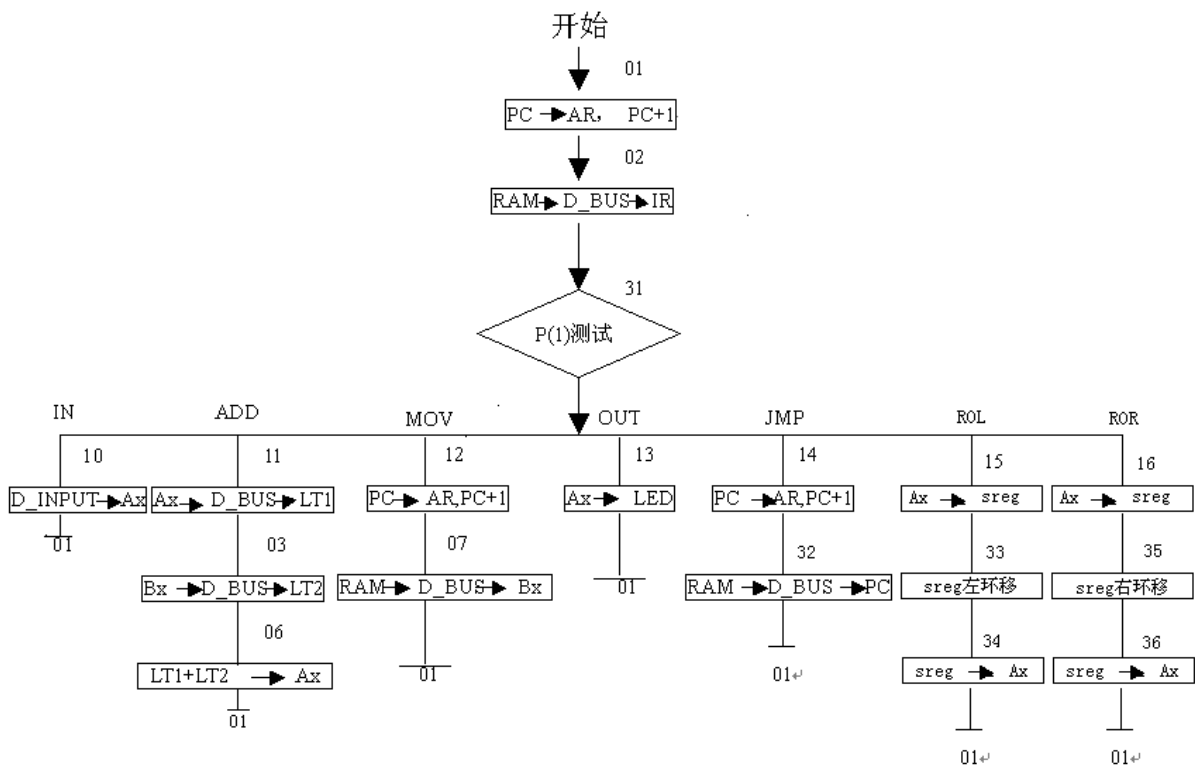


图 7-2 (a) sreg 表示移位寄存器

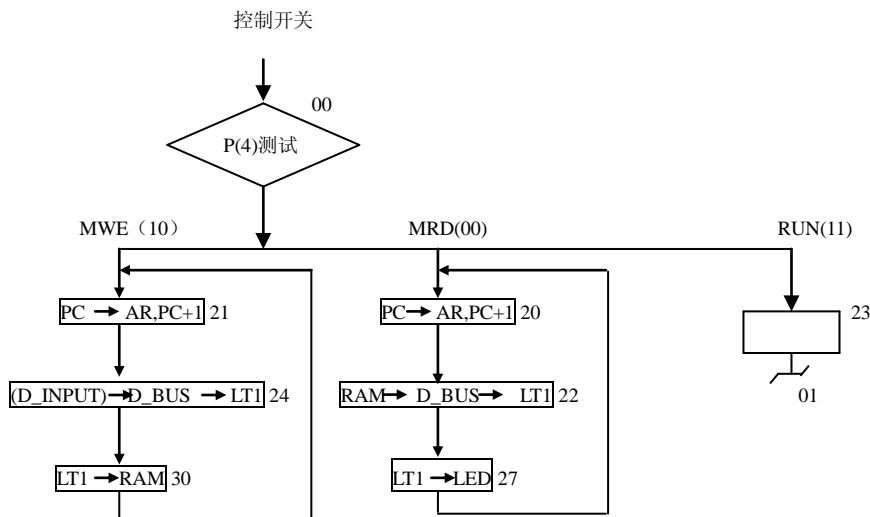


图 7-2 (b) 微指令流程图

机器指令的执行过程如下：首先将指令在外存储器的地址送上地址总线，然后将该地址上的指令传送到指令寄存器，这就是“取指”过程。之后必须对操作码进行 P1 测试，根据指令的译码将后续微地址中的某几位强制置位，使下一条微指令指向相应的微程序首地址，这就是“译码”过程。然后才顺序执行该段微程序，这是真正的指令执行过程。

在所有机器指令的执行过程中，“取指”和“译码”是必不可少的，而且微指令执行的操作也是相同的，这些微指令称为公用微指令，对应于图 7-2 中 01、02、31 地址的微指令。31 地址为“译码”微指令，该微指令的操作为 P（1）测试，测试结果出现多路分支。本实验用指令寄存器的前 4 位（I7-I4）作为测试条件，出现 7 路分支，占用 7 个固定微地址单元。

当全部微程序流程图设计完毕后，应将每条微指令代码化，表 7-2 即为将图 7-2 的微程序流程按微指

令格式转化而成的“二进制微代码表”。

表 7-2 二进制微代码表

微地址	S3	S2	S1	S0	M	CN	WE	1A	1B	F1	F2	F3	UA5...UA0
000000		0	0	0	0	0	0	0	0	111	111	110	010000
000001		0	0	0	0	0	0	0	0	101	101	101	000010
000010		0	0	0	0	0	0	1	0	110	111	111	011001
000011		0	0	0	0	0	0	0	0	010	100	111	000110
000110		1	0	0	1	0	1	0	0	000	001	111	000001
000111		0	0	0	0	0	0	1	0	000	111	111	000001
001000		0	0	0	0	0	0	1	1	000	111	000	000001
001001		0	0	0	0	0	0	0	0	100	000	111	000011
001010		0	0	0	0	0	0	0	0	101	101	101	000111
001011		0	0	0	0	0	0	1	0	111	000	111	000001
001100		0	0	0	0	0	0	0	0	101	101	101	011010
001101		0	0	1	1	0	0	0	0	000	000	111	011011
001110		0	0	1	1	0	0	0	0	000	000	111	011101
010000		0	0	0	0	0	0	0	0	101	101	101	010010
010001		0	0	0	0	0	0	0	0	101	101	101	010100
010010		0	0	0	0	0	0	1	0	100	111	111	010111
010011		0	0	0	0	0	0	0	0	111	111	111	000001
010100		0	0	0	0	0	0	1	1	100	111	111	011000
010111		0	0	0	0	0	1	1	0	111	001	111	010000
011000		1	1	1	1	1	1	1	0	111	001	111	010001
011001		0	0	0	0	0	0	1	0	110	111	000	001000
011010		0	0	0	0	0	0	1	0	001	111	101	000001
011011		0	0	0	1	0	0	0	0	111	110	111	011100
011100		0	0	0	0	0	0	0	0	000	110	111	000001
011101		0	0	1	0	0	0	0	0	111	110	111	011110
011110		0	0	0	0	0	0	0	0	000	110	111	000001

## 六、实验参考代码

本实验采用条机器指令，根据上面所说的工作原理，设计参考实验程序如下：

地址（二进制）	机器指令码	助记符	说 明
0000 0000	0000 0000	IN AX, KIN	数据输入电路 → AX
0000 0001	0010 0001	MOV Bx, 01H	0001H → Bx
0000 0010	0000 0001		
0000 0011	0101 0000	ROL Ax	Ax 循环左移一位
0000 0100	0001 0000	ADD Ax, Bx	Ax+Bx → Ax
0000 0101	0110 0000	ROR Ax	Ax 循环右移一位
0000 0110	0011 0000	OUT DISP, Ax	Ax → 输出显示电路
0000 0111	0100 0000	JMP 00H	00H → PC
0000 0110	0000 0000		

注：其中 MOV、JMP 为双字长（32 位），其余为单字长指令。对于双字长指令，第一字为操作码，第二字为操作数；对于单字长指令只有操作码，没有操作数。上述所有指令的操作码均为低 8 位有效，高

八位默认为 0。而操作数 8 位和 16 位均可。KIN 和 DISP 分别为本系统专用输入、输出设备。

## 七、实验步骤

### 单片机键盘操作方式实验

在进行单片机键盘操作实验时，必须把 K4 开关置于“OFF”状态，否则系统处于自锁状态，无法进行实验。

#### 1. 实验连线

实验连线图如图 7-3 所示。

连线时应按如下方法：对于横排座，应使排线插头上的箭头面向自己插在横排座上；对于竖排座，应使排线插头上的箭头面向左边插在竖排座上。

#### 2. 写微代码

(1) 将开关 K1K2K3K4 拨到写状态即 K1 off、K2 on、K3 off、K4 off，其中 K1、K2、K3 在微程序控制电路，K4 在 24 位微代码输入及显示电路上。在监控指示灯滚动显示【CLASS SELEct】状态下按【实验选择】键，显示【ES--\_ \_】输入 07 或 7，按【确认】键，显示为【ES07】，再按下【确认】键。

(2) 监控显示为【CtL1=\_】，表示对微代码进行操作。输入 1 显示【CtL1\_1】，表示写微代码，按【确认】。

(3) 监控显示【U-Addr】，此时输入【000000】6 位二进制数表示的微地址，然后按【确认】键，监控指示灯显示【U\_CodE】，此时输入微代码【007F90】，该微代码是用 6 位十六进制数来表示前面的 24 位二进制数，注意输入微代码的顺序，先右后左，按【确认】键则显示【PULSE】，按【单步】完成一条微代码的输入，重新显示【U-Addr】提示输入表 7-3 第二条微代码地址。

(4) 按照上面的方法输入表 7-3 微代码,观察微代码与微地址显示灯的对应关系（注意输入微代码的顺序是由右至左）。

表 7-3 十六进制微代码表

微地址（八进制）	微地址（二进制）	微代码（十六进制）
00	000000	007F90
01	000001	005B42
02	000010	016FD9
03	000011	0029C6
06	000110	9403C1
07	000111	010FC1
10	001000	018E01
11	001001	0041C3
12	001010	005B47
13	001011	02F1C1
14	001100	005B5A
15	001101	3001DB
16	001110	3001DD
20	010000	005B52
21	010001	005B54
22	010010	014FD7
23	010011	007FC1
24	010100	01CFD8

25	010101	06F3C1
26	010110	011F41
27	010111	06F3D0
30	011000	FF73D1
31	011001	016E08
32	011010	011F41
33	011011	107DDC
34	011100	000DC1
35	011101	207DDE
36	011110	000DC1

### 3. 读微代码及校验微代码

(1) 将开关 K1K2K3K4 拨到读状态即 K1 off、K2 off、K3 on、K4 off，按【RESET】按钮对单片机复位，使监控指示灯滚动显示【CLASS SELEcT】状态。

(2) 按【实验选择】键，显示【ES--\_ \_】输入 07 或 7，按【确认】键，显示【ES07】。按【确认】键，显示【CtL1=\_】时，输入 2，按【确认】进入读代码状态。

(3) 监控显示【U\_Addr】，此时输入 6 位二进制微地址，再按【确认】显示【PULSE】，此时按【单步】键，微地址指示灯显示输入的微地址，微代码显示电路上显示该地址对应的微代码，至此完成一条微指令的读过程。

(4) 此时监控显示【U\_Addr】，按照上述步骤继续输入微地址，对照表 7-2 表检查微代码是否有错误，如有错误，可按步骤 2 重新输入微代码。

### 4. 写机器指令

(1) 将 K1K2K3K4 拨到运行状态即 K1 on、K2 off、K3 on、K4 off，按【RESET】按钮对单片机复位，使监控指示灯滚动显示【CLASS SELEcT】状态。

(2) 按【实验选择】键，显示【ES--\_ \_】输入 07 或 7，按【确认】键，显示【ES07】，再按【确认】。

(3) 监控显示【CtL1=\_】，按【取消】键，监控指示灯显示【CtL2=\_】，输入 1 显示【CtL2\_1】表示进入对机器指令写操作状态，此时拨动 CLR 清零开关对地址寄存器、指令寄存器清零。

(4) 确定清零后，按【确认】显示闪烁的【PULSE】，连续按【单步】键，当微地址显示灯显示“010100”时，按【确认】键，监控指示灯显示【data】，提示输入机器指令“00”或“0000”（两位或四位十六进制数），输入后按【确认】，显示【PULSE】，再按【单步】，微地址显示灯显示“011000”，数据总线显示灯显示“0000000000000000”，即输入的机器指令。此时第一条机器指令完成。

(5) 连续按【单步】，当微地址显示灯再次显示“010100”时，按【确认】输入第二条机器指令。依此规律逐条输入表 7-4 的机器指令，输完后，可连续按【取消】或【RESET】退出写机器指令状态。**注意，每当微地址显示灯显示“010100”时，地址指示灯自动加 1 显示。如输入指令为 8 位，则高 8 位自动变为 0。**

表 7-4 机器指令表

地址（十六进制）	机器指令（十六进制）
00	0000
01	0021
02	0001
03	0050
04	0010
05	0060

06	0030
07	0040
08	0000

### 5. 读机器指令及校验机器指令

在监控指示灯显示【Ctl2=\_】状态下，输入 2，显示【Ctl2\_2】，表示进入读机器指令状态，按步骤 4 的方法拨动 CLR 开关对地址寄存器和指令寄存器进行清零，然后按【确认】键，显示【PULSE】，连续按【单步】键，微地址显示灯显示从“000000”开始，然后按“010000”、“010010”、“010111”方式循环显示。当微地址灯再次显示为“010000”时，输出显示数码管上显示写入的机器指令。读的过程注意微地址显示灯，地址显示灯和数据总线指示灯的对应关系。如果发现机器指令有误，则需重新输入机器指令。

注意：机器指令存放在 RAM 里，掉电丢失，故断电后需重新输入。

### 6. 运行程序

在监控指示灯显示【Ctl2=\_】状态下，输入 3，显示【Ctl2\_3】，表示进入运行机器指令状态，按步骤 4 的方法拨动 CLR 开关对地址寄存器和指令寄存器进行清零，使程序入口地址为 00H，可以【单步】运行程序也可以【全速】运行，运行过程中提示输入相应的量，运行结束后从输出显示电路上观察结果。

### 7. 实验结果说明

注意，进位指示灯 Z 在运算器电路上，亮表示为“1”，灭表示为“0”。

本实验结果为：输入一数据，循环左移一位，然后执行加“1”运算，接着对结果循环右移一位，可从数据总线观察结果，也可从输出显示电路数码管观察执行结果。

#### 1) 单步运行结果

在监控指示灯显示【run CodE】状态下，连续按【单步】键，可以单步运行程序。当微地址显示灯显示“001 000”时，按【单步】，监控显示灯显示【dAtA】，提示输入数据，输入 8000，按【确认】。再连续按【单步】，在微地址灯显示“000 111”时，按【单步】，数据总线显示灯显示“0000000000000001”，即立即数 0001H；再连续按【单步】，在微地址灯显示“011 100”时，按【单步】，数据总线显示灯显示“0000000000000001”，即 8000H 循环左移一位；再连续按【单步】，在微地址灯显示“000 110”时，按【单步】，数据总线显示灯显示“0000000000000010”，即 8000H 循环左移后加 1；再连续按【单步】，在微地址灯显示“011 110”时，按【单步】，数据总线显示灯显示“0000000000000001”，即循环右移后的最终结果。

#### 2) 全速运行结果

在监控指示灯显示【run CodE】状态下，按【全速】键，则开始自动运行程序，在监控指示灯显示【dAtA】时输入数据“8000”，按【确定】键，程序继续运行，此时可由输出显示电路的数码管显示最终运算结果“0001”。

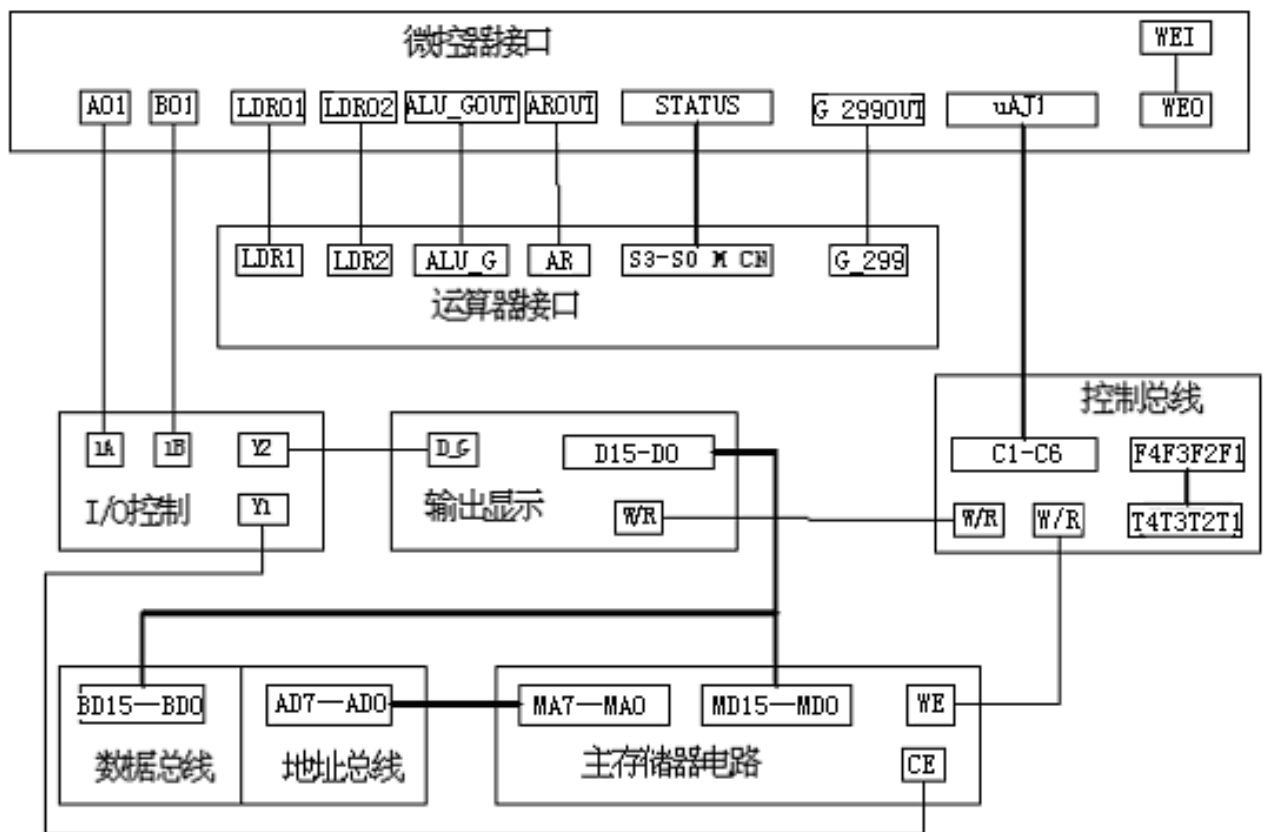


图 7-3 键盘连线图

## 实验八 复杂模型机组成原理实验

### 一、实验目的

在实验七的基础上，构造一个指令系统，实现比较完整的模型机功能。

### 二、预习要求

认真预习本实验的相关知识和内容。

### 三、实验设备

EL-JY-II 型计算机组成原理实验系统一套，排线若干。

### 四、模型机结构

模型机结构见图 8-1：

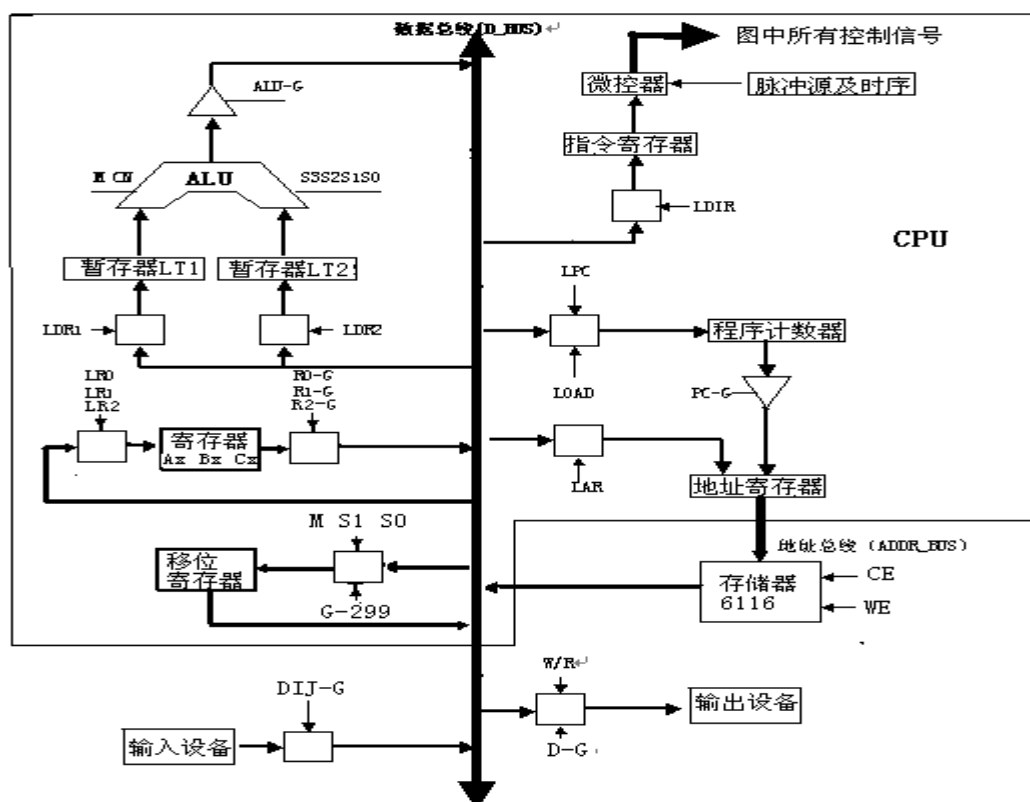


图 8-1 模型机结构框图

图中运算器 ALU 由 U7—U10 四片 74LS181 构成，暂存器 1 由 U3、U4 两片 74LS273 构成，暂存器 2 由 U5、U6 两片 74LS273 构成。微控器部分控存由 U13—U15 三片 2816 构成。除此之外，CPU 的其它部分都由 EP1K10 集成（其原理见系统介绍部分）。

存储器部分由两片 6116 构成 16 位存储器，地址总线只有低八位有效，因而其存储空间为 00H—FFH。

输出设备由底板上的四个 LED 数码管及其译码、驱动电路构成，当 D-G 和 W/R 均为低电平时将数据总线的数据送入数码管显示。在开关方式下，输入设备由 16 位电平开关及两个三态缓冲芯片 74LS244 构成，当 DIJ-G 为低电平时将 16 位开关状态送上数据总线。在键盘方式或联机方式下，数据可由键盘或上位机输入，然后由监控程序直接送上数据总线，因而外加的数据输入电路可以不用。

注：本系统的数据总线为 16 位，指令、地址和程序计数器均为 8 位。当数据总线上的数据打入指令



寄存器、地址寄存器和程序计数器时，只有低 8 位有效。

## 五、工作原理

### 1、数据格式：

本实验计算机采用定点补码表示法表示数据，字长为 16 位，其格式如下：

15	14	13.....	0
符号	尾	数	

其中第 16 位为符号位，数值表示范围是： $-32768 \leq X < 32767$ 。

### 2、指令格式：

#### 1) 算术逻辑指令

设计 9 条单字长算术逻辑指令，寻址方式采用寄存器直接寻址。其格式如下：

7	6	5	4	3	2	1	0
OP-CODE				rs		rd	

其中 OP-CODE 为操作码，rs 为源寄存器，rd 为目的寄存器，并规定：

OP-CODE	0111	1000	1001	1010	1011	1100	1101	1110	1111
指令	CLR	MOV	ADD	SUB	INC	AND	NOT	ROR	ROL

Rs 或 rd	选定寄存器
00	Ax
01	Bx
10	Cx

9 条算术逻辑指令的名称、功能和具体格式见表 6-2。

#### 2) 存储器访问及转移指令

存储器的访问有两种，即存数和取数。它们都使用助记符 MOV，但其操作码不同。转移指令只有一种，即无条件转移 (JMP)。指令格式如下：

7	6	5	4	3	2	1	0
00		M	OP-CODE		rd		
D							

其中 OP-CODE 为操作码，rd 为寄存器。M 为寻址模式，D 随 M 的不同其定义也不相同，如下表所示：

OP-CODE	00	01	10
指令说明	写存储器	读存储器	转移指令

寻址模式 M	有效地址 E	D 定义	说 明
00	$E = (PC) + 1$	立即数	立即寻址
10	$E = D$	直接地址	直接寻址

11	E=100H +D	直接地址	扩展直接寻址
----	-----------	------	--------

注：扩展直接寻址用于面包板上扩展的存储器的寻址。

### 3) I/O 指令

输入 (IN) 和输出 (OUT) 指令采用单字节指令，其格式如下：

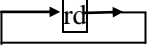
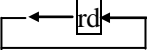
7 6 5 4	3 2	1 0
OP-CODE	addr	rd

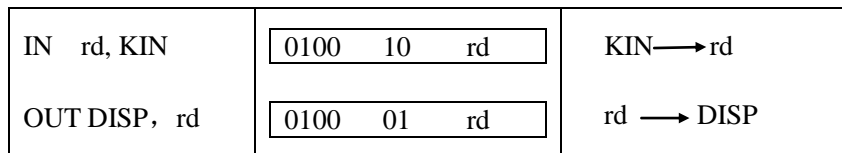
其中，当 OP-CODE=0100 且 addr=10 时，从“数据输入电路”中的开关组输入数据；当 OP-CODE=0100 且 addr=01 时，将数据送到“输出显示电路”中的数码管显示。

### 3、指令系统

本实验共有 14 条基本指令，其中算术逻辑指令 8 条，访问内存指令和程序控制指令 4 条。输入输出指令 2 条。表 8-1 列出了各条指令的格式、汇编符号和指令功能。

表 8-1 指令格式

汇编符号	指令的格式	功 能								
MOV rd, rs	<table border="1"><tr><td>1000</td><td>rs</td><td>rd</td></tr></table>	1000	rs	rd	rs → rd					
1000	rs	rd								
ADD rd, rs	<table border="1"><tr><td>1001</td><td>rs</td><td>rd</td></tr></table>	1001	rs	rd	rs+rd → rd					
1001	rs	rd								
SUB rd, rs	<table border="1"><tr><td>1010</td><td>rs</td><td>rd</td></tr></table>	1010	rs	rd	rd-rs → rd					
1010	rs	rd								
INC rd	<table border="1"><tr><td>1011</td><td>rd</td><td>rd</td></tr></table>	1011	rd	rd	rd+1 → rd					
1011	rd	rd								
AND rd, rs	<table border="1"><tr><td>1100</td><td>rs</td><td>rd</td></tr></table>	1100	rs	rd	rs ∧ rd → rd					
1100	rs	rd								
NOT rd	<table border="1"><tr><td>1101</td><td>rd</td><td>rd</td></tr></table>	1101	rd	rd	$\overline{rd} \rightarrow rd$					
1101	rd	rd								
ROR rd	<table border="1"><tr><td>1110</td><td>rd</td><td>rd</td></tr></table>	1110	rd	rd						
1110	rd	rd								
ROL rd	<table border="1"><tr><td>1111</td><td>rd</td><td>rd</td></tr></table>	1111	rd	rd						
1111	rd	rd								
MOV [D], rd	<table border="1"><tr><td>00</td><td>10</td><td>00</td><td>rd</td></tr><tr><td colspan="4">D</td></tr></table>	00	10	00	rd	D				rd → [D]
00	10	00	rd							
D										
MOV rd, [D]	<table border="1"><tr><td>00</td><td>10</td><td>01</td><td>rd</td></tr><tr><td colspan="4">D</td></tr></table>	00	10	01	rd	D				[D] → rd
00	10	01	rd							
D										
MOV rd, D	<table border="1"><tr><td>00</td><td>00</td><td>01</td><td>rd</td></tr><tr><td colspan="4">D</td></tr></table>	00	00	01	rd	D				D → rd
00	00	01	rd							
D										
JMP D	<table border="1"><tr><td>00</td><td>00</td><td>10</td><td>00</td></tr><tr><td colspan="4">D</td></tr></table>	00	00	10	00	D				D → PC
00	00	10	00							
D										



#### 4、设计微代码

设计三个控制操作微程序：

存储器读操作 (MRD)：拨动清零开关 CLR 对地址、指令寄存器清零后，指令译码输入 CA1、CA2 为“00”时，按“单步”键，可对 RAM 连续读操作。

存储器写操作 (MWE)：拨动清零开关 CLR 对地址、指令寄存器清零后，指令译码输入 CA1、CA2 为“10”时，按“单步”键，可对 RAM 连续写操作。

启动程序 (RUN)：拨动开关 CLR 对地址、指令寄存器清零后，指令译码输入 CA1、CA2 为“11”时，按“单步”键，即可转入到第 01 号“取指”微指令，启动程序运行。

**注：CA1、CA2 由控制总线的 E4、E5 给出。键盘操作方式时由监控程序直接对 E4、E5 赋值，无需接线。**

开关方式时可将 E4、E5 接至控制开关 CA1、CA2，由开关来控制。

本系统设计的微程序字长共 24 位，其控制位顺序如下：

24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
S3	S2	S1	S0	M	Cn	WE	1A	1B	F1	F2	F3	uA5	uA4	uA3	uA2	uA1	uA0						

F 1、F 2、F 3 三个字段的编码方案如表 8-2：

**表 8-2 编码表**

F1 字段		F2 字段		F3 字段	
15 14 13	选择	12 11 10	选择	9 8 7	选择
0 0 0	LDRi	0 0 0	RAG	0 0 0	P1
0 0 1	LOAD	0 0 1	ALU-G	0 0 1	AR
0 1 0	LDR2	0 1 0	RCG	0 1 0	P3
0 1 1	自定义	0 1 1	自定义	0 1 1	自定义
1 0 0	LDR1	1 0 0	RBG	1 0 0	P2
1 0 1	LAR	1 0 1	PC-G	1 0 1	LPC
1 1 0	LDIR	1 1 0	299-G	1 1 0	P4
1 1 1	无操作	1 1 1	无操作	1 1 1	无操作

系统涉及到的微程序流程见图 8-2（图中各方框内为微指令所执行的操作，方框外的标号为该条微指令所处的八进制微地址）。控制操作为 P4 测试，它以 CA1、CA2 作为测试条件，出现了写机器指令、读机器指令和运行机器指令 3 路分支，占用 3 个固定微地址单元。当分支微地址单元固定后，剩下的其它地方就可以一条微指令占用控存一个微地址单元随意填写。

机器指令的执行过程如下：首先将指令在外存储器的地址送上地址总线，然后将该地址上的指令传送至指令寄存器，这就是“取指”过程。之后必须对操作码进行 P1 测试，根据指令的译码将后续微地址中的某几位强制置位，使下一条微指令指向相应的微程序首地址，这就是“译码”过程。然后才顺序执行该段微程序，这是真正的指令执行过程。

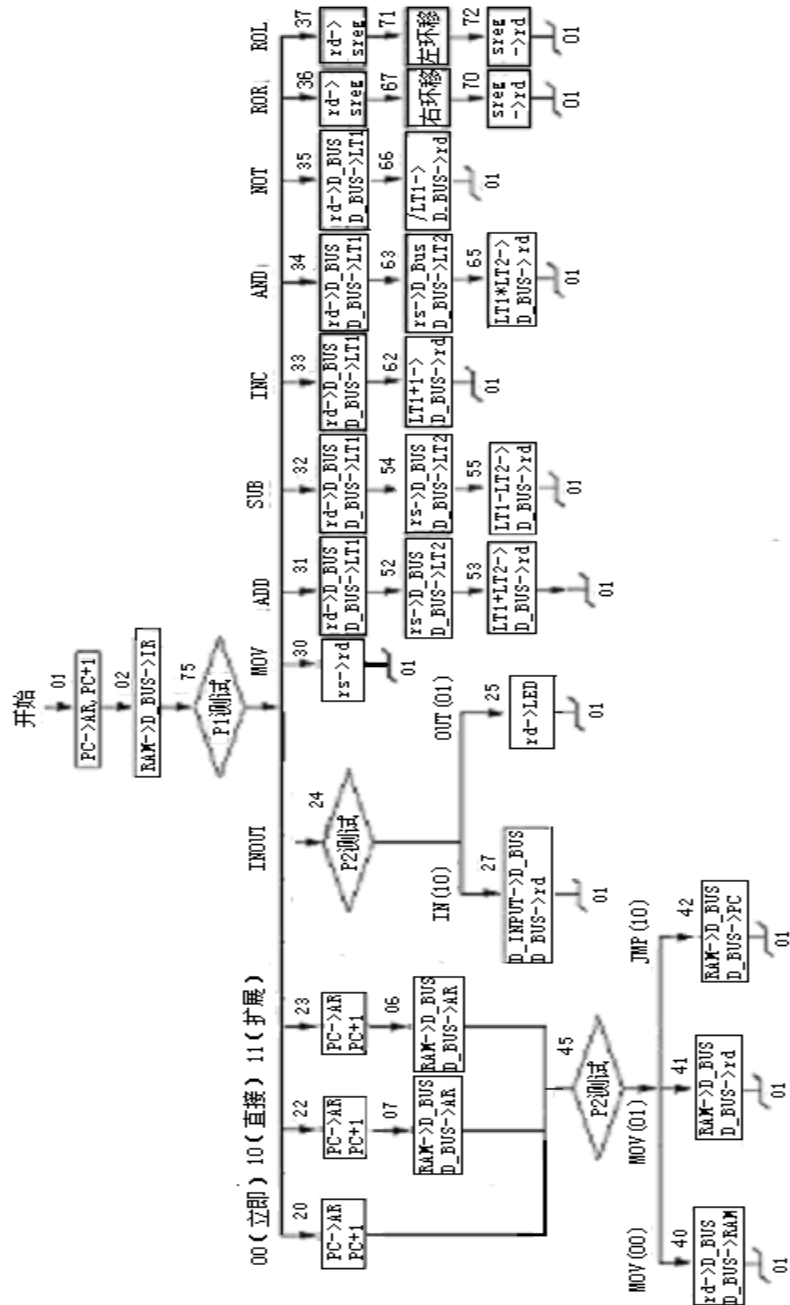
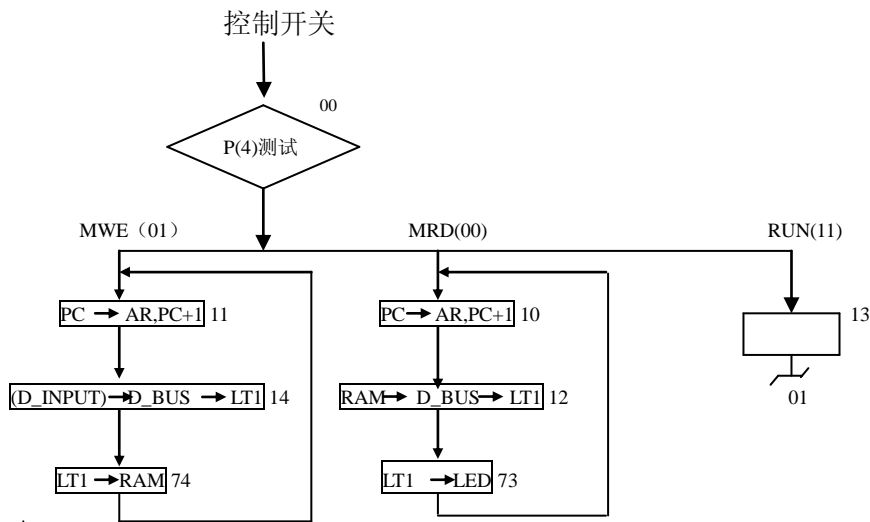


图 8-2 微程序流程图

在所有机器指令的执行过程中，“取指”和“译码”是必不可少的，而且微指令执行的操作也是相同的，这些微指令称为公用微指令，对应于图 8-2 中 01、02、75 地址的微指令。75 地址为“译码”微指令，该微指令的操作为 P（1）测试，测试结果出现多路分支。本实验用指令寄存器的前 4 位（I7-I4）作为测试条件，出现 12 路分支，占用 12 个固定微地址单元。如 I7—I4 相同，则还需进行 P2 测试，以指令寄存器的 I3、I2 位作为测试条件，以区分不同的指令，如 MOV 指令和 IN、OUT 指令。

表 8-3 即为将图 8-2 的微程序流程按微程序格式转化而成的二进制微代码表。

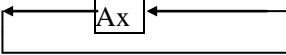
表 8-3 二进制微代码表

微地址	S3S2S1 S0 M CN WE 1A 1B	F1	F2	F3	UA5--UA0
000000	000000000	111	111	110	001000
000001	000000000	101	101	101	000010
000010	000000010	110	111	111	111101
000110	000000010	101	111	111	100101
000111	000000010	101	111	111	100101
001000	000000000	101	101	101	001010
001001	000000000	101	101	101	001100
001010	000000010	100	111	111	111011
001011	000000000	111	111	111	000001
001100	000000011	100	111	111	111100
010000	000000000	101	101	101	100101
010010	000000000	101	101	101	000111
010011	000000000	101	101	101	000110
010100	000000000	111	111	100	010101
010101	000000101	111	010	111	000001
010111	000000011	000	111	111	000001
011000	000000000	000	000	111	000001
011001	000000000	100	000	111	101010
011010	000000000	100	000	111	101100
011011	000000000	100	000	111	110010
011100	000000000	100	000	111	110011
011101	000000000	100	000	111	110110
011110	001100000	111	000	111	110111
011111	001100000	000	000	111	111001
100000	000000110	111	100	111	000001
100001	000000010	000	111	111	000001
100010	000000010	001	111	101	000001
100101	000000000	111	111	100	100000
101010	000000000	010	010	111	101011
101011	100101000	000	001	111	000001
101100	000000000	010	100	111	101101
101101	011000000	000	001	111	000001

110010	00000000	000	001	111	000001
110011	00000000	010	010	111	110101
110101	10111000	000	001	111	000001
110110	00001100	000	001	111	000001
110111	00100000	111	110	111	111000
111000	00000000	000	110	111	000001
111001	00010000	111	110	111	111010
111010	00000000	000	110	111	000001
111011	000001101	111	001	111	001000
111100	111111110	111	001	111	001001
111101	000000010	110	111	000	010000

## 六、实验参考代码

我们按程序流程图译出适合本实验系统所有机器指令的微代码，供学生自己编程实验，加深对较完整的模型机的认识。这里只提供以下简单的实验程序：

地址（二进制）	指令（二进制）	助记符	说明
0000 0000	0100 1000	IN Ax, KIN	“开关输入” → Ax
0000 0001	0000 0101	MOV Bx, 01H	01H → Bx
0000 0010	0000 0001		
0000 0011	1001 0100	ADD Ax, Bx	Ax+Bx → Ax
0000 0100	1111 0000	ROL Ax	
0000 0101	1000 0010	MOV Cx, Ax	Ax → Cx
0000 0110	0100 0110	OUT DISP, Cx	Cx → LED
0000 0110	0000 1000	JMP 00H	00H → PC
0000 0111	0000 0000		

注：其中 MOV、JMP 为双字长（32 位），其余为单字长指令。对于双字长指令，第一字为操作码，第二字为操作数；对于单字长指令只有操作码，没有操作数。上述所有指令的操作码均为低 8 位有效，高 8 位默认为 0。而操作数 8 位和 16 位均可。KIN 和 DISP 分别为本系统专用输入、输出设备。

## 七、实验步骤

单片机键盘操作方式实验。

在进行单片机键盘控制实验时，必须把 K4 开关置于“OFF”状态，否则系统处于自锁状态，无法进行实验。

### 1. 实验连线

实验连线图如图 8-3 所示。

连线时应按如下方法：对于横排座，应使排线插头上的箭头面向自己插在横排座上；对于竖排座，应使排线插头上的箭头面向左边插在竖排座上。

### 4. 写微代码

(1) 将开关 K1K2K3K4 拨到写状态即 K1 off、K2 on、K3 off、K4 off，其中 K1、K2、K3 在微程序控制电路，K4 在 24 位微代码输入及显示电路上。

(2) 在监控指示灯滚动显示【CLASS SELEcT】状态下按【实验选择】键，显示【ES--\_\_】输入 08 或 8，按【确认】键，显示为【ES08】，再按下【确认】键。

(3) 监控显示为【CtL1=\_】，输入 1 显示【CtL1\_1】，按【确认】。

(4) 监控显示【U-Addr】，此时输入【000000】6 位二进制数表示的微地址，然后按【确认】键，监控指示灯显示【U\_CodE】，显示这时输入微代码【007F88】，注意输入微代码的顺序，先右后左。按【确认】键则显示【PULSE】，按【单步】完成一条微代码的输入。

(5) 监控重新显示【U-Addr】提示输入表 8-4 第二条微代码地址。按照上面的方法输入表 8-4 微代码,观察微代码与微地址显示灯的对应关系（注意输入微代码的顺序是由右至左）。

表 8-4 微代码表

微地址（八进制）	微地址（二进制）	微代码（十六进制）
00	000000	007F88
01	000001	005B42
02	000010	016FFD
06	000110	015FE5
07	000111	015FE5
10	001000	005B4A
11	001001	005B4C
12	001010	014FFB
13	001011	007FC1
14	001100	01CFFC
20	010000	005B65
22	010010	005B47
23	010011	005B46
24	010100	007F15
25	010101	02F5C1
27	010111	018FC1
30	011000	0001C1
31	011001	0041EA
32	011010	0041EC
33	011011	0041F2
34	011100	0041F3
35	011101	0041F6
36	011110	3071F7
37	011111	3001F9
40	100000	0379C1
41	100001	010FC1
42	100010	011F41
45	100101	007F20
52	101010	0029EB
53	101011	9403C1
54	101100	0029ED

55	101101	3003C1
62	110010	0003C1
63	110011	0025F5
65	110101	B803C1
66	110110	0C03C1
67	110111	207DF8
70	111000	000DC1
71	111001	107DFA
72	111010	000DC1
73	111011	06F3C8
74	111100	FF73C9
75	111101	016E10

### 3. 读微代码及校验微代码

(1) 先将开关 K1K2K3K4 拨到读状态即 K1 off、K2 off、K3 on、K4 off，按【RESET】按钮对单片机复位，使监控指示灯滚动显示【CLASS SELEcT】状态。

(2) 按【实验选择】键，显示【ES--\_ \_】输入 08 或 8，按【确认】键，显示【ES08】。再按【确认】键。

(3) 监控显示【CtL1=\_】时，输入 2，按【确认】显示【U\_Addr】，此时输入 6 位二进制微地址，进入读代码状态。再按【确认】显示【PULSE】，此时按【单步】键，显示【U\_Addr】，微地址指示灯显示输入的微地址，微代码显示电路上显示该地址对应的微代码，至此完成一条微指令的读过程。

(4) 对照表 8-3 表检查微代码是否有错误，如有错误，可按步骤 2 重新输入微代码。

### 4. 写机器指令

(1) 先将 K1K2K3K4 拨到运行状态即 K1 on、K2 off、K3 on、K4 off，按【RESET】按钮对单片机复位，使监控指示灯滚动显示【CLASS SELEcT】状态。

(2) 按【实验选择】键，显示【ES--\_ \_】输入 08 或 8，按【确认】键，显示【ES08】，再按【确认】。

(3) 监控显示【CtL1=\_】，按【取消】键，监控指示灯显示【CtL2=\_】，输入 1 显示【CtL2\_1】表示进入对机器指令操作状态，此时拨动 CLR 清零开关（在控制开关电路上，注意对应的 JUI 应短接）对地址寄存器、指令寄存器清零。确定清零后，按【确认】显示闪烁【PULSE】。连续按【单步】键，当微地址显示灯显示“001100”时，按【确认】键，监控指示灯显示【data】，提示输入机器指令“48”或“0048”（两位或 4 位十六进制数），输入后按【确认】，显示【PULSE】，再按【单步】，微地址显示灯显示“111100”，数据总线显示灯显示“000000001001000”至此完成第一条机器指令的输入。

(4) 再连续按【单步】，微地址显示灯显示“001100”时，按【确认】输入第二条机器指令。依此规律逐条输入表 8-5 的机器指令，输完后，可连续按【取消】或【RESET】键退出写机器指令状态。**注意，每当微地址显示灯显示“001100”时，地址指示灯自动加 1 显示。如输入指令为 8 位，则高 8 位自动变为 0。**

表 8-5 机器指令表

地址（十六进制）	机器指令（十六进制）
00	0048
01	0005
02	0001
03	0094



04	00F0
05	0082
06	0046
07	0008
08	0000

### 5. 读机器指令及校验机器指令

在监控指示灯显示【Ctl2=\_】状态下，输入 2，显示【Ctl2\_2】，表示进入读机器指令状态，按步骤 4 的方法拨动 CLR 开关对地址寄存器和指令寄存器进行清零，然后按【确认】键，显示【PULSE】，连续按【单步】键，微地址显示灯显示从“000000”开始，然后按“001000”、“001010”、“111011”方式循环显示。当微地址灯再次显示为“001000”时，输出显示数码管上显示写入的机器指令。读的过程注意微地址显示灯，地址显示灯和数据总线指示灯的对应关系。如果发现机器指令有误，则需重新输入机器指令。

**注意：**机器指令存放在 RAM 里，掉电丢失，故断电后需重新输入。

### 6. 运行程序

在监控指示灯显示【Ctl2=\_】状态下，输入 3，显示【Ctl2\_3】，表示进入运行机器指令状态，按步骤 4 的方法拨动 CLR 开关对地址寄存器和指令寄存器进行清零，使程序入口地址为 00H，可以【单步】运行程序也可以【全速】运行，运行过程中提示输入相应的量，运行结束后从输出显示电路上观察结果。

## 八、实验结果说明

根据本实验的微程序流程图（图 8-2）来观察程序运行的过程，并验证运行结果是否正确。参考结果：输入“1111H”，输出显示“2224H”。

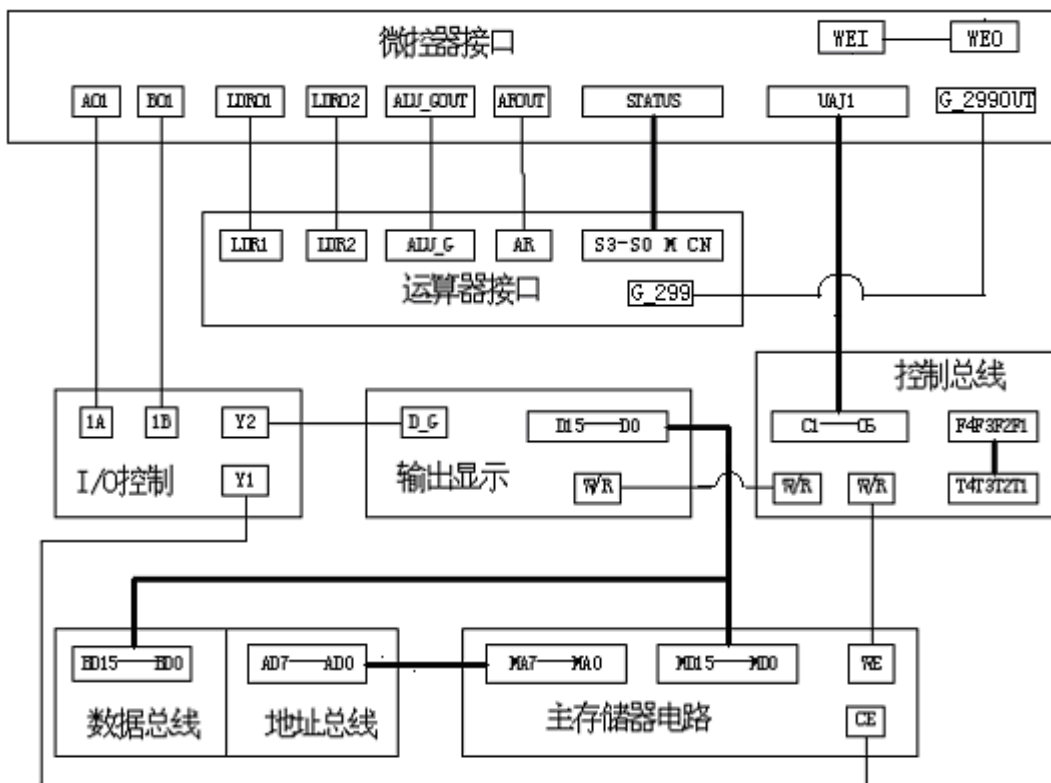


图 8-3 键盘实验连线图