

# 微机原理及应用

## DJ8086K 实验部分

## 目 录

<b>第一章 系统概述</b>	1
引言	
1.1 系统主要特点	1
1.2 系统资源分配	2
1.3 系统配置	3
<b>第二章 系统组成和结构</b>	4
2.1 系统接口定义	4
2.2 系统硬件组成	6
<b>第三章 系统安装与使用</b>	11
<b>第四章 键盘监控使用简介</b>	12
4.1 引言	12
4.2 键盘显示	12
4.3 功能键操作说明—缆表	12
4.4 总操作过程	13
4.5 监控程序命令及操作	13
<b>第五章 8086K 实验系统与 PC 机联机操作</b>	16
5.1 DJ-8086k 软件概述	16
5.2 DJ-8086k 软件安装	16
5.3 DJ-8086k 软件启动和联机	16
5.4 DJ-8086k 软件主窗口	17
5.5 菜单栏和工具栏命令简介	17
5.5.1 文件菜单栏	17
5.5.2 编辑菜单栏	18
5.5.3 调试菜单栏	18
5.5.4 设置	19
5.5.5 窗口	19
<b>第六章 实验指导</b>	20
实验说明	20
软件实验	21
实验一 二进制多位加法运算	21
实验二 二进制码转换为 BCD 码	22
实验三 BCD 码转换为二进制码	25
实验四 十进制数的 BCD 码相减运算	27
实验五 内存清零	29
实验六 数码显示	30
实验七 求最大值和最小值	33
实验八 数据块移动	37
实验九 多分支程序	40

<b>硬件实验</b>	43
实验一 A/D 转换实验	43
实验二 D/A 转换实验(一)	47
实验三 D/A 转换实验(二)	50
实验四 8255A 并行口实验(一)	53
实验五 8255A 并行口实验(二)	56
实验六 定时器 / 计数器	60
实验七 8259 单级中断控制器实验	63
实验八 串行接口和应用(一) 串行发送	69
实验九 串行接口和应用(二) 串行接收	79
实验十 小直流电机调速实验	82
实验十一 步进电机控制	86
实验十二 继电器控制	95
实验十三 存储器读写实验	97
实验十四 电子琴实验	100
实验十五 简单 I/O 口扩展实验	107
实验十六 8251 可编程通讯接口与 PC 机通讯	109
实验十七 LED16*16 点阵显示实验	116
实验十八 128×64 LCD 液晶显示实验	127
实验十九 8237 DMA 传送实验	138
实验二十 8250 串口实验	147
实验二十一 8279 键盘显示实验	155
实验二十二 温度控制实验	159
实验二十三 压力测量实验	164
附录一 实验程序目录	165
附录二 88 部分实验接线汇总表	167
附录三 DJ-8086K 字形字位表	168
附录四 DJ-8086K 键值表	169
附录五 DJ-598K 结构框图	170

## 第一章 系统概述

**引言** 本《实验指导书》适用于 DJ 系列 8086/88 微机实验开发系统, 本书就 DJ-598K 实验开发系统的 8086 部分作详细介绍, 其它机型均可参考。在使用 DJ 系列单片机、微机实验开发系统之前, 请务必先阅读本章内容, 以便对实验系统的硬件结构、硬件信息有个全面了解, 对后面实验有必不可少的帮助。

### 1.1 系统主要特点

- 1、598K 是集 51、96、8088 (以下简称 88) 三大系列 CPU 于一体的三合一实验系统。内置 5196 仿真器和 8086 实验系统。该系统以 8086 实验系统作为基板, 自成一体, 可单独作为 8086 微机原理和接口实验使用, 也可分别叠插 51 或 96CPU 卡, 构成 51/96 单片机实验开发系统, 598K 系统结构紧凑, 资源充分共享, 无总线开关切换, 自动识别 CPU 的类型。
- 2、8086 实验系统采用主频为 14.7MHZ 的 8088CPU 为 CPU, 并以最小工作方式构成系统。
- 3、配有二片 61C256 静态 RAM 构成系统的 64K 基本内存, 存放用户程序/数据, 地址范围为 00000H-0FFFFH, 其中 00000H-00FFFH 监控占用。  
配有二片 W27C512 EEPROM 64K, EP1 存放监控程序, 地址范围 F0000H—FFFFFH; EP2 存放实验程序, 为实验系统独立运行时下载实验程序提供方便。
- 4、自带键盘、显示器, 能独立运行, 为实验程序调试带来方便。
- 5、配备 Windows/XP 仿真调试软件, 支持机器码、汇编、C 等三种语言的开发和调试。
- 6、提供标准 RS232 异步通讯口, 以连接 IBM-PC 机。
- 7、配有各种微机常用 I/O 接口芯片, 定时/计数接口芯片 8253、A/D 转换接口芯片 0809、D/A 转换接口芯片 0832、中断控制器 8259、键盘显示接口 8279、并行 I/O 接口芯片 8255、通讯接口芯片 8250、8251、485 和 DMA 控制器 8237 等。
- 8、配备键盘、数码显示、发光二极管显示、开关量、LED16×16、LCD12864、温度压力、步进电机、直流电机、扬声器等输入输出设备。
- 9、带有脉冲发生器、计数器电路、单脉冲发生器等常用电路。

10、可以单步、断点、全速调试各实验程序。

11、内置稳压电源，为实验提供+5V/3A、±12V/0.5A 直流稳压电源。

12、使用环境：环境温度 0~+40℃，无明显潮湿，无明显振动碰撞。

## 1.2 系统资源分配

8088 有 1 兆存储空间，系统提供用户使用的空间为 00000H-0FFFFH，用于存放、调试实验程序。具体分配如下：

(1)存储器地址分配

系统监控程序区	F0000H-FFFFFH
监控/用户中断矢量	00000H-0000FH
用户中断矢量	00010H-000FFH
监控数据区	00100H-00FFFH
默认用户栈	00683H
用户数据/程序区	01000H-0FFFFH

(2)I/O 地址分配(一)：

地址	地址范围	用途
0FE00H	FE00H-FEFFFH	实验用口地址
0FF00H	FF00H-FF1FH	实验用口地址
0FF30H	FF30H-FF7FH	实验用口地址
0FF80H	FF80H-FF8FH	实验用口地址
0FF90H	FF90H-FF9FH	实验用口地址
0FFA0H	FFA0H-FFFFFH	实验用口地址

I/O 地址分配(二)：

0FF20H	系统8255PA口	键扫/字位口
0FF21H	系统8255PB口	字形口
0FF22H	系统8255PC口	键入口
0FF23H	系统8255控制口	控制口
0FF28H	扩展8255PA口	扩展用
0FF29H	扩展8255PB口	扩展用
0FF2AH	扩展8255PC口	扩展用
0FF2BH	扩展8255控制口	控制口

监控占用 00004H-0000FH 作为单步 (T)、断点 (INT3)、无条件暂停 (NMI) 中断矢量区, 用户也可以更改这些矢量, 指向用户的处理, 但失去了相应的单步、断点、暂停等监控功能。

F0000H—FFFFFH 监控程序区系统占用。

### 1.3 系统配置

系统内含串行通讯和键盘显示监控的全部软硬件设施, 典型配置如下:

(1) 主机板+PC 机

(2) 主机板

二种配置的使用特点如下:

配置一: 主机板和 PC 机通过 RS232 串行口相连。这种配置形式可以充分利用 PC 机的软硬件资源, 实现源程序的编辑、汇编和链接和调试。实验程序源文件 (•ASM) 在随机光盘 598K8ASM 文件夹, 你可以利用 PC 机的键盘显示等输入输出设备完成各种软硬件实验。

配置二: 为最经济型配置, 它不需要和 PC 机联机, 自成一体, 通过一组键盘命令完成所有软硬件实验, 每个实验程序固化在实验监控中。这种配置为没有 PC 机的用户带来极大方便。

第二章 系统组成和结构

8086（部分）实验系统由 8088 控制单元、实验单元、开关电源和 DJ8086K 集成调试软件组成。8088 控制单元由 8088CPU、CPLD（1032）、监控 EP1（27512）、内存 RAM1，2 和键盘显示电路组成。实验单元采用一体化、分模块设计，结构合理清晰。

2.1 系统接口定义

（1）CZ11：用户实验通讯接口；

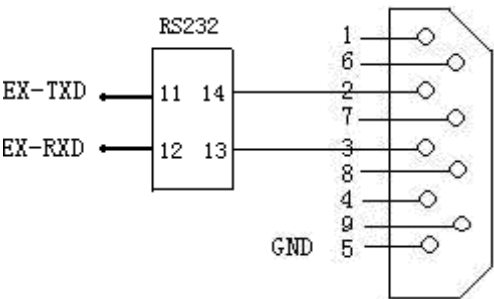


图 2-1

（2）CZ4：打印接口；

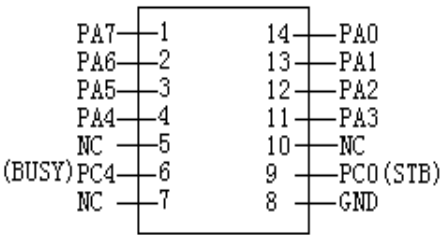


图 2-2

（3）JX0，JX17 为系统提供的数据库接口；

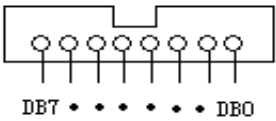


图 2-3

(4) CZ7：系统提供的扩展接口；

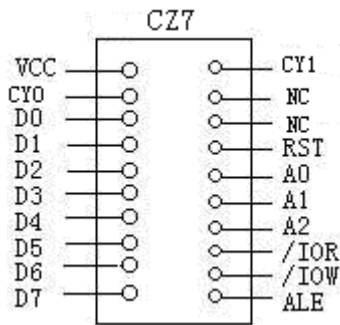


图 2-4

(5) ● JX12、JX14：液晶显示接口。

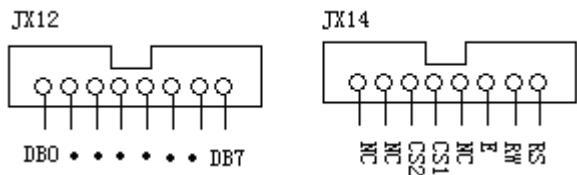
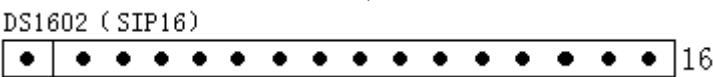


图 2-5—1

● LCD1602 引脚定义：

通过 JX12、JX14 接口，DJ8086 系列实验仪可以驱动一个标准的点阵字符液晶显示屏(16\*1 行、16\*2 行、16\*4 行)等，引脚信号如下图：



LCD1602 点阵字符液晶显示屏通用接口定义

引脚	符号	功能说明
1	Vss	电源地:0V
2	Vdd	电源:5V
3	Vadj	LCD 驱动电压:0V~5V
4	RS	寄存器选择:“0”指令寄存器;“1”数据寄存器
5	R/W	读写操作:“1”读操作;“0”写操作
6	E	LCD 使能信号
7~14	D0~D7	8 位双向数据信号线
15~16	V+、V-	背光照明电源输入正、负极



### ● LCD12864 引脚定义:

通过 JX12、JX14 接口, DJ8086 系列实验仪可以驱动显示一个标准的点阵液晶显示屏(128\*64)等, 点阵 LCD 液晶显示屏通用接口 J12864 的 20 个引脚信号的管脚定义如表 2.2 所示。

表 2.2 LCD12864 点阵液晶显示屏通用接口定义

引脚	符号	功能说明
1	V <sub>SS</sub>	电源地:0V
2	V <sub>DD</sub>	电源:5V
3	V <sub>ADJ</sub>	LCD 驱动电压:0.5~5V(正端)
4	RS	寄存器选择:“0”指令寄存器;“1”数据寄存器
5	R/W	读写操作:“1”读操作;“0”写操作
6	E	LCD 使能信号
7~14	D0~D7	8 位双向数据信号线
15	CS1	左半屏片选:“1”有效
16	CS2	右半屏片选:“1”有效
17	RST	复位脚:高复位
18	VEE	LCD 驱动电压:0V~5V(负端)
19~20	V <sub>+</sub> 、V <sub>-</sub>	背光照明电源输入正、负极

### (6) 开关及跳线:

JK: 为键盘/显示选择开关。JK 置系统, 键盘/显示选择系统配置的 8255 接口芯片, 进入监控状态; JK 置外接, 由用户选择自定义的 I/O 接口芯片控制。

KB6: 为通信选择开关。KB6 置 59, 选择 51/96 单片机系统和 PC 机通信; KB6 置 88, 选择 8086 系统和 PC 机通信(单 8086 系统无 KB6)。

JXT: 为 ISP 单元晶振选择跳线器。跳线器短接 XT, 锁紧插座 18, 19 脚和晶振源相连, 反之, 锁紧插座 18, 19 脚和晶振源断开。

## 2.2 系统硬件组成

- (1) LED 发光二极管指示电路: 实验台上包括 16 只发光二极管及相应驱动电路。见图 2-6, L1~L16 为相应发光二极管驱动信号输入端, 该输入端为低电平“0”时发光二极管亮。(箱式实验机简化为 12 只发光二极管)

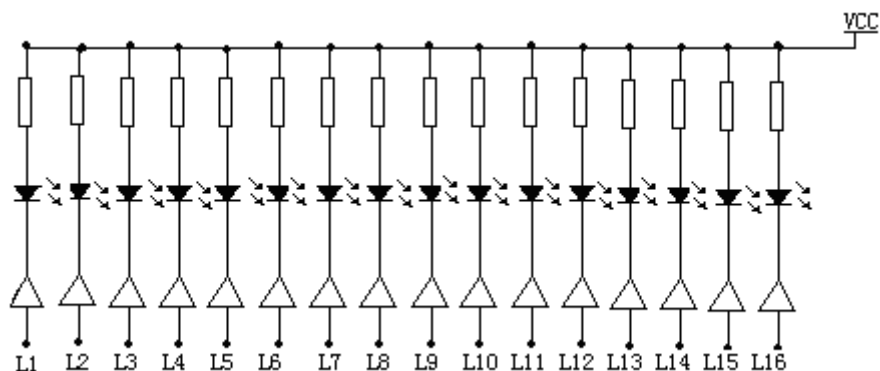


图 2-6

- (2) 逻辑电平开关电路：见图 2-7。实验台上有 8 只开关 K1-K8，与之相对应的 K1-K8 各引线孔为逻辑电平输出端。开关向上拨相应插孔输出高电平“1”，向下拨相应插孔输出低电平“0”。

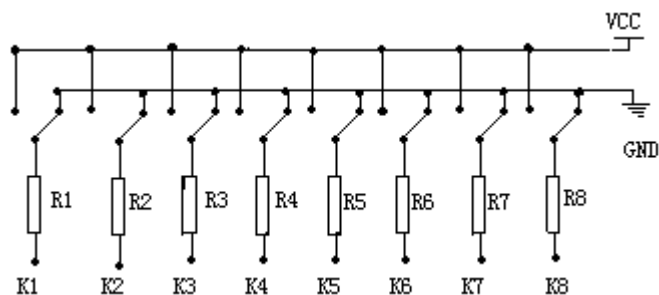




图 2-7

- (3) 单脉冲电路：实验台上单脉冲产生电路如图 2-8，标有“”和“”的两个引线插孔为正负单脉冲输出端。AN0 为单脉冲产生开关，每拨动一次来回产生一个单脉冲。

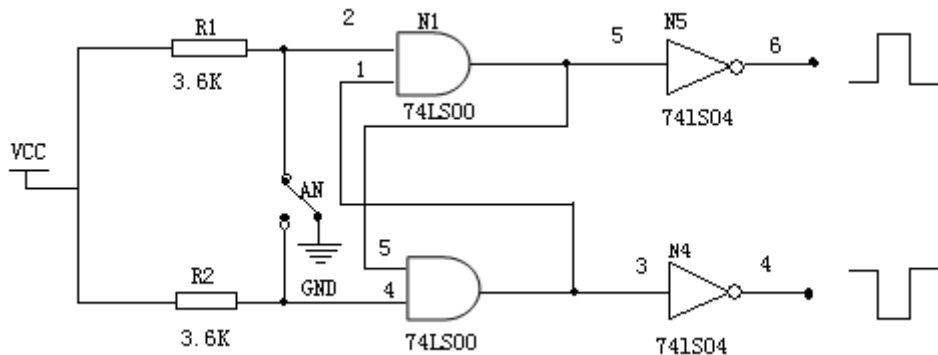


图 2-8

- (4) 分频电路：该电路由一片 74LS393 组成，见图 2-9。T0-T7 为分频输出插孔。该计数器在加电时由 RESET 信号清零。当脉冲输入为 8.0MHZ 时，T0-T7 输出脉冲频率依次为 4.0MHZ，2.0MHZ，1.0MHZ，500KHZ，250KHZ，125KHZ，62500HZ，31250HZ。

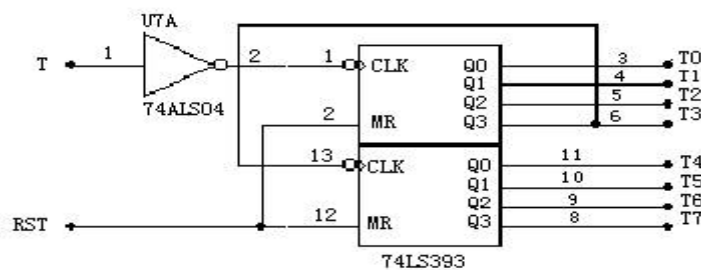


图 2-9

- (5) 脉冲发生电路：实验台上提供 8MHZ 的脉冲源，见图 2-10，实验台上标有 8MHZ 的插孔，即为脉冲的输出端。

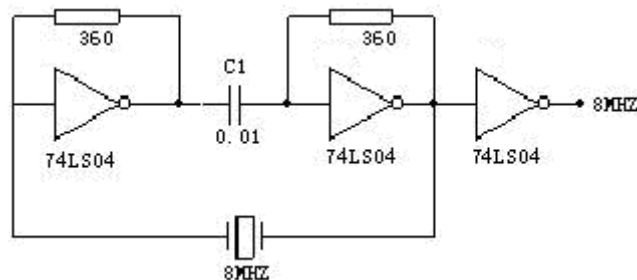


图 2-10

(6) 485 接口电路:

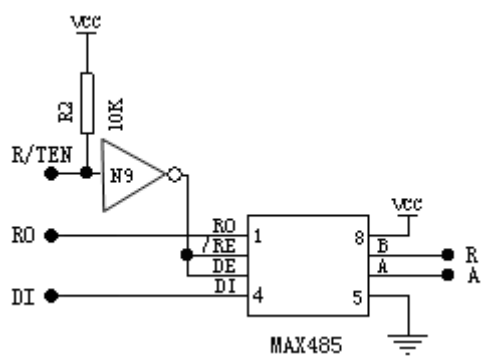


图 2-11

### (7) 系统通讯接口电路

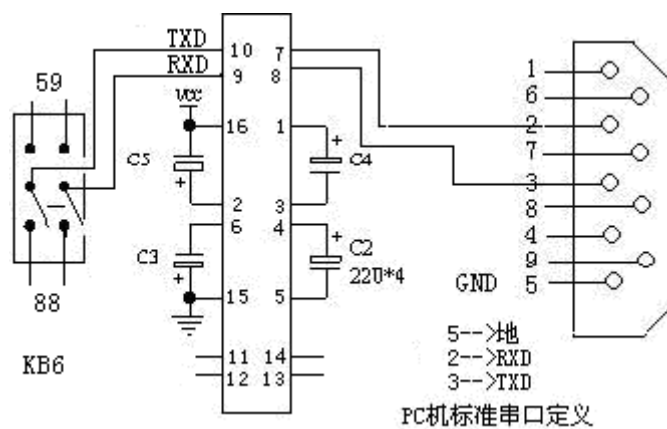


图 2-12

### (8) 数码管显示电路

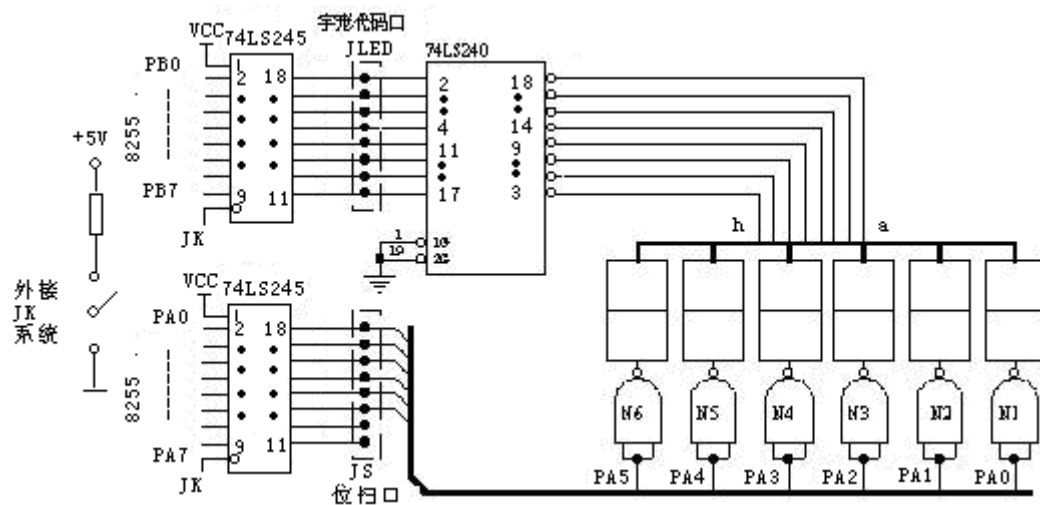


图 2-13

### (9) 键盘电路

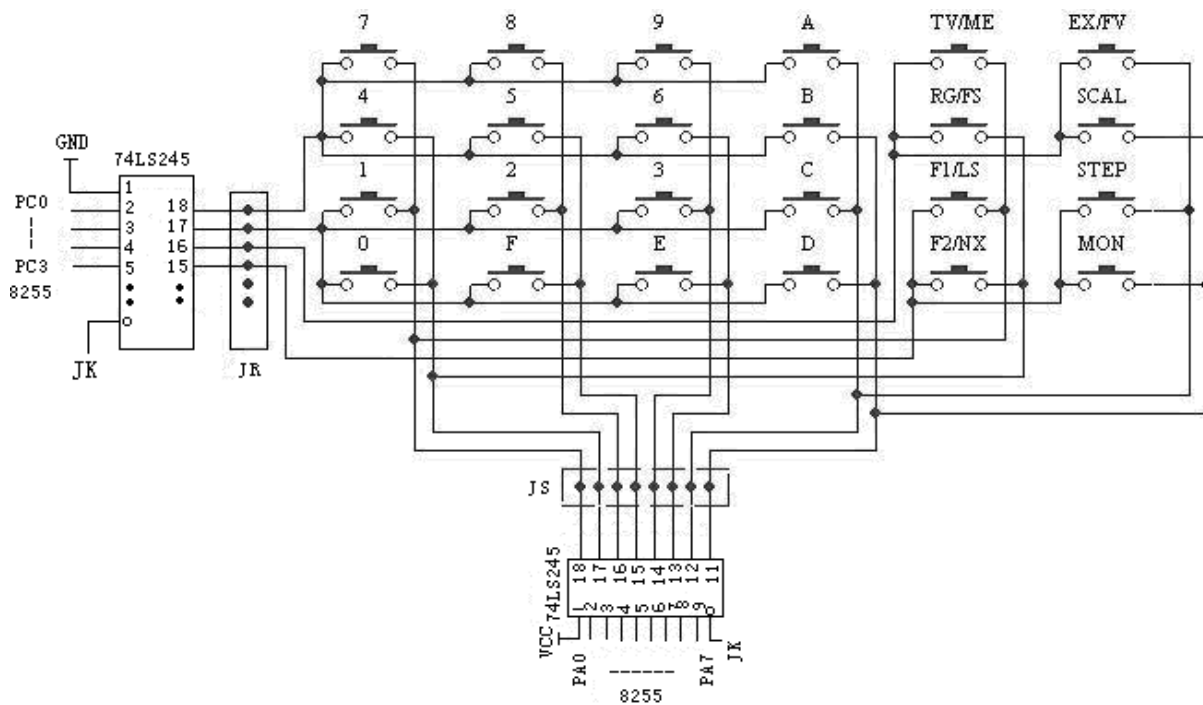


图 2-14

### 第三章 系统安装与使用

#### 8086 实验系统安装与使用

##### 1、键盘、显示设置：

JK（开关）置‘系统’：键盘/显示接口选择系统配置的 8255 接口芯片，进入监控状态。

- 2、KB6：通讯选择开关，KB6 开关拨向 88 一侧（单 8086 实验仪无 KB6），选择 PC 机和 8086 系统通讯。将随机配送的串行通讯线，一端与实验仪的 RS232D 型插座 CZ1 相连，另一端与 PC 机 COM1 或 COM2 串行口相连。
- 3、不插或卸掉已装的 CPU 卡（单 8086 实验仪无 CPU 卡安装、插座），接通实验系统电源，+5V LED 指示灯应正常发光，实验仪数码管应显示闪动 P.，说明实验仪初始化成功，处于待命状态。（否则应及时关闭电源，待修正常后使用）
- 4、打开 PC 机电源，执行 DJ8086K 的集成调试软件，关于该软件的详细说明请参阅《第五章 DJ8086K 实验系统与 PC 机联机操作》。

## 第四章 键盘监控使用简介

### 4.1 引言

本章详细叙述键盘监控程序的功能和操作方法。

系统一经接通电源，数码管上显示监控提示符“P.”。

### 4.2 键盘显示

- (1) 系统配备 6 位 LED 显示器，左边 4 位显示地址，右边 2 位显示该地址内容。
- (2) 系统具有 24 个注塑键盘，左边 16 个是数字键，右边 8 个是功能键。

在键盘监控状态下用户可以通过一组键命令完成下列操作

- 读写寄存器内容
- 读写存储器内容
- EPROM 传送
- 断点设置/清除
- 通过单步、断点、连续等功能来调试运行实验程序。

### 4.3 功能键操作说明一览表

键名	功 能	操 作
ME	存贮器读写	先输入四位数字的地址，再按ME，读存贮器单元的内容，此时输入数字键，则修改该单元的内容。
RG	寄存器读写	先输入寄存器代号，再按RG键读出该代号所表示的寄存器单元的内容。此时再输入数字键，则修改该寄存器的内容。
SCAL	EPROM传送	在“P”态，按SCAL键，把EP2中内容传送到内存RAM中。
STEP	单步运行	先输入四位起始地址，再按STEP键或在存贮器读写状态下按STEP键或在“P.”监控提示符状态下按STEP键，按一次执行一条指令语句。
EXEC	连续运行	先输入四位起始地址，按EXEC键或在存贮器读写状态下按EXEC键或在“P.”监控提示符下按EXEC键。
EXEC	断点运行	先输入四位断点地址后，再输入四位起始地址后按EXEC键。
F1	断点查看/设置	P.态时，按F1为断点查看，P.态时，输入四位数字的地址，再按F1，为断点设置。
F2	断点清除/ “8”字传送	在断点查看状态时，按F2为断点清除，在P.态时按F2，把8字循环程序传送到首址为FFC1开始的内存中。

#### 4.4 总操作过程

8086 十六位微机实验系统上电总清以后，显示器上显示监控提示符“P.”，各寄存器的初始化值如下：

SP=0683H, CS=0000H, DS=0000H, SS=0000H, ES=0000H, IP=1000H, FL=0000H

注意：• 所有命令均在提示符“P.”状态输入。

• 在键盘监控状态，段地址缺省值为 0000H。

#### 4.5 监控程序命令及操作

(1) 存贮器单元内容显示修改



操作：××××ME 即在提示符“P.”下，先输入四位存贮单元地址，再按 ME 键，显示器左边 4 位显示地址，右边 2 位显示该单元的内容。此时

- 按 NX 键使地址加 1 再显示。
- 按 LS 键使地址减 1 再显示。
- 键入十六进制数据，则将改写现行地址单元的内容。
- 按 MON 键则返回监控，显示提示符“P.”。

## (2) 寄存器内容显示修改

操作：×RG 即在“P.”提示符下，先输入寄存器代号，再按 RG 键，显示器右边 4 位显示寄存器中内容，左 1、2 位显示寄存器名。此时：

- 按 NX 键，则依次循环显示下一个寄存器中的内容。
- 按 LS 键，则依次显示上一个寄存器中的内容。
- 输入十六进制数字，则该寄存器中的内容被修改。
- 按 MON 键则返回监控，显示提示符“P.”。

## (3) EPROM 传送

操作：在提示符“P.”状态下，按 SCAL 键，开始 EPROM 传送，把 EPROM 中 1000H~FFFFH 内容传送到相应的内存中，传送时数码管显示闪动的“——”，传送完毕返回监控，显示提示符“P.”。

## (4) 单步运行

操作：有三种状态都可以进入单步操作(段地址固定为 0000H)

- 在监控提示符“P.”状态下，按 STEP 键，从默认的起始地址 IP=1000H 单元开始单步运行一条指令语句。

- 在存贮器单元读写状态(即左边 4 位显示存贮器地址，右边 2 位显示存贮器单元内容)下，从当前存贮器地址开始单步运行一条指令语句。

- 先输入四位起始地址，再按 STEP 键，则从输入的起始地址开始运行一条指令语句。每运行一条语句，显示器显示下一条指令语句的地址及其内容。

#### (5) 连续运行

操作：有三种状态可以进入连续运行

- 在监控提示符“P.”状态下，按 EXEC 键，从默认起始地址 CS=0000H，IP=1000H 开始连续运行用户程序。

- 在存储器读写状态下，按 EXEC 键，从现行存储器地址开始（CS=0000H）连续运行用户程序。

- 先输入 4 位起始地址，再按 EXEC 键，则从输入的起始地址开始连续运行用户程序（CS=0000H）。

要想从运行用户程序返回监控，可以按主机 RST 键，则重新进入监控状态，初始化部分寄存器。

#### (8) 断点运行（默认段地址 CS=0000H）

操作：××××F1, ××××EXEC，即在提示符“P.”状态下，输入 4 位断点地址，按 F1 键，再输入 4 位起始地址，按 EXEC 键，程序从起始地址开始运行，显示器显示“┐”执行符。当程序运行到断点地址处会自动停下来，并显示断点地址和内容，同时保护所有寄存器的值。

## 第五章 8086K 实验系统与 PC 机联机操作

### 5.1 DJ-8086k 软件概述

- (1) 该软件适用于 DJ-8086K、DJ-8086KP、DJ-598K 等实验仪。
- (2) 该软件运行环境：该软件适宜在安装 Windows98/2000/XP 操作系统的 PC 机上运行。

### 5.2 DJ-8086k 软件安装

运行随机光盘上“DJ-8086k.EXE”安装文件，根据提示完成软件安装，然后双击桌面上‘DJ-8086k’快捷图标，即可运行 DJ-8086k 软件。

### 5.3 DJ-8086k 软件启动和联机

#### 5.3.1 启动和联机

双击桌面上‘DJ-8086k’快捷图标，即可运行 DJ-8086k 软件，屏幕显示 DJ-8086k 软件的工作窗口，底边状态栏由“下位机没有连接…”变成“连接上下位机”，说明联机成功，否则会弹出对话框提示：“下位机没有连接…”，此时单击“OK”，实验系统和 PC 机处于脱机状态，然后用户根据连接硬件实际情况来设定 COM1 或 COM2，设定方法如下：依次单击菜单栏：“设置(X)、通讯口设置(Z)、COM1 或 COM2”确定，然后点击‘打开端口’，状态设定后，系统会自动检测连接。如果此时确定端口正确，但系统仍然没有连接，可按以下两种方法解决：（1）单击工具栏上的“重新连接”按钮，即可联机；（2）复位实验系统，使得数码管上显示监控提示符“P.”，然后关闭 DJ-8086k 软件，重新运行，故障即可排除。

#### 5.3.2 联机失败原因及解决办法

- (1) 实验系统与 PC 机没有连接；
- (2) 实验系统电源没有打开；
- (3) 串口实际位置与软件设置不一致；
- (4) 通讯开关位置不正确，检查实验系统各状态设置开关是否正确；

(5) 软件有病毒；

(6) 实验系统或 PC 机处于非正常状态。

对于(1)(2)(4)(5)的错误，只要在硬件上检查一下，或用杀毒软件清毒一下即可；如文件已被破坏或需软件升级，可到本公司网站下载，网址：WWW.QD-DJ.CN。对于第(3)种错误，重复 5.3.1 操作，对第(6)种错误，联机之前先按一下实验系统 RST 复位按钮，使实验系统处于 P 态，否则说明实验系统不正常，应送当地销售部门或返厂维修。如 PC 机工作失常也会引起联机失败，可关闭 DJ-8086k 软件，重新运行或调换 PC 机试一下。

#### 5.4 DJ-8086k 软件主窗口

① 顶部为菜单栏和工具栏，提供调试全部命令和功能。

② 中间部分为工作窗口区，提供软件调试、寄存器、标志位、存储器、汇编代码对照、编译信息显示等窗口。

③ 底部为状态栏，提示软件联机状态等信息。

#### 5.5 菜单栏和工具栏命令简介

工具栏使用：只要移动鼠标，指向工具栏中选定的图标，即会提示该图标所执行的命令。

##### 5.5.1 文件菜单栏

###### ● 新建文件

单击菜单栏“文件”或工具栏“新建”图标，即会建立一个新的源程序编辑窗口，此时可在此窗口编辑、输入源程序。

###### ● F3 打开文件

单击菜单栏“文件”或工具栏“打开”图标，弹出“打开文件”的对话框，然后选择要装入的源文件，单击“确定”即可装入源文件。

###### ● 关闭

单击菜单栏“文件”下的“关闭”即可关闭源程序编辑窗口。

● 保存

单击菜单栏“文件”下的“保存”或按 CTRL+S，即可保存文件。

● N 建立新文件

分别单击“F 文件”、“建立新文件”即可创建新文件，再选择“换名存盘”填入所存文件名即可。

● 另存为

单击菜单栏“文件”下的“另存为”，可弹出另存为对话框，在此可选择新的文件名和保存文件夹，单击“保存”完成保存操作。

● 退出

依次单击“文件”、“退出”或单击菜单栏上的最后一个门形按钮“关闭按钮”即可退出 DJ-8086k 软件调试环境。

5.5.2 编辑菜单栏

剪贴 CTRL+X	删除程序中选定的正文，同时将它复制到剪贴板中
复制 CTRL+C	保留选定的正文，同时将它复制到剪贴板中
粘贴 CTRL +V	将剪贴板中内容复制在光标处
删除 DEL	删除选中的正文
全选 CTRL+A	选中源程序编辑窗口中所有的正文
ALT+BackSpace	可恢复上几次误操作

5.5.3 调试菜单栏

● 编译装载 F9

打开源文件，依次单击：调试”\“编译装载 F9”或工具栏上的编译装载图标，状态栏会提示正在编译、编译成功、正在装载数据、装载数据结束，几秒钟后便会弹出‘源文件调试窗口’，如编译出错，会提示错误信息，待修改源文件正确后，重新编译装载即可。

● 连续运行

单击菜单栏“调试”下的“连续运行”或单击工具栏的连续运行图标，即全速连续运行程序。

- 程序单步

单击菜单栏“调试”下的“程序单步”或单击工具栏的程序单步图标，即可单步运行程序。

- 重新连接

可单击菜单栏中“重新连接“，也可单击工具栏的”重新连接“图标，即可重新连接上下位机。

- 复位

菜单栏中“复位”无效，可单击工具栏的“复位”图标，即可复位软件，寄存器置初值，指令指针返回到程序首地址。

#### 5.5.4 设置

- 单击“设置”下的“通讯口设置”或工具栏上的“通讯口设置”即可打开“通讯设置”对话框，在此可设置端口。

#### 5.5.5 窗口

在此菜单可设置窗口的排列顺序，可打开相应的工作窗口。具体详见中文菜单。

## 第六章 实验指导

### 实验说明

本章是为 8086 十六位微机实验系统编写的详细实验指导,系统上提供的全套实验是为《微机原理和接口》课程配置的。手册中详细叙述了各实验的实验目的、实验内容、实验线路原理图、实验步骤、实验程序框图和程序清单,减轻和免除了主讲教师和实验辅导教师为设计、准备、调试实验线路和实验程序所需的工作量,节约了宝贵的时间。书中所有软硬件实验都已经过调试运行。需要特别说明的三点是:

一、所有实验程序用两种方式存放:其一放在随机光盘中,源程序(.ASM)在 598K8ASM 目录下;其二源程序的目标代码存放在实验监控 EP2 中,各个实验程序在 EPROM 中的存放入口地址详见附表。

当选择主板+PC 系统配置时,你既可以运行存放在 EPROM 中的实验程序,也可以运行 598K8ASM 目录下的实验程序。

当你选择主板这种系统配置时,你可以通过系统自带的键盘显示输入各种命令运行实验监控中的实验程序,显示实验结果,完成各个实验,这种配置为没有 PC 机的用户带来极大方便。

二、EPROM 内的实验程序通过 EPROM 传送命令把 EPROM 中内容传送到内存才能运行。(按下实验系统键盘上的 SCAL 键,即可把 EPROM 中的内容传送到内存 RAM、或按下 DJ-8086K 软件工具栏上的“读 EPROM 到内存 RAM”图标,完成传送。)

三、所有实验都是相互独立的,次序上也没有固定的先后关系,在使用本系统进行教学时,教师可以根据本校(院)的教学要求,选择相应的实验。

对同一问题的解决办法往往不是唯一的,我们欢迎用户在使用本系统过程中提出最为优秀的实验方案,并对本系统提出改进意见。

四、各实验原理图中的粗黑线表示在实验时需要连接的线。

五、以下实验请严格按照“实验步骤”来操作,操作次序不要颠倒。

## 软件实验

### 实验一 二进制多位加法运算

#### 一、实验目的

- 1、熟悉实验系统键盘显示的使用。
- 2、掌握使用加法类运算指令编程及调试方法。
- 3、掌握加法类指令对状态标志位的影响。

#### 二、实验内容

计算  $X+Y+Z=?$  ( $X$ 、 $Y$ 、 $Z$  为 16 位二进制数)

#### 三、编程指南

由于本实验是三个十六位二进制数相加运算，因此，当  $X+Y$  时要考虑用 ADC 指令，把进位 C 加到结果的高 16 位中，当  $(X+Y)+Z$  时，再把进位 C 加到结果的高 16 位中，本实验设定三个加数 0FFFFH，计算结果应为 2FFFDH。

#### 四、实验步骤

联机模式：

(1) 在 PC 机和实验系统联机状态下，运行该实验程序，可用鼠标左键单击菜单栏“文件”或工具栏“打开图标”，弹出“打开文件”的对话框，然后打开 598k8Asm 文件夹，点击 S1.ASM 文件，单击“确定”即可装入源文件，再单击工具栏中编译装载，即可完成源文件自动编译、装载目标代码功能，再单击“调试”中“连续运行”或工具图标运行，即开始运行程序。

(2) 运算结果保留在内存 4100H~4102H 单元中，点击 DJ-8086K 软件中存储器 RAM 窗口，输入 RAM 的起始地址 4100、4101、4102，单元内容应为 FD、FF、02。

脱机模式：

(1)在 P. 态下，按 SCAL 键，输入 2CA0，按 EXEC 键。



(2)复位 RST 键,运算结果保留在内存 4100H~4102H 单元中,共内容分别为:FD、FF、02,低位在前,高位在后,具体操作方法:输入键 [4100]→[ME]→[NX]→[NX]。

## 五、实验程序清单

```
CODE          SEGMENT                ;S1.ASM-->BIN ADD
ASSUME        CS:CODE
              ORG 2CA0H                ;this is a program of add
START:        CLC
              MOV SI,4000H              ;result in [4100]
              MOV [SI],0ffffH
              MOV [SI+2],0ffffH
              MOV [SI+4],0ffffH
              MOV AX,0000H
              MOV [SI+102H],AX
              MOV AX,[SI]
              ADD AX,[SI+2]
              ADC [SI+102H],0000
              ADD AX,[SI+4]
              MOV [SI+100H],AX
              ADC [SI+102H],0000
              JMP $
CODE ENDS
END  START
```

## 实验二 二进制码转换为 BCD 码

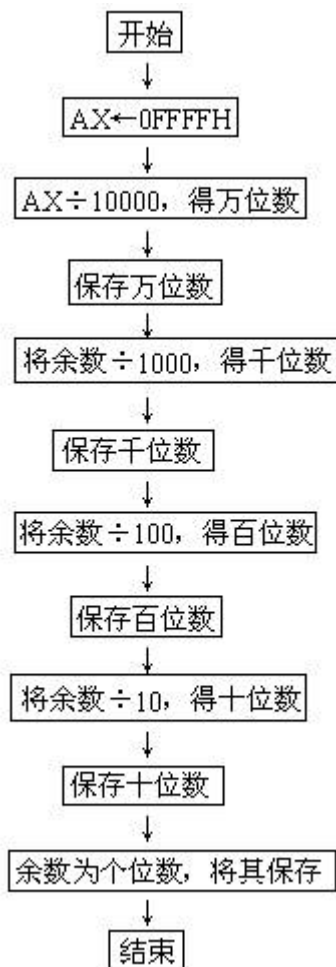
### 一、实验目的

- 1、掌握数码转换基本方法,加深对数码的理解。
- 2、用于十进制 BCD 码显示。

### 二、实验内容

将 AX 的内容转换为十进制 BCD 码。

### 三、实验程序框图



#### 四、实验步骤

联机模式：

(1) 在 PC 机和实验系统联机状态下，运行该实验程序，可用鼠标左键单击菜单栏“文件”或工具栏“打开图标”，弹出“打开文件”的对话框，然后打开 598k8Asm 文件夹，点击 S2.ASM 文件，单击“确定”即可装入源文件，再单击工具栏中编译装载，即可完成源文件自动编译、装载目标代码功能，再单击“调试”中“连续运行”或工具图标运行，即开始运行程序。

(2) 复位 RST 键，由于 AX 中给定数为 0FFFF，查看 BCD 码结果保留在 4100H～4104H 单元中，故其值应为 06、05、05、03、05。

脱机模式：

(1)在 P. 态，按 SCAL 键，输入 2CE0，按 EXEC 键。

(2)复位 RST 键，由于 AX 中给定数为 0FFFF，查看 BCD 码结果保留在 4100H~4104H 单元中，故其值应为 06、05、05、03、05。

## 五、实验程序清单

```
;将 AX 拆为 5 个 BCD 码，并存入 Result 开始的 5 个单元
data    segment AT 0                                ;S2. ASM, BIN-->BCD
        ORG 4100H
        Result db 5 dup(?)
        data    ends
code     segment
assume  cs:code, ds:data
        ORG 2CE0H
        start  proc  near
        mov     ax, data
        mov     ds, ax
        mov     dx, 0000h
        mov     ax, 65535
        mov     cx, 10000
        div     cx
        mov     Result, al                          ; 除以 10000, 得 wan 位数
        mov     ax, dx
        mov     dx, 0000h
        mov     cx, 1000
        div     cx
        mov     Result+1, al                        ; 除以 1000, 得 qian 位数
        mov     ax, dx
        mov     dx, 0000h
        mov     cx, 100
        div     cx
        mov     Result+2, al                        ; 除以 100, 得 bai 位数
        mov     ax, dx
        mov     dx, 0000h
        mov     cx, 10
        div     cx
        mov     Result+3, al                        ; 除以 10, 得 shi 位数
        mov     Result+4, dl                        ; 得 ge 位数
```

```

        jmp      $
code    ends
end start

```

### 实验三 BCD 码转换为二进制码

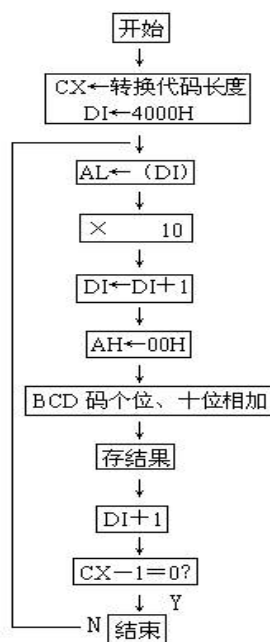
#### 一、实验目的

- 1、掌握 BCD 码到二进制码的转换方法。
- 2、用于 BCD 码输入信号处理。

#### 二、实验内容

对于内存中给定的几个十进制 BCD 码数转换为二进制数。

#### 三、实验程序框图



#### 四、实验步骤

联机模式：

(1) 在内存单元 4000H~4006H 中依次写入四个十进制数 (12、34、56、78) 的 BCD 码，即输入 01、02、03、04、05、06、07，按 MON 键，返回 P 状态。

(2)在 PC 机和实验系统联机状态下，运行该实验程序，可用鼠标左键单击菜单栏“文件”或工具栏“打开图标”，弹出“打开文件”的对话框，然后打开 598K8ASM 文件夹，点击 S3.ASM 文件，单击“确定”即可装入源文件，再单击工具栏中编译装载，即可完成源文件自动编译、装载目标代码功能，再单击“调试”中“连续运行”或工具图标运行，即开始运行程序。

(3)复位 RST 键，查看结果在 4100~4107 中，其内容应为：0C, 00, 22, 00, 38, 00, 4E, 00。

脱机模式：

(1)在 P. 态下，按 SCAL 键，然后在内存单元 4000H~4006H 中依次写入四个十进制数（12、34、56、78）的 BCD 码，即输入 01、02、03、04、05、06、07，按 MON 键，返回 P 状态。

(2)在 P. 态下，输入 2D40，按 EXEC 键。

(3)复位 RST 键，查看结果在 4100~4107 中，其内容应为：0C, 00, 22, 00, 38, 00, 4E, 00。

## 五、实验程序清单

```
CODE          SEGMENT                ;S3.ASM ,BCD 码转换为 2 进制码
ASSUME        CS:CODE
              ORG 2D40H
START:        MOV CX,0004H            ;INPUT DATA IN [4000]:01,02..
              MOV DI,4000H            ;result in [4100]: 0C,00,..
CON1:         MOV AL,[DI]
              ADD AL,AL                ;*2
              MOV BL,AL
              ADD AL,AL                ;*2
              ADD AL,AL                ;*2
              ADD AL,BL                ;*10
              INC DI
              MOV AH,00
              ADD AL,[DI]
              MOV [DI+0FFH],AX
              INC DI
              LOOP CON1
              JMP $
```

```
CODE ENDS
```

```
END START
```

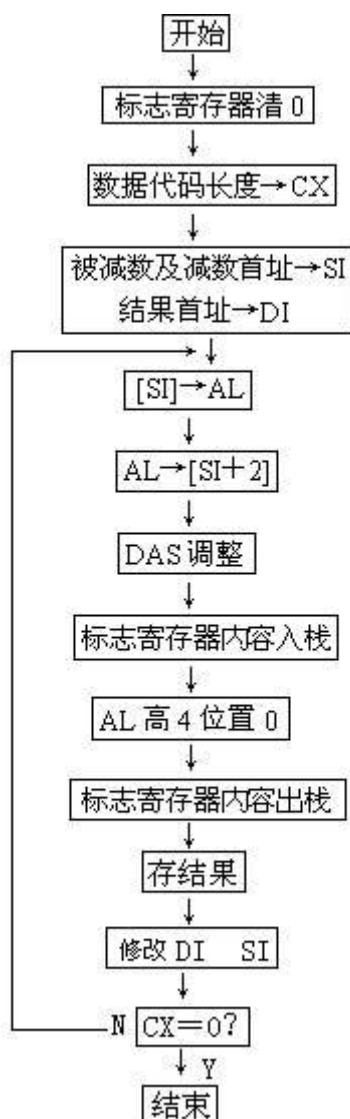
## 实验四 十进制数的 BCD 码相减运算

一、实验目的：熟悉微机对十进制数减法的运算方法。

二、实验内容：

计算  $X - Y = Z$ ，其中 X、Y、Z 为 BCD 码，设 X、Y 为 40，12，则 Z 为 28。

三、实验程序框图



## 四、实验步骤

联机模式：

(1) 在 PC 机和实验系统联机状态下，运行该实验程序，可用鼠标左键单击菜单栏“文件”或工具栏“打开图标”，弹出“打开文件”的对话框，然后打开 598K8ASM 文件夹，点击 S4.ASM 文件，单击“确定”即可装入源文件，再单击工具栏中编译装载，即可完成源文件自动编译、装载目标代码功能，再单击“调试”中“连续运行”或工具图标运行，即开始运行程序。

(2) 复位 RST 键，查看减法运算结果在 4100H~4101H 单元中，其内容应为 08、02。

脱机模式：

(1) 在 P. 态下，按 SCAL 键，输入 2D80，按 EXEC 键。

(2) 复位 RST 键，查看减法运算结果在 4100H~4101H 单元中，其内容应为 08、02。

## 五、实验程序清单

```
CODE          SEGMENT          ;S4.ASM, BCD-BCD=?
ASSUME        CS:CODE
              ORG 2D80H          ;this is a program of bcd sbb, 40-12=?
START:        MOV AH, 00H        ;result in [4100]=28
              SAHF
              MOV CX, 0002
              MOV SI, 4000H
              MOV DI, 4100H
              MOV [SI], 0400H
              MOV [SI+2], 0102H
CON1:         MOV AL, [SI]
              SBB AL, [SI+2]
              DAS
              PUSHF
              AND AL, 0FH
              POPF
              MOV [DI], AL
              INC DI
```

```
        INC SI
        LOOP CON1
        JMP $

CODE ENDS
END     START
```

## 实验五 内存清零

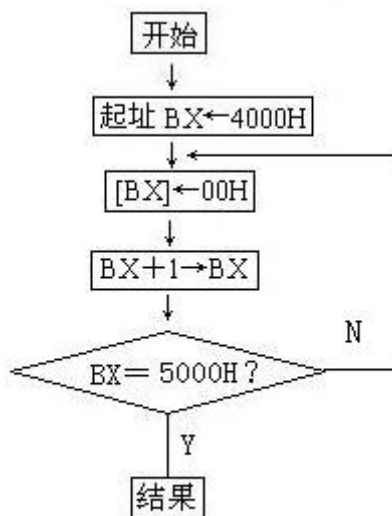
### 一、实验目的

- 1、掌握内存读写方法。
- 2、了解内存的块操作方法。

### 二、实验内容

对指定内存起址和长度的块，要求将其内容清零，设起址为 4000H，长度为 1000H。

### 三、实验程序框图



### 四、实验步骤

联机模式：



(1) 在 PC 机和实验系统联机状态下, 运行该实验程序, 可用鼠标左键单击菜单栏“文件”或工具栏“打开图标”, 弹出“打开文件”的对话框, 然后打开 598K8ASM 文件夹, 点击 S5.ASM 文件, 单击“确定”即可装入源文件, 再单击工具栏中编译装载, 即可完成源文件自动编译、装载目标代码功能, 再单击“调试”中“连续运行”或工具图标运行, 即开始运行程序。

(2) 按复位键 RST, 查看内存 4000H~4FFFH 单元内容应为 00

脱机模式:

1、在 P. 态下, 按 SCAL 键, 输入 2DC0, 按 EXEC 键。

2、按复位键 RST, 查看内存 4000H~4FFFH 单元内容应为 00

## 五、实验程序清单

```
CODE      SEGMENT                ;S5. ASM, CLR
ASSUME     CS:CODE
          ORG 2DC0H                ;this is a program of clear "0"
START:     MOV AL, 00H              ;CLEAR RAM:4000--4FFFH
          MOV BX, 4000H
CON1:      MOV [BX], AL
          INC BX
          CMP BX, 5000H
          JNE CON1
          JMP $
CODE ENDS
END  START
```

## 实验六 数码显示

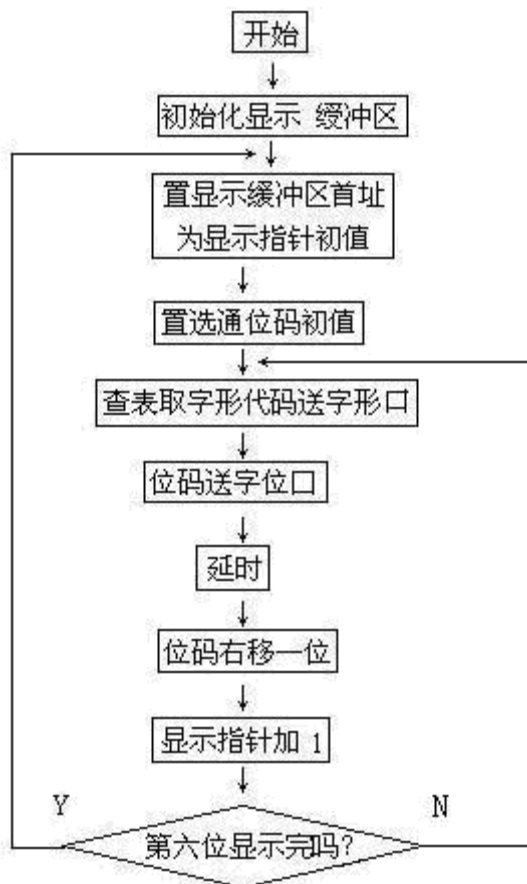
### 一、实验目的

了解 LED 数码管动态显示的工作原理及编程方法。

### 二、实验内容

编制程序, 使数码管显示“DJ--88”字样。

### 三、实验程序框图



### 四、实验步骤

联机模式：

(1) 在 PC 机和实验系统联机状态下，运行该实验程序，可用鼠标左键单击菜单栏“文件”或工具栏“打开图标”，弹出“打开文件”的对话框，然后打开 598K8ASM 文件夹，点击 S6.ASM 文件，单击“确定”即可装入源文件，再单击工具栏中编译装载，即可完成源文件自动编译、装载目标代码功能，再单击“调试”中“连续运行”或工具图标运行，即开始运行程序。

(2) 数码管显示“DJ--88”字样。

脱机模式：

1、在 P. 态下，按 SCAL 键，输入 2DF0，按 EXEC 键。

## 2、数码管显示“DJ--88”字样。

## 五、实验程序清单

```

CODE          SEGMENT                ;S6.ASM display "DJ--88"
ASSUME        CS:CODE
              ORG 2DF0H
START:        JMP START0
              PA      EQU 0FF20H      ;字位口
              PB      EQU 0FF21H      ;字形口
              PC      EQU 0FF22H      ;键入口
              BUF     DB ?, ?, ?, ?, ?, ?

data1:
              db0c0h, 0f9h, 0a4h, 0b0h, 99h, 92h, 82h, 0f8h, 80h, 90h, 88h, 83h,
              0c6h, 0a1h
              db
              86h, 8eh, 0ffh, 0ch, 89h, 0deh, 0c7h, 8ch, 0f3h, 0bfh, 8FH, 0F0H
START0:       CALL BUF1
CON1:         CALL DISP
              JMP CON1
DISP:         MOV AL, 0FFH              ;00H
              MOV DX, PA
              OUT DX, AL
              MOV CL, 0DFH              ;显示子程序 , 5ms
              MOV BX, OFFSET BUF
DIS1:         MOV AL, [BX]
              MOV AH, 00H
              PUSH BX
              MOV BX, OFFSET DATA1
              ADD BX, AX
              MOV AL, [BX]
              POP BX
              MOV DX, PB
              OUT DX, AL
              MOV AL, CL
              MOV DX, PA
              OUT DX, AL
              PUSH CX
DIS2:         MOV CX, 00A0H

```

```

        LOOP $
        POP CX
        CMP CL, 0FEH          ;01H
        JZ LX1
        MOV AL, 0FFH          ;00H
        MOV DX, PA
        OUT DX, AL
        INC BX
        ROR CL, 1              ;SHR CL, 1
        JMP DIS1
LX1:    MOV AL, 0FFH
        MOV DX, PB
        OUT DX, AL
        RET
BUF1:   MOV BUF, 0DH
        MOV BUF+1, 19H
        MOV BUF+2, 17H
        MOV BUF+3, 17H
        MOV BUF+4, 08H
        MOV BUF+5, 08H
        RET
CODE ENDS
END START

```

## 实验七 求最大值和最小值

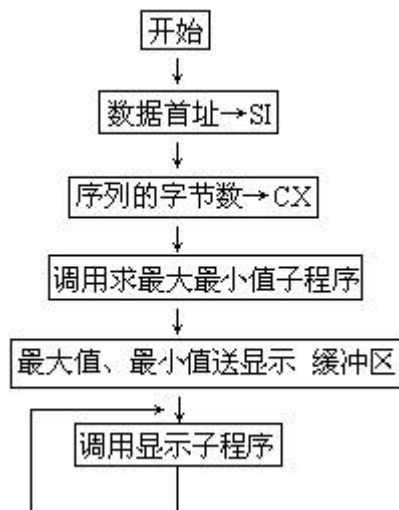
### 一、实验目的

- 1、学习子程序的定义和调用方法。
- 2、掌握子程序设计、调试。

### 二、实验内容

对内存中给定的几个无符号字节数，求其最大值和最小值并在数码管上显示。

### 三、实验程序框图



#### 四、实验步骤

联机模式：

(1) 在内存 4000H~4007H 中写入任意八个字节的数，按下 MON 键，返回 P 状态。

(2) 在 PC 机和实验系统联机状态下，运行该实验程序，可用鼠标左键单击菜单栏“文件”或工具栏“打开图标”，弹出“打开文件”的对话框，然后打开 598K8ASM 文件夹，点击 S7.ASM 文件，单击“确定”即可装入源文件，再单击工具栏中编译装载，即可完成源文件自动编译、装载目标代码功能，再单击“调试”中“连续运行”或工具图标运行，即开始运行程序。

(3) 数码管显示为：“XX——XX”，最左两位为最大值，最右两位为最小值。

脱机模式：

1、在 P. 态下，按 SCAL 键，然后在内存 4000H~4007H 中写入任意八个字节的数，按下 MON 键，返回 P 状态。

2、在 P. 态下，输入 2E70，按 EXEC 键。

3、数码管显示为：“XX——XX”，最左两位为最大值，最右两位为最小值。

## 五、实验程序清单

```

CODE          SEGMENT                ;S7. ASM, LOOK FOR MAX & MIN
ASSUME        CS:CODE
              ORG 2E70H                ;INPUT DATA 4000H--4007H
START:        JMP START0
              PA      EQU 0FF20H        ;字位口
              PB      EQU 0FF21H        ;字形口
              PC      EQU 0FF22H        ;键入口
              BUF DB  ?, ?, ?, ?, ?, ?

data1:
              db0c0h, 0f9h, 0a4h, 0b0h, 99h, 92h, 82h, 0f8h, 80h, 90h, 88h, 83h,
              0c6h, 0a1h
              db 86h, 8eh, 0ffh, 0ch, 89h, 0deh, 0c7h, 8ch, 0f3h, 0bfh, 8FH
START0:       MOV SI, 4000H
              MOV CX, 0008H
              CALL MAXMIN
              CALL BUF1
CON1:         CALL DISP
              JMP CON1
MAXMIN:       JCXZ EXIT
              PUSH SI
              PUSH CX
              PUSH BX
              MOV BH, [SI]
              MOV BL, BH
CON2:         LODSB
              CMP AL, BH
              JNA X1
              MOV BH, AL
              JMP X2
X1:           CMP AL, BL
              JNB X2
              MOV BL, AL
X2:           LOOP CON2
              MOV AX, BX
              POP BX
              POP CX
              POP SI

```

```

EXIT:      RET
DISP:      MOV AL, 0FFH                ;00H
           MOV DX, PA
           OUT DX, AL
           MOV CL, 0DFH                ;显示子程序 , 5ms
           MOV BX, OFFSET BUF
DIS1:      MOV AL, [BX]
           MOV AH, 00H
           PUSH BX
           MOV BX, OFFSET DATA1
           ADD BX, AX
           MOV AL, [BX]
           POP BX
           MOV DX, PB
           OUT DX, AL
           MOV AL, CL
           MOV DX, PA
           OUT DX, AL
           PUSH CX
DIS2:      MOV CX, 00A0H
           LOOP $
           POP CX
           CMP CL, 0FEH                ;01H
           JZ LX1
           MOV AL, 0FFH                ;00H
           MOV DX, PA
           OUT DX, AL
           INC BX
           ROR CL, 1                    ;SHR CL, 1
           JMP DIS1
LX1:      MOV AL, 0FFH
           MOV DX, PB
           OUT DX, AL
           RET
BUF1:      MOV BH, AH
           CALL ZH
           MOV BUF+5, AL
           MOV BUF+4, AH
           MOV AL, BH

```

```
CALL ZH
MOV BUF, AH
MOV BUF+1, AL
MOV BUF+2, 17H
MOV BUF+3, 17H
RET
;-----
ZH:    MOV BL, AL
        MOV CL, 4
        ROL AL, CL
        AND AL, 0FH
        MOV AH, AL
        MOV AL, BL
        AND AL, 0FH
        RET
;-----
CODE ENDS
END START
```

## 实验八 数据块移动

### 一、实验目的

- 1、了解内存中数据块移动方法。
- 2、掌握分支程序的设计。

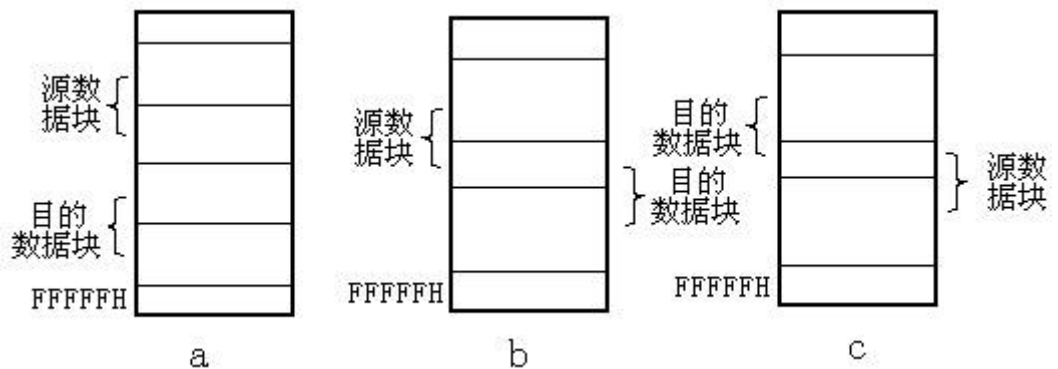
### 二、实验内容

设计一通用数据块间数据搬移程序。

### 三、编程指南

程序要求把内存中一数据区（称为源数据块）传送到内存另一数据区（称为目的数据块）。源数据块和目的数据块在存贮中可能有三种情况，如下图所示。

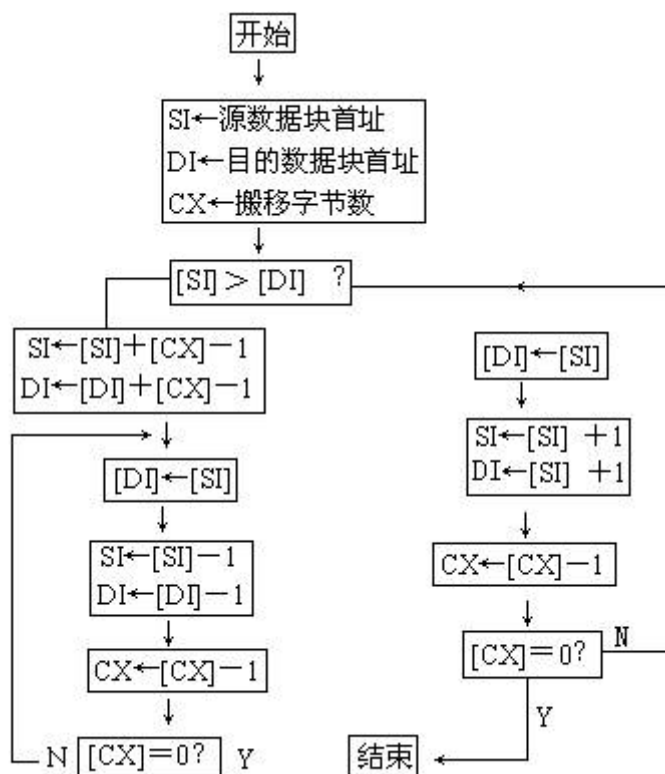




对于两个数据块分离的情况，如图（a），数据的传送从数据块的首址开始，或者从数据块的末址开始均可。但对于有部分重叠的情况，则要加以分析，否则重叠部分会因“搬移”而遭破坏。

可以得出如下结论：当源数据块首址大于目的块首址时，从数据块首地址开始传送数据。当源数据块首址小于目的块首址时，从数据块末址开始传送数据。

#### 四、实验程序框图



## 五、实验步骤

联机模式：

(1)在源数据块 4000H~4FFFH 中首址、末址几个单元，填入几个标志性字节，按下 MON 键，返回 P 状态。

(2)在 PC 机和实验系统联机状态下，运行该实验程序，可用鼠标左键单击菜单栏“文件”或工具栏“打开图标”，弹出“打开文件”的对话框，然后打开 598K8ASM 文件夹，点击 S8.ASM 文件，单击“确定”即可装入源文件，再单击工具栏中编译装载，即可完成源文件自动编译、装载目标代码功能，再单击“调试”中“连续运行”或工具图标运行，即开始运行程序。

(3)复位 RST 键，查看目的数据块 4100H~4FFFH 数据是否和源数据块 4000H~4FFFH 单元相一致。

脱机模式：

(1)在 P. 态下，按 SCAL 键，然后在源数据块 4000H~4FFFH 中首址、末址几个单元，填入几个标志性字节，按下 MON 键，返回 P 状态。

(2)在 P. 态下，输入 2EF0，按 EXEC 键。

(3)复位 RST 键，查看目的数据块 4100H~4FFFH 数据是否和源数据块 4000H~4FFFH 单元相一致。

## 六、实验程序清单

```
CODE          SEGMENT          ;S8.ASM
ASSUME        CS:CODE          ;MOV RAM:4000-40FFH TO 4100-41FFH
ORG 2EF0H
START:        MOV CX, 0100H
              MOV SI, 4000H
              MOV DI, 4100H      ;
              CMP SI, DI
              JA  FADR           ;SI>DI JMP
              ADD SI, CX
              ADD DI, CX
              DEC SI
              DEC DI
```

```
CON1:    MOV AL, [SI]
          MOV [DI], AL
          DEC SI
          DEC DI
          DEC CX
          JNE CON1
          JMP $
FADR:    MOV AL, [SI]
          INC SI
          INC DI
          DEC CX
          JNE FADR
          JMP $
CODE ENDS
END START
```

## 实验九 多分支程序

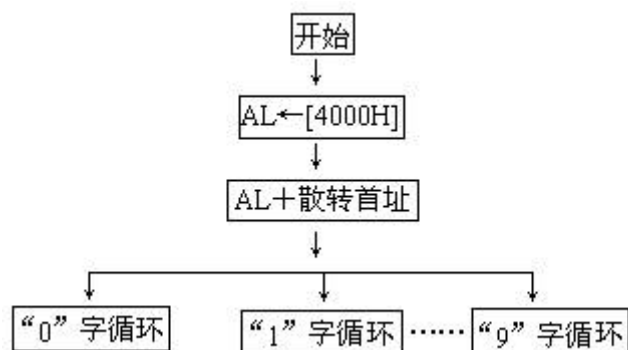
### 一、实验目的

掌握程序散转的方法，实现程序的多分支转移。

### 二、实验内容

编制程序，根据送入的数据转移运行。

### 三、实验程序框图



## 四、实验步骤

联机模式：

(1)在 4000H 单元写入 00, 01, ……09 中任一个数, 按下 MON 键, 返回 P 状态。

(2)在 PC 机和实验系统联机状态下, 运行该实验程序, 可用鼠标左键单击菜单栏“文件”或工具栏“打开图标”, 弹出“打开文件”的对话框, 然后打开 598K8ASM 文件夹, 点击 S9.ASM 文件, 单击“确定”即可装入源文件, 再单击工具栏中编译装载, 即可完成源文件自动编译、装载目标代码功能, 再单击“调试”中“连续运行”或工具图标运行, 即开始运行程序。

(3)数码管应根据 4000H 单元中内容作不同的循环显示。

脱机模式：

1、在 P. 态下, 按 SCAL 键。

2、在 4000H 单元分别写入 00, 01, ……09, 在 P. 态下, 输入 2F40, 按 EXEC 键, 数码管应根据 4000H 单元中内容作不同的循环显示。

## 五、实验程序清单

```
CODE          SEGMENT          ;S9.ASM
ASSUME        CS:CODE          ;this is a program of 9 way jmp
                ORG 2F40H        ;[4000H]=0, 1, 2, 3... 9
START:        JMP START0        ;DISP:0, 1, 2, 3...9
ADDR          DW DP0, DP1, DP2, DP3, DP4, DP5, DP6, DP7, DP8, DP9
START0:        MOV SI, 4000H
                MOV AL, [SI]
                SUB AH, AH
                SHL AL, 1
                MOV BX, OFFSET ADDR
                ADD BX, Ax
                JMP [BX]
DP0:           mov bl, 0c0h        ;DISP "0"
                jmp disp
DP1:           mov bl, 0f9h        ;DISP "1"
                jmp disp
```

```

DP2:      mov bl,0a4h          ;DISP "2"
          jmp disp
DP3:      mov bl,0b0h          ;DISP "3"
          jmp disp
DP4:      mov bl,99h           ;DISP "4"
          jmp disp
DP5:      mov bl,92h           ;DISP "5"
          jmp disp
DP6:      mov bl,82h           ;DISP "6"
          jmp disp
DP7:      mov bl,0f8h          ;DISP "7"
          jmp disp
DP8:      mov bl,80h           ;DISP "8"
          jmp disp
DP9:      mov bl,90h           ;DISP "9"
          jmp disp
;-----
disp:      mov ah,0DFh
disp0:     mov dx,0ff21h
          mov al,bl
          OUT DX,AL
          mov dx,0ff20h
          mov al,ah
          OUT DX,AL
          CALL DLY
          ror ah,01h
          jmp disp0
DLY:      mov cx,0001h
dly1:     push cx
          mov cx,0ffffh
displ:    loop displ
          pop cx
          loop dly1
          ret
CODE ENDS
END START

```

## 硬件实验

### 实验一 A/D 转换实验

#### 一、实验目的

了解模/数转换基本原理，掌握 ADC0809 的使用方法。

#### 二、实验内容

利用实验系统上电位器提供的可调电压作为 0809 模拟信号的输入，编制程序，将模拟量转换为数字量，通过数码管显示出来。

#### 三、实验接线图

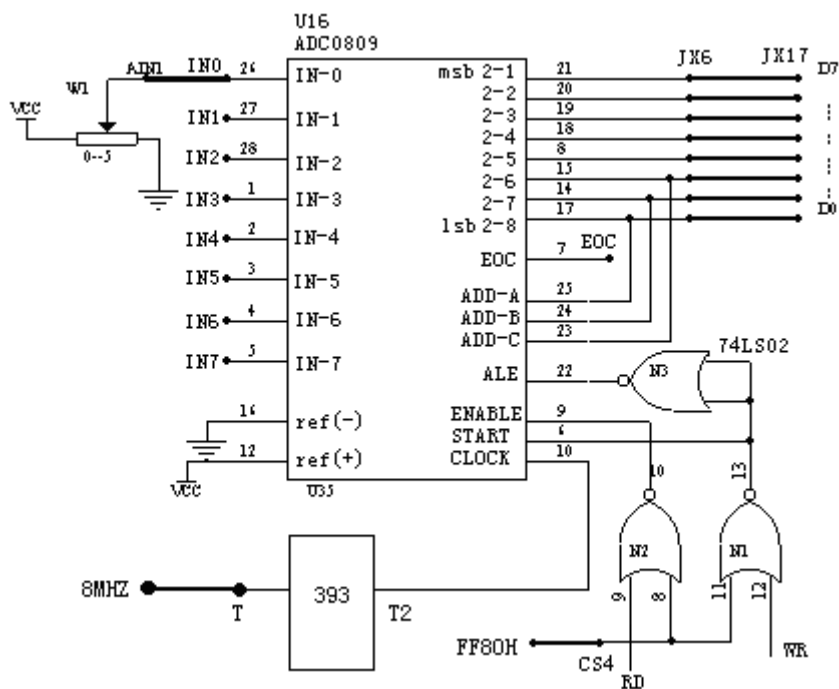


图 6-1

#### 四、编程指南

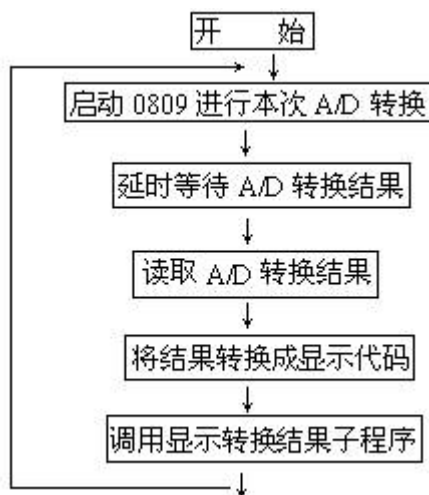
(1)ADC0809 的 START 端为 A/D 转换启动信号,ALE 端为通道选择地址的锁存信号，实验电路中将其相连，以便同时锁存通道地址并开始 A/D 采样转换，其输入控制信号为 CS 和 WR，故启动 A/D 转换只须如下两条指令：

MOV DX, ADPORT ; ADC0809 端口地址  
 OUT DX, AL ; 发 CS 和 WR 信号并送通道地址  
 (2)用延时方式等待 A/D 转换结果, 使用下述指令读取 A/D 转换结果。

MOV DX, ADPORT ; ADC0809 端口地址  
 IN AL, DX

(3)循环不断采样 A/D 转换的结果, 边采样边显示 A/D 转换后的数字量。

## 五、实验程序框图



## 六、实验步骤

①将 0809 CS4 插孔连到译码输出 FF80H 插孔、连 JX6→JX17。

②将通道 0 模拟量输入端 IN0 连电位器 W1 的中心插头 AOUT1 (0—5V) 插孔, 8MHZ→T。

③运行实验程序

如在 PC 机和实验系统联机状态下, 运行该实验程序, 可用鼠标左键单击菜单栏“文件”或工具栏“打开图标”, 弹出“打开文件”的对话框, 然后打开 598K8ASM 文件夹, 点击 H0809.ASM 文件, 单击“确定”即可装入 A/D 转换实验源文件, 再单击工具栏中编译装载, 即可完成源文件自动编译、装载目标代码功能, 再单击“调试”中“R 运行”或工具图标运行, 即开始运行程序。如果在没有 PC 机的情况下, 则在 8086K 系统显示监控提示符“P.”时, 用户段地址固定为 0000 (省略输入), 按 SCAL 键, 输入起始偏移地址 1000, 按 EXEC 键, 在系统上显示“0809 XX”。“XX”表示输入模拟量转换后的数字量。

④调节电位器 W1，显示器上会不断显示新的转换结果。

模拟量和数字量对应关系的典型值为：

0V→00H    +2.5V→80H +5V→FFH

⑤按 RST 键退出。

以上两种方式运行结果完全一致，因此，在下面的实验中，不管按何种方式均能运行实验程序。

## 七、实验程序清单

```
CODE          SEGMENT                ;H0809. ASM
ASSUME        CS:CODE
              ADPORT EQU 0FF80h
              PA      EQU 0FF20H      ;字位口
              PB      EQU 0FF21H      ;字形口
              PC      EQU 0FF22H      ;键入口
              ORG 1000H
START:        JMP START0
BUF           DB  ?, ?, ?, ?, ?, ?
data1:
              db 0c0h, 0f9h, 0a4h, 0b0h, 99h, 92h, 82h, 0f8h, 80h, 90h, 88h, 83h,
              0c6h, 0a1h
              db 86h, 8eh, 0ffh, 0ch, 89h, 0deh, 0c7h, 8ch, 0f3h, 0bfh, 8FH
START0:       CALL BUF1
ADCON:        MOV AX, 00
              MOV DX, ADPORT
              OUT DX, AL
              MOV CX, 0500H
;DELAY:       LOOP DELAY
              MOV DX, ADPORT
              IN AL, DX
              CALL CONVERS
              CALL DISP
              JMP ADCON
CONVERS:      MOV AH, AL
              AND AL, 0FH
              MOV BX, OFFSET BUF
```



```

        MOV [BX+5], AL
        MOV AL, AH
        AND AL, 0F0H
        MOV CL, 04H
        SHR AL, CL
        MOV [BX+4], AL
        RET
DISP:   MOV AL, 0FFH           ;00H
        MOV DX, PA
        OUT DX, AL
        MOV CL, 0DFH         ;20H      ;显示子程序 ,5ms
        MOV BX, OFFSET BUF
DIS1:   MOV AL, [BX]
        MOV AH, 00H
        PUSH BX
        MOV BX, OFFSET DATA1
        ADD BX, AX
        MOV AL, [BX]
        POP BX
        MOV DX, PB
        OUT DX, AL
        MOV AL, CL
        MOV DX, PA
        OUT DX, AL
        PUSH CX
DIS2:   MOV CX, 00A0H
        LOOP $
        POP CX
        CMP CL, 0FEH         ;01H
        JZ LX1
        INC BX
        ROR CL, 1           ;SHR CL, 1
        JMP DIS1
LX1:   MOV AL, 0FFH
        MOV DX, PB
        OUT DX, AL
        RET
BUF1:  MOV BUF, 00H
        MOV BUF+1, 08H

```

```

MOV BUF+2, 00H
MOV BUF+3, 09H
MOV BUF+4, 00H
MOV BUF+5, 00H
RET
CODE ENDS
END START

```

## 实验二 D/A 转换实验（一）

### 一、实验目的

了解数/模转换的基本原理，掌握 DAC0832 芯片的使用方法。

### 二、实验内容

利用 0832 通过 D/A 转换交替产生方波和锯齿波。

### 三、实验接线图

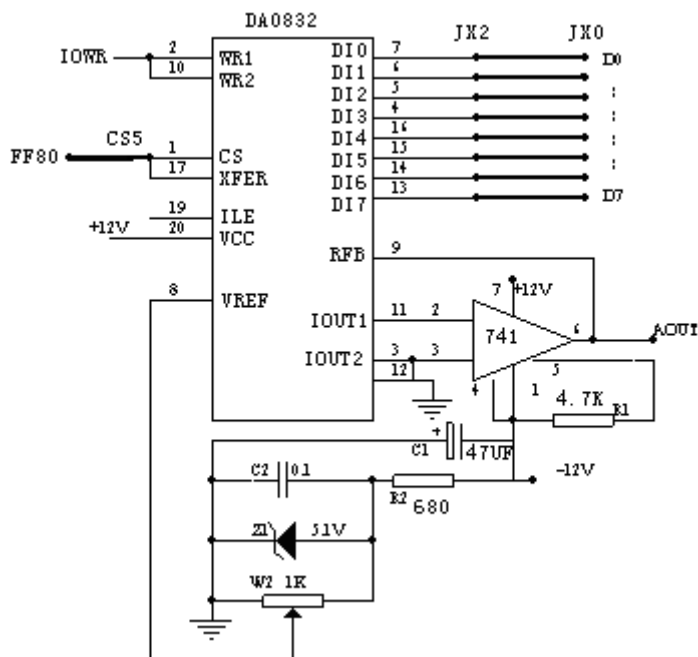


图 6-2

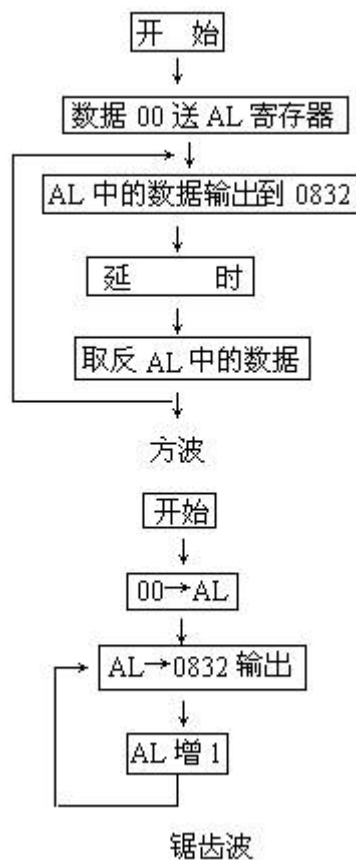
## 四、编程指南

(1) 首先须由 CS 片选信号确定量 DAC 寄存器的端口地址, 然后锁存一个数据通过 0832 输出, 典型程序如下:

```
MOV DX, DAPORT      ; 0832 口地址
MOV AL, DATA        ; 输出数据到 0832
OUT DX, AL
```

(2) 产生波形信号的周期由延时常数确定。

## 五、实验程序框图



## 六、实验步骤

1. 0832 片选信号 CS5 插孔和译码输出 FF80H 插孔相连。
2. 连 JX2→JX0。

### 3. 运行实验程序

在系统显示监控提示符“P.”时，按 SCAL 键，待系统返 P 后，输入 10D0，再按 EXEC 键，在系统上显示执行符“┐”。

4. 用示波器测量 0832 左侧 AOUT 插孔，应有方波和锯齿波交替输出。

## 七、实验程序清单

```

CODE          SEGMENT                ;H0832-1. ASM
ASSUME        CS:CODE
              DAPORT EQU 0FF80h
              ORG 10D0h
START:        MOV AL, 0FFH
              MOV CX, 0800H
DACON1:       MOV DX, DAPORT
              OUT DX, AL
              PUSH CX
              MOV CX, 0400H
              LOOP $
              POP CX
              NOT AL
              LOOP DACON1
;-----
              MOV DX, DAPORT
              MOV AL, 00H
              MOV CX, 0008H
              MOV BX, 0FFFFH
DACON2:       OUT DX, AL
              INC AL
              DEC BX
              CMP BX, 0000H
              JNZ DACON2
              LOOP DACON2
              JMP START
CODE ENDS
END  START

```

### 实验三 D/A 转换实验(二)

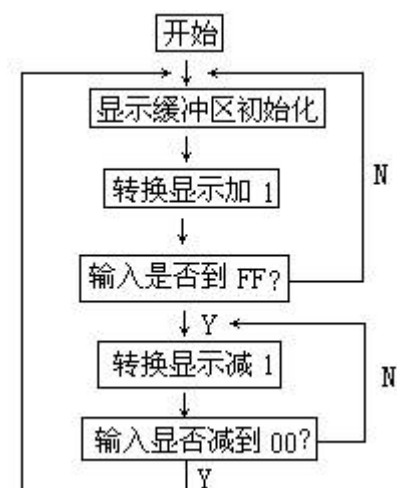
#### 一、实验目的

进一步掌握数/模转换的基本原理。

#### 二、实验内容

通过 0832D/A 转换输出一个从 0V 开始逐渐升至 5V，再从 5V 降至 0V 的可变电压输出。

#### 三、实验程序框图



#### 四、实验步骤

1. 接线同实验(二)
2. 运行实验程序

在系统显示监控提示符“P.”时，按 SCAL 键，返 P 后，输入 1110，按 EXEC 键，系统显示 0832 ---××，“××”从 00→FF，FF→00 不断加大或减小的数字量循环显示。

3. 用万用表测量 0832 左侧 AOUT 插孔，应不断加大或减小的电压。

#### 五、实验程序清单

---

```

CODE          SEGMENT                ;H0832-2. ASM 0-->5v
ASSUME        CS:CODE
DAPORT EQU 0FF80h
PA          EQU 0FF20H                ;字位口
PB          EQU 0FF21H                ;字形口
PC          EQU 0FF22H                ;键入口
ORG 1110H
START:        JMP START0
BUF           DB ?, ?, ?, ?, ?, ?
data1:
                db 0c0h, 0f9h, 0a4h, 0b0h, 99h, 92h, 82h, 0f8h, 80h, 90h, 88h, 83h,
                0c6h, 0a1h
                db 86h, 8eh, 0ffh, 0ch, 89h, 0deh, 0c7h, 8ch, 0f3h, 0bfh, 8FH
START0:       call buf1
DACON0:       MOV AL, 00H
DACON1:       MOV DX, DAPORT
                OUT DX, AL
                push ax
                call conv
                MOV CX, 0010H
DISCON:       PUSH CX
                call disp
                POP CX
                LOOP DISCON
                pop ax
                INC AL
                CMP AL, 00H
                JNZ DACON1
                MOV AL, 0FFH
DACON2:       MOV DX, DAPORT
                OUT DX, AL
                push ax
                call conv
                MOV CX, 0010H
DISCON2:      PUSH CX
                call disp
                POP CX
                LOOP DISCON2
                pop ax

```

```

        DEC AL
        CMP AL, 0FFH
        JNZ DAON2
        JMP DAON0
CONV:   MOV AH, AL
        AND AL, 0FH
        MOV BX, OFFSET BUF
        MOV [BX+5], AL
        MOV AL, AH
        AND AL, 0F0H
        MOV CL, 04H
        SHR AL, CL
        MOV [BX+4], AL
        RET
DISP:   MOV AL, 0FFH           ; 00H
        MOV DX, PA
        OUT DX, AL
        MOV CL, 0DFH           ; 20H      ; 显示子程序 , 5ms
        MOV BX, OFFSET BUF
DIS1:   MOV AL, [BX]
        MOV AH, 00H
        PUSH BX
        MOV BX, OFFSET DATA1
        ADD BX, AX
        MOV AL, [BX]
        POP BX
        MOV DX, PB
        OUT DX, AL
        MOV AL, CL
        MOV DX, PA
        OUT DX, AL
        PUSH CX
DIS2:   MOV CX, 00A0H
DELAY:  LOOP DELAY
        POP CX
        CMP CL, 0FEH           ; 01H
        JZ LX1
        INC BX
        ROR CL, 1              ; SHR CL, 1

```

```
JMP DIS1
LX1:  MOV AL, 0FFH
      MOV DX, PB
      OUT DX, AL
      RET
BUF1:  MOV BUF, 00H
      MOV BUF+1, 08H
      MOV BUF+2, 03H
      MOV BUF+3, 02H
      MOV BUF+4, 00H
      MOV BUF+5, 00H
      RET
DELY:  PUSH CX
DEL2:  PUSH CX
DEL3:  PUSH CX
      LOOP $
      POP CX
      LOOP DEL3
      POP CX
      LOOP DEL2
      POP CX
      LOOP DELY
      RET
CODE ENDS
END START
```

## 实验四 8255A 并行口实验（一）

### 一、实验目的

1. 掌握 8255A 和微机接口方法。
2. 掌握 8255A 的工作方式和编程原理。

### 二、实验内容

用 8255PA 口控制 PB 口。



### 三、实验接线图

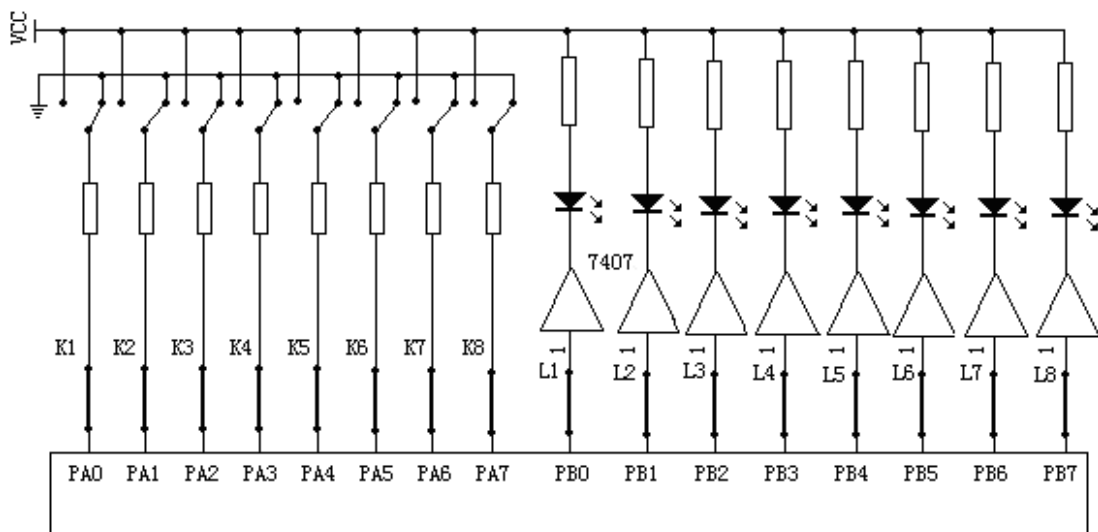


图 6-3

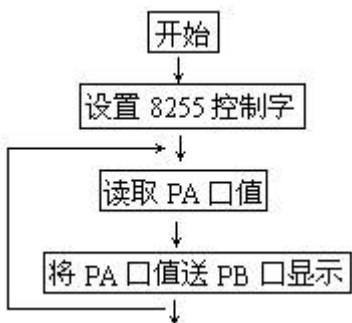
### 四、编程指南

1. 8255A 芯片简介： 8255A 可编程外围接口芯片是 INTEL 公司生产的通用并行接口芯片，它具有 A、B、C 三个并行接口，用+5V 单电源供电，能在以下三种方式下工作：

- 方式 0：基本输入/ 输出方式
- 方式 1：选通输入/ 输出方式
- 方式 2：双向选通工作方式

2. 使 8255A 端口 A 工作在方式 0 并作为输入口，读取 K1-K8 个开关量，PB 口工作在方式 0 作为输出口。

### 五、实验程序框图



## 六、实验步骤

1. 在系统显示监控提示符“P.”时，按 SCAL 键，传送 EPROM 中的实验程序到内存中。（注：必须先传送 EPROM 后，再往下操作）

2. 8255A 芯片 A 口的 PA0-PA7 依次和开关量输入 K1-K8 相连。

3. 8255A 芯片 B 口的 PB0-PB7 依次接 L1-L8。

4. 运行实验程序。

在系统显示监控提示符“P.”时，输入 11B0，按 EXEC 键，系统显示执行提示符“┐”拨动 K1-K8，LI-L8 会跟着亮灭。

## 七、实验程序清单

```
CODE          SEGMENT                                ;H8255-1. ASM
ASSUME        CS:CODE
              IOCONPT  EQU 0FF2BH
              IOBPT    EQU 0FF29H
              IOAPT     EQU 0FF28H
              ORG 11B0H
START:        MOV AL, 90H
              MOV DX, IOCONPT
              OUT DX, AL
              NOP
              NOP
              NOP
IOLED1:       MOV DX, IOAPT
              IN AL, DX
              MOV DX, IOBPT
              OUT DX, AL
              MOV CX, 0FFFFH
DELAY:        LOOP DELAY
              JMP IOLED1
CODE ENDS
END  START
```

## 实验五 8255A 并行口实验(二)

### 一、实验目的

掌握通过 8255A 并行口传输数据的方法，以控制发光二极管的亮与灭。

### 二、实验内容

用 8255 做输出口，控制十二个发光管亮灭，模拟交通灯管理。

### 三、实验接线图

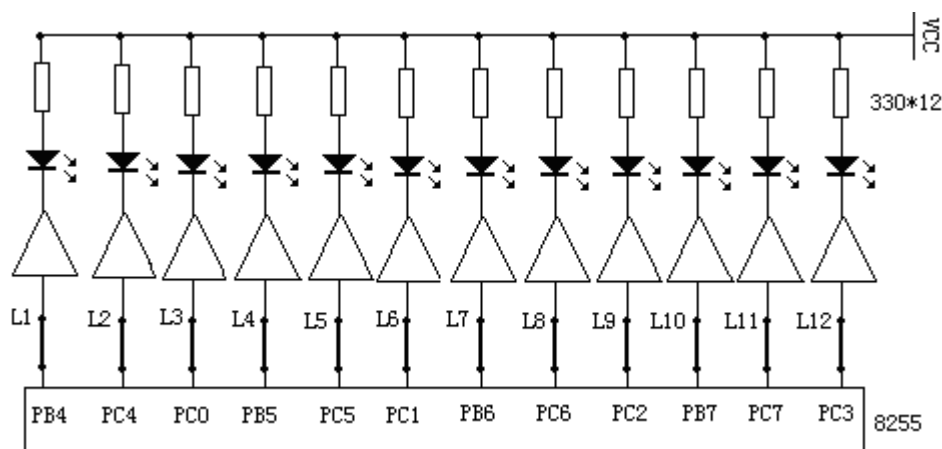


图 6-4

### 四、编程指南

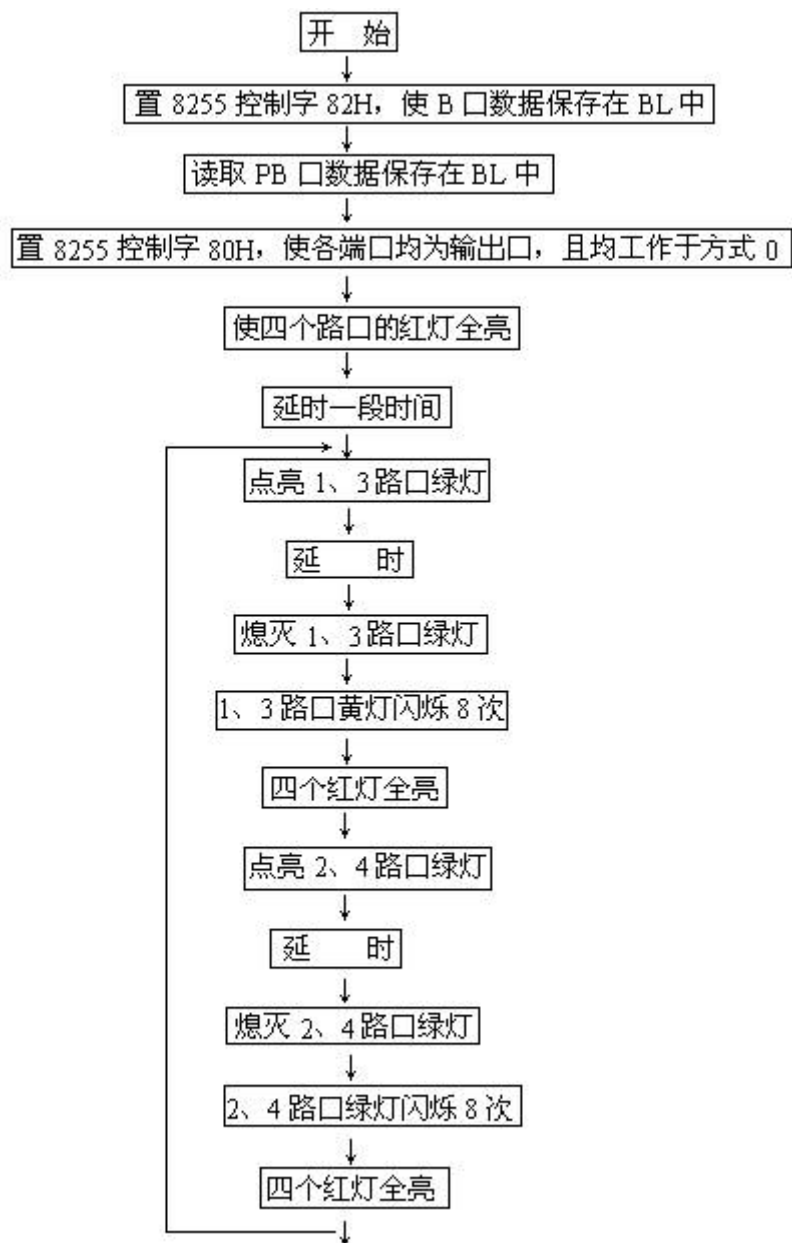
(1) 通过 8255A 控制发光二极管，PB4-PB7 对应黄灯，PC0-PC3 对应红灯，PC4-PC7 对应绿灯，以模拟交通路灯的管理。

(2) 要完成本实验，必须先了解交通路灯的亮灭规律，设有一个十字路口 1、3 为南北方向，2、4 为东西方向，初始状态为四个路口的红灯全亮，之后，1、3 路口的绿灯亮，2、4 路口的红灯亮，1、3 路口方向通车。延时一段时间后，1、3 路口的绿灯熄灭，而 1、3 路口的黄灯开始闪烁，闪烁若干次以后，1、3 路口红灯亮，而同时 2、4 路口的绿灯亮，2、4 路口方向通车，延时一段时间后，2、4 路口的绿灯熄灭，而黄灯开始闪烁，闪烁若干次以后，再切换到 1、3 路口方向，之后，重复上述过程。

(3) 程序中设定好 8255A 的工作模式及三个端口均工作在方式 0，并处于输出状态。

(4) 各发光二极管共阳极，使其点亮应使 8255A 相应端口的位清 0。

## 五、实验程序框图



## 六、实验步骤

1. 在系统显示监控提示符“P.”时，按 SCAL 键。

2. 按图 6-4 连好实验线路

8255A: PC0—>L3, PC1—>L6, PC2—>L9, PC3—>L12, PC4—>L2, PC5—>L5,  
PC6—>L8, PC7—>L11;

PB4—>L1, PB5—>L4, PB6—>L7, PB7—>L10

3. 在“P.”态时，输入 11E0 后，按 EXEC 键；系统显示执行符“┐”，同时 L1-L15 发光二极管模拟交通灯显示。

## 七、实验程序清单

```
CODE          SEGMENT                                ;H8255-2. ASM
ASSUME        CS:CODE
              IOCONPT EQU 0FF2BH
              IOAPT   EQU 0FF28H
              IOBPT   EQU 0FF29H
              IOCPT   EQU 0FF2AH
              ORG 11e0H
START:        MOV AL, 82H
              MOV DX, IOCONPT
              OUT DX, AL
              MOV DX, IOBPT
              IN AL, DX
              MOV BYTE PTR DS:[0601H], AL
              MOV DX, IOCONPT
              MOV AL, 80H
              OUT DX, AL
              MOV DX, IOBPT
              MOV AL, DS:[0601H]
              OR AL, 0F0H
              OUT DX, AL
              MOV DX, IOCPT
              MOV AL, 0F0H
              OUT DX, AL
              CALL DELAY1
```

```
IOLED0:  MOV AL, 10100101B
          MOV DX, IOCPT
          OUT DX, AL
          CALL DELAY1
          CALL DELAY1
          OR AL, 0F0H
          OUT DX, AL
          MOV CX, 8H
IOLED1:  MOV DX, IOBPT
          MOV AL, DS:[0601H]
          AND AL, 10101111B
          OUT DX, AL
          CALL DELAY2
          OR AL, 01010000B
          OUT DX, AL
          CALL DELAY2
          LOOP IOLED1
          MOV DX, IOCPT
          MOV AL, 0F0H
          OUT DX, AL
          CALL DELAY2
          MOV AL, 01011010B
          OUT DX, AL
          CALL DELAY1
          CALL DELAY1
          OR AL, 0F0H
          OUT DX, AL
          MOV CX, 8H
IOLED2:  MOV DX, IOBPT
          MOV AL, DS:[0601H]
          AND AL, 01011111B
          OUT DX, AL
          CALL DELAY2
          OR AL, 10100000B
          OUT DX, AL
          CALL DELAY2
          LOOP IOLED2
          MOV DX, IOCPT
          MOV AL, 0F0H
```

```
OUT DX, AL
CALL DELAY2
JMP IOLED0
DELAY1:  PUSH AX
        PUSH CX
        MOV CX, 0030H
DELY2:   CALL DELAY2
        LOOP DELY2
        POP CX
        POP AX
        RET
DELAY2:  PUSH CX
        MOV CX, 8000H
DELA1:   LOOP DELA1
        POP CX
        RET
CODE ENDS
END  START
```

## 实验六 定时器 / 计数器

### 一、实验目的

1. 学会 8253 芯片和微机接口的原理和方法。
2. 掌握 8253 定时器/计数器的工作方式和编程原理。

### 二、实验内容

用 8253 的 0 通道工作在方式 3，产生方波。

### 三、实验接线图

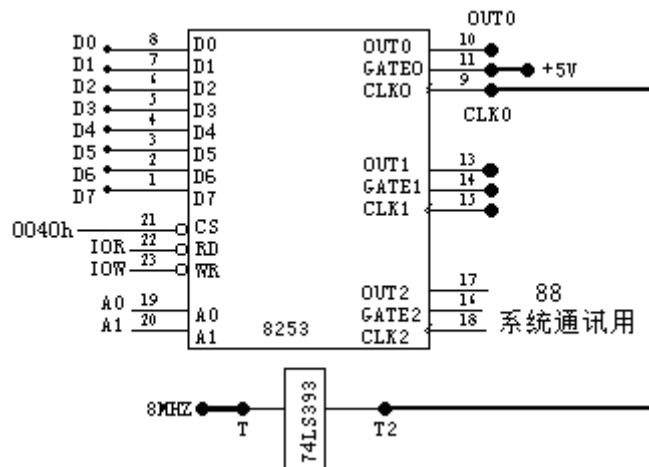


图 6-5

## 四、编程指南

### 1. 8253 芯片介绍

8253 是一种可编程定时/计数器，有三个十六位计数器，其计数频率范围为 0-2MHz，用+5V 单电源供电。

8253 的功能用途：

- |              |              |
|--------------|--------------|
| (1) 延时中断     | (5) 实时时钟     |
| (2) 可编程频率发生器 | (6) 数字单稳     |
| (3) 事件计数器    | (7) 复杂的电机控制器 |
| (4) 二进制倍频器   |              |

2, 8253 的六种工作方式：

- |                  |                    |
|------------------|--------------------|
| (1) 方式 0：计数结束中断  | (4) 方式 3：方波频率发生器   |
| (2) 方式 1：可编程频率发生 | (5) 方式 4：软件触发的选通信号 |
| (3) 方式 2：频率发生器   | (6) 方式 5：硬件触发的选通信号 |



## 五、实验程序框图



## 六、实验步骤

- 按图 6-5 连好实验线路
  - 8253 的 GATE0 接+5V。
  - 8253 的 CLK0 插孔接分频器 74LS393（左下方）的 T2 插孔，分频器的频率源为 8.0MHZ，T→8.0MHZ。
- 运行实验程序
 

在系统提示符“P.”状态下，按 SCAL 键，输入 1290，按 EXEC 键。

用示波器测量 8253 的 OUT0 输出插孔有方波产生。

## 七、实验程序清单

```

CODE          SEGMENT                                ;H8253. ASM
ASSUME        CS: CODE
              ORG 1290H
START:        JMP TCONT
              TCONTRO EQU 0043H
              TCON0   EQU 0040H
TCONT:        MOV DX, TCONTRO
              MOV AL, 36H
              OUT DX, AL
              MOV DX, TCON0
              MOV AL, 00H
              OUT DX, AL
              MOV AL, 04H
              OUT DX, AL
              MOV DX, TCONTRO
              MOV AL, 36H

```

```
OUT DX, AL
MOV DX, TCON0
MOV AL, 00H
OUT DX, AL
MOV AL, 02H
OUT DX, AL
JMP $

CODE ENDS
END START
```

实验七 8259 单级中断控制器实验

一、实验目的

- 1. 掌握 8259 中断控制器的接口方法。
- 2. 掌握 8259 中断控制器的应用编程。

二、实验内容

利用 8259 实现对外部中断的响应和处理，要求程序对每次中断进行计数，并将计数结果送数码显示。

三、实验接线图

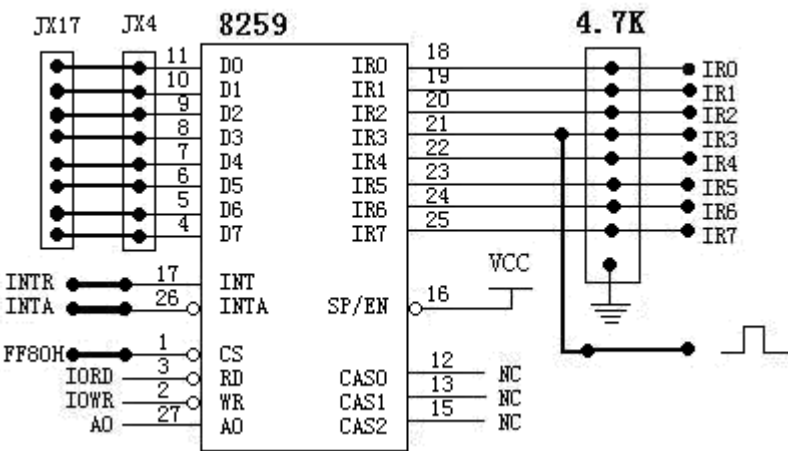



图 6—6

## 四、编程指南

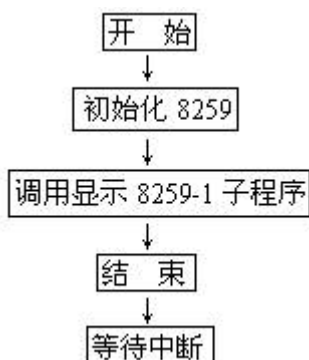
### (1) 8259 芯片介绍

中断控制器 8259A 是专为控制优先级中断而设计的芯片。它将中断源优先级排队、辨别中断源以及提供中断矢量的电路集于一片中。因此无需附加任何电路，只需对 8259A 进行编程，就可以管理 8 级中断，并选择优先模式和中断请求方式。即中断结构可以由用户编程来设定。同时，在不需要增加其它电路的情况下，通过多片 8259A 的级联，能构成多达 64 级的矢量中断系统。

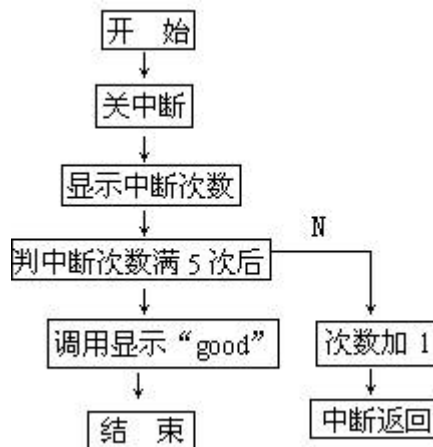
中断序号	0	1	2	3	4	5	6	7
变量地址	20H	24H	28H	2CH	30H	34H	38H	3CH
	23H	27H	2BH	2FH	33H	37H	3BH	3FH

(2) 本实验中使用 3 号中断源 IR3，“”插孔和 IR3 相连，中断方式为边沿触发方式，每拨二次 AN 开关产生一次中断，满 5 次中断，显示“8259—good”。如果中断源电平信号不符合规定要求，则自动转到 7 号中断，显示“Err”。

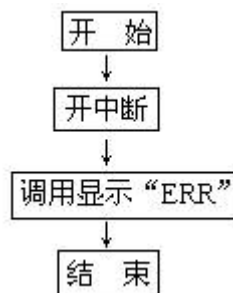
## 五、实验程序框图



IR3 中断服务程序：





IR7 中断服务程序:



## 六、实验步骤

1、按图 6—6 连好实验线路图。

(1)8259 的 INT 连 8088 的 INTR; (2)8259 的 INTA 连 8088 的 INTA; (3)“”

插孔和 8259 的 3 号中断 IR3 插孔相连, “”端初始为低电平; (4)8259 的 CS 端接 FF80H 孔; (5)连 JX4→JX17。

2、运行实验程序, 在系统处于命令提示符“P.”状态下, 按 SCAL 键, 输入 12D0, 按 EXEC 键, 系统显示 8259—1。

3、拨动 AN 开关按钮, 按满 10 次显示 good。

## 七、实验程序清单

```

CODE          SEGMENT                                ;H8259. ASM
ASSUME        CS: CODE
INTPORT1 EQU 0FF80H
  
```

```

        INTPORT2 EQU 0FF81H
        INTQ3     EQU INTREEUP3
        INTQ7     EQU INTREEUP7
        PA        EQU 0FF20H           ;字位口
        PB        EQU 0FF21H           ;字形口
        PC        EQU 0FF22H           ;键入口
        ORG 12D0H
START:   JMP START0
        BUF       DB ?, ?, ?, ?, ?, ?
        intcnt    db ?
        data1:
                db 0c0h, 0f9h, 0a4h, 0b0h, 99h, 92h, 82h, 0f8h, 80h, 90h, 88h, 83h,
                0c6h, 0a1h
                db 86h, 8eh, 0ffh, 0ch, 89h, 0deh, 0c7h, 8ch, 0f3h, 0bfh, 8FH
START0:  CLD
        CALL BUF1
        CALL WRINTVER                   ;WRITE INTRRUPT
        MOV AL, 13H
        MOV DX, INTPORT1
        OUT DX, AL
        MOV AL, 08H
        MOV DX, INTPORT2
        OUT DX, AL
        MOV AL, 09H
        OUT DX, AL
        MOV AL, 0F7H
        OUT DX, AL
        MOV intcnt, 01H                 ;TIME=1
        STI
WATING:  CALL DISP                      ;DISP 8259-1
        JMP WATING
WRINTVER: MOV AX, 0H
        MOV ES, AX
        MOV DI, 002CH
        LEA AX, INTQ3
        STOSW
        MOV AX, 0000h
        STOSW
        MOV DI, 003CH

```

```
        LEA AX, INTQ7
        STOSW
        MOV AX, 0000h
        STOSW
        RET
INTREEUP3: CLI
        push ax
        push bx
        push cx
        push dx
        MOV AL, INTCNT
        CALL CONVERS
        MOV BX, OFFSET BUF           ;077BH
        MOV AL, 10H
        MOV CX, 05H
INTRE0:  MOV [BX], AL
        INC BX
        LOOP INTRE0
        MOV AL, 20H
        MOV DX, INTPORT1
        OUT DX, AL
        ADD INTCNT, 01H
        CMP INTCNT, 06H
        JNA INTRE2
        CALL BUF2                   ;DISP:good
INTRE1:  CALL DISP
        JMP INTRE1
CONVERS: AND AL, 0FH
        MOV BX, offset buf         ;077AH
        MOV [BX+5], AL
        RET
INTRE2:  MOV AL, 20H
        MOV DX, INTPORT1
        OUT DX, AL
        pop dx
        pop cx
        pop bx
        pop ax
        STI
```

```

                IRET
INTREEUP7: CLI
                MOV AL, 20H
                MOV DX, INTPORT1
                OUT DX, AL
                call buf3                                ;disp:err
INTRE3:         CALL DISP
                JMP INTRE3
DISP:           MOV AL, OFFH                            ;00H
                MOV DX, PA
                OUT DX, AL
                MOV CL, 0DFH                            ;20H  ;显示子程序 ,5ms
                MOV BX, OFFSET BUF
DIS1:           MOV AL, [BX]
                MOV AH, 00H
                PUSH BX
                MOV BX, OFFSET DATA1
                ADD BX, AX
                MOV AL, [BX]
                POP BX
                MOV DX, PB
                OUT DX, AL
                MOV AL, CL
                MOV DX, PA
                OUT DX, AL
                PUSH CX
DIS2:           MOV CX, 00A0H
                LOOP $
                POP CX
                CMP CL, 0FEH                            ;01H
                JZ LX1
                INC BX
                ROR CL, 1                                ;SHR CL, 1
                JMP DIS1
LX1:            MOV AL, OFFH
                MOV DX, PB
                OUT DX, AL
                RET
BUF1:           MOV BUF, 08H

```

```
        MOV BUF+1, 02H
        MOV BUF+2, 05H
        MOV BUF+3, 09H
        MOV BUF+4, 17H
        MOV BUF+5, 01H
        RET
BUF2:    MOV BUF, 09H
        MOV BUF+1, 00H
        MOV BUF+2, 00H
        MOV BUF+3, 0dH
        MOV BUF+4, 10H
        MOV BUF+5, 10H
        RET
BUF3:    MOV BUF, 0eH
        MOV BUF+1, 18H
        MOV BUF+2, 18H
        MOV BUF+3, 10H
        MOV BUF+4, 10H
        MOV BUF+5, 10H
        RET
CODE ENDS
END START
```

## 实验八 串行接口和应用（一）串行发送

### 一、实验目的

- 1、了解串行通讯的一般原理和 8251A 的工作原理。
- 2、初步了解 RS-232 串行接口标准及 TTL 电路的连接方法。
- 3、掌握 8251A 编程方法。

### 二、实验内容

用二台 8086K 通过 8251 进行双机通讯，一台作为发送，另一台作为接收，发送方读入按键值，并发送给接收方，接收方收到数据后在数码管上显示。



三、实验接线图

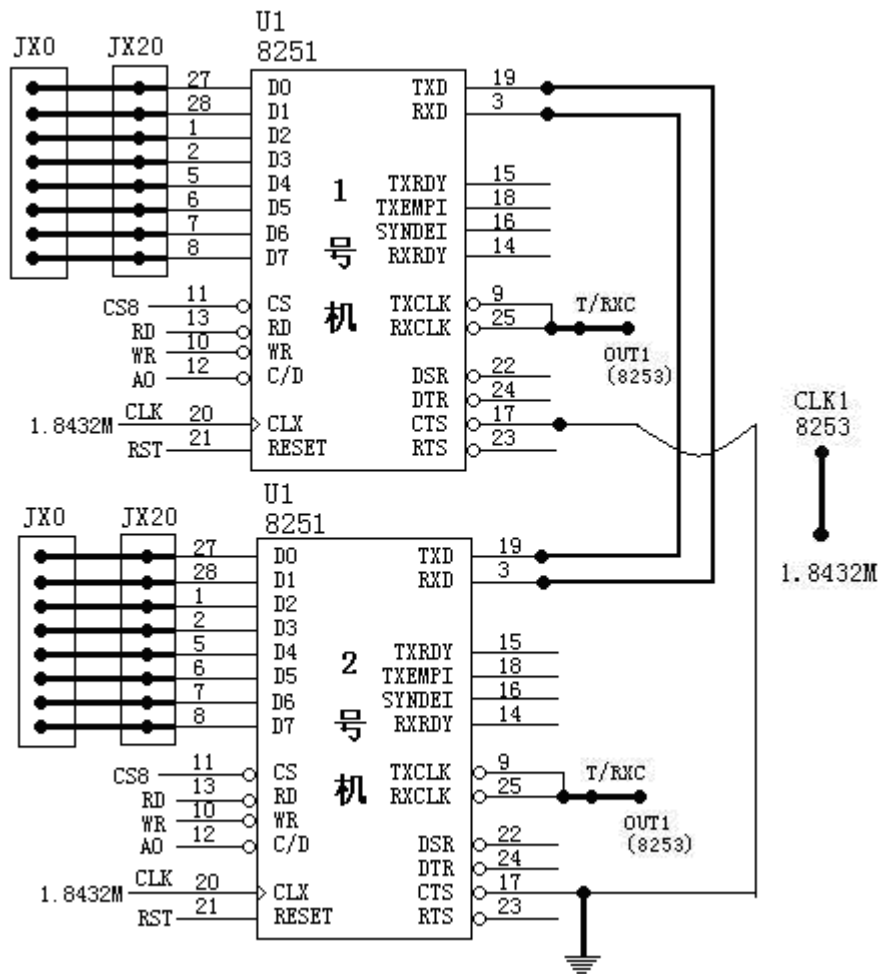


图 6-7

电路图说明：

- (1)TxC 和 RxC 分别为 8251A 的发送时钟和接收时钟，由片外 8253 的 OUT1 提供。
- (2)8251 片选信号 8251 CS 接译码输出 3F8H-3FFH（系统上已连接）。
- (3)CTS 端必须接低电平（已接好），8251A 才可向外发送数据，其它回答信号 RTS、DTR、DSR 不用。

(4)RxRDY、TxRDY 为允许接收和允许发送信号，由于在本系统中使用查询方式进行通讯，这两个信号线不用，中断方式时，这两个信号分别作发送/接收时的中断申请信号。

#### 四、编程指南

(1) 8251 状态口地址：03F9H，8251 数据口地址：03F8H；

(2) 8253 命令口地址：43H，8253 计数器#1 口地址：41H；

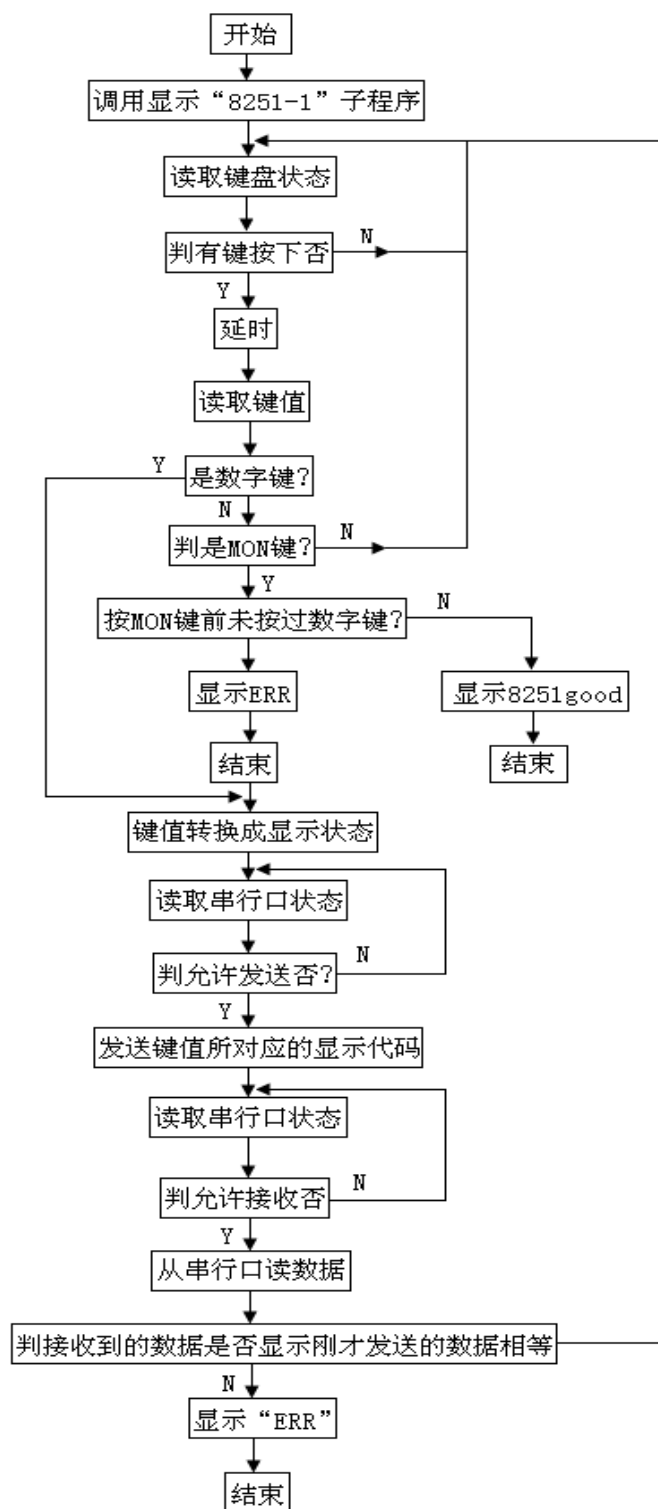
(3)8255 命令口地址:0FF23H,键扫口/字位口:0FF20H,键入口 PC:0FF22H,字形口 PB: 0FF21H;

(4) 通讯约定：异步方式，字符 8 位，一个起始位，一个停止位，波特率因子为 16，波特率为 9600；

(5) 计算 T/RXC，收发时钟  $f_c$ ， $f_c=16*9600=153.6K$ ；

(6) 8253 分频系数： $1843.2K / 153.6K=12$ 。

#### 五、实验程序框图



## 六、实验步骤

1、准备 2 台 8086K 实验机，确定 1 号机为发送，2 号机为接收。

2、连接：CLK1→1.8432M，GATE1→+5V，OUT1→T/RXC，JX0→JX20。1 号机和 2 号机的 RXD、TXD 交叉相连，且两机共地。

3、先运行 2 号机，在 2 号机处于命令提示符“P.”状态下，按 SCAL 键，再输入 13F0，按 EXEC 键，即进入等待接收状态，显示器显示 8251-2。

注意：串行接收实验的有关内容详见实验九。

4、再运行 1 号机，在 1 号机处于命令提示符“P.”状态下，按 SCAL 键，再输入 1510，按 EXEC 键，即可进入串行发送状态，显示器显示 8251-1。

5、在 1 号机键盘上按动数字键，在 2 号机的显示器上应显示对应数字键值。当 1 号机上按“MON”键时，1 号机即显示 good，此时可按 RST 键退出。

## 七、实验程序清单

```
CODE          SEGMENT                ;H8251T.ASM
ASSUME        CS:CODE
                SECOPORT EQU 03F9H
                SEDAPORT EQU 03F8H
                PA        EQU 0FF20H        ;字位口
                PB        EQU 0FF21H        ;字形口
                PC        EQU 0FF22H        ;键入口
                ORG 13F0H
START:         JMP START0
                BUF        DB ?, ?, ?, ?, ?, ?
                KZ         DB ?
                ltime      db ?
                lkey       db ?
data1:         db 0c0h, 0f9h, 0a4h, 0b0h, 99h, 92h, 82h, 0f8h, 80h, 90h, 88h, 83h,
                0c6h, 0a1h
                db 86h, 8eh, 0ffh, 0ch, 89h, 0deh, 0c7h, 8ch, 0f3h, 0bfh, 8FH
START0:        call for8251
                CALL BUF1                ;DISP:8251-1
redikey:       call dispkey
```

```

        cmp KZ, 10h
        jc wattxd
        jmp funckey
WATTXD:  MOV DX, SECOPORT
        IN AL, DX
        TEST AL, 01H
        JZ WATTXD
        MOV AL, KZ
        MOV DX, SEDAPORT
        OUT DX, AL
WATRXD:  MOV DX, SECOPORT
        IN AL, DX
        TEST AL, 02H
        JZ WATRXD
        MOV DX, SEDAPORT
        IN AL, DX
        CMP KZ, AL
        JZ seri2
        CALL BUF3                      ;DISP:err
sererr:  CALL DISP
        JMP sererr
seri2:   mov cx, 0018h
ser3:    push cx
        call disp
        pop cx
        loop ser3
        jmp redikey
funckey: CMP KZ, 1FH
        JNZ REDIKEY
        call buf2                      ;good
monit:   CALL DISP
        JMP monit
dispkey: call disp
        call key
        mov ah, al                      ;newkey
        mov bl, ltime                   ;ltime
        mov bh, lkey                    ;lkey
        mov al, 01h
        mov dx, PA                      ;Off21h

```

```

        out dx, al
        cmp ah, bh
        mov bh, ah           ;bh=new key
        mov ah, bl          ;al=time
        jz disk4
        mov bl, 88h
        mov ah, 88h
disk4:   dec ah
        cmp ah, 82h
        jz disk6
        cmp ah, 0eh
        jz disk6
        cmp ah, 00h
        jz disk5
        mov ah, 20h
        dec bl
        jmp disk7
disk5:   mov ah, 0fh
disk6:   mov bl, ah
        mov ah, bh
disk7:   mov ltime, bl
        mov lkey, bh
        mov KZ, bh
        mov al, ah
        ret
key:     mov al, 0ffh
        mov dx, PB          ;0ff22h
        out dx, al
        mov bl, 00h
        mov ah, 0feh
        mov cx, 08h
key1:    mov al, ah
        mov dx, PA          ;0ff21h
        out dx, al
        rol al, 01h
        mov ah, al
        nop
        nop
        nop

```

```

        nop
        nop
        nop
        mov dx, PC                      ;0ff23h
        in al, dx
        not al
        nop
        nop
        and al, 0fh
        jnz key2
        inc bl
        loop key1
        jmp nkey
key2:    test al, 01h
        je key3
        mov al, 00h
        jmp key6
key3:    test al, 02h
        je key4
        mov al, 08h
        jmp key6
key4:    test al, 04h
        je key5
        mov al, 10h
        jmp key6
key5:    test al, 08h
        je nkey
        mov al, 18h
key6:    add al, bl
        cmp al, 10h
        jnc fkey
        mov bl, al
        mov bh, 0h
        mov si, offset data2
        mov al, [bx+si]
        ret
nkey:    mov al, 20h
fkey:    ret
data2:   db 07h, 04h, 08h, 05h, 09h, 06h, 0ah, 0bh

```

---

 DB 01h, 00h, 02h, 0fh, 03h, 0eh, 0ch, 0dh

```

for8251:  call t8253
          mov al, 65h
          out dx, al
          mov dx, 03f9h
          mov al, 25h
          out dx, al
          mov dx, 03f9h
          mov al, 65h
          out dx, al
          mov dx, 03f9h
          mov al, 4eh
          out dx, al
          mov dx, 03f9h
          mov al, 25h
          out dx, al
          ret

```

```

T8253:    MOV DX, 43H                ;9600
          MOV AL, 76H
          out dx, al
          MOV DX, 41H
          MOV AL, 0CH
          out dx, al
          MOV DX, 41H
          MOV AL, 00H
          out dx, al
          mov dx, 03F9H
          mov dx, 03f9h
          RET

```

```

DISP:     MOV AL, 0FFH              ;00H
          MOV DX, PA
          OUT DX, AL
          MOV CL, 0DFH              ;20H   ;显示子程序 , 5ms
          MOV BX, OFFSET BUF

```

```

DIS1:     MOV AL, [BX]
          MOV AH, 00H
          PUSH BX
          MOV BX, OFFSET DATA1

```



```

        ADD BX, AX
        MOV AL, [BX]
        POP BX
        MOV DX, PB
        OUT DX, AL
        MOV AL, CL
        MOV DX, PA
        OUT DX, AL
        PUSH CX
DIS2:   MOV CX, 0180H
        LOOP $
        POP CX
        CMP CL, 0FEH                ;01H
        JZ LX1
        INC BX
        ROR CL, 1                    ;SHR CL, 1
        JMP DIS1
LX1:    MOV AL, 0FFH
        MOV DX, PB
        OUT DX, AL
        RET
BUF1:   MOV BUF, 08H
        MOV BUF+1, 02H
        MOV BUF+2, 05H
        MOV BUF+3, 01H
        MOV BUF+4, 17H
        MOV BUF+5, 01H
        RET
BUF2:   MOV BUF, 09H
        MOV BUF+1, 00H
        MOV BUF+2, 00H
        MOV BUF+3, 0dH
        MOV BUF+4, 10H
        MOV BUF+5, 10H
        RET
BUF3:   MOV BUF, 0eH
        MOV BUF+1, 18H
        MOV BUF+2, 18H
        MOV BUF+3, 10H

```

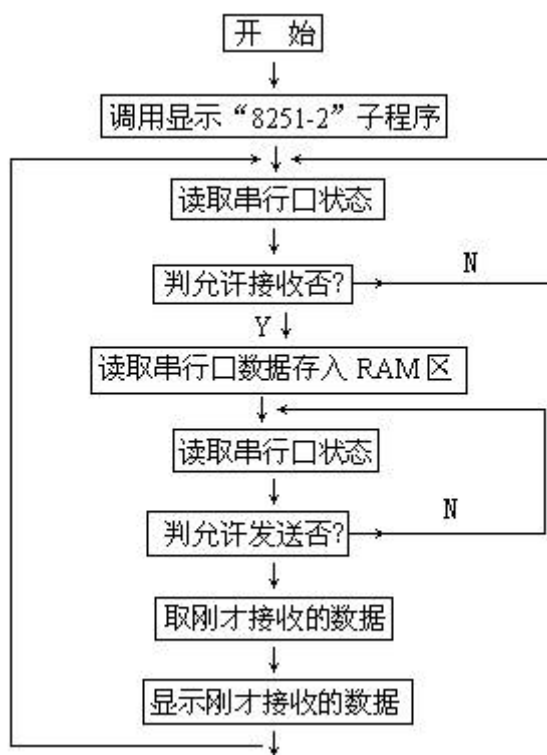
```

MOV BUF+4, 10H
MOV BUF+5, 10H
RET
CODE ENDS
END START

```

## 实验九 串行接口和应用（二）串行接收

### 一、实验程序框图



### 二、实验程序清单

```

CODE          SEGMENT                                ;H8251R. ASM
ASSUME        CS:CODE
SECOPORT EQU 03F9H
SEDAPORT EQU 03F8H
PA          EQU 0FF20H                                ;字位口
PB          EQU 0FF21H                                ;字形口

```

---

```

PC      EQU 0FF22H          ;键入口
ORG 1510H
START:  JMP START0
        BUF      DB  ?, ?, ?, ?, ?, ?
        ZP       DW  ?

data1:

        db0c0h, 0f9h, 0a4h, 0b0h, 99h, 92h, 82h, 0f8h, 80h, 90h, 88h, 83h,
        0c6h, 0a1h
        db 86h, 8eh, 0ffh, 0ch, 89h, 0deh, 0c7h, 8ch, 0f3h, 0bfh, 8FH
START0:  call for8251
        MOV ZP, OFFSET BUF
        CALL BUF1
watrx:   call disp
        MOV DX, SECOPORT
        IN AL, DX
        TEST AL, 02H
        JZ watrx
        MOV DX, SEDAPORT
        IN AL, DX
        PUSH AX
wattx:   MOV DX, SECOPORT
        IN AL, DX
        TEST AL, 01H
        JZ WATTXD
        MOV DX, SEDAPORT
        POP AX
        OUT DX, AL
        MOV BX, ZP
        mov [BX], AL
        CMP BX, OFFSET BUF+5
        jz serial1
        INC BX
        MOV ZP, BX
        jmp watrx
serial1: mov ZP, OFFSET BUF
        jmp watrx
DISP:   MOV AL, 0FFH          ;00H
        MOV DX, PA
        OUT DX, AL

```

```

MOV CL, 0DFH ;20H ;显示子程序 ,5ms
MOV BX, OFFSET BUF
DIS1: MOV AL, [BX]
MOV AH, 00H
PUSH BX
MOV BX, OFFSET DATA1
ADD BX, AX
MOV AL, [BX]
POP BX
MOV DX, PB
OUT DX, AL
MOV AL, CL
MOV DX, PA
OUT DX, AL
PUSH CX
DIS2: MOV CX, 00A0H
LOOP $
POP CX
CMP CL, 0FEH ;01H
JZ LX1
INC BX
ROR CL, 1 ;SHR CL, 1
JMP DIS1
LX1: MOV AL, 0FFH
MOV DX, PB
OUT DX, AL
RET
for8251: call t8253
mov al, 65h
out dx, al
mov dx, 03f9h
mov al, 25h
out dx, al
mov dx, 03f9h
mov al, 65h
out dx, al
mov dx, 03f9h
mov al, 4eh
out dx, al

```

```
        mov dx, 03f9h
        mov al, 25h
        out dx, al
        ret
T8253:  MOV DX, 43H
        MOV AL, 76H
        out dx, al
        MOV DX, 41H
        MOV AL, 0CH
        out dx, al
        MOV DX, 41H
        MOV AL, 00H
        out dx, al
        mov dx, 03F9H
        mov dx, 03f9h
        RET
BUF1:   MOV BUF, 08H
        MOV BUF+1, 02H
        MOV BUF+2, 05H
        MOV BUF+3, 01H
        MOV BUF+4, 17H
        MOV BUF+5, 02H
        RET
code ends
END  START
```

## 实验十 小直流电机调速实验

### 一、实验目的

- 1、掌握直流电机的驱动原理
- 2、了解直流电机调速的方法

### 二、实验内容

- 1、用 0832D/A 转换电路的输出经放大后驱动直流电机。

2、编制程序，改变 0832 输出经放大后的电压信号来控制电机转速。

### 三、实验接线图

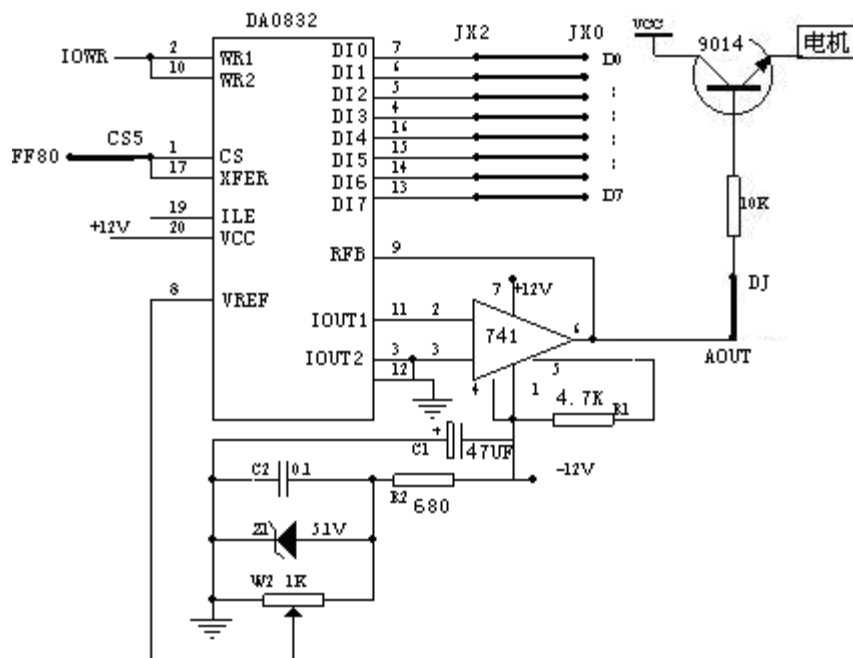


图 6-8

#### 四、实验步骤

- 1、0832 的片选信号 CS5 连到译码输出 FF80H。
- 2、连 JX2→JX0。
- 3、0832 输出 AOUT 连 DJ 插孔。
- 4、在“P.”状态下，按 SCAL 键，再输入 1590 后，按 EXEC 键。
- 5、观察直流电机的转速。

## 五、实验程序清单

```
CODE          SEGMENT          ;DJ. ASM 0-->5v
ASSUME        CS: CODE
DAPORT        EQU 0FF80h
```

---

```

        PA      EQU 0FF20H      ;字位口
        PB      EQU 0FF21H      ;字形口
        PC      EQU 0FF22H      ;键入口
        ORG 1590H
START:   JMP START0
BUF      DB  ?, ?, ?, ?, ?, ?
data1:
        db 0c0h, 0f9h, 0a4h, 0b0h, 99h, 92h, 82h, 0f8h, 80h, 90h, 88h, 83h, 0c6h, 0
        alh
        db 86h, 8eh, 0ffh, 0ch, 89h, 0deh, 0c7h, 8ch, 0f3h, 0bfh, 8FH, 0F1H
START0:  call buf1
DACON0:  MOV AL, 00H
DACON1:  MOV DX, DAPORT
        OUT DX, AL
        push ax
        call conv
        MOV CX, 0010H
DISCON:  PUSH CX
        call disp
        POP CX
        LOOP DISCON
        pop ax
        INC AL
        CMP AL, 00H
        JNZ DACON1
        MOV AL, 0FFH
DACON2:  MOV DX, DAPORT
        OUT DX, AL
        push ax
        call conv
        MOV CX, 0010H
DISCON2: PUSH CX
        call disp
        POP CX
        LOOP DISCON2
        pop ax
        DEC AL
        CMP AL, 0FFH
        JNZ DACON2

```

```

                                JMP DAON0
CONV:    MOV AH, AL
                                AND AL, 0FH
                                MOV BX, OFFSET BUF
                                MOV [BX+5], AL
                                MOV AL, AH
                                AND AL, 0F0H
                                MOV CL, 04H
                                SHR AL, CL
                                MOV [BX+4], AL
                                RET
DISP:    MOV AL, 0FFH                                ;00H
                                MOV DX, PA
                                OUT DX, AL
                                MOV CL, 0DFH                                ;20H;显示子程序 , 5ms
                                MOV BX, OFFSET BUF
DIS1:    MOV AL, [BX]
                                MOV AH, 00H
                                PUSH BX
                                MOV BX, OFFSET DATA1
                                ADD BX, AX
                                MOV AL, [BX]
                                POP BX
                                MOV DX, PB
                                OUT DX, AL
                                MOV AL, CL
                                MOV DX, PA
                                OUT DX, AL
                                PUSH CX
DIS2:    MOV CX, 00A0H
DELAY:   LOOP DELAY
                                POP CX
                                CMP CL, 0FEH                                ;01H
                                JZ LX1
                                INC BX
                                ROR CL, 1                                ;SHR CL, 1
                                JMP DIS1
LX1:    MOV AL, 0FFH
                                MOV DX, PB

```



```
OUT DX, AL
RET
BUF1:  MOV BUF, 0DH
        MOV BUF+1, 19H
        MOV BUF+2, 17H
        MOV BUF+3, 17H
        MOV BUF+4, 00H
        MOV BUF+5, 00H
        RET
DELY:   PUSH CX
DEL2:   PUSH CX
DEL3:   PUSH CX
        LOOP $
        POP CX
        LOOP DEL3
        POP CX
        LOOP DEL2
        POP CX
        LOOP DELY
        RET
CODE ENDS
END START
```

## 实验十一 步进电机控制

### 一、实验目的

- 1、了解步进电机控制的基本原理。
- 2、掌握步进电机转动编程方法。

### 二、实验内容

1、用 8255 的 PA0~PA3 输出脉冲信号，驱动步进电机转动，通过键盘设定来控制步进电机正转、反转、停止。

2、实验预备知识，步进电机驱动原理是通过对它每相线圈中的电流的顺序切换来使电机作步进式旋转。驱动电路由脉冲信号来控制，所以调节脉冲信号的频率便可改变步进电机的转速，用微电脑控制步进电机最适合。

### 三、实验接线图

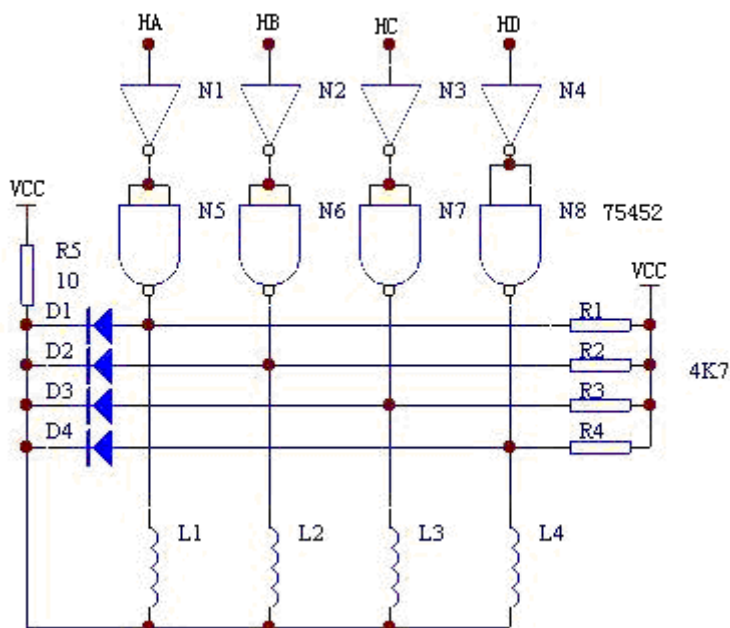


图 6-9

### 四、实验步骤

- 1、在系统处于命令提示符“P.”态下，按 SCAL 键。
- 2、按图 6-9 连好实验线路图，8255 的 PA0~PA3 依次连到 HA-HD 插孔。
- 3、运行实验程序。

在系统处于命令提示符“P.”态下，输入 1630，按 EXEC 键，显示 BJ---，按“1”键正转；按“2”键反转；按“3”停止。

- 4、观察步进电机转动情况。

## 五、实验程序清单

```

CODE                SEGMENT                ;BJ. ASM
ASSUME               CS: CODE
                    IOCONPT EQU 0FF2BH
                    IOBPT   EQU 0FF29H
                    IOAPT   EQU 0FF28H
                    PA      EQU 0FF20H      ;字位口
                    PB      EQU 0FF21H      ;字形口
                    PC      EQU 0FF22H      ;键入口
                    ORG 1630H
START:              JMP START0
                    BUF      DB ?, ?, ?, ?, ?, ?
                    KZ       DB ?
                    ltime    db ?
                    lkey     db ?

data1:
db0c0h, 0f9h, 0a4h, 0b0h, 99h, 92h, 82h, 0f8h, 80h, 90h, 88h, 83h, 0c6h, 0a1h
db 86h, 8eh, 0ffh, 0ch, 89h, 0deh, 0c7h, 8ch, 0f3h, 0bfh, 8fh, 0f1h
START0:             CALL BUF1
                    MOV AL, 88H
                    MOV DX, IOCONPT
                    OUT DX, AL

redikey:            call dispkey
                    cmp KZ, 01h
                    JZ ZZ
                    cmp KZ, 02h
                    JZ FZ
                    cmp KZ, 03h
                    JZ STX
                    JMP REDIKEY

STX:                JMP ST
ZZ:                 CALL BUFZ
ZZ1:                MOV DX, IOAPT
                    MOV AL, 03H
                    MOV DX, IOAPT
                    OUT DX, AL
                    CALL DELPZ
                    MOV AL, 06H

```

```
MOV DX, IOAPT
OUT DX, AL
CALL DELPZ
MOV AL, 0CH
MOV DX, IOAPT
OUT DX, AL
CALL DELPZ
MOV AL, 09H
MOV DX, IOAPT
OUT DX, AL
CALL DELPZ
MOV AL, 03H
MOV DX, IOAPT
OUT DX, AL
CALL DELPZ
MOV AL, 06H
MOV DX, IOAPT
OUT DX, AL
CALL DELPZ
MOV AL, 0CH
MOV DX, IOAPT
OUT DX, AL
CALL DELPZ
MOV AL, 09H
MOV DX, IOAPT
OUT DX, AL
CALL DELPZ
JMP ZZ1
```

;-----

```
FZ:      CALL BUFF
FZ1:     MOV DX, IOAPT
          MOV AL, 0CH
          OUT DX, AL
          CALL DELPF
          MOV AL, 06H
          MOV DX, IOAPT
          OUT DX, AL
          CALL DELPF
          MOV DX, IOAPT
```

```
MOV AL, 03H
OUT DX, AL
CALL DELPF
MOV AL, 09H
MOV DX, IOAPT
OUT DX, AL
CALL DELPF
MOV AL, 0CH
MOV DX, IOAPT
OUT DX, AL
CALL DELPF
MOV AL, 06H
MOV DX, IOAPT
OUT DX, AL
CALL DELPF
MOV AL, 03H
MOV DX, IOAPT
OUT DX, AL
CALL DELPF
MOV AL, 09H
MOV DX, IOAPT
OUT DX, AL
CALL DELPF
JMP FZ1
;-----
ST:      CALL BUFS
        MOV DX, IOAPT
        MOV AL, 00H
        OUT DX, AL
ST1:     call dispkey
        cmp KZ, 01h
        JZ ZZMON
        cmp KZ, 02h
        JZ FZMON
        JMP ST1
delpZ:   mov cx, 02h
con1:    push cx
        CALL dispkey
        pop cx
```

```

                                cmp KZ, 02h
                                JZ FZMON
                                cmp KZ, 03h
                                JZ STMON
                                loop con1
                                RET
delpF:                          mov cx, 02h
con2:                            push cx
                                CALL dispkey
                                pop cx
                                cmp KZ, 01h
                                JZ ZZMON
                                cmp KZ, 03h
                                JZ STMON
                                loop con2
                                RET
ZZMON:                          POP CX
                                JMP ZZ
FZMON:                          POP CX
                                JMP FZ
STMON:                          POP CX
                                JMP ST
;-----
dispkey:                         call disp
                                call key
                                mov ah, al                                ;newkey
                                mov bl, ltime                            ;ltime
                                mov bh, lkey                              ;lkey
                                mov al, 01h
                                mov dx, PA                                ;0ff21h
                                out dx, al
                                cmp ah, bh
                                mov bh, ah                                ;bh=new key
                                mov ah, bl                                ;al=time
                                jz disk4
                                mov bl, 88h
                                mov ah, 88h
disk4:                           dec ah
                                cmp ah, 82h

```

```
        jz disk6
        cmp ah, 0eh
        jz disk6
        cmp ah, 00h
        jz disk5
        mov ah, 20h
        dec bl
        jmp disk7
disk5:   mov ah, 0fh
disk6:   mov bl, ah
        mov ah, bh
disk7:   mov ltime, bl
        mov lkey, bh
        mov KZ, bh
        mov al, ah
        ret
key:     mov al, 0ffh
        mov dx, PB
        out dx, al
        mov bl, 00h
        mov ah, 0feh
        mov cx, 08h
key1:    mov al, ah
        mov dx, PA
        out dx, al
        rol al, 01h
        mov ah, al
        nop
        nop
        nop
        nop
        nop
        nop
        mov dx, PC
        in al, dx
        not al
        nop
        nop
        and al, 0fh
```

```

                                jnz key2
                                inc bl
                                loop key1
                                jmp nkey
key2:                          test al, 01h
                                je key3
                                mov al, 00h
                                jmp key6
key3:                          test al, 02h
                                je key4
                                mov al, 08h
                                jmp key6
key4:                          test al, 04h
                                je key5
                                mov al, 10h
                                jmp key6
key5:                          test al, 08h
                                je nkey
                                mov al, 18h
key6:                          add al, bl
                                cmp al, 10h
                                jnc fkey
                                mov bl, al
                                mov bh, 0h
                                mov si, offset data2
                                mov al, [bx+si]
                                ret
nkey:                          mov al, 20h
fkey:                          ret
data2:                         db 07h, 04h, 08h, 05h, 09h, 06h, 0ah, 0bh
                                DB 01h, 00h, 02h, 0fh, 03h, 0eh, 0ch, 0dh
DISP:                          MOV AL, 0FFH                ;00H
                                MOV DX, PA
                                OUT DX, AL
                                MOV CL, 0DFH                ;20H;显示子程序 , 5ms
                                MOV BX, OFFSET BUF
DIS1:                          MOV AL, [BX]
                                MOV AH, 00H
                                PUSH BX

```



```
MOV BX, OFFSET DATA1
ADD BX, AX
MOV AL, [BX]
POP BX
MOV DX, PB
OUT DX, AL
MOV AL, CL
MOV DX, PA
OUT DX, AL
PUSH CX
DIS2: MOV CX, 0a0H
      LOOP $
      POP CX
      CMP CL, 0FEH
      JZ LX1
      INC BX
      ROR CL, 1
      JMP DIS1
LX1:  MOV AL, 0FFH
      MOV DX, PB
      OUT DX, AL
      RET
BUF1: MOV BUF, 0BH
      MOV BUF+1, 019H
      MOV BUF+2, 17H
      MOV BUF+3, 17H
      MOV BUF+4, 17H
      MOV BUF+5, 17H
      RET
BUFZ: MOV BUF, 0BH
      MOV BUF+1, 19H
      MOV BUF+2, 17H
      MOV BUF+3, 17H
      MOV BUF+4, 17H
      MOV BUF+5, 0FH
      RET
BUFF: MOV BUF, 0BH
      MOV BUF+1, 19H
      MOV BUF+2, 17H
```

```
MOV BUF+3, 17H
MOV BUF+4, 17H
MOV BUF+5, 18H
RET
BUFS: MOV BUF, 0BH
      MOV BUF+1, 19H
      MOV BUF+2, 17H
      MOV BUF+3, 17H
      MOV BUF+4, 17H
      MOV BUF+5, 05H
      RET
CODE ENDS
END START
```

## 实验十二 继电器控制

### 一、实验目的

掌握用继电器控制的基本方法和编程。

### 二、实验内容

1、利用 8255 的 PA0 输出高/低电平，控制继电器的开合，以实现对外部装置的控制。

2、实验预备知识：现代自动化控制设备中都存在一个电子与电气电路的互相联结问题。一方面要使电子电路的控制信号能够控制电气电路的执行元件(电动机、电磁铁、电灯等),一方面又要为电子电路的电气提供良好的电隔离,以保护电子电路和人身的安全。电子继电器便能完成这一桥梁作用。

### 三、实验接线图

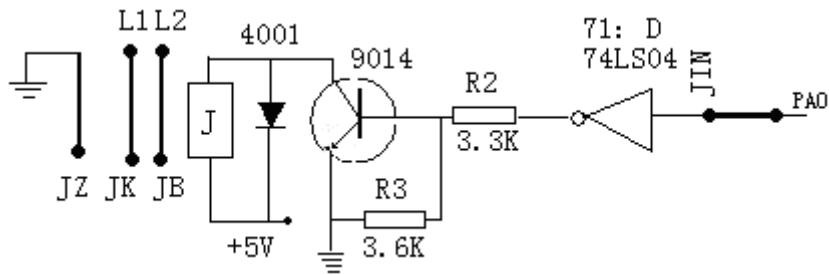


图 6-10

#### 四、实验步骤

- 1、在系统处于命令提示符“P.”下，按 SCAL 键。
- 2、连 8255 的 PA0 到 JIN 插孔；继电器常开触点 JK 接 L2，常闭触点 JB 接 L1，中心抽头 JZ 接地。
- 3、运行实验程序，在系统处于命令提示符“P.”下，输入 1810，按 EXEC 键。
- 4、继电器应循环吸合。

#### 五、实验程序清单

```

CODE          SEGMENT          ;JDQ. ASM
ASSUME        CS:CODE
              IOCONPT EQU 0FF2BH
              IOBPT   EQU 0FF29H
              IOAPT    EQU 0FF28H
              ORG 1810H
START:        MOV AL, 88H
              MOV DX, IOCONPT
              OUT DX, AL
              NOP
              NOP
              NOP
IOLED1:       MOV DX, IOAPT
IODE2:        MOV AL, 01H
              OUT DX, AL
              CALL DELAY

```

```

        MOV AL, 00H
        OUT DX, AL
        CALL DELAY
        JMP IODE2
DELAY:   MOV CX, 0FFFFH
DELY:    LOOP DELY
        RET
CODE ENDS
END START

```

### 实验十三 存贮器读写实验

#### 一、实验目的

- 1、熟悉静态 RAM 的使用方法, 掌握 8088 微机系统扩展 RAM 的方法。
- 2、掌握静态 RAM 读写数据编程方法。

#### 二、实验内容

对指定地址区间的 RAM (4000H~4FFH) 先进行写数据 55AAH, 然后将其内容读出再写到 5000H~5FFH 中。

#### 三、实验接线图 (系统中已连接好)

#### 四、实验步骤

- 1、在系统提示符 “P” 状态下, 按 SCAL 键。
- 2、输入执行地址 1850 后, 按 EXEC 键。
- 3、稍后按 RST 键退出, 用存贮器读方法检查 4000H~43FFH 中的内容和 5000~53FFH 中的内容应都是 55AA。

#### 五、实验程序清单

```

CODE      SEGMENT                ;RAM. ASM
ASSUME    CS:CODE
PA        EQU 0FF20H              ;字位口
PB        EQU 0FF21H              ;字形口
PC        EQU 0FF22H              ;键入口

```

```

        ORG 1850h
START:  JMP START0
        BUF      DB ?, ?, ?, ?, ?, ?

data1:
        db0c0h, 0f9h, 0a4h, 0b0h, 99h, 92h, 82h, 0f8h, 80h, 90h, 88h, 83h,
        0c6h, 0a1h
        db 86h, 8eh, 0ffh, 0ch, 89h, 0deh, 0c7h, 8ch, 0f3h, 0bfh, 8FH
START0: MOV AX, 0H
        MOV DS, AX
        MOV BX, 4000H
        MOV AX, 55AAH
        MOV CX, 0200H
RAMW1:  MOV DS: [BX], AX
        ADD BX, 0002H
        LOOP RAMW1
        MOV AX, 4000H
        MOV SI, AX
        MOV AX, 5000H
        MOV DI, AX
        MOV CX, 0400H
        CLD
        REP MOVSB
        call buf1
        mov cx, 0ffh
con1:   push cx
        call disp
        pop cx
        loop con1
        call buf2
con2:   call disp
        jmp con2
DISP:  MOV AL, 0FFH                ;00H
        MOV DX, PA
        OUT DX, AL
        MOV CL, 0DFH              ;20H ;显示子程序 , 5ms
        MOV BX, OFFSET BUF
DIS1:  MOV AL, [BX]
        MOV AH, 00H
        PUSH BX

```

```

MOV BX, OFFSET DATA1
ADD BX, AX
MOV AL, [BX]
POP BX
MOV DX, PB
OUT DX, AL
MOV AL, CL
MOV DX, PA
OUT DX, AL
PUSH CX
DIS2: MOV CX, 00A0H
      LOOP $
      POP CX
      CMP CL, 0FEH          ; 01H
      JZ LX1
      INC BX
      ROR CL, 1             ; SHR CL, 1
      JMP DIS1
LX1:  MOV AL, 0FFH
      MOV DX, PB
      OUT DX, AL
      RET
BUF1: MOV BUF, 06H
      MOV BUF+1, 02H
      MOV BUF+2, 02H
      MOV BUF+3, 05H
      MOV BUF+4, 06H
      MOV BUF+5, 17H
      RET
BUF2: MOV BUF, 17H
      MOV BUF+1, 17H
      MOV BUF+2, 09H
      MOV BUF+3, 00H
      MOV BUF+4, 00H
      MOV BUF+5, 0dH
      RET
CODE ENDS
END START

```

## 实验十四 电子琴实验

### 一、实验目的

- 1、进一步掌握定时器编程和应用。
- 2、了解微机利用定时器产生音乐的基本方法。

### 二、实验内容

利用实验仪键盘，使数字键 1、2、3、4、5、6、7 作为电子琴键，按下数字键发出相应的音调，PBO 作为定时器门控信号，OUT1 发出音频信号，驱动喇叭。

### 三、实验接线图

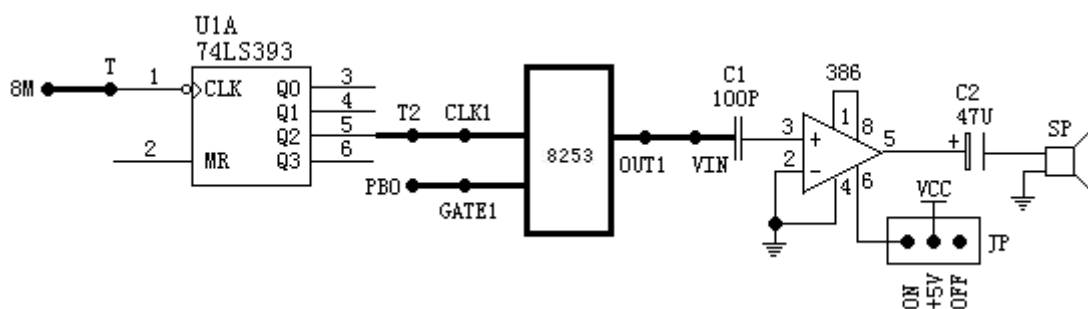


图 6-11

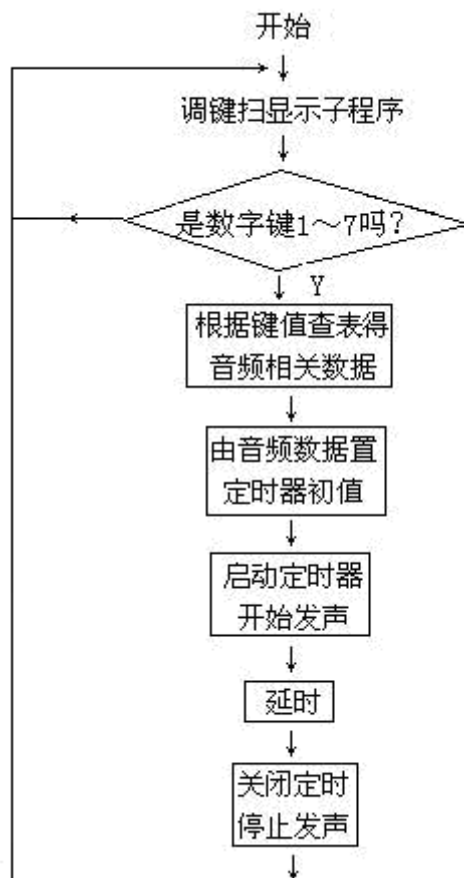
### 四、编程指南

1、根据音阶频率表，利用定时器可以产生相应频率的脉冲信号，不同频率的脉冲信号经驱动电路放大后，就会发生不同的音调。

2、对于每个按键的音调发音时间由软件延时控制，如键一直按下，就会连续发音各音阶标称频率值：

音阶	1	2	3	4	5	6	7
频率 (HZ)	440.00	493.88	554.37	587.33	659.26	739.99	830.61

## 五、实验程序框图



## 六、实验步骤

(1)8253 的 CLK1 接分频输出端 T2。

(2)GATE1 接 PB0。

(3)OUT1 接 VIN。

(4)8MHZ→T。

(5)JP 向下短接。

(6)在 P. 态, 按 SCAL 键, 输入 18F0, 按 EXEC 键, 显 P., 按 1、2……7 键, 依次发出 1、2……7 音调声。



## 七、实验程序清单

```

CODE          SEGMENT                                ;DZQ. ASM
ASSUME        CS:CODE, ds:code
CONTROL equ 43h
COUNT0 equ 40h
COUNT1 equ 41h
COUNT2 equ 42h
IOCONPT EQU 0FF2BH
IOBPT EQU 0FF29H
IOAPT EQU 0FF28H
PA EQU 0FF20H ;字位口
PB EQU 0FF21H ;字形口
PC EQU 0FF22H ;键入口
ORG 18F0H
START:        JMP START0
BUF DB ?, ?, ?, ?, ?, ?
KZ DB ?
ltime db ?
lkey db ?
ZP DW ?

data1:
db 0c0h, 0f9h, 0a4h, 0b0h, 99h, 92h, 82h, 0f8h, 80h, 90h, 88h, 83h, 0c6h, 0a1h
db 86h, 8eh, 0ffh, 0ch, 89h, 0deh, 0c7h, 8ch, 0f3h, 0bfh, 8FH, 0F1H
data3:        dw 2273, 2024, 1805, 1704
               dw 1517, 1353, 1205, 1136
START0:       mov ax, cs
               mov ds, ax
               CALL BUF1
               MOV AL, 88H
               MOV DX, IOCONPT
               OUT DX, AL
               mov dx, IOBPT
               mov al, 00
               out dx, al
               mov zp, offset buf
redikey:      call dispkey
               cmp KZ, 09h
               JNC redikey

```

```

        cmp kz, 01h
        jc redikey
        mov bx, zp
        mov al, kz
        mov [bx], al
        cmp bx, offset buf+5
        jz zpl
        inc bx
        mov zp, bx
        jmp outtone
zpl:    mov zp, offset buf
outtone: mov dx, IOBPT
        mov al, 0ffh
        out dx, al
        mov al, kz
        mov ah, 00h
        dec ax
        shl ax, 1
        mov bx, offset data3
        add bx, ax
        mov ax, [bx]
        call t8253
        mov cx, 20h
con1:   push cx
        call disp
        pop cx
        loop con1
        mov dx, IOBPT
        mov al, 00
        out dx, al
        jmp redikey
t8253:  push ax
        mov al, 76h
        mov dx, CONTROL
        out dx, al
        pop ax
        mov dx, COUNT1
        out dx, al
        mov al, ah

```

; 计数器 1, 16 位二进制, 方式 3。

```

        out    dx, al
        ret

;-----
dispkey:  call disp
          call key
          mov  ah,al                ;newkey
          mov  bl,ltime            ;ltime
          mov  bh,lkey             ;lkey
          mov  al,01h
          mov  dx,PA               ;0ff21h
          out  dx,al
          cmp  ah,bh
          mov  bh,ah               ;bh=new key
          mov  ah,bl               ;al=time
          jz   disk4
          mov  bl,88h
          mov  ah,88h
disk4:    dec  ah
          cmp  ah,82h
          jz   disk6
          cmp  ah,0eh
          jz   disk6
          cmp  ah,00h
          jz   disk5
          mov  ah,20h
          dec  bl
          jmp  disk7
disk5:    mov  ah,0fh
disk6:    mov  bl,ah
          mov  ah,bh
disk7:    mov  ltime,bl
          mov  lkey,bh
          mov  KZ,bh
          mov  al,ah
          ret
key:      mov  al,0ffh
          mov  dx,PB
          out  dx,al
          mov  bl,00h

```

```
        mov ah, 0feh
        mov cx, 08h
key1:    mov al, ah
        mov dx, PA
        out dx, al
        rol al, 01h
        mov ah, al
        nop
        nop
        nop
        nop
        nop
        nop
        mov dx, PC
        in al, dx
        not al
        nop
        nop
        and al, 0fh
        jnz key2
        inc bl
        loop key1
        jmp nkey
key2:    test al, 01h
        je key3
        mov al, 00h
        jmp key6
key3:    test al, 02h
        je key4
        mov al, 08h
        jmp key6
key4:    test al, 04h
        je key5
        mov al, 10h
        jmp key6
key5:    test al, 08h
        je nkey
        mov al, 18h
key6:    add al, bl
```

```

        cmp al, 10h
        jnc fkey
        mov bl, al
        mov bh, 0h
        mov si, offset data2
        mov al, [bx+si]
        ret
nkey:   mov al, 20h
fkey:   ret
data2:  db 07h, 04h, 08h, 05h, 09h, 06h, 0ah, 0bh
        DB 01h, 00h, 02h, 0fh, 03h, 0eh, 0ch, 0dh
DISP:   MOV AL, 0FFH                ;00H
        MOV DX, PA
        OUT DX, AL
        MOV CL, 0DFH                ;20H ;显示子程序 , 5ms
        MOV BX, OFFSET BUF
DIS1:   MOV AL, [BX]
        MOV AH, 00H
        PUSH BX
        MOV BX, OFFSET DATA1
        ADD BX, AX
        MOV AL, [BX]
        POP BX
        MOV DX, PB
        OUT DX, AL
        MOV AL, CL
        MOV DX, PA
        OUT DX, AL
        PUSH CX
DIS2:   MOV CX, 0a0H
        LOOP $
        POP CX
        CMP CL, 0FEH
        JZ LX1
        INC BX
        ROR CL, 1
        JMP DIS1
LX1:   MOV AL, 0FFH
        MOV DX, PB

```

```
OUT DX, AL
RET
BUF1:  MOV BUF, 11H
        MOV BUF+1, 10H
        MOV BUF+2, 10H
        MOV BUF+3, 10H
        MOV BUF+4, 10H
        MOV BUF+5, 10H
        RET
CODE ENDS
END START
```

## 实验十五 简单 I/O 口扩展实验

### 一、实验目的

- 1、学习微机系统中扩展简单 I/O 口的方法。
- 2、学习数据输入输出程序的编制方法。

### 二、实验内容

利用 74LS244 作为输入口，读取开关状态，并将此状态通过 74LS273 再驱动发光二极管显示出来。

### 三、实验接线图

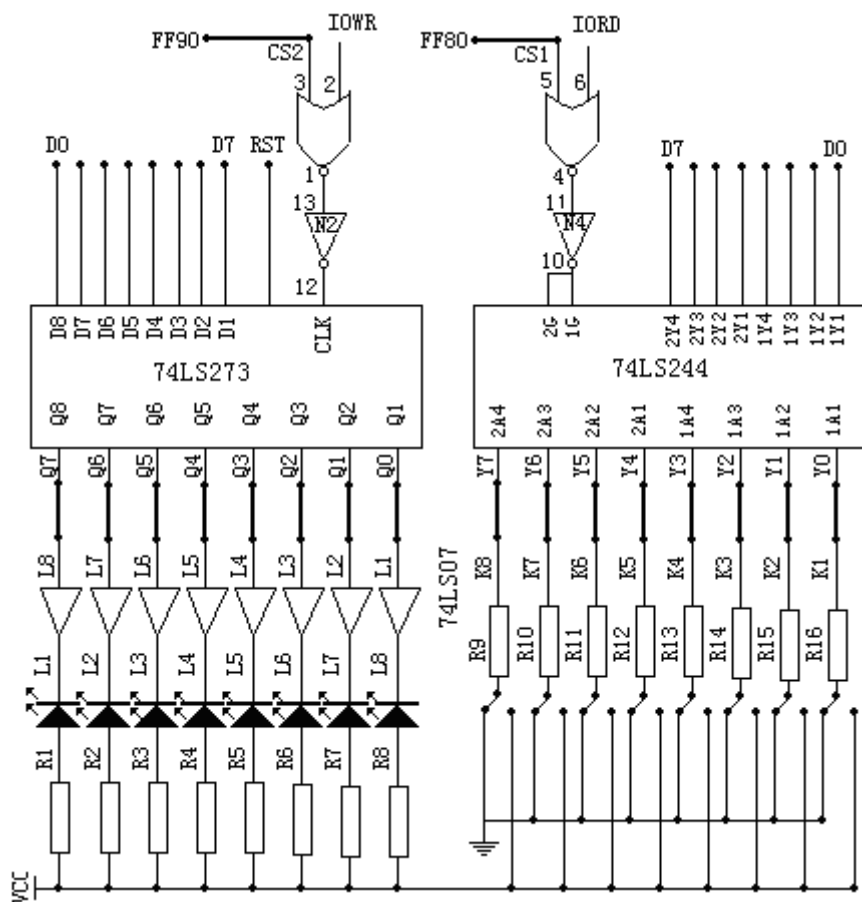
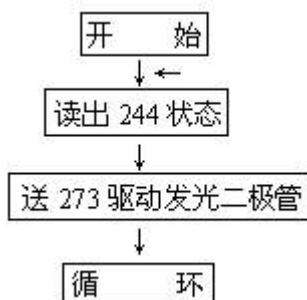


图 6-12

#### 四、实验程序框图



## 五、实验步骤

1、按图 6-12 接线，Y0~Y7 接 K1~K8，Q0~Q7 接 L1~L8，CS1 接 FF80H 孔，CS2 接 FF90H 孔。

2、运行实验程序，在系统处于命令提示符“P.”状态下，按 SCAL 键，输入 19D0，按 EXEC 键。

3、按动 K1~K8，观察 L1~L8 是否对应点亮。

## 六、实验程序清单

```
CODE          SEGMENT          ; IO. ASM
ASSUME        CS: CODE
              ORG 19D0H
START:        MOV DX, 0FF80H
              IN  AL, DX
              MOV DX, 0FF90H
              OUT DX, AL
              JMP START
CODE ENDS
END START
```

## 实验十六 8251 可编程通讯接口与 PC 机通讯

### 一、实验目的

利用实验机内的 8251 芯片，实现与 PC 机通讯。

### 二、实验内容

(1) 利用实验机内的 8253 芯片的分频作为 8251 的收发时钟频率。

(2) 利用实验机内小键盘，每按动一次任一数字键，就把该键值通过 8251 发送给 PC 机接收，并在 PC 机屏幕上显示出该键值。

### 三、实验接线图



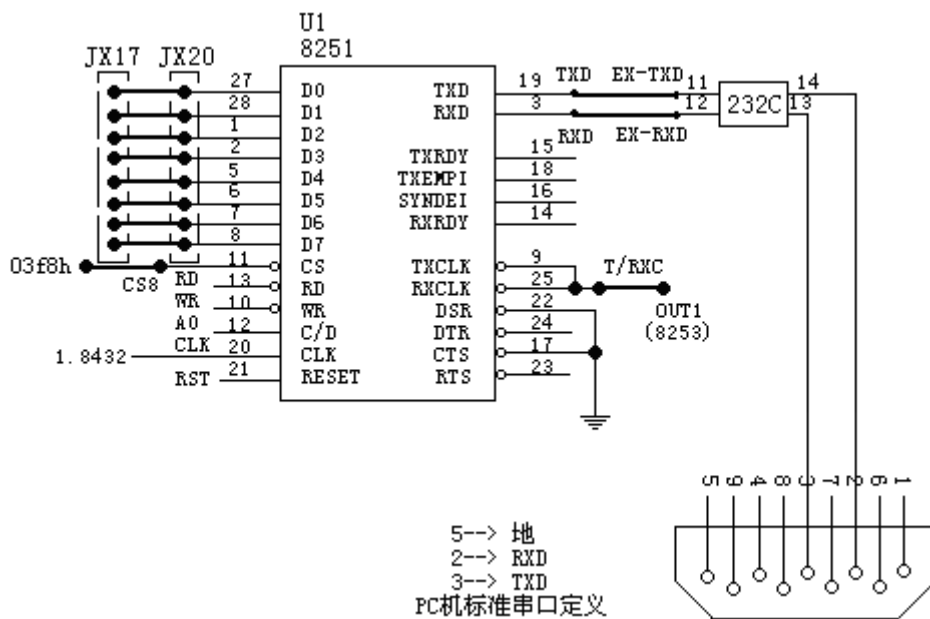


图 6-13

#### 四、编程指南

- (1) 8251 状态口地址: 03F9H, 8251 数据口地址: 03F8H;
- (2) 8253 命令口地址: 43H, 8253 计数器#1 口地址: 41H;
- (3) 8255 命令口地址: 0FF23H, 键扫口/字位口: 0FF20H, 键入口 PC: 0FF22H, 字形口 PB: 0FF21H;
- (4) 通讯约定: 异步方式, 字符 8 位, 一个起始位, 一个停止位, 波特率因子为 16, 波特率为 9600;
- (5) 计算 T/RXC, 收发时钟  $f_c$ ,  $f_c = 16 * 9600 = 153.6K$ ;
- (6) 8253 分频系数:  $1843.2K / 153.6K = 12$ 。

#### 五、实验步骤

- (1) 8251 单元: T/RXC→OUT1, TXD→EX-TXD, RXD→EX-RXD, JX20→JX17;
- (2) 8253 单元: GATE1→+5V, CLK1→1.8432MHZ;

(3) CZ11 用户通信口→PC 机串口;

(4) 在 P. 态下, 按 SCAL 键, 输入 1A00, 按 EXEC 键;

(5) 数码管显示 8251—1, 等待按键, 发送键值;

(6) 运行“串口调试助手”, 按动小键盘数字键, 在 PC 机屏幕上显示相应的数字, 按 MON 键发送结束, 按 RST 键, 返回 P. 态。

## 六、实验程序清单

```

CODE          SEGMENT                ;PC8251. ASM, 8251TXD-->PCRXD
ASSUME        CS: CODE
                SECOPORT EQU 03F9H
                SEDAPORT EQU 03F8H
                PA        EQU 0FF20H    ;字位口
                PB        EQU 0FF21H    ;字形口
                PC        EQU 0FF22H    ;键入口
                ORG 1A00H
START:         JMP START0
                BUF        DB ?, ?, ?, ?, ?, ?
                KZ         DB ?
                lcntkz     dw ?
                zp         dw ?

data1:         db0c0h, 0f9h, 0a4h, 0b0h, 99h, 92h, 82h, 0f8h, 80h, 90h, 88h, 83h,
                0c6h, 0a1h
                db 86h, 8eh, 0ffh, 0ch, 89h, 0deh, 0c7h, 8ch, 0f3h, 0bfh, 8FH
START0:        call for8251
                mov zp, offset buf
                CALL BUF1                ;DISP:8251-1
redikey:       call dispkey
                cmp KZ, 10h
                jc wattxd
                jmp funckey
WATTXD:        MOV DX, SECOPORT
                IN AL, DX
                TEST AL, 01H
                JZ WATTXD
                MOV AL, KZ

```

```

MOV DX, SEDAPORT
OUT DX, AL
mov bx, zp
mov [bx], al
cmp bx, offset buf+5
jz zp1
inc bx
mov zp, bx
jmp redikey
zp1:    mov zp, offset buf
        jmp redikey
funckey: CMP KZ, 1FH
        JNZ REDIKEY
        call buf2                ;good
monit:  CALL DISP
        JMP monit
dispkey: call disp
        call key
XD:     MOV BX, lcntkz
        MOV CX, [BX]
        MOV AH, AL
        CMP AL, CH
        JE XD1
        MOV CL, 88H
XD1:    DEC CL
        CMP CL, 82H
        JE XD3
        CMP CL, 0EH
        JE XD3
        CMP CL, 00H
        JE XD2
        MOV AL, 20H
        JMP XD3
XD2:    MOV CL, 0FH
XD3:    MOV BX, lcntkz
        MOV CH, AH
        MOV [BX], CX
        mov kz, al
        RET

```

```
key:      mov al, 0ffh
          mov dx, PB
          out dx, al
          mov bl, 00h
          mov ah, 0feh
          mov cx, 08h
key1:     mov al, ah
          mov dx, PA
          out dx, al
          rol al, 01h
          mov ah, al
          nop
          nop
          nop
          nop
          nop
          mov dx, PC
          in al, dx
          not al
          nop
          nop
          and al, 0fh
          jnz key2
          inc bl
          loop key1
          jmp nkey
key2:     test al, 01h
          je key3
          mov al, 00h
          jmp key6
key3:     test al, 02h
          je key4
          mov al, 08h
          jmp key6
key4:     test al, 04h
          je key5
          mov al, 10h
          jmp key6
```

```
key5:    test al, 08h
          je nkey
          mov al, 18h
key6:    add al, bl
          cmp al, 10h
          jnc fkey
          mov bl, al
          mov bh, 0h
          mov si, offset data2
          mov al, [bx+si]
          ret
nkey:    mov al, 20h
fkey:    ret
data2:    db 07h, 04h, 08h, 05h, 09h, 06h, 0ah, 0bh
          DB 01h, 00h, 02h, 0fh, 03h, 0eh, 0ch, 0dh
for8251: call t8253
          mov al, 65h
          out dx, al
          mov dx, 03f9h
          mov al, 25h
          out dx, al
          mov dx, 03f9h
          mov al, 65h
          out dx, al
          mov dx, 03f9h
          mov al, 4eh
          out dx, al
          mov dx, 03f9h
          mov al, 25h
          out dx, al
          ret
T8253:    MOV DX, 43H                ;9600
          MOV AL, 76H
          out dx, al
          MOV DX, 41H
          MOV AL, 0CH
          out dx, al
          MOV DX, 41H
          MOV AL, 00H
```

```

        out dx, al
        mov dx, 03F9H
        mov dx, 03f9h
        RET
DISP:   MOV AL, 0FFH                ;00H
        MOV DX, PA
        OUT DX, AL
        MOV CL, 0DFH                ;20H ;显示子程序 , 5ms
        MOV BX, OFFSET BUF
DIS1:   MOV AL, [BX]
        MOV AH, 00H
        PUSH BX
        MOV BX, OFFSET DATA1
        ADD BX, AX
        MOV AL, [BX]
        POP BX
        MOV DX, PB
        OUT DX, AL
        MOV AL, CL
        MOV DX, PA
        OUT DX, AL
        PUSH CX
DIS2:   MOV CX, 00A0H
        LOOP $
        POP CX
        CMP CL, 0FEH                ;01H
        JZ LX1
        INC BX
        ROR CL, 1                    ;SHR CL, 1
        JMP DIS1
LX1:   MOV AL, 0FFH
        MOV DX, PB
        OUT DX, AL
        RET
BUF1:   MOV BUF, 08H
        MOV BUF+1, 02H
        MOV BUF+2, 05H
        MOV BUF+3, 01H
        MOV BUF+4, 17H

```

```
                MOV BUF+5, 01H
                RET
BUF2:           MOV BUF, 09H
                MOV BUF+1, 00H
                MOV BUF+2, 00H
                MOV BUF+3, 0dH
                MOV BUF+4, 10H
                MOV BUF+5, 10H
                RET
BUF3:           MOV BUF, 0eH
                MOV BUF+1, 18H
                MOV BUF+2, 18H
                MOV BUF+3, 10H
                MOV BUF+4, 10H
                MOV BUF+5, 10H
                RET
CODE ENDS
END    START
```

## 实验十七 LED16\*16 点阵显示实验

### 一、实验目的

- (1) 了解 16X16 矩阵 LED 基本原理和功能。
- (2) 掌握微机接口芯片与 LED 点阵块之间接口电路设计及编程。

### 二、实验内容

利用取模软件建立标准字库，编制程序实现点阵循环左移显示汉字。

### 三、编程指南

1、16X16 矩阵 LED 为共阴极显示，由四个 8X8 LED 点阵块组成，根据提供 I/O 地址、功能，由不同 I/O 口分别提供字形代码送行，列扫描信号送列扫描行，凡字形代码位“1”、列扫描信号“0”该点点亮，否则熄灭，通过逐列扫描，循环点亮字形或曲线。

2、用 8255 的 PA、PB 输出接口提供扫描列信号；用 8255 的 PC 口和 273 输出接口提供扫描行信号，输出字形代码，完成  $16 \times 16$  的点阵显示。

#### 四、实验接线图

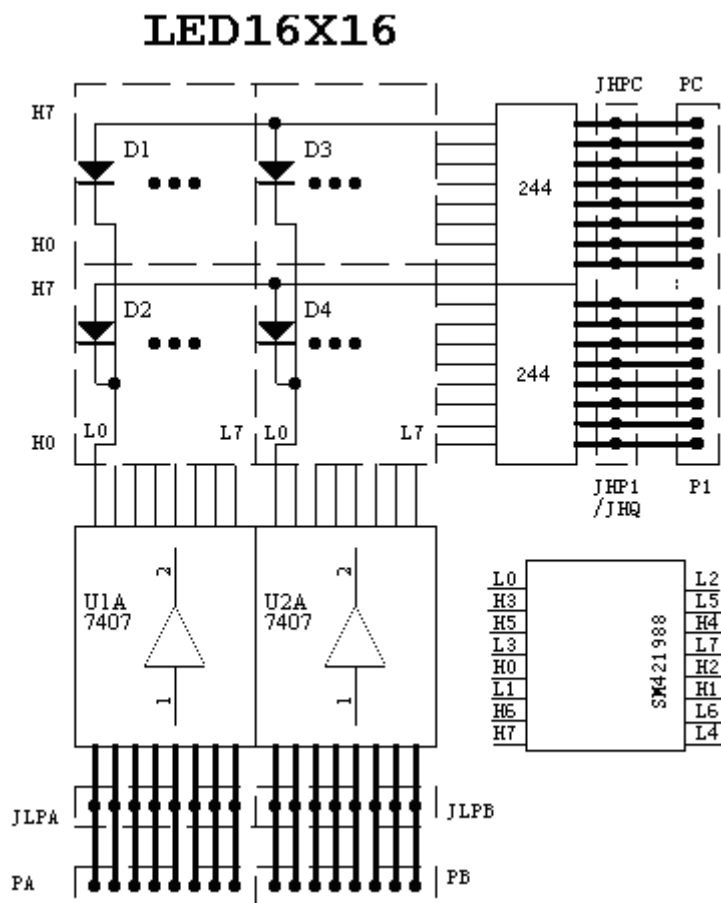


图 6-14

#### 五、实验步骤

1、在 P. 态下，按 SCAL 键。

2、将 JHPC 接 JX16 (PC)，JX7→JX17，CS2→FF80H，JQ→JHQ，JLPA 接 JX9 (PA)，JLPB 接 JX15 (PB)。

3、运行程序，输入 1B90，按 EXEC 键，即可左移显示“启东东疆计算机公司 . . . . .”字样。



## 六、实验程序清单

```

;LEDQD. ASM, USR-LED 16X16, TAB=ffff (2048x32); XPA=0D000H
XPA    EQU 0FF28H          ;L-SAO-1
XPB    EQU 0FF29H          ;L-SAO-2
XPC    EQU 0FF2AH          ;H-CODE-2
XPCTL  EQU 0FF2BH
;-----
PO      EQU 0FF80H          ;H-CODE-1
DATA    SEGMENT          AT 0
        ORG 1B90H
FL      DB ?               ;BL SAO FLAG
RH      DW ?               ;H-CODE
RL      DB ?               ;L-CODE
PAG     DW ?               ;XPA/XPB
DATA    ENDS
CODE     SEGMENT
ASSUME   CS:CODE, DS:DATA
        ORG 2DC0H
START:   MOV DX, XPCTL
        MOV AL, 80H
        OUT DX, AL
        CALL OFFLED
        MOV AL, 00H
        MOV FL, AL
        MOV PAG, XPA        ;0FF28H    XPA/XPB
X0:      MOV RH, 0000H        ;H-CODE
X1:      CALL DEL1
        INC RH
        MOV AX, RH
        AND AX, 000FH
        CMP AX, 0000H
        JNZ X2
        ADD RH, 0010H
x2:      CMP RH, 0641h        ;0ffe1H
        JZ X0
        jmp x1
DISPW:   CLC
        PUSH RH              ;H-NUM

```

```

MOV RL, 01H                ;L-NUM
DISP1:  MOV SI, RH
        MOV BX, OFFSET TAB
        MOV AL, CS:[BX+SI]
        MOV DX, XPC          ;PA, CODE->SHANG BAN BU
        OUT DX, AL
        MOV AL, CS:[BX+SI+10H];XPC, CODE->XIA BAN BU
        MOV DX, PO
        OUT DX, AL
        MOV DX, PAG
        MOV AL, RL
        NOT AL
        OUT DX, AL
        MOV CX, 0080H
        LOOP $
        CALL OFFLED
        INC RH
;-----
        MOV AX, RH
        AND AX, 000FH
        CMP AX, 0000H
        JNZ CON2
        ADD RH, 0010H
CON2:   RCL RL, 1
        JNC DISP1
        CMP FL, 00H
        JNZ EXIT
        INC FL
        CLC
        MOV RL, 01H
        MOV PAG, XPB          ;0FF29H
        JMP DISP1
EXIT:   MOV PAG, XPA          ;0FF28H
        MOV FL, 00H
        POP RH
        RET
OFFLED: MOV DX, XPA
        MOV AL, 0FFH
        OUT DX, AL

```

```

MOV DX, XPB
OUT DX, AL
RET
DEL1: MOV CX, 0020H
CON1: PUSH CX
CALL DISPW
POP CX
LOOP CON1
RET
TAB: ;-- 文字: 启 --
;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --
DB 00H, 00H, 00H, 1FH, 12H, 12H, 12H, 92H, 72H, 52H, 12H, 12H, 12H, 1EH, 00H, 00H
DB 02H, 04H, 18H, 0E0H, 00H, 7EH, 44H, 44H, 44H, 44H, 44H, 44H, 44H, 7EH, 00H, 00H

;-- 文字: 东 --
;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --
DB 00H, 20H, 20H, 23H, 2DH, 31H, 0E1H, 21H, 2FH, 21H, 21H, 21H, 21H, 20H, 00H, 00H
DB 00H, 00H, 04H, 18H, 70H, 20H, 04H, 02H, 0FFH, 00H, 40H, 20H, 18H, 0CH, 00H, 00H

;-- 文字: 东 --
;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --
DB 00H, 20H, 20H, 23H, 2DH, 31H, 0E1H, 21H, 2FH, 21H, 21H, 21H, 21H, 20H, 00H, 00H
DB 00H, 00H, 04H, 18H, 70H, 20H, 04H, 02H, 0FFH, 00H, 40H, 20H, 18H, 0CH, 00H, 00H

;-- 文字: 疆 --
;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --
DB 80H, 9EH, 92H, 92H, 0F3H, 00H, 80H, 0BEH, 0AAH, 0AAH, 0BEH, 0AAH, 0AAH, 0BEH, 80H, 00H
DB 48H, 0F4H, 52H, 04H, 0F8H, 02H, 82H, 0FAH, 0AAH, 0AAH, 0FAH, 0AAH, 0AAH, 0FAH, 82H, 00H

;-- 文字: 计 --
;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --
DB 04H, 84H, 74H, 27H, 00H, 00H, 04H, 04H, 04H, 04H, 0FFH, 04H, 04H, 04H, 04H, 00H
DB 00H, 00H, 00H, 0FEH, 04H, 08H, 10H, 00H, 00H, 00H, 0FFH, 00H, 00H, 00H, 00H, 00H

;-- 文字: 算 --
;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --
DB 10H, 0E0H, 5FH, 55H, 75H, 55H, 55H, 15H, 35H, 0D5H, 55H, 7FH, 50H, 40H, 40H, 00H
DB 10H, 10H, 0D1H, 56H, 78H, 50H, 50H, 50H, 50H, 7FH, 50H, 0D0H, 10H, 10H, 10H, 00H

```

;-- 文字： 机 --

;-- 宋体 12; 此字体下对应的点阵为： 宽 x 高=16x16 --

DB 10H, 10H, 13H, 0FFH, 12H, 11H, 10H, 00H, 7FH, 40H, 40H, 40H, 7FH, 00H, 00H, 00H

DB 20H, 0C0H, 00H, 0FFH, 00H, 82H, 0CH, 30H, 0C0H, 00H, 00H, 00H, 0FCH, 02H, 1EH, 00H

;-- 文字： 公 --

;-- 宋体 12; 此字体下对应的点阵为： 宽 x 高=16x16 --

DB 00H, 00H, 01H, 02H, 0CH, 70H, 21H, 00H, 00H, 70H, 08H, 06H, 03H, 01H, 01H, 00H

DB 00H, 80H, 04H, 0EH, 14H, 24H, 0C4H, 8CH, 08H, 08H, 28H, 1EH, 0CH, 80H, 00H, 00H

;-- 文字： 司 --

;-- 宋体 12; 此字体下对应的点阵为： 宽 x 高=16x16 --

DB 00H, 08H, 49H, 49H, 49H, 49H, 49H, 49H, 49H, 48H, 40H, 40H, 7FH, 00H, 00H

DB 00H, 00H, 0F8H, 20H, 20H, 20H, 20H, 20H, 20H, 0F0H, 00H, 04H, 02H, 0FCH, 00H, 00H

;-- 文字： 好 --

;-- 宋体 12; 此字体下对应的点阵为： 宽 x 高=16x16 --

;DB 08H, 08H, 0FH, 0F8H, 08H, 0FH, 01H, 41H, 41H, 41H, 4FH, 51H, 61H, 41H, 01H, 00H

;DB 01H, 0C2H, 44H, 28H, 30H, 0CEH, 04H, 00H, 02H, 01H, 0FEH, 00H, 00H, 00H, 00H, 00H

;-- 文字： ！ --

;-- 宋体 12; 此字体下对应的点阵为： 宽 x 高=16x16 --

;DB 00H, 00H, 00H, 1FH, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H

;DB 00H, 00H, 00H, 0CCH, 0CH, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H

;-- 文字： ， --

;-- 宋体 12; 此字体下对应的点阵为： 宽 x 高=16x16 --

DB 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H

DB 00H, 00H, 1AH, 1CH, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H

;-- 文字： 主 --

;-- 宋体 12; 此字体下对应的点阵为： 宽 x 高=16x16 --

DB 00H, 00H, 08H, 08H, 08H, 08H, 88H, 6FH, 48H, 08H, 08H, 08H, 19H, 08H, 00H, 00H

DB 00H, 04H, 84H, 84H, 84H, 84H, 84H, 0FCH, 84H, 84H, 84H, 84H, 84H, 8CH, 04H, 00H

;-- 文字： 要 --

;-- 宋体 12; 此字体下对应的点阵为： 宽 x 高=16x16 --

DB 40H, 40H, 5FH, 52H, 52H, 7FH, 52H, 52H, 52H, 7EH, 52H, 52H, 5FH, 40H, 40H, 00H

DB 40H, 41H, 41H, 42H, 72H, 0D4H, 54H, 48H, 48H, 48H, 54H, 64H, 43H, 42H, 40H, 00H

;-- 文字： 生 --

;-- 宋体 12; 此字体下对应的点阵为： 宽 x 高=16x16 --

DB 00H, 01H, 06H, 78H, 08H, 08H, 08H, 08H, 0FFH, 48H, 08H, 08H, 19H, 08H, 00H, 00H

DB 80H, 02H, 02H, 82H, 82H, 82H, 82H, 82H, 0FEH, 82H, 82H, 82H, 82H, 86H, 02H, 00H

;-- 文字： 产 --

;-- 宋体 12; 此字体下对应的点阵为： 宽 x 高=16x16 --

DB 00H, 20H, 27H, 22H, 32H, 2EH, 2AH, 0A2H, 62H, 26H, 2AH, 32H, 22H, 26H, 22H, 00H

DB 02H, 0CH, 0F0H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H

;-- 文字： 单 --

;-- 宋体 12; 此字体下对应的点阵为： 宽 x 高=16x16 --

DB 00H, 00H, 1FH, 14H, 94H, 74H, 54H, 1FH, 14H, 34H, 0D4H, 54H, 1FH, 00H, 00H, 00H

DB 10H, 10H, 0D0H, 90H, 90H, 90H, 90H, 0FFH, 90H, 90H, 90H, 90H, 0D0H, 10H, 10H, 00H

;-- 文字： 片 --

;-- 宋体 12; 此字体下对应的点阵为： 宽 x 高=16x16 --

DB 00H, 00H, 00H, 7FH, 08H, 08H, 08H, 08H, 08H, 0F8H, 08H, 08H, 08H, 18H, 08H, 00H

DB 01H, 02H, 0CH, 0F0H, 80H, 80H, 80H, 80H, 80H, 80H, 80H, 0FFH, 00H, 00H, 00H, 00H

;-- 文字： 机 --

;-- 宋体 12; 此字体下对应的点阵为： 宽 x 高=16x16 --

DB 10H, 10H, 13H, 0FFH, 12H, 11H, 10H, 00H, 7FH, 40H, 40H, 40H, 7FH, 00H, 00H, 00H

DB 20H, 0C0H, 00H, 0FFH, 00H, 82H, 0CH, 30H, 0C0H, 00H, 00H, 00H, 0FCH, 02H, 1EH, 00H

;-- 文字： ， --

;-- 宋体 12; 此字体下对应的点阵为： 宽 x 高=16x16 --

DB 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H

DB 00H, 00H, 1AH, 1CH, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H

;-- 文字： 微 --

;-- 宋体 12; 此字体下对应的点阵为： 宽 x 高=16x16 --

DB 08H, 11H, 0EFH, 44H, 3AH, 0AH, 0FAH, 0AH, 3AH, 04H, 1FH, 0E8H, 48H, 0FH, 08H, 00H

DB 80H, 00H, 0FFH, 02H, 04H, 0F8H, 80H, 80H, 0FDH, 0AH, 84H, 68H, 10H, 0EFH, 02H, 00H

;-- 文字： 机 --

;-- 宋体 12; 此字体下对应的点阵为： 宽 x 高=16x16 --

DB 10H, 10H, 13H, 0FFH, 12H, 11H, 10H, 00H, 7FH, 40H, 40H, 40H, 7FH, 00H, 00H, 00H

DB 20H, 0C0H, 00H, 0FFH, 00H, 82H, 0CH, 30H, 0C0H, 00H, 00H, 00H, 0FCH, 02H, 1EH, 00H

;-- 文字： 原 --

;-- 宋体 12; 此字体下对应的点阵为： 宽 x 高=16x16 --

DB 00H, 00H, 7FH, 40H, 40H, 4FH, 4AH, 7AH, 6AH, 4AH, 4AH, 4FH, 40H, 40H, 00H, 00H

DB 03H, 0CH, 0F0H, 02H, 04H, 0D8H, 4AH, 41H, 7EH, 40H, 50H, 0C8H, 0EH, 04H, 00H, 00H

;-- 文字： 理 --

;-- 宋体 12; 此字体下对应的点阵为： 宽 x 高=16x16 --

DB 22H, 22H, 3FH, 22H, 22H, 00H, 7FH, 49H, 49H, 7FH, 49H, 49H, 49H, 7FH, 00H, 00H

DB 08H, 08H, 0F0H, 10H, 12H, 02H, 0A2H, 22H, 22H, 0FEH, 22H, 22H, 22H, 0A2H, 02H, 00H

;-- 文字： 与 --

;-- 宋体 12; 此字体下对应的点阵为： 宽 x 高=16x16 --

DB 00H, 00H, 00H, 00H, 7EH, 12H, 12H, 12H, 12H, 12H, 12H, 12H, 12H, 33H, 10H, 00H

DB 00H, 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H, 24H, 62H, 22H, 04H, 0F8H, 00H, 00H

;-- 文字： 接 --

;-- 宋体 12; 此字体下对应的点阵为： 宽 x 高=16x16 --

DB 10H, 10H, 10H, 0FFH, 11H, 16H, 24H, 34H, 2DH, 0A4H, 64H, 2CH, 34H, 24H, 04H, 00H

DB 40H, 42H, 81H, 0FEH, 40H, 41H, 51H, 72H, 0CAH, 4CH, 48H, 74H, 42H, 43H, 40H, 00H

;-- 文字： 口 --

;-- 宋体 12; 此字体下对应的点阵为： 宽 x 高=16x16 --

DB 00H, 00H, 3FH, 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H, 3FH, 00H, 00H, 00H

DB 00H, 00H, 0FCH, 10H, 10H, 10H, 10H, 10H, 10H, 10H, 10H, 10H, 0FCH, 00H, 00H, 00H

;-- 文字： ， --

;-- 宋体 12; 此字体下对应的点阵为： 宽 x 高=16x16 --

DB 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H

DB 00H, 00H, 1AH, 1CH, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H

;-- 文字： 计 --

;-- 宋体 12; 此字体下对应的点阵为： 宽 x 高=16x16 --

DB 04H, 84H, 74H, 27H, 00H, 00H, 04H, 04H, 04H, 04H, 0FFH, 04H, 04H, 04H, 04H, 00H

DB 00H, 00H, 00H, 0FEH, 04H, 08H, 10H, 00H, 00H, 00H, 0FFH, 00H, 00H, 00H, 00H, 00H

;-- 文字： 算 --

;-- 宋体 12; 此字体下对应的点阵为： 宽 x 高=16x16 --

DB 10H, 0E0H, 5FH, 55H, 75H, 55H, 55H, 15H, 35H, 0D5H, 55H, 7FH, 50H, 40H, 40H, 00H

DB 10H, 10H, 0D1H, 56H, 78H, 50H, 50H, 50H, 50H, 7FH, 50H, 0D0H, 10H, 10H, 10H, 00H

;-- 文字: 机 --

;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --

DB 10H, 10H, 13H, 0FFH, 12H, 11H, 10H, 00H, 7FH, 40H, 40H, 40H, 7FH, 00H, 00H, 00H

DB 20H, 0C0H, 00H, 0FFH, 00H, 82H, 0CH, 30H, 0C0H, 00H, 00H, 00H, 0FCH, 02H, 1EH, 00H

;-- 文字: 组 --

;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --

DB 00H, 04H, 1CH, 0E5H, 46H, 18H, 00H, 7FH, 44H, 44H, 44H, 44H, 7FH, 00H, 00H, 00H

DB 00H, 44H, 0C4H, 44H, 48H, 48H, 04H, 0FCH, 44H, 44H, 44H, 44H, 0FCH, 04H, 04H, 00H

;-- 文字: 成 --

;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --

DB 00H, 00H, 1FH, 12H, 12H, 12H, 13H, 10H, 0FFH, 10H, 90H, 50H, 13H, 11H, 10H, 00H

DB 02H, 0CH, 0F0H, 00H, 10H, 0AH, 0F2H, 04H, 08H, 0D0H, 30H, 48H, 84H, 02H, 0FH, 00H

;-- 文字: 原 --

;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --

DB 00H, 00H, 7FH, 40H, 40H, 4FH, 4AH, 7AH, 6AH, 4AH, 4AH, 4FH, 40H, 40H, 00H, 00H

DB 03H, 0CH, 0F0H, 02H, 04H, 0D8H, 4AH, 41H, 7EH, 40H, 50H, 0C8H, 0EH, 04H, 00H, 00H

;-- 文字: 理 --

;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --

DB 22H, 22H, 3FH, 22H, 22H, 00H, 7FH, 49H, 49H, 7FH, 49H, 49H, 49H, 7FH, 00H, 00H

DB 08H, 08H, 0F0H, 10H, 12H, 02H, 0A2H, 22H, 22H, 0FEH, 22H, 22H, 22H, 0A2H, 02H, 00H

;-- 文字: , --

;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --

DB 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H

DB 00H, 00H, 1AH, 1CH, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H

;-- 文字: E --

;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --

DB 10H, 1FH, 11H, 11H, 17H, 10H, 08H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H

DB 04H, 0FCH, 04H, 04H, 0C4H, 04H, 18H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H

;-- 文字: D --

;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --

DB 10H, 1FH, 10H, 10H, 10H, 08H, 07H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H  
DB 04H, 0FCH, 04H, 04H, 04H, 08H, 0F0H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H

;-- 文字: A --

;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --

DB 00H, 00H, 03H, 1CH, 07H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H  
DB 04H, 3CH, 0C4H, 40H, 40H, 0E4H, 1CH, 04H, 00H, 00H, 00H, 00H, 00H, 00H, 00H

;-- 文字: , --

;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --

DB 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H  
DB 00H, 00H, 1AH, 1CH, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H

;-- 文字: 数 --

;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --

DB 08H, 49H, 2AH, 1CH, 0FFH, 1CH, 2AH, 4AH, 01H, 0FH, 0F8H, 48H, 08H, 0FH, 08H, 00H  
DB 42H, 42H, 54H, 74H, 0C8H, 58H, 64H, 40H, 02H, 04H, 0C8H, 30H, 0CCH, 06H, 04H, 00H

;-- 文字: 电 --

;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --

DB 00H, 00H, 1FH, 12H, 12H, 12H, 12H, 0FFH, 12H, 12H, 12H, 12H, 1FH, 00H, 00H, 00H  
DB 00H, 00H, 0F0H, 20H, 20H, 20H, 20H, 0FCH, 22H, 22H, 22H, 22H, 0F2H, 02H, 0EH, 00H

;-- 文字: 模 --

;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --

DB 08H, 0BH, 0FFH, 0AH, 09H, 20H, 2FH, 2AH, 0FAH, 2AH, 2AH, 0FAH, 2FH, 20H, 00H, 00H  
DB  
0C0H, 00H, 0FFH, 00H, 00H, 21H, 0A1H, 0A2H, 0ACH, 0F0H, 0A8H, 0A4H, 0A6H, 23H, 22H, 00H

;-- 文字: 电 --

;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --

DB 00H, 00H, 1FH, 12H, 12H, 12H, 12H, 0FFH, 12H, 12H, 12H, 12H, 1FH, 00H, 00H, 00H  
DB 00H, 00H, 0F0H, 20H, 20H, 20H, 20H, 0FCH, 22H, 22H, 22H, 22H, 0F2H, 02H, 0EH, 00H

;-- 文字: 等 --

;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --

DB 09H, 11H, 0E5H, 45H, 65H, 5DH, 45H, 1FH, 0E5H, 45H, 65H, 5DH, 45H, 41H, 01H, 00H  
DB 00H, 20H, 20H, 20H, 20H, 30H, 2CH, 20H, 22H, 21H, 0FEH, 20H, 20H, 20H, 00H, 00H

;-- 文字: 实 --



```

;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16  --
DB 00H, 08H, 30H, 20H, 32H, 2DH, 29H, 0A0H, 6FH, 20H, 20H, 20H, 28H, 30H, 20H, 00H
DB 00H, 41H, 41H, 42H, 42H, 0C4H, 48H, 50H, 0E0H, 50H, 48H, 47H, 42H, 40H, 40H, 00H
;-- 文字: 验 --
;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16  --
DB 40H, 5FH, 41H, 41H, 7FH, 01H, 02H, 06H, 1AH, 62H, 12H, 0AH, 04H, 04H, 04H, 00H
DB 10H, 10H, 20H, 24H, 02H, 0FCH, 44H, 34H, 84H, 74H, 04H, 0CH, 34H, 0C4H, 04H, 00H
;-- 文字: 仪 --
;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16  --
DB 02H, 04H, 0FH, 30H, 0C0H, 00H, 1CH, 03H, 80H, 70H, 20H, 07H, 38H, 00H, 00H, 00H
DB 00H, 00H, 0FFH, 00H, 02H, 02H, 04H, 08H, 0D0H, 20H, 0D0H, 08H, 04H, 06H, 04H, 00H
;-- 文字: 器 --
;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16  --
DB 02H, 02H, 0F2H, 92H, 92H, 93H, 0F3H, 0EH, 03H, 0F3H, 92H, 9AH, 96H, 0F2H, 00H, 00H
DB 40H, 40H, 7EH, 0A2H, 0A2H, 22H, 3EH, 00H, 3EH, 22H, 0A2H, 0A2H, 7EH, 60H, 40H, 00H

;-- 文字: 的 --
;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16  --
;DB 00H, 1FH, 31H, 0D1H, 11H, 1FH, 02H, 0CH, 0F1H, 10H, 10H, 10H, 10H, 1FH, 00H, 00H
;DB 00H, 0FEH, 08H, 08H, 08H, 0FCH, 00H, 00H, 00H, 0C0H, 64H, 02H, 04H, 0F8H, 00H, 00H
;-- 文字: 开 --
;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16  --
DB 02H, 42H, 42H, 42H, 42H, 7FH, 42H, 42H, 42H, 42H, 7FH, 42H, 42H, 42H, 42H, 00H
DB 00H, 02H, 04H, 08H, 30H, 0C0H, 00H, 00H, 00H, 00H, 0FEH, 00H, 00H, 00H, 00H, 00H
;-- 文字: 发 --
;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16  --
DB 00H, 08H, 7CH, 08H, 08H, 0FH, 0F9H, 09H, 09H, 49H, 29H, 38H, 08H, 08H, 08H, 00H
DB 02H, 04H, 08H, 11H, 0E1H, 82H, 62H, 14H, 08H, 14H, 0E4H, 02H, 03H, 02H, 00H, 00H
;-- 文字: 制 --
;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16  --
DB 00H, 0AH, 0F2H, 52H, 12H, 0FFH, 12H, 12H, 12H, 00H, 3FH, 00H, 00H, 0FFH, 00H, 00H
DB 00H, 00H, 0FCH, 80H, 80H, 0FFH, 84H, 86H, 0FCH, 00H, 0F0H, 02H, 01H, 0FEH, 00H, 00H
;-- 文字: 造 --
;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16  --
DB 02H, 82H, 63H, 00H, 02H, 0AH, 72H, 12H, 12H, 0FEH, 12H, 12H, 12H, 12H, 02H, 00H
DB 02H, 04H, 0F8H, 04H, 02H, 02H, 0FAH, 92H, 92H, 92H, 92H, 92H, 0FAH, 02H, 02H, 00H
CODE ENDS
END START

```

实验十八 128×64 LCD 液晶显示实验

一、实验目的

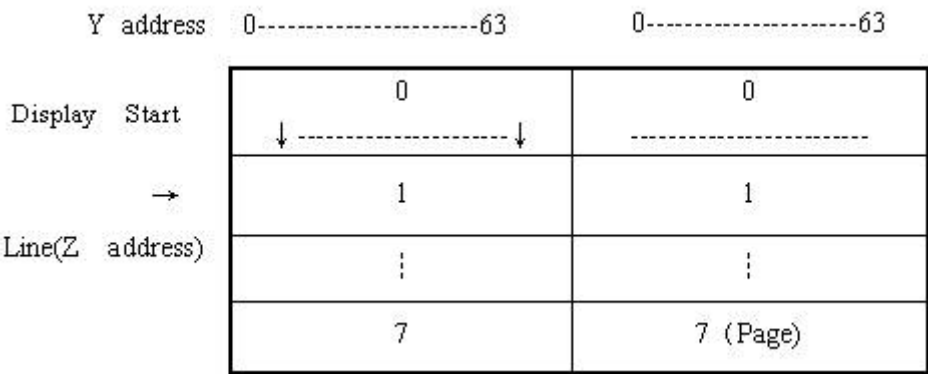
- (1) 了解点阵式液晶显示器工作原理和显示方法。
- (2) 掌握 8088 微机系统与液晶显示器之间接口设计与编程。

二、实验内容

利用取模软件建立标准字库，编制程序，在液晶显示器上显示汉字。

三、编程指南

- (1) 显示内存与液晶显示屏关系



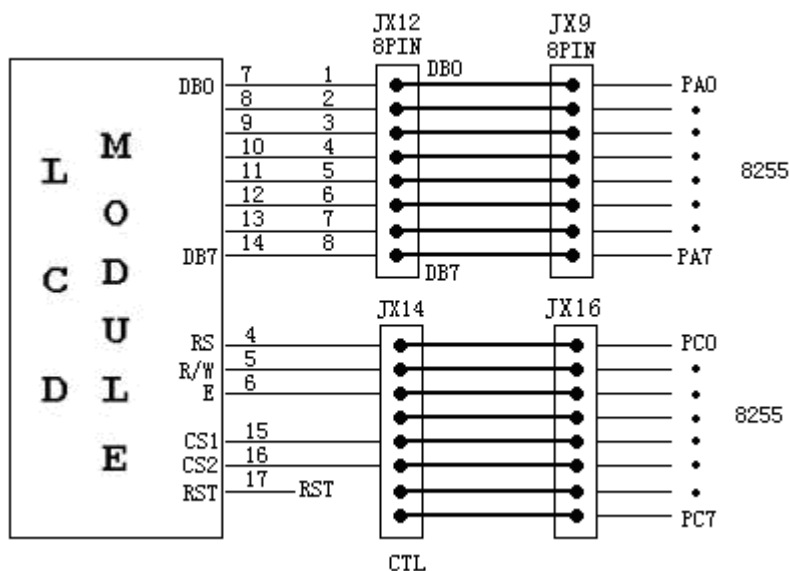
(2) 利用 8255 芯片的 PA 口作为液晶显示器接口的数据线，PC 口作为其控制线，自建字库后，通过查表程序依次将字库中的字形代码送显示内存显示汉字或图形。

(3) 编程流程：开显示→设置页地址→设置 Y 地址→写数据表 1→写数据表 2。

- (4) 显示控制指令表

Instruction	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Function
Display ON/OFF	L	L	L	L	H	H	H	H	H	L/H	Controls the Display on or off Internal status and display RAM data is not affected L:OFF, H:ON
Set address (Y address)	L	L	L	H		Y address(0~63)					Sets the Y address in the Y address counter
Set page (X address)	L	L	H	L	H	H	H	Page(0~7)			Sets the X address at the X address register
Display Start Line (Z address)	L	L	H	H	Display start line(0~63)						Indicates the display data RAM displayed at the top of the screen
Status Read	L	H	B U S Y	L	O N / O F F	R E S E T	L	L	L	L	BUSY L:Ready H:In operation ON/OFF L:Display OFF RESET L:Normal H:Reset
Write Display Data	H	L	Write Data								Writes data (DB0:7) into display data RAM, After writing instruction, Y address is increased by 1 automatically
Read Display Data	H	H	Read Data								Reads data (DB0:) from display data RAM to the data bus

## 四、实验接线图



## 五、实验步骤

(1) 在 P. 态下，按 SCAL 键。

(2) 用扁平线连 JX9 座 (PA 口) 到 JX12 座 (DB0~DB7)，连 JX16 座 (PC 口) 到 JX14 座。

(3) 运行程序 LCDQD-ASM; 或在实验系统处于 P. 态下，输入 [22E0]→[EXEC] 即可。

## 六、实验程序清单

```
CODE      SEGMENT
ASSUME    CS:   CODE
          PA     EQU 0FF28H
          PCTL    EQU 0FF2BH
          RSN     EQU 00H           ;PC0
          RS      EQU 01H
          RWN     EQU 02H           ;PC1
          RW      EQU 03H
          EN      EQU 04H           ;PC2
          E       EQU 05H
          CS1N    EQU 08H           ;PC4
```

```
CS1    EQU 09H
CS2N    EQU 0ah          ;PC5
CS2     EQU 0bh
ORG     22E0h            ;
JMP START
YR      DB ?
ZR      DB ?
PAG     DB ?
Y16     DB ?
pag7    db ?
FL      DB ?
START:  MOV DX, PCTL
        MOV AL, 80H
        OUT DX, AL
        MOV FL, 00H
        MOV CX, 0003H
        CALL DELAYXMS
        MOV DX, PCTL
        MOV AL, EN
        OUT DX, AL
        MOV AL, RS
        OUT DX, AL
        MOV AL, CS1
        OUT DX, AL
        MOV AL, CS2N
        OUT DX, AL
        MOV AL, RSN
        OUT DX, AL
        MOV DX, PA
        MOV AL, 3FH      ;WR ON
        OUT DX, AL
        CALL WRITE
        MOV CX, 0001H
        CALL DELAYXMS
        MOV DX, PCTL
        MOV AL, CS2
        OUT DX, AL
        MOV AL, CS1N
        OUT DX, AL
```

```

MOV AL, RSN
OUT DX, AL
MOV DX, PA
MOV AL, 3FH
OUT DX, AL
CALL WRITE
MAIN: MOV BX, OFFSET TAB5
CALL ZXL ;DISP 1-SCREEN
JMP $
ZXL: MOV DX, PCTL ;DISP 1-SCREEN
MOV AL, CS2N
OUT DX, AL
MOV AL, CS1 ;DISP L-SCREEN
OUT DX, AL
CALL XPAGE ;SET X, Y, PAG
CALL COM ;DISP
MOV DX, PCTL
MOV AL, CS1N
OUT DX, AL
MOV AL, CS2 ;DISP R-SCREEN
OUT DX, AL
CALL XPAGE ;SET X, Y, PAG
CALL COM ;DISP
RET
XPAGE: MOV ZR, 0c0H ;SET Z, Y, PAG
MOV YR, 40H
MOV PAG, 0B8H
mov pag7, 08h
MOV DX, PCTL
MOV AL, EN
OUT DX, AL
MOV AL, RSN ;wr instruction
OUT DX, AL
MOV DX, PA
MOV AL, ZR ;set z=0
OUT DX, AL
CALL WRITE
MOV DX, PA
MOV AL, YR ;set y

```

```

        OUT DX, AL
        CALL WRITE
        MOV DX, PA
        MOV AL, PAG                ;SET PAG
        OUT DX, AL
        CALL WRITE
        RET
COM:    MOV Y16, 10H
        MOV CX, 0040H
        MOV DX, PCTL
        MOV AL, RS
        OUT DX, AL
J44:    push bx
J4:     MOV AL, CS: [BX]
        MOV DX, PA
        OUT DX, AL
        push cx
        CALL WRITE
        POP CX
        inc bx                    ;!!!!
        DEC Y16
        CMP Y16, 0000H
        JNZ J42
        MOV Y16, 10H
        ADD BX, 0010H
J42:    LOOP J4
        NOT FL
        CMP FL, 00H
        JZ J45
        pop bx
        add bx, 0010h
J46:    INC PAG
        dec pag7
        cmp pag7, 00H
        jz j43
        MOV AL, RSN
        MOV DX, PCTL
        OUT DX, AL
        MOV DX, PA

```

```
MOV AL, YR                                ;set y
OUT DX, AL
CALL WRITE
MOV DX, PA
MOV AL, PAG                                ;SET PAG
OUT DX, AL
CALL WRITE
JMP COM
J43: RET
J45: pop bx
      add bx, 0070h
      JMP J46
;-----
WRITE: MOV DX, PCTL
        MOV AL, RWN
        OUT DX, AL
        MOV AL, EN
        OUT DX, AL
        MOV AL, E
        OUT DX, AL
        CALL DELAY2MS
        MOV AL, EN
        OUT DX, AL
        RET
DELAY2MS:
        MOV CX, 78H
        LOOP $
        RET
DELAYXMS:
        PUSH CX
        MOV CX, 0040H
DELY2:  PUSH CX
        MOV CX, 3E8H
        LOOP $
        POP CX
        LOOP DELY2
        POP CX
        LOOP DELAYXMS
```



## RET

TAB5:

DB 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H

DB 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H

DB 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H

DB 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H

;-- 文字: 欢 --

;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --

DB 14H, 24H, 44H, 84H, 64H, 1CH, 20H, 18H, 0FH, 0E8H, 08H, 08H, 28H, 18H, 08H, 00H

DB 20H, 10H, 4CH, 43H, 43H, 2CH, 20H, 10H, 0CH, 03H, 06H, 18H, 30H, 60H, 20H, 00H

;-- 文字: 迎 --

;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --

DB 40H, 41H, 0CEH, 04H, 00H, 0FCH, 04H, 02H, 02H, 0FCH, 04H, 04H, 04H, 0FCH, 00H, 00H

DB 40H, 20H, 1FH, 20H, 40H, 47H, 42H, 41H, 40H, 5FH, 40H, 42H, 44H, 43H, 40H, 00H

;-- 文字: 启 --

;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --

DB 00H, 00H, 00H, 0F8H, 48H, 48H, 48H, 49H, 4EH, 4AH, 48H, 48H, 48H, 78H, 00H, 00H

DB 40H, 20H, 18H, 07H, 00H, 7EH, 22H, 22H, 22H, 22H, 22H, 22H, 22H, 7EH, 00H, 00H

;-- 文字: 东 --

;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --

DB 00H, 04H, 04H, 0C4H, 0B4H, 8CH, 87H, 84H, 0F4H, 84H, 84H, 84H, 84H, 04H, 00H, 00H

DB 00H, 00H, 20H, 18H, 0EH, 04H, 20H, 40H, 0FFH, 00H, 02H, 04H, 18H, 30H, 00H, 00H

;-- 文字: 东 --

;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --

DB 00H, 04H, 04H, 0C4H, 0B4H, 8CH, 87H, 84H, 0F4H, 84H, 84H, 84H, 84H, 04H, 00H, 00H

DB 00H, 00H, 20H, 18H, 0EH, 04H, 20H, 40H, 0FFH, 00H, 02H, 04H, 18H, 30H, 00H, 00H

;-- 文字: 疆 --

;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --

DB 01H, 79H, 49H, 49H, 0CFH, 00H, 01H, 7DH, 55H, 55H, 7DH, 55H, 55H, 7DH, 01H, 00H

DB 12H, 2FH, 4AH, 20H, 1FH, 40H, 41H, 5FH, 55H, 55H, 5FH, 55H, 55H, 5FH, 41H, 00H

;-- 文字: 限 --

;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --

DB 0FEH, 02H, 32H, 4EH, 82H, 00H, 0FEH, 4AH, 0CAH, 4AH, 4AH, 4AH, 7EH, 00H, 00H, 00H

DB 0FFH, 00H, 02H, 04H, 03H, 00H, 0FFH, 40H, 20H, 03H, 0CH, 12H, 21H, 60H, 20H, 00H

;-- 文字: 公 --

;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --

DB 00H, 00H, 80H, 40H, 30H, 0EH, 84H, 00H, 00H, 0EH, 10H, 60H, 0C0H, 80H, 80H, 00H

DB 00H, 01H, 20H, 70H, 28H, 24H, 23H, 31H, 10H, 10H, 14H, 78H, 30H, 01H, 00H, 00H

;-- 文字: 司 --

;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --

DB 00H, 10H, 92H, 92H, 92H, 92H, 92H, 92H, 92H, 12H, 02H, 02H, 0FEH, 00H, 00H

DB 00H, 00H, 1FH, 04H, 04H, 04H, 04H, 04H, 04H, 0FH, 00H, 20H, 40H, 3FH, 00H, 00H

;-- 文字: 生 --

;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --

DB 00H, 80H, 60H, 1EH, 10H, 10H, 10H, 10H, 0FFH, 12H, 10H, 10H, 98H, 10H, 00H, 00H

DB 01H, 40H, 40H, 41H, 41H, 41H, 41H, 41H, 7FH, 41H, 41H, 41H, 41H, 61H, 40H, 00H

;-- 文字: 系 --

;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --

DB 00H, 00H, 02H, 22H, 0B2H, 0AAH, 66H, 62H, 22H, 11H, 4DH, 81H, 01H, 01H, 00H, 00H

DB 00H, 40H, 21H, 13H, 09H, 05H, 41H, 81H, 7FH, 01H, 05H, 09H, 13H, 62H, 00H, 00H

;-- 文字: 列 --

;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --

DB 02H, 82H, 42H, 0BAH, 16H, 12H, 92H, 7AH, 12H, 00H, 0F8H, 00H, 00H, 0FFH, 00H, 00H

DB 41H, 40H, 20H, 10H, 09H, 06H, 01H, 00H, 00H, 00H, 07H, 20H, 40H, 3FH, 00H, 00H

;-- 文字: 实 --

;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --

DB 00H, 10H, 0CH, 04H, 4CH, 0B4H, 94H, 05H, 0F6H, 04H, 04H, 04H, 14H, 0CH, 04H, 00H

DB 00H, 82H, 82H, 42H, 42H, 23H, 12H, 0AH, 07H, 0AH, 12H, 0E2H, 42H, 02H, 02H, 00H

;-- 文字: 验 --

;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --

DB 02H, 0FAH, 82H, 82H, 0FEH, 80H, 40H, 60H, 58H, 46H, 48H, 50H, 20H, 20H, 20H, 00H

DB 08H, 08H, 04H, 24H, 40H, 3FH, 22H, 2CH, 21H, 2EH, 20H, 30H, 2CH, 23H, 20H, 00H

;-- 文字: 使 --

;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --

DB 40H, 20H, 0F0H, 1CH, 07H, 0F2H, 94H, 94H, 94H, 0FFH, 94H, 94H, 94H, 0F4H, 04H, 00H

DB 00H, 00H, 7FH, 00H, 40H, 41H, 22H, 14H, 0CH, 13H, 10H, 30H, 20H, 61H, 20H, 00H

;-- 文字: 用 --

;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --

DB 00H, 00H, 00H, 0FEH, 22H, 22H, 22H, 22H, 0FEH, 22H, 22H, 22H, 22H, 0FEH, 00H, 00H

DB 80H, 40H, 30H, 0FH, 02H, 02H, 02H, 02H, 0FFH, 02H, 02H, 42H, 82H, 7FH, 00H, 00H

DB 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H

DB 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H

DB 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H

DB 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H

;-- 文字: 计 --

;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --

DB 20H, 21H, 2EH, 0E4H, 00H, 00H, 20H, 20H, 20H, 20H, 0FFH, 20H, 20H, 20H, 20H, 00H

DB 00H, 00H, 00H, 7FH, 20H, 10H, 08H, 00H, 00H, 00H, 0FFH, 00H, 00H, 00H, 00H, 00H

;-- 文字: 算 --

;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --

DB

08H, 07H, 0FAH, 0AAH, 0AEH, 0AAH, 0AAH, 0A8H, 0ACH, 0ABH, 0AAH, 0FEH, 0AH, 02H, 02H, 00H

DB 08H, 08H, 8BH, 6AH, 1EH, 0AH, 0AH, 0AH, 0AH, 0FEH, 0AH, 0BH, 08H, 08H, 08H, 00H

;-- 文字: 机 --

;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --

DB 08H, 08H, 0C8H, 0FFH, 48H, 88H, 08H, 00H, 0FEH, 02H, 02H, 02H, 0FEH, 00H, 00H, 00H

DB 04H, 03H, 00H, 0FFH, 00H, 41H, 30H, 0CH, 03H, 00H, 00H, 00H, 3FH, 40H, 78H, 00H

;-- 文字: 有 --

;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --

DB 00H, 04H, 84H, 44H, 0E4H, 34H, 2CH, 27H, 24H, 24H, 24H, 0E4H, 04H, 04H, 04H, 00H

DB 02H, 01H, 00H, 00H, 0FFH, 09H, 09H, 09H, 29H, 49H, 0C9H, 7FH, 00H, 00H, 00H, 00H

;-- 文字: 产 --

;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --

DB 00H, 04H, 0E4H, 44H, 4CH, 74H, 54H, 45H, 46H, 64H, 54H, 4CH, 44H, 64H, 44H, 00H

DB 40H, 30H, 0FH, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H

;-- 文字: 的 --

;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --

DB 00H, 0F8H, 8CH, 8BH, 88H, 0F8H, 40H, 30H, 8FH, 08H, 08H, 08H, 08H, 0F8H, 00H, 00H

DB 00H, 7FH, 10H, 10H, 10H, 3FH, 00H, 00H, 00H, 03H, 26H, 40H, 20H, 1FH, 00H, 00H

;-- 文字: D --

;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --

DB 08H, 0F8H, 08H, 08H, 08H, 10H, 0E0H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H

DB 20H, 3FH, 20H, 20H, 20H, 10H, 0FH, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H

;-- 文字: J --

;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --

DB 00H, 00H, 08H, 08H, 0F8H, 08H, 08H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H

DB 0C0H, 80H, 80H, 80H, 7FH, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H

;-- 文字: 仪 --

;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --

DB 40H, 20H, 0F0H, 0CH, 03H, 00H, 38H, 0C0H, 01H, 0EH, 04H, 0E0H, 1CH, 00H, 00H, 00H

DB 00H, 00H, 0FFH, 00H, 40H, 40H, 20H, 10H, 0BH, 04H, 0BH, 10H, 20H, 60H, 20H, 00H

;-- 文字: 器 --

;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --

DB 40H, 40H, 4FH, 49H, 49H, 0C9H, 0CFH, 70H, 0C0H, 0CFH, 49H, 59H, 69H, 4FH, 00H, 00H

DB 02H, 02H, 7EH, 45H, 45H, 44H, 7CH, 00H, 7CH, 44H, 45H, 45H, 7EH, 06H, 02H, 00H

;-- 文字: ! --

;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --

DB 00H, 00H, 00H, 0F0H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H

DB 00H, 00H, 00H, 5FH, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H

;-- 文字: ! --

;-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --

DB 00H, 00H, 00H, 0F0H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H

DB 00H, 00H, 00H, 5FH, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H

CODE ENDS

END START

## 实验十九 8237 DMA 传送实验

### 一、实验目的

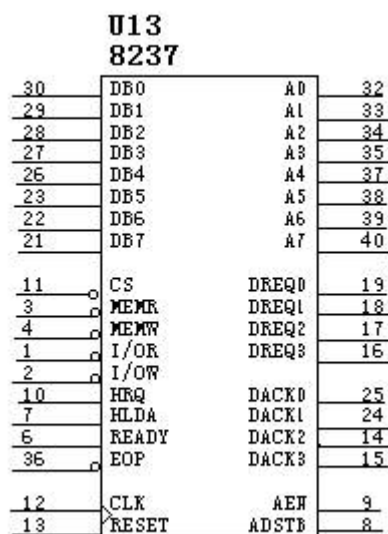
- (1) 掌握 DMA 方式的工作原理和 8237DMA 控制器的编程使用方法。
- (2) 掌握如何在实验系统环境下，使用 DMA 的 CH0 进行数据传送。

### 二、实验内容

利用本实验系统提供的 8237A-5 DMA 控制的 CH0，实现 DMA 的内存到内存的读、写传送，采用 DMA 软件请求，把内存中 3000H~4FFFH 单元内容传送到 5000H~6FFFH 单元中。

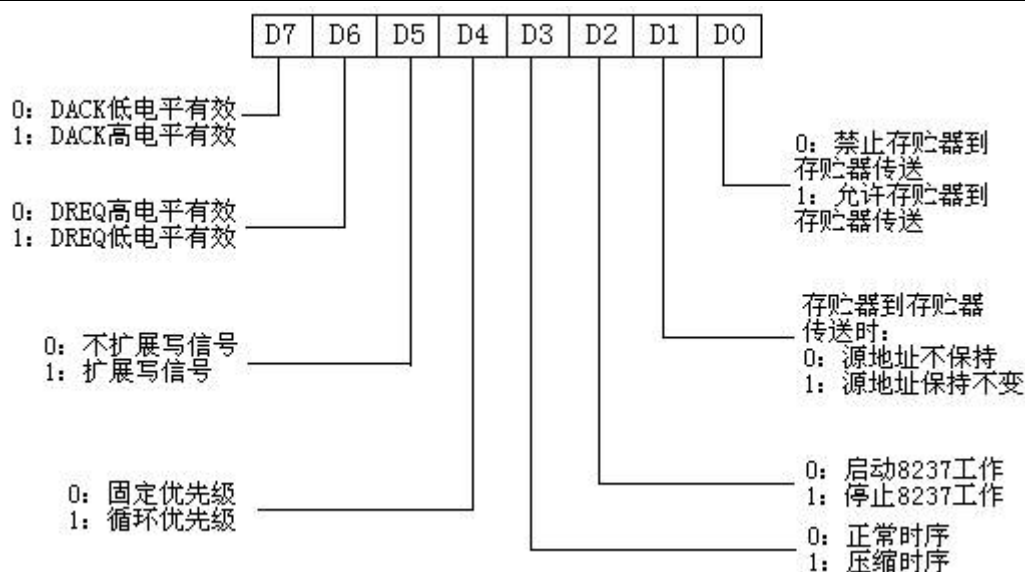
### 三、编程指南

#### 1、8237A 引脚图

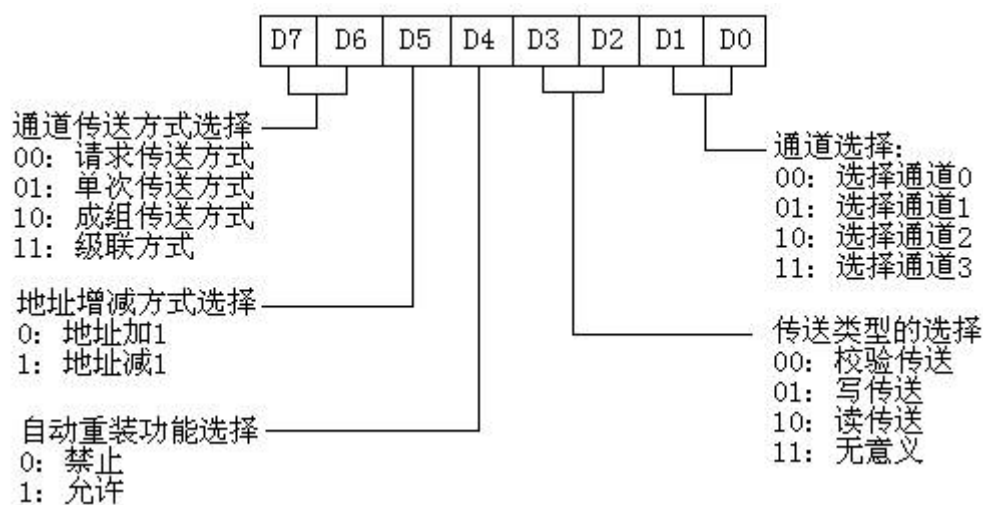


#### 2、8237A 的内部寄存器格式

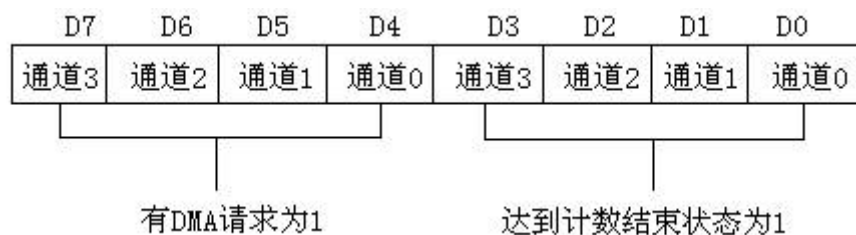
##### (1)8237A 控制寄存器格式



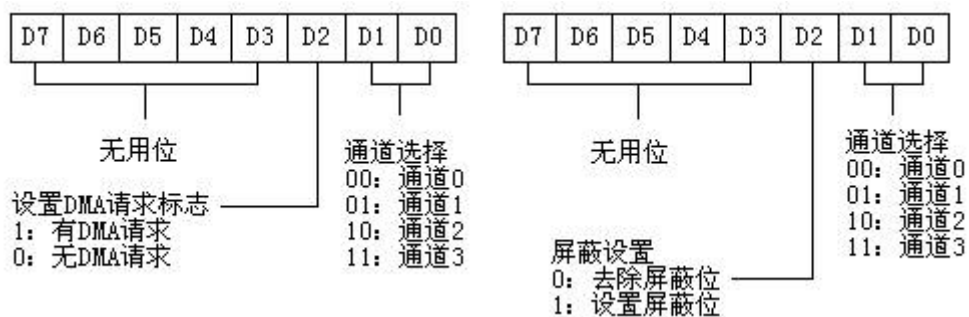
## (2) 8237A 模式寄存器格式



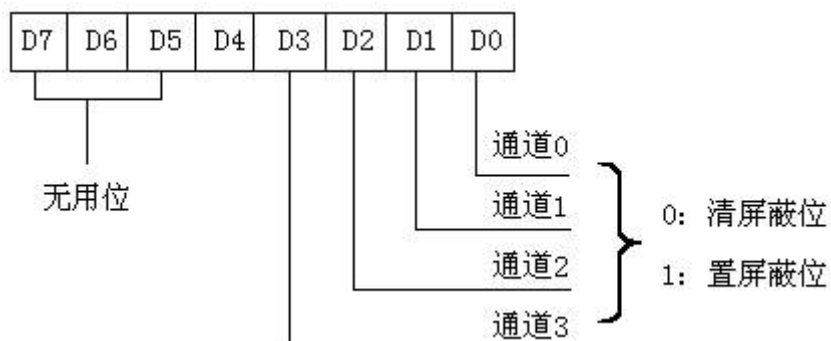
## (3) 8237A 状态寄存器格式



## (4) 请求寄存器和屏蔽寄存器格式



## (5) 多通道屏蔽寄存器格式



## (6) 8237A 端口地址

I/O口地址	寄存器	
	读 (IOR) 有效	写 (IOW) 有效
F000H	CH0当前地址寄存器	CH0基址与当前地址寄存器
F001H	CH0当前字节计数器	CH0基字节与当前字节计数器
F002H	CH1当前地址寄存器	CH1基址与当前地址寄存器
F003H	CH1当前字节计数器	CH1基字节与当前字节计数器
F004H	CH2当前地址寄存器	CH2基址与当前地址寄存器
F005H	CH2当前字节计数器	CH2基字节与当前字节计数器
F006H	CH3当前地址寄存器	CH3基址与当前地址寄存器
F007H	CH3当前字节计数器	CH3基字节与当前字节计数器
F008H	状态寄存器	控制寄存器
F009H	_____	请求寄存器
F00AH	_____	写屏蔽寄存器
F00BH	_____	模式寄存器
F00CH	_____	清除先/后触发器
F00DH	暂存器	发复位命令
F00EH	_____	清屏蔽寄存器
F00FH	_____	写多通道屏蔽寄存器

### 3、8237A 的初始化编程

(1)命令字写入控制寄存器

(2)方式字写入模式寄存器

(3)屏蔽字写入屏蔽寄存器

(4)写先/后触发器

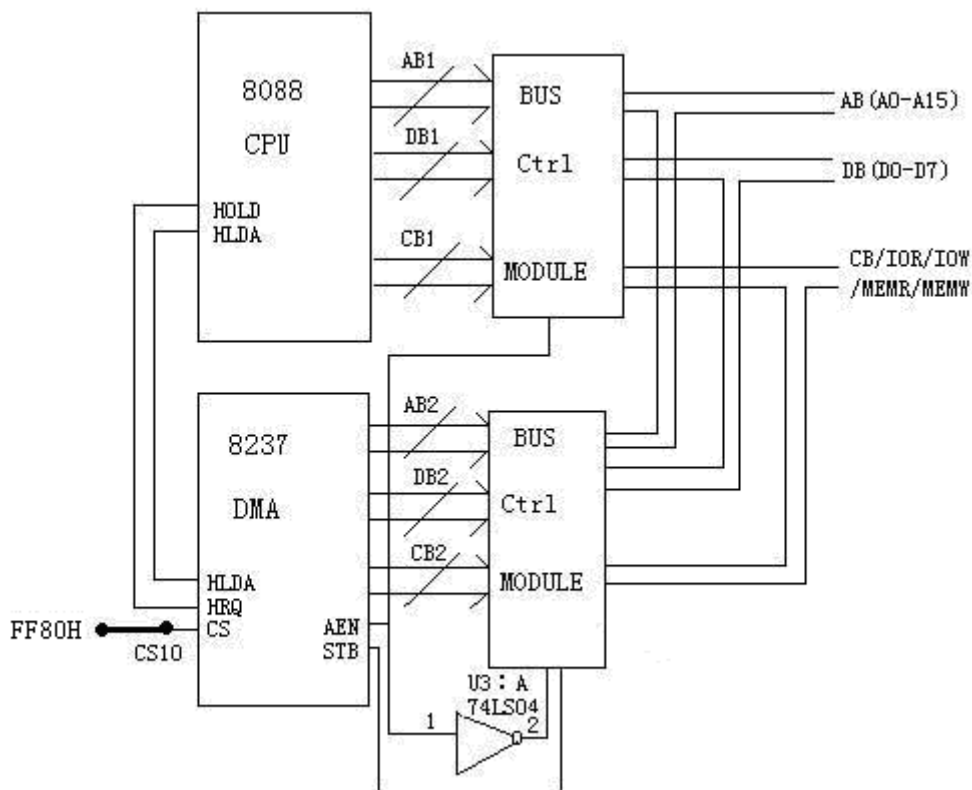
(5)写入基地址和当前地址寄存器以及基字节和当前字节计数器。

(6)写入请求寄存器

如果采用软件 DMA 请求，在适当的时候令通道的请求触发器置 1。



## 四、实验接线图



## 五、实验步骤

连CS10到FF80H,在系统内存3000H~4FFFH单元中填充10,11,12,13……,运行程序8237·ASM或在实验系统处于P态下,按SCAL键,然后在系统内存3000H~4FFFH单元中填充10,11,12,13……,输入2860,按EXEC键即可,当系统显示“8237——good”表示DMA传送结束,按RST钮,返回P态,用内存读写命令检查5000H~6FFFH单元内容是否和3000H~4FFFH单元内容相一致,验证其正确性。

## 六、实验程序清单

```
;8237.ASM,MOV:RAM FROM 3000H TO 5000H ;BYTE-MOV
CODE      SEGMENT
ASSUME    CS:CODE
PCTL      EQU    0FF23H          ;控制口
PA        EQU    0FF20H          ;字位口
PB        EQU    0FF21H          ;字形口
```

```

PC      EQU    0FF22H          ;键入口
CLEARF  EQU    0FF8CH
CH0A    EQU    0FF80h
CH1A    EQU    0FF82H
CH1C    EQU    0FF83H
MODE    EQU    0FF8BH
CMMD    EQU    0FF88H
MASKS   EQU    0FF8FH
REQ      EQU    0FF89H
STATUS  EQU    0FF88H
rst      equ    0FF8dh
LATCH   EQU    9000H
ORG      2860H          ;FORM EPROM FILE (.COM), SET ORG=0100H
START:   JMP START0
BUF      DB    ?, ?, ?, ?, ?, ?

data1:
db
0c0h, 0f9h, 0a4h, 0b0h, 99h, 92h, 82h, 0f8h, 80h, 90h, 88h, 83h, 0
c6h, 0a1h
db 86h, 8eh, 0ffh, 0ch, 89h, 0deh, 0c7h, 8ch, 0f3h, 0bfh, 8FH
START0:  MOV SI, 3000H
MOV DI, 5000H
MOV CX, 1FFFH
MOV AL, 00
MOV DX, LATCH
OUT DX, AL
NOP
NOP
mov dx, rst
out dx, al
MOV DX, CLEARF
OUT DX, AL
MOV AL, 0FH          ;MASK_CH_0-3
MOV DX, MASKS
OUT DX, AL
NOP
NOP
MOV DX, CH0A          ;L-SAD
MOV AX, SI

```

```

OUT DX, AL
MOV AL, AH
OUT DX, AL
NOP
NOP
MOV DX, CH1A                ;L-DAD
MOV AX, DI
OUT DX, AL
MOV AL, AH
OUT DX, AL
NOP
NOP
MOV AX, CX                  ;COUNT
MOV DX, CH1C
OUT DX, AL
MOV AL, AH
OUT DX, AL
MOV AL, 48H                 ;P_MODE
MOV DX, MODE
OUT DX, AL
MOV AL, 45H
OUT DX, AL
MOV AL, 01H                 ;ON-8237
MOV DX, CMMD
OUT DX, AL
L1:  MOV AL, 0EH             ;UNMASK_CH_0
MOV DX, MASKS
OUT DX, AL
MOV AL, 04H                 ;START_DMA_TRANSFER
MOV DX, REQ
OUT DX, AL
MOV DX, STATUS
NOP
NOP
WAIT1: IN AL, DX
TEST AL, 03H
JZ WAIT1
MOV DX, CH1C
IN AL, DX

```

```

        MOV AH, AL
        IN AL, DX
        CMP AX, 0FFFFH
        JNZ L1
        CALL BUF1
        MOV CX, 0080H
L2:     PUSH CX
        CALL DISP
        POP CX
        LOOP L2
        CALL BUF2
L3:     CALL DISP
        JMP L3
;-----
CMP1:   MOV DX, CH0A
        CALL CMP10
        MOV SI, AX
        MOV DX, CH1A
        CALL CMP10
        MOV DI, AX
        MOV AX, [SI]
        CMP [DI], AX
        RET
CMP10:  IN AL, DX
        MOV BL, AL
        IN AL, DX
        MOV AH, AL
        MOV AL, BL
        DEC AX
        RET
ERR:    CALL BUF3
ERR0:   CALL DISP
        JMP ERR0
;-----
DISP:   MOV AL, OFFH           ;00H
        MOV DX, PA
        OUT DX, AL
        MOV CL, 0DFH          ;20H  ;显示子程序 ,5ms
        MOV BX, OFFSET BUF

```

---

```
DIS1:    MOV AL, [BX]
          MOV AH, 00H
          PUSH BX
          MOV BX, OFFSET DATA1
          ADD BX, AX
          MOV AL, [BX]
          POP BX
          MOV DX, PB
          OUT DX, AL
          MOV AL, CL
          MOV DX, PA
          OUT DX, AL
          PUSH CX
DIS2:     MOV CX, 00A0H
DELAY:   LOOP DELAY
          POP CX
          CMP CL, 0FEH                ;01H
          JZ LX1
          INC BX
          ROR CL, 1                   ;SHR CL, 1
          JMP DIS1
LX1:     MOV AL, 0FFH
          MOV DX, PB
          OUT DX, AL
          RET
BUF1:    MOV BUF, 08H
          MOV BUF+1, 02H
          MOV BUF+2, 03H
          MOV BUF+3, 07H
          MOV BUF+4, 17H
          MOV BUF+5, 17H
          RET
;-----
BUF2:    MOV BUF, 09H
          MOV BUF+1, 00H
          MOV BUF+2, 00H
          MOV BUF+3, 0DH
          MOV BUF+4, 10H
          MOV BUF+5, 10H
```

```
                RET
BUF3:          MOV BUF, 08H
                MOV BUF+1, 02H
                MOV BUF+2, 03H
                MOV BUF+3, 07H
                MOV BUF+4, 18H
                MOV BUF+5, 18H
                RET
CODE ENDS
END    START
```

## 实验二十 8250 串口实验

### 一、实验目的

- (1) 系统扩展以 8250 为核心的可编程串行异步通讯接口芯片。
- (2) 对扩展的串行通讯接口实现自发自收。

### 二、实验内容

将寄存器 AH 的内容从 10H 开始通过 8250 发送、接收，每次收发后 AH 的内容自动增 1 直到 FFH 为止，同时将每次接收到的数据，依次写到内存 4000H~40EFH 单元中，实现自发自收。

### 三、编程指南

#### 1、通讯协议

1 个起始位，8 位数据位，1 个停止位，波特率：4800。

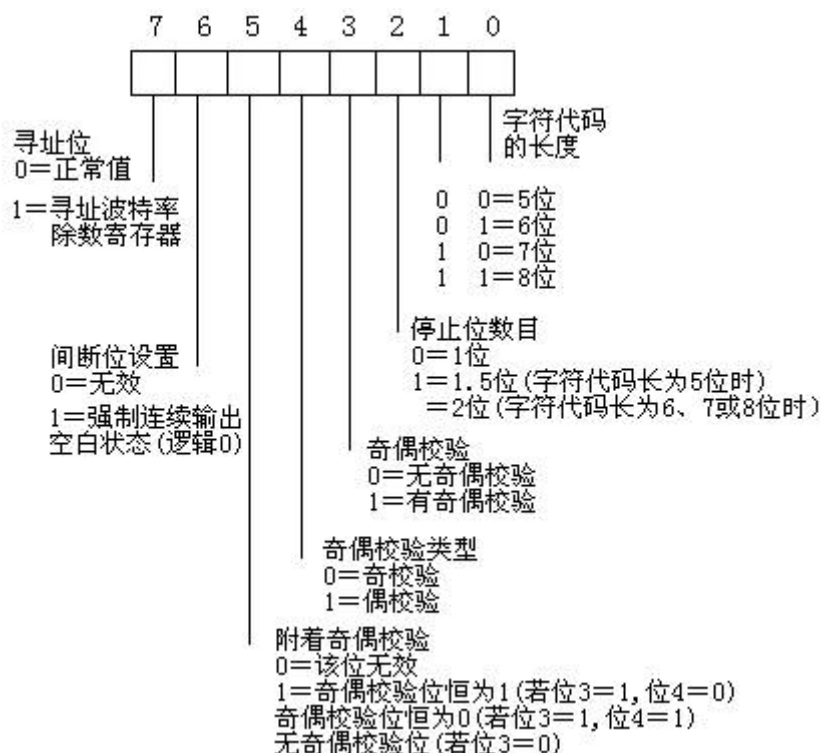
#### 2、8250 的内部寄存器的端口地址

\* 通讯线控制寄存器第 7 位 (DLAB)=0 , \*\* DLAB=1

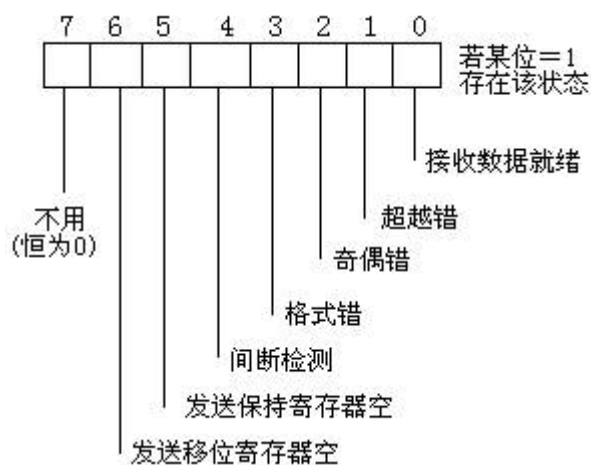
I/O端口	IN/OUT	寄存器名称
FF80H*	OUT	数据发送保持寄存器
FF80H*	IN	数据接收缓冲寄存器
FF80H**	OUT	除数锁存器（低位）
FF80H**	OUT	除数锁存器（高位）
FF80H*	OUT	中断允许寄存器
FF80H	IN	中断识别寄存器
FF80H	OUT	通讯线控制寄存器
FF80H	OUT	MODEM控制寄存器
FF80H	IN	通讯线状态寄存器
FF80H	IN	MODEM状态寄存器

### 3、8250 的内部寄存器格式

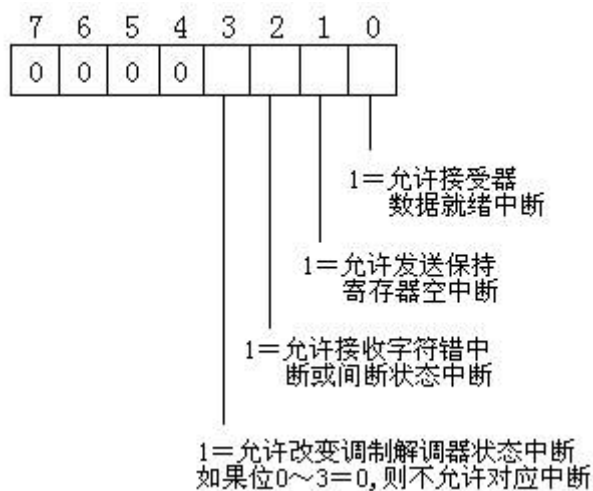
#### (1) 通讯线控制寄存器格式



## (2) 通讯线状态寄存器格式

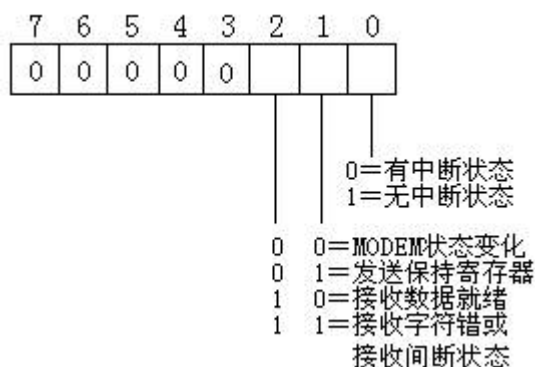


## (3) 断允许寄存器格式

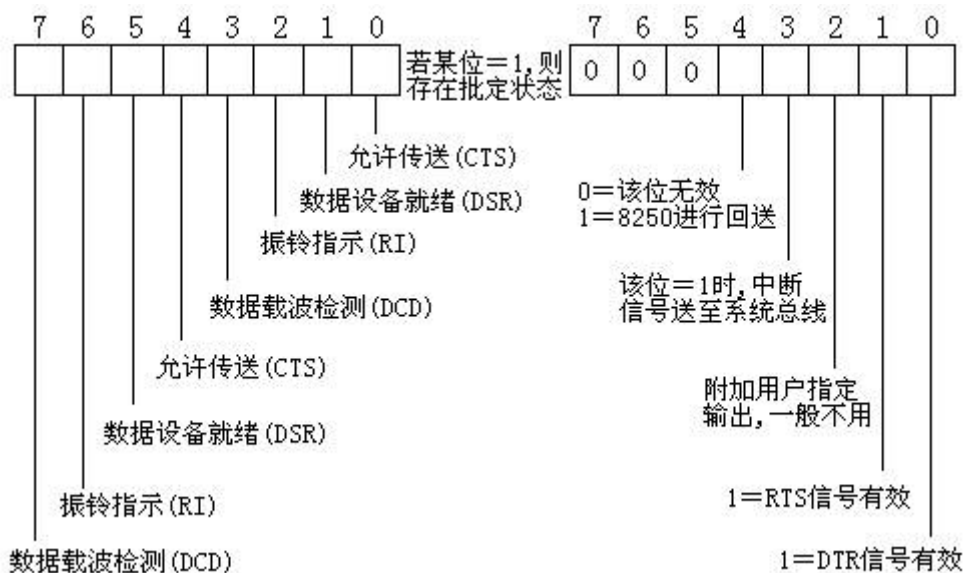


## (4) 中断识别寄存器格式





#### (5) MODEM 控制寄存器与 MODEM 状态寄存器格式



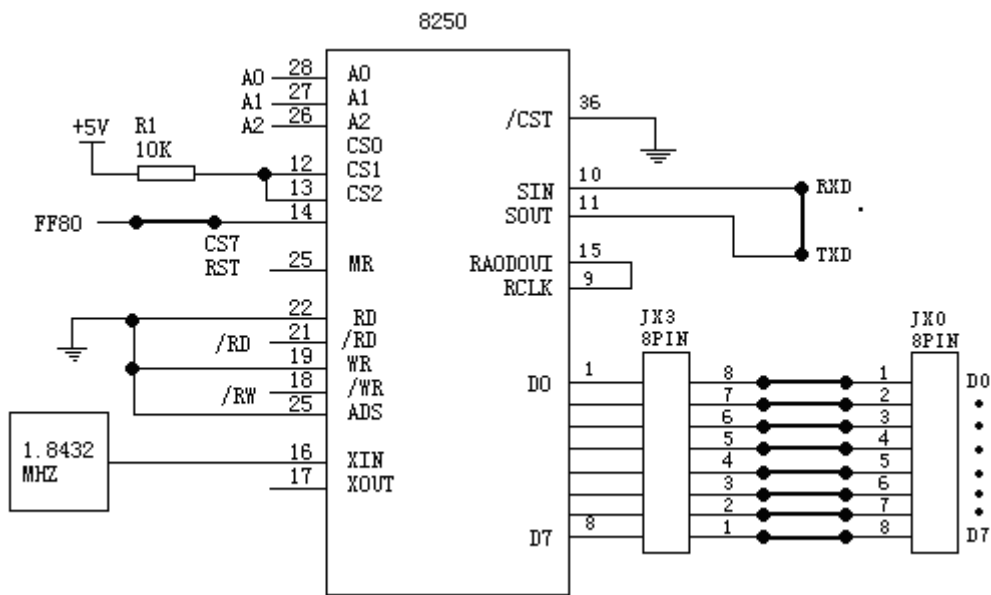
#### 4、波特率除数锁存器的值与波特率的对应关系 (16 进制数)

波特率	波特率除数锁存器的值	
	高8位 (H)	低8位 (M)
300	01	80
1200	00	60
2400	00	30
4800	00	18
9600	00	0C

#### 5、8250 的初始化编程

- (1) 设置波特率
- (2) 设置通讯数据格式
- (3) 设置操作方式
- (4) 设置中断允许寄存器

四、实验接线图



五、实验步骤

用扁平线连 JX0 (BUS) 到 JX3 (D0~D7)，连 FF80H 孔到 CS7，连 TXD 到 RXD (8250 实验区)。运行程序 8250 • ASM 或在实验系统处于 P 态下，按 SCAL 键，输入 29A0，按 EXEC 即可。当系统显示 “8250——good” 表示自发自收结束，按 RST 键，系统返回 P 态，用内存读写命令检查 4000H~40EFH 内容是否为 10H~FFH，验证其正确性。

六、实验程序清单

```
CODE          SEGMENT                                ;H8250. ASM
ASSUME        CS: CODE                                ;H8250. ASM
DATA EQU      0ff80H                                  ;BTS-LSB
MSB EQU       0ff81H
```

```

        LINE    EQU    0ff83H
        LSTAT   EQU    0ff85H
        PA      EQU    0FF20H      ;字位口
        PB      EQU    0FF21H      ;字形口
        PC      EQU    0FF22H      ;键入口
        ORG 29A0H
START:   JMP START0
BUF      DB  ?, ?, ?, ?, ?, ?
data1:

        db
        0c0h, 0f9h, 0a4h, 0b0h, 99h, 92h, 82h, 0f8h, 80h, 90h, 88h, 83h, 0
        c6h, 0a1h
        db 86h, 8eh, 0ffh, 0ch, 89h, 0deh, 0c7h, 8ch, 0f3h, 0bfh, 8FH
START0:  MOV AL, 80H                ;DLAB=1
        MOV DX, LINE
        OUT DX, AL
        MOV AL, 18H                ;BTS=4800
        MOV DX, DATA              ;ff80H
        OUT DX, AL
        MOV DX, MSB
        MOV AL, 00
        OUT DX, AL

;-----
        MOV AL, 03H                ;8- BIT , 1-STOP
        MOV DX, LINE
        OUT DX, AL

;-----
        MOV AL, 00                ;NO-INT
        MOV DX, MSB                ;8001H
        OUT DX, AL
        MOV AH, 10H
        MOV BX, 4000H
MAIN:    CALL TXD
        CALL RCV
        MOV [BX], AL
        INC BX
        INC AH
        CMP AH, 00H
        JNZ MAIN

```

```

        CALL BUF1
        MOV CX, 00FFH
S3:      PUSH CX
        CALL DISP
        POP CX
        LOOP S3
        CALL BUF3
S1:      CALL DISP
        JMP S1
TXD:     MOV DX, LSTAT
WAIT1:   IN AL, DX
        TEST AL, 20H
        JZ WAIT1
        MOV AL, AH
        MOV DX, DATA
        OUT DX, AL
        RET
RCV:     MOV DX, LSTAT
WAIT2:   IN AL, DX
        TEST AL, 01H
        JZ WAIT2
        TEST AL, 0EH
        JNZ ERR
        MOV DX, DATA
        IN AL, DX
        RET
ERR:     CALL BUF2
S2:      CALL DISP
        JMP S2
;-----
DISP:    MOV AL, 0FFH                ;00H
        MOV DX, PA
        OUT DX, AL
        MOV CL, 0DFH                ;20H ;显示子程序 , 5ms
        MOV BX, OFFSET BUF
DIS1:    MOV AL, [BX]
        MOV AH, 00H
        PUSH BX
        MOV BX, OFFSET DATA1

```

---

```

        ADD BX, AX
        MOV AL, [BX]
        POP BX
        MOV DX, PB
        OUT DX, AL
        MOV AL, CL
        MOV DX, PA
        OUT DX, AL
        PUSH CX
DIS2:   MOV CX, 00A0H
DELAY:  LOOP DELAY
        POP CX
        CMP CL, 0FEH                ;01H
        JZ LX1
        INC BX
        ROR CL, 1                    ;SHR CL, 1
        JMP DIS1
LX1:    MOV AL, 0FFH
        MOV DX, PB
        OUT DX, AL
        RET
BUF1:   MOV BUF, 08H
        MOV BUF+1, 02H
        MOV BUF+2, 05H
        MOV BUF+3, 00H
        MOV BUF+4, 17H
        MOV BUF+5, 17H
        RET
;-----
BUF2:   MOV BUF, 08H
        MOV BUF+1, 02H
        MOV BUF+2, 05H
        MOV BUF+3, 00H
        MOV BUF+4, 0EH
        MOV BUF+5, 18H
        RET
BUF3:   MOV BUF, 09H
        MOV BUF+1, 00H
        MOV BUF+2, 00H

```

```
MOV BUF+3, 0DH
MOV BUF+4, 10H
MOV BUF+5, 10H
RET

CODE ENDS
END START
```

实验二十一 8279 键盘显示实验

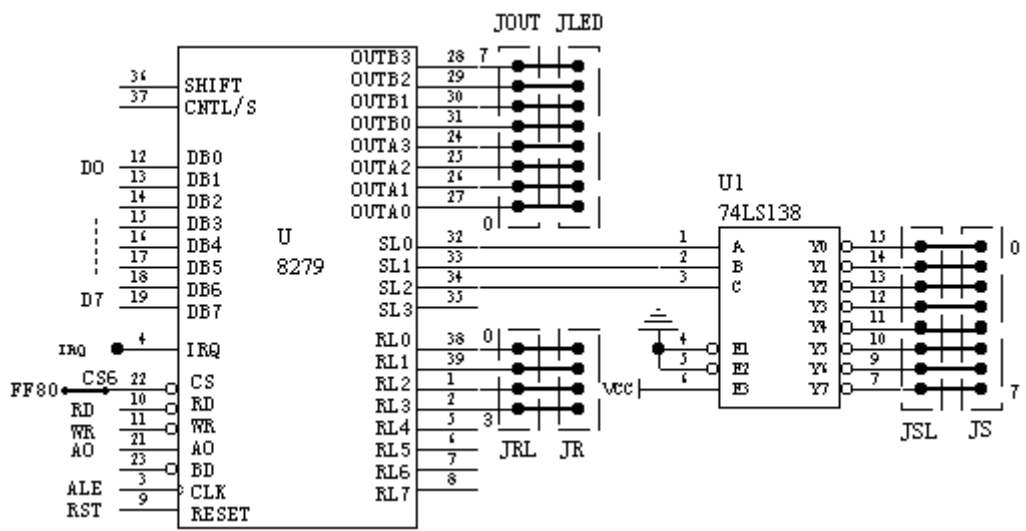
一、实验目的

掌握 8088 微机系统中，扩展 8279 键盘显示 接口的方法。

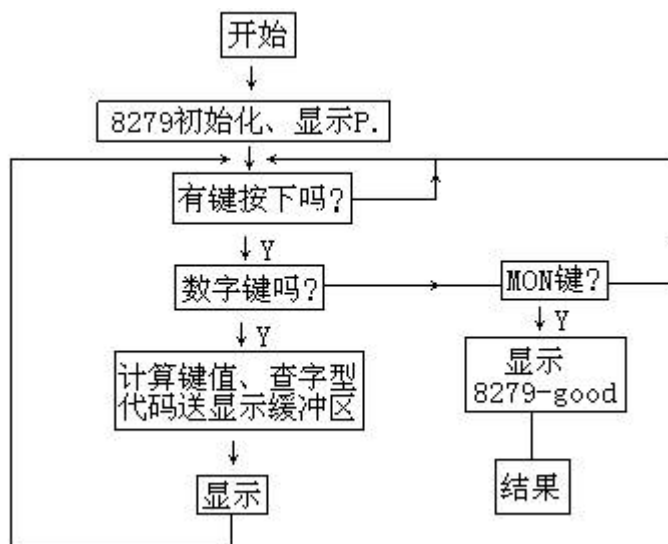
二、实验内容

用 8279 接口芯片来控制实验系统键盘显示，按下数字键，在数码管上应显示相应的数字，按 MON 键，显示 8279—good，按其它功能键不响应。

三、实验接线图



四、实验程序框图



## 五、实验步骤

①连 CS6→FF80H 孔, JSL→JS, JRL→JR, JOUT→JLED, 开关 JK 置外接 (实验完后置系统)。

②建议联 PC 机进行调试, 打开源文件 8279 • ASM (EP2: 2A90H), 然后编译、装载, 点击运行命令, 数码管上显示 P, 按数字键, 数码管上显示相应数字, 按 MON 键, 数码管显示 8279-good, 按 RST 键, 退出运行。

## 六、实验程序清单

```

CODE          SEGMENT                ;H8279. ASM
ASSUME        CS: CODE
D8279        EQU        0FF80H
C8279        EQU        0FF81H
ORG 2A90H
JMP START
KH           DB ?                    ;KEY HAO
ZW           DB ?
ZX           DB ?
START:       MOV DX, C8279            ;8001H      ;WR-->MODE
             MOV AL, 00H              ;8BIT L-IN 2KEY
             OUT DX, AL
             MOV AL, 32H              ;FENG PIN
             OUT DX, AL

```

```

        MOV AL, 0DFH                ;CLR BUF
        OUT DX, AL
WAIT:    IN AL, DX                    ;clr buf end ?
        TEST AL, 80H
        JNZ WAIT
        MOV AL, 85H ;WR DISRAM ADR Y5 (PA5);100 0 0101-->com no-inc y5
        OUT DX, AL
        MOV DX, D8279                ;ff80H      ;WR DATA 'P'
        MOV AL, 0C8H
        OUT DX, AL
START0:  MOV ZW, 85H                  ;BEST H BIT
NEXT:    MOV KH, 00H                  ;KEY HAO 0..FH, 10..13H
        MOV DX, C8279                ;8001H      ;RD STATUS KEY ?
NOKEY:   IN AL, DX
        AND AL, 07H
        CMP AL, 00H
        JZ NOKEY
        MOV DX, D8279                ;ff80H      ;rd key zhi
        IN AL, DX
        MOV AH, AL                    ;SAVE KEY ZHI
        MOV BX, OFFSET TABK
CMPK:    MOV AL, KH
        XLAT
        CMP AH, AL
        JZ KEY
        INC KH
        CMP KH, 14H
        JNC KEY0
        JMP CMPK
KEY:     CMP KH, 10H
        JNC FUN
        CALL DIS
        DEC ZW
KEY0:    CMP ZW, 7FH
        JNZ NEXT
        JMP START0
FUN:     CMP KH, 13H
        JNZ KEY0
        MOV ZW, 85H

```



```
MOV KH, 08H
CALL DIS
MOV ZW, 84H
MOV KH, 02H
CALL DIS
MOV ZW, 83H
MOV KH, 07H
CALL DIS
MOV ZW, 82H
MOV KH, 09H
CALL DIS
MOV ZW, 81H
MOV KH, 11H
CALL DIS
MOV ZW, 80H
MOV KH, 11H
CALL DIS
```

;-----

```
CALL DELY
MOV ZW, 85H
MOV KH, 09H
CALL DIS
MOV ZW, 84H
MOV KH, 00H
CALL DIS
MOV ZW, 83H
MOV KH, 00H
CALL DIS
MOV ZW, 82H
MOV KH, 0DH
CALL DIS
MOV ZW, 81H
MOV KH, 10H
CALL DIS
MOV ZW, 80H
MOV KH, 10H
CALL DIS
JMP $
```

DIS: MOV DX, C8279

;WR BIT, BIT 85H, 84H, .. 80H

```

MOV AL, ZW
OUT DX, AL
MOV AL, KH                      ;WR CODE
MOV BX, OFFSET TABC
XLAT
MOV DX, D8279                   ;ff80H
OUT DX, AL
RET
DELY:  MOV BX, 00FFH
DELY1:  DEC BX
        CMP BX, 0000
        JZ  DELY2
        MOV CX, 04FFH
        LOOP $
        JMP DELY1
DELY2:  RET
TABK:   DB 0C9H, 0C1H, 0D1H, 0E1H, 0C8H, 0D8H, 0E8H, 0C0H, 0D0H
        DB 0E0H, 0F0H, 0F8H, 0F1H, 0F9H, 0E9H, 0D9H
        DB 0F2H, 0FAH, 0F3H, 0FBH
TABC:   DB 0CH, 9FH, 4AH, 0BH, 99H, 29H, 28H, 8FH, 08H, 09H, 88H
        DB 38H, 6CH, 1AH, 68H, 0E8H, 0FFH, 0FBH

CODE ENDS
END START

```

## 实验二十二 温度控制实验

### 一、实验目的

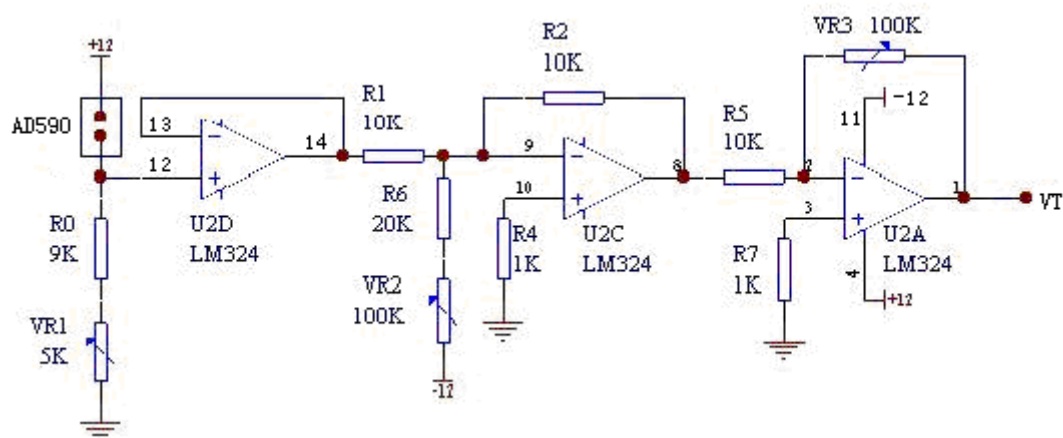
1、了解集成温度传感器 AD590 的工作原理和应用，熟悉小信号放大器的工作原理和零点、增益的调整方法。

2、了解微机对温度采样控制的基本方法。

### 二、实验内容

设定单点温度控制点为 30° C，当小于 30° C 时，L1 指示灯亮，模拟电热器加热，当大于 30° C 时 L1 灭，L2 亮，模拟关掉电热器。

### 三、实验接线图



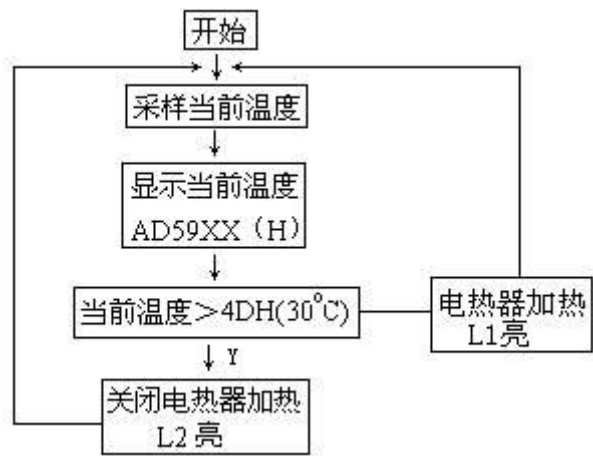
### 四、编程指南

AD590 是电流输出型温度传感器，工作电压 4~30V，检测温度范围  $-55^{\circ}\text{C}$  ~  $+150^{\circ}\text{C}$ ，它有非常好的线性，灵敏度为  $1\mu\text{A/K}$ 。AD590 传感器输出信号通过 10K 电阻取出的电压信号，经零点调整，小信号放大后，输出的电压信号 VT，供 ADC0809 采样用。

AD590 温度与有关参数一览表

温度	AD590 电流	经10K $\Omega$ 电压	零点调整后 电压	放大5倍后 电压VT	ADC 数字量
0 $^{\circ}\text{C}$	273.2 $\mu\text{A}$	2.732V	0.0V	0V	00H
10 $^{\circ}\text{C}$	283.2 $\mu\text{A}$	2.832V	0.1V	0.5V	1AH
20 $^{\circ}\text{C}$	293.2 $\mu\text{A}$	2.932V	0.2V	1V	33H
30 $^{\circ}\text{C}$	303.2 $\mu\text{A}$	3.032V	0.3V	1.5V	4DH
40 $^{\circ}\text{C}$	313.2 $\mu\text{A}$	3.132V	0.4V	2V	66H
50 $^{\circ}\text{C}$	323.2 $\mu\text{A}$	3.232V	0.5V	2.5V	80H
60 $^{\circ}\text{C}$	333.2 $\mu\text{A}$	3.332V	0.6V	3V	99H
70 $^{\circ}\text{C}$	343.2 $\mu\text{A}$	3.432V	0.7V	3.5V	B3H
80 $^{\circ}\text{C}$	353.2 $\mu\text{A}$	3.532V	0.8V	4V	CCH
90 $^{\circ}\text{C}$	363.2 $\mu\text{A}$	3.632V	0.9V	4.5V	E6H
100 $^{\circ}\text{C}$	373.2 $\mu\text{A}$	3.732V	1V	5V	FFH

五、实验程序框图



六、实验步骤

- ①备一温度计，根据室温调整 AD590 的输出电压，假如室温为 25° C，则调整 VR1 使 LM324（下）14 脚输出为 2.982V，再调 VR2，使 LM324（下）8 脚输出为－0.25V，再调 VR3，使 VT 为 1.25。
- ②连 JX6→JX17，CS4→F880H，IN0→VT，PA0→L1，PA1→L2，8MHZ→T。
- ③在 P 态下，按 SCAL 键，输入 2C20，接 EXEC 键，显示“AD59XX”，“XX”为当前温度对应的十六进制值（对应关系见上表）。
- ④用手温摸测 AD590 的外壳，观察 L1、L2 灯的亮灭情况。

七、实验程序清单

CODE	SEGMENT		;T. ASM
ASSUME	CS: CODE		
	ADPORT EQU	0ff80h	
	PA EQU	0FF20H	;字位口
	PB EQU	0FF21H	;字形口
	PC EQU	0FF22H	;键入口
	P2CTL EQU	0FF2BH	
	P2A EQU	0FF28H	
	P2B EQU	0FF29H	
	P2C EQU	0FF2AH	
	ORG 2C20H		

```

START:      JMP START0
BUF         DB ?, ?, ?, ?, ?, ?
data1:

            db0c0h, 0f9h, 0a4h, 0b0h, 99h, 92h, 82h, 0f8h, 80h, 90h, 88h,
            83h, 0c6h, 0a1h
            db 86h, 8eh, 0ffh, 0ch, 89h, 0deh, 0c7h, 8ch, 0f3h, 0bfh, 8FH

START0:     CALL BUF1
            MOV DX, P2CTL
            MOV AL, 80H
            OUT DX, AL

ADCON:      MOV AX, 00
            MOV DX, ADPORT
            OUT DX, AL
            MOV CX, 0500H

DELAY:      LOOP DELAY
            MOV DX, ADPORT
            IN AL, DX
            PUSH AX
            cmp al, 4dh
            jnc ct1
            MOV AL, 0FEH

CON:        MOV DX, P2A                ;pa0-->11
            OUT DX, AL
            POP AX
            CALL CONVERS
            CALL DISP
            JMP ADCON

CT1:        MOV AL, 0FDH                ;pa1-->12
            JMP CON

CONVERS:    MOV AH, AL
            AND AL, 0FH
            MOV BX, OFFSET BUF
            MOV [BX+5], AL
            MOV AL, AH
            AND AL, 0F0H
            MOV CL, 04H
            SHR AL, CL
            MOV [BX+4], AL
            RET

```

---

```

DISP:      MOV AL, 0FFH          ;00H
            MOV DX, PA
            OUT DX, AL
            MOV CL, 0DFH          ;20H ;显示子程序 , 5ms
            MOV BX, OFFSET BUF
DIS1:      MOV AL, [BX]
            MOV AH, 00H
            PUSH BX
            MOV BX, OFFSET DATA1
            ADD BX, AX
            MOV AL, [BX]
            POP BX
            MOV DX, PB
            OUT DX, AL
            MOV AL, CL
            MOV DX, PA
            OUT DX, AL
            PUSH CX
DIS2:      MOV CX, 00A0H
            LOOP $
            POP CX
            CMP CL, 0FEH          ;01H
            JZ LX1
            INC BX
            ROR CL, 1              ;SHR CL, 1
            JMP DIS1
LX1:      MOV AL, 0FFH
            MOV DX, PB
            OUT DX, AL
            RET
BUF1:      MOV BUF, 0AH
            MOV BUF+1, 0DH
            MOV BUF+2, 05H
            MOV BUF+3, 09H
            MOV BUF+4, 00H
            MOV BUF+5, 00H
            RET
CODE ENDS
END START

```

## 实验二十三 压力测量实验

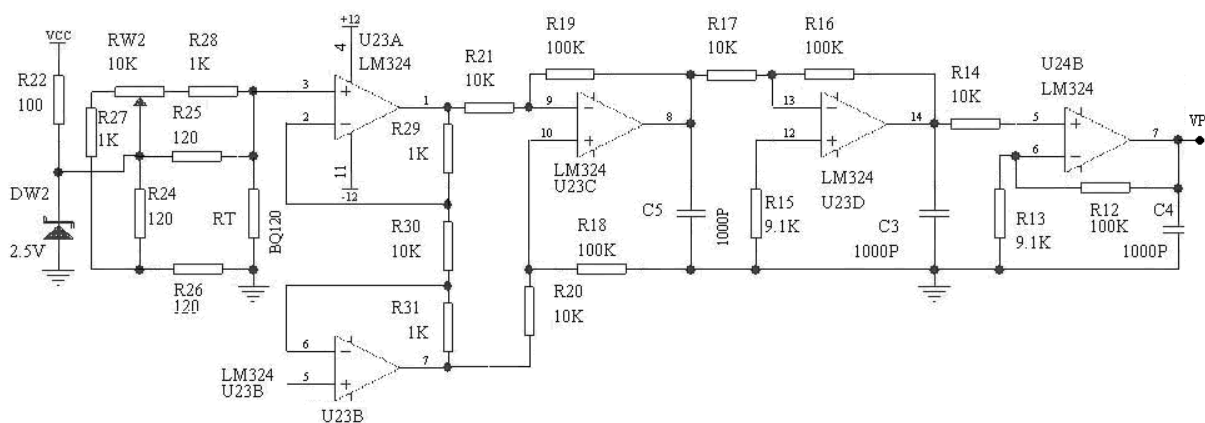
### 一、实验目的

了解压力传感器的工作原理以及调整方法。

### 二、实验内容

压力测量方法。

### 三、实验接线图



### 四、编程指南

在实验板的中间部分有一金属片，它是组合型压力传感器。压力传感器的输出经测量电路和放大电路放大后产生标准电压信号，VP 就是压力信号→电压信号（0~5V）的输出端点。压力传感器表面金属片在外力的作用下发生变形，导致电阻应变片电阻值的变化，用力越大，电阻变化越大，上图为应变片电桥测量电路，由应变片电阻 R1 和另外三个电阻 R2、R3、R4 构成桥路，当电桥平衡时（电阻应变片未受力作用时）， $R_1=R_2=R_3=R_4=R$ ，此时电桥的输出  $U_0 \approx 0$ ，当应变片受力后，R1 发生变化，使  $R_1、R_3 \neq R_2、R_4$ ，电桥输出  $U_0 \neq 0$ ，并有：

$$U_0 = \pm 1/4 \cdot \Delta R/R \cdot U \approx K_0 \varepsilon / 4 \cdot U$$

该微弱的电压信号经 LM324 运算放大器放大到 0~5V，0~5V 对应 0~1020g，RW2 为调零电位器，当电桥平衡时，VP 输出应调到 0V，由 VP 端输出，作为 ADC0809 的模拟量输入信号。模拟量到数字量转换实验请参见《实验一》中有关 A/D 转换实验一。当压力变化时，数码管上显示的电压值作相应变化，学生根据上述工作原理，当被测压力是 0~1020g 时，先将压力分度值制成分度值表，由电压值的变化通过软件查表法查出对应力值，显示在数码管上。假设 0~5V 电压近似正比于 0~1020g。(调整 R12 的大小可改变 VP 输出增益的大小)。

## 附录一 实验程序目录

### 软件实验：

实验序号	实验名称	文件名	入口地址
实验一	二进制多位加法运算	S1.asm	2CA0H
实验二	二进制码转换为 BCD 码	S2.asm	2CE0H
实验三	BCD 码转换为二进制码	S3.asm	2D40H
实验四	十进制数的 BCD 码相减运算	S4.asm	2D80H
实验五	内存清零	S5.asm	2DC0H
实验六	数码显示	S6.asm	2DF0H
实验七	求最大值和最小值	S7.asm	2E70H
实验八	数据块移动	S8.asm	2EF0H
实验九	多分支程序	S9.asm	2F40H

### 硬件实验：

实验一	A/D 转换实验	H0809.asm	1000H
实验二	D/A 转换实验（一）	H0832-1.asm	10D0H



实验三	D/A 转换实验 (二)	H0832-2. asm	1110H
实验四	8255A 并行口实验 (一)	H8255-1. asm	11B0H
实验五	8255A 并行口实验 (二)	H8255-2. asm	11E0H
实验六	定时器/器	H8253. asm	1290H
实验七	8259 单级中断控制器实验	H8259. asm	12D0H
实验八	串行接口和应用(一) 串行发送(双机通讯)	H8251T. asm	13F0H
实验九	串行接口和应用(二) 串行接收(双机通讯)	H8251R. asm	1510H
实验十	小直流电机调速实验	DJ. asm	1590H
实验十一	步进电机控制	BJ. asm	1630H
实验十二	继电器控制	JDQ. asm	1810H
实验十三	存贮器读写实验	RAM. asm	1850H
实验十四	电子琴实验	DZQ. asm	18F0H
实验十五	简单 I/O 口扩展实验	IO. asm	1850H
实验十六	8251 可编程通讯接口和 PC 机通讯	PC8251. asm	1A00H
实验十七	LED16×16 点阵显示实验	LEDQD. asm	1B90H
实验十八	128×64LCD 液晶显示实验	LCDQD. asm	22E0H
实验十九	8237DMA 传送实验	H8237. asm	2860H
实验二十	8250 可编程异步通讯接口实验	H8250. asm	29A0H
实验二十一	8279 键盘显示实验	H8279. asm	2A90H
实验二十二	温度控制实验	T. asm	2C20H

## 附录二、88 部分实验接线汇总表

实验名称	入口地址	接线步骤
实验一(A/D 转换)	1000H	CS4→FF80H, JX6→JX17, IN0→AOUT1, 8MHZ→T
实验二、三(D/A 转换)	10D0H 1110H	CS5→FF80H, JX2→JX17
实验四(8255A)	11B0H	PA0~PA7→K1~K8, PB0~PB7→L1~L8
实验五(8255A 交通灯)	11E0H	PB4~PB7→L1, L4, L7, L10 PC0~PC7→L3, L6, L9, L12, L2, L5, L8, L11
实验六(定时/计数)	1290H	CLK0→T2, GATE0→+5V, 8MHZ→T
实验七(8259)	12D0H	INT→INTR, INTA→INTA, IR3→单脉冲输出端, CS→FF80H, JX4→JX17
实验八、九(串行接口)	13F0H 1510H	CLK1→1.8432MHZ, OUT1→T/RXC, GATE1→+5V, JX0→JX20, 1 号机, 2 号机 TXD、RXD 交叉相连, 两机共地
实验十(小直流电机)	1590H	CS5→FF80H, JX2→JX17, AOUT→DJ
实验十一(步进电机)	1630H	PA0~PA3→HA~HD
实验十二(继电器)	1810H	PA0→JIN, JK→L2, JB→L1, JZ→地
实验十三(存贮器)	1850H	无
实验十四(电子琴)	18F0H	CLK1→T2, OUT1→VIN, GATE1→PB0, 8MHZ→T, JP 向下短接。
实验十五(I/O)	19D0H	CS1→FF80H, CS2→FF90H, JX7→JX17 Y0~Y7→K1~K8, Q0~Q7→L1~L8
实验十六(8251 与 PC 机通信)	1A00H	T/RXC→OUT1, TXD→EX-TXD, RXD→EX-RXD, JX20→JX17, CLK1→1.8432MHZ, GATE1→+5V, CZ11 用户通信口→PC 机串口
实验十七(LED 点阵)	2DC0H	JLPA→JX9, JLPB→JX15, JHPC→JX16, JX7→JX17, CS2→FF80H, JQ→JHQ。
实验十八(12864LCD)	22E0H	JX9→JX12, JX16→JX14
实验十九(8237DMA)	2860H	CS10→FF80H
实验二十(8250 串口)	29A0H	CS7→FF80H, JX0→JX3, TXD→RXD
实验二十一(8279)	2A90H	CS6→FF80H, JRL→JR, JSL→JS, JOUT→JLED, JK 置外接, (实验完后置系统)
实验二十二(温度)	2C20H	CS4→FF80H, JX6→JX17, 8MHZ→T, IN0→VT, PA0→L1, PA1→L2。

## 附录三 DJ-8086K 字形字位表

字形表

代码	D7	D6	D5	D4	D3	D2	D1	D0	字形
H	H	G	f	e	d	c	b	a	
C0	1	1	0	0	0	0	0	0	0
F9	1	1	1	1	1	0	1	0	1
A4	1	0	1	0	0	1	0	0	2
B0	1	0	1	1	0	0	0	0	3
99	1	0	0	1	1	0	0	1	4
92	1	0	0	1	0	0	1	0	5
82	1	0	0	0	0	0	1	0	6
F8	1	1	1	1	1	0	0	0	7
80	1	0	0	0	0	0	0	0	8
90	1	0	0	1	0	0	0	0	9
88	1	0	0	0	1	0	0	0	A
83	1	0	0	0	0	0	1	1	B
C6	1	1	0	0	0	1	1	0	C
A1	1	0	1	0	0	0	0	1	D
86	1	0	0	0	0	1	1	0	E
8E	1	0	0	0	1	1	1	0	F
FF	1	1	1	1	1	1	1	1	
0C	0	0	0	0	1	1	0	0	P.
89	1	0	0	0	1	0	0	1	H
7F	0	1	1	1	1	1	1	1	.
BF	1	0	1	1	1	1	1	1	—

字位表

		LED5	LED4	LED3	LED2	LED1	LED0
		DF	EF	F7	FB	FD	FE

附录四 DJ-8086K 键值表

07	08	09	0A	10	16
04	05	06	0B	11	17
01	02	03	0C	18	1E
00	0F	0E	0D	19	1F