



Dais PRO 系列

达爱思教学实验系统

8086 微机原理与接口 实验教程

无锡华文默克仪器有限公司

<http://www.merke.com.cn>

目 录

第 1 章 系统综述.....	1
1.1 系统特性.....	1
1.2 资源分配及接口定义.....	2
1.3 端口地址的译码及选通定义.....	3
1.4 控制信号的名称及说明.....	4
1.5 接口的实验连接定义.....	5
1.6 限制性控制与接口连接.....	6
1.7 CPU 与 I/O 扩展接口.....	7
1.8 实验系统通用电路.....	8
1.9 实验环境的认识（重点阅读）.....	13
第 2 章 微机原理及其程序设计实验.....	14
实验一 系统认识实验.....	14
实验二 数制转换实验.....	20
1. 十六进制数转换为十进制数.....	20
2. 十进制数转换为十六进制数.....	21
实验三 码制转换实验.....	23
1. ASCII 码（数字符）转换为十六进制数.....	23
2. 十六进制数转换为 ASCII 码.....	24
3. ASCII 码（数字符）转换为十进制数.....	25
4. 十进制数转换为 ASCII 码.....	26
5. 十进制数的 ASCII 码转换为 BCD 码.....	27
6. 十进制 BCD 码转换为二进制数.....	28
实验四 运算类编程实验.....	30
1. 二进制双精度加法运算.....	30
2. 十进制的 BCD 码减法运算.....	32
3. 乘法运算.....	34
实验五 分支程序设计实验.....	36
实验六 循环程序设计实验.....	39
1. 计算 $S=1+2\times 3+3\times 4+4\times 5+\cdots+N(N+1)$	39
2. 求某数据区内负数的个数.....	40
实验七 排序程序设计实验.....	42
1. 气泡排序法.....	42
2. 学生成绩名次表.....	43

实验八 子程序设计实验.....	46
1. 求无符号字节序列中的最大值和最小值.....	46
2. 求 N!	48
实验九 查表程序设计实验.....	50
实验十 输入输出程序设计实验.....	52
1. 显示 A~Z 共 26 个大写英文字母	52
2. INT 21H 功能调用示例程序实验	53
3. PC 键盘输入、实验系统数码管输出	55
4. 实验系统键盘输入、PC 屏幕输出	57
第 3 章 微机接口技术及其应用实验.....	60
实验一 存储器扩展实验.....	60
实验二 I/O 扩展实验	62
1. 8 位 I/O 扩展	62
2. 16 位 I/O 扩展	63
实验三 8255 并行口实验.....	64
1. 8255 A/B/C 口输出方波.....	64
2. 8255 PA 输入 / PB 输出	65
3. 8255 控制交通灯	65
实验四 A/D 0809 模数转换实验.....	67
实验五 D/A 0832 数模转换实验	68
实验六 键盘与显示设计实验.....	69
1. 8279 键盘与显示设计.....	69
2. 8255 键盘与显示设计	70
实验七 8237 可编程 DMA 控制器实验.....	72
实验八 8251 串行通信应用实验.....	73
实验九 8253 定时 / 计数器应用实验.....	74
实验十 8259 中断控制器实验.....	75
实验十一 LCD 128×64 液晶显示实验.....	76
实验十二 LED 16×16 点阵显示实验.....	77
实验十三 音频驱动实验.....	78
实验十四 继电器控制.....	79
实验十五 步进电机控制.....	80
实验十六 直流电机控制.....	81
实验十七 数字温度传感器实验.....	82
实验十八 V/F 转换实验	83
实验十九 PWM 输出实验	84

第 1 章 系统综述

1.1 系统特性

Dais-PRO 系列教学实验系统全面支持 8086/8088 微机原理与接口扩展技术的教学实验，为微处理器在教学中的运用构建了一个全开放、可开发、易拓展式的实验环境。

● 硬件特性

Dais-PRO 系列 16 位微机原理与接口实验系统采用 Intel 8086/8088 微处理器作为系统核心，全面支持 8086/8088 微机原理与接口技术的实验教学，为 16 位微处理器在微机教学中的运用构建了一个全开放、可开发、易拓展式的实验环境。

Dais-PRO 系列 16 位微机原理与接口实验系统的内外总线均为 16 位，支持 16 位 I/O 和内存读写，同时拥有集成微处理器和外挂微处理器两个硬件版本。

● 软件特性

系统配有 MKStudio 集成开发环境，支持 80X86 汇编语言和 C 语言的源程序级编程与调试，支持寄存器、内存和外设接口芯片的非编程读写操作，支持常用的 INT 21h 功能调用，满足微机原理与接口的实验需求。

● 总线特性

系统开放了总线宽度的动态选择机制，把总线宽度控制“BS8”列入用户可定义的范围，为了简化电路连接，系统对该输入信号的缺省定义为“16 位”，即当用户扩展 16 位存储器或输入输出接口时可忽略对“BS8”的定义与连接。

● 寻址能力

系统构建了全覆盖的寻址空间，系统内存可融入任意段的 0~1FFFh，可寻址范围为 16 个段的 0~1FFFh（其中 F000 为系统 BIOS 段），当 IP 大于等于 2000h 时，系统的寻址目标指向外部存储器扩展空间。系统对于 I/O 的扩展寻址无制约，0~FFFFh 共 64K 全空间开放。

● 扩展能力

系统的内存和 I/O 扩展以字节操作为基准，把高低字节允许信号 BHE、BLE 列入扩展定义的范围，支持字与字节指令的扩展寻址，并把 DMA 操作期间的字节定义融入到 BHE、BLE 选通端，使其扩展特性与微机实际应用无缝结合。

- 辅助功能

系统可选配高速 USB 接口的数字虚拟示波器模块，可用于 D/A 输出、串行数据、定时 / 计数器、输入输出等实验中输出波形或信号的测量。

- 安全功能

系统引出的扩展总线均由隔离器件驱动，“隔离”设计保障了 CPU 的安全，避免了误操作、误连线对 CPU 造成的损伤；“驱动”设计的目的是提高了系统扩展总线的稳定性和可靠性。另外系统内置了具有短路保护、过流保护的高性能稳压开关电源，进一步保障了系统的安全性。

1.2 资源分配及接口定义

1.2.1 存储器空间

段地址	寻址范围	寻址目标
0000h~E000h	0000h~1FFFh	用户工作区
0000h~E000h	2000h~FFFFh	用户扩展区
F000h	0000h~FFFFh	BIOS 监控区

注：BIOS 监控区由系统定义，用户程序可以调用其资源，但不可对其进行读出或写入操作。

1.2.2 I/O 空间

地址	寻址目标	说明
0000h~02EFh	I/O 扩展区	由用户定义
02F0h~02FFh	8279 选通区	不可变定义
0300h~03FFh	端口译码区	可选性定义
0400h~FFFFh	I/O 扩展区	由用户定义

说明：

1. 可选性定义仅为简化操作而设置，在舍弃的情况下该区域允许用户另行定义；
2. 不可变定义是指由系统定义的公共资源，用户可以使用但不能更改其端口地址。

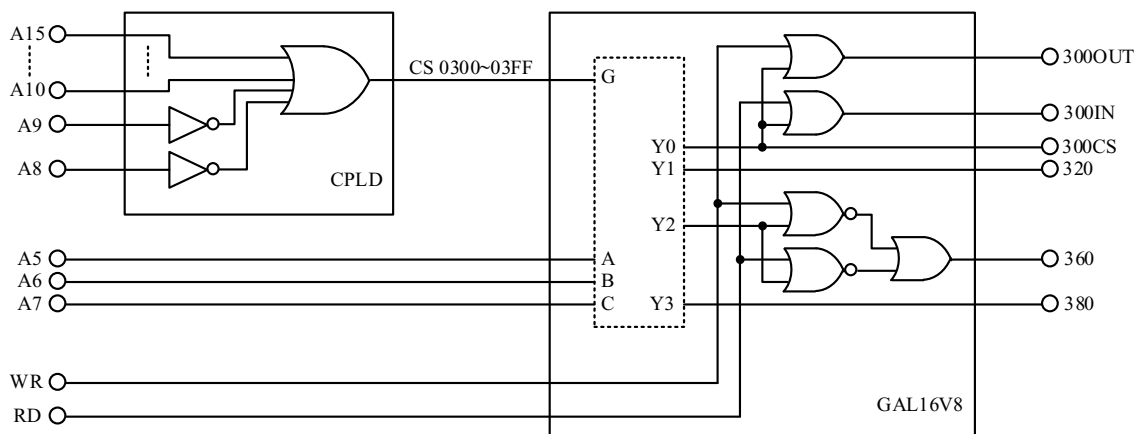
1.3 端口地址的译码及选通定义

端口地址	功能说明	适用对象
300CS	低电平有效的片选控制信号	常规接口器件
300IN	带 300 译码选通的“读”控制信号	244 缓冲输入
300OUT	带 300 译码选通的“写”控制信号	273 锁存输出
320	低电平有效的片选控制信号	常规接口器件
360	在 360 译码选通的“读”或“写”周期输出高电平的控制信号	液晶显示模块
380	低电平有效的片选控制信号	常规接口器件

译码及选通控制的逻辑表达式描述：

$$\begin{aligned}
 !CS &= !A15 \ \& \ !A14 \ \& \ !A13 \ \& \ !A12 \ \& \ !A11 \ \& \ !A10 \ \& \ A9 \ \& \ A8; \\
 !X300 &= !CS \ \& \ !A7 \ \& \ !A6 \ \& \ !A5; \\
 !X320 &= !CS \ \& \ !A7 \ \& \ !A6 \ \& \ A5; \\
 !X380 &= !CS \ \& \ A7 \ \& \ !A6 \ \& \ !A5; \\
 !X300in &= !X300 \ \& \ !RD; \\
 !X300out &= !X300 \ \& \ !WR; \\
 X360 &= !CS \ \& \ !A7 \ \& \ A6 \ \& \ A5 \ \& \ (!RD \ \# \ !WR);
 \end{aligned}$$

译码及选通控制的电路原理图描述：



1.4 控制信号的名称及说明

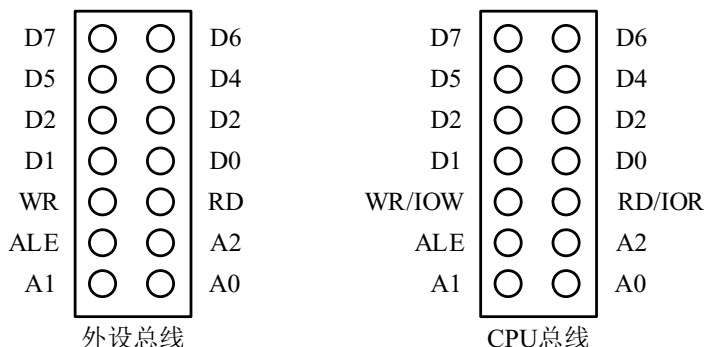
Dais-PRO 系列按用途分为专用机型和通用机型两大类（均有集成仿真器和外挂仿真器两种版本），下表列出了专用机型和通用的控制总线信号名称的区别。

控制信号名称	控制信号说明
MER	存储器扩展读选通信号，低电平有效。
MEW	存储器扩展写选通信号，低电平有效。
IOR	I/O 读控制信号，低电平有效。
IOW	I/O 写控制信号，低电平有效。
ALE	地址锁存控制信号，下降沿有效。
RST / RESET	复位控制信号，高电平有效。
CLK	系统 8284 提供的 PCLK 时钟信号。
INTR	IRQ0~IRQ7 中断请求输入端，高电平有效。
INTA	中断响应控制输出端，低电平有效。
AEN	DMA 请求输入端，低电平有效。
BLE	偶字节允许控制信号，低电平有效。
BHE	奇字节允许控制信号，低电平有效。
BS8	Dais-86PRO/PRO3C 机型的总线宽度选择信号，低电平有效。当 BS8 为高电平（缺省状态）时，为 16 位存储器和 8 位 I/O；当 BS8 为低电平时，为 8 位存储器和 16 位 I/O。
MECS16	Dais-PRO3C+ 机型的存储器扩展总线宽度选择信号，低电平有效。当 MECS16 为高电平（缺省状态）时，存储器扩展总线宽度为 16 位；当 MECS16 为低电平时，存储器扩展总线宽度为 8 位。
IOCS16	Dais-PRO3C+ 机型的 I/O 扩展总线宽度选择信号，低电平有效。当 IOCS16 为高电平（缺省状态）时，I/O 总线扩展宽度为 8 位；当 IOCS16 为低电平时，I/O 扩展总线宽度为 16 位。

1.5 接口的实验连接定义

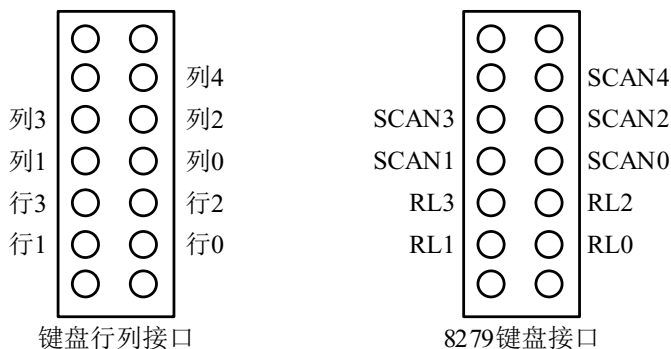
1.5.1 总线连接接口（位于键盘左上方）

该接口为简化连接和在线测试而设置，适用于多模块综合实验及受课时制约的单一实验。当拆除该接口的连接电缆，则各接口实验单元的数据总线、地址总线及控制信号均需要实验者手动连接。由于受 PCB 面积所限，每个接口单元不一定全部预留 8 位数据总线接口，因此对于接口实验时数据总线的连接可通过“ADC0809 模数转换”单元或“8255 并行口”单元所标识的数据总线接口与 CPU 总线连接。该接口在出厂时默认为连接状态。



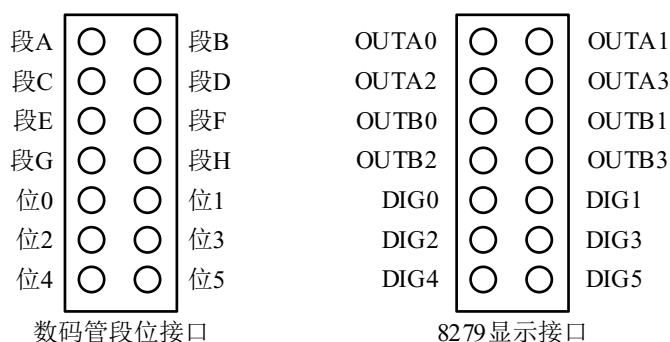
1.5.2 键盘连接接口（位于键盘右上方）

该接口为 8279 专用键盘连接接口。若要定义 8255 键盘接口可拆除该接口的连接电缆，通过“键盘动态扫描电路”单元（位于键盘上方）的“行”、“列”引出端与 8255 并行口相连。



1.5.3 数码显示连接接口（位于数码管右下方）

该接口为 8279 专用数码显示连接接口。若要定义 8255 数码显示接口可拆除该接口的连接电缆，通过“七段显示器动态扫描电路”单元左侧的“位”、“段”引出端与 8255 并行口相连。



1.6 限制性控制与接口连接

1.6.1 直流电机控制开关

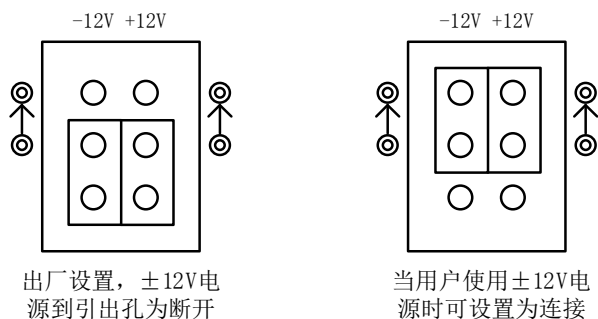
在直流电机下方设有控制开关，通常处于 OFF 位置，仅在涉及直流电机实验时才拨动至 ON 位置，实验完成后，须拨回“OFF”位置。

1.6.2 0~10V 模拟电压的限制性调节

实验系统中所用的模拟量为 0~5V，因此出厂时已将 0~10V 模拟电压的输出限制在 0~5V，以防止大于 5V 的电压被误操作而损坏器件。若在扩展实验中需要 0~10V 的模拟量，可通过调节位于 0~10V 模拟电压下方、继电器左侧的微型限压电位器，使其输出 0~10V 模拟量。

1.6.3 $\pm 12V$ 电压的输出连接

为了避免 $\pm 12V$ 电压因误连接而造成器件的损坏，系统的实验器件所需的 $\pm 12V$ 已内部连接。我们保留了 $\pm 12V$ 的引出插孔，但出厂时限制其输出，若在自行扩展的实验中需要其供电，可卸去左上方的液晶模块，用跳线帽将 $\pm 12V$ 电源与其引出插孔导通（以箭头方向为准）。



1.7 CPU 与 I/O 扩展接口

1.7.1 多 CPU 集成控制接口定义

通用 CPU 模块接口单元可支持 Intel 80386、8088 微处理器及 MCS-51/96 系列单片机。

HOLD	①	②	VCC	CLK	①	②	D0
VCC	③	④	RESET	ADY	③	④	D1
HLDA	⑤	⑥	CLK2	BS8	⑤	⑥	D2
READY	⑦	⑧	NC	BHE	⑦	⑧	D3
IRQ6	⑨	⑩	M/IO	ADS	⑨	⑩	D4
IRQ7	⑪	⑫	D/C	MNI	⑪	⑫	D5
P3.1	⑬	⑭	W/R	A19	⑬	⑭	D6
P3.0	⑮	⑯	D8	A18	⑮	⑯	D7
INTA	⑰	⑱	D9	A17	⑰	⑱	A16
	⑲	⑳	D10	DBG	⑲	㉑	A15
	㉑	㉒	D11	A7	㉑	㉒	A14
	㉓	㉔	D12	A6	㉓	㉔	A13
	㉕	㉖	D13	A5	㉕	㉖	A12
	㉗	㉘	D14	A4	㉗	㉘	A11
	㉙	㉚	D15	A3	㉙	㉚	A10
Z1	㉛	㉜	GND	A2	㉛	㉜	A9
Z0	㉝	㉞	RD	A1	㉝	㉞	A8
GND	㉟	㊱	WR	A0	㉟	㊱	GND

1.7.2 PACK 扩展接口定义

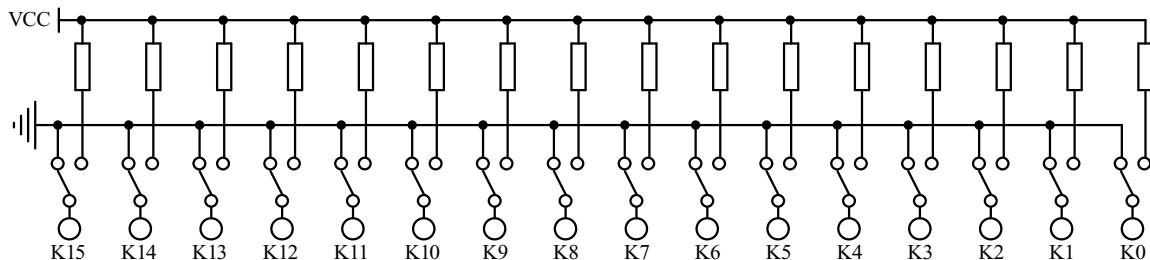
PACK 单元用来扩展增加的实验模块，可扩展 USB1.1、USB2.0、CAN-bus 现场总线、TCP/IP 以太网接口、16C550 串行通信接口、RS485 通信等、压力传感器等模块。

D6	D4	D2	D0	IOW	IO2	INT	IO1	ALE	GND
○	○	○	○	○	○	○	○	○	○
○	○	○	○	○	○	○	○	○	○
D7	D5	D3	D1	IOR	IO5	IO4	IO3	CS0	VCC

1.8 实验系统通用电路

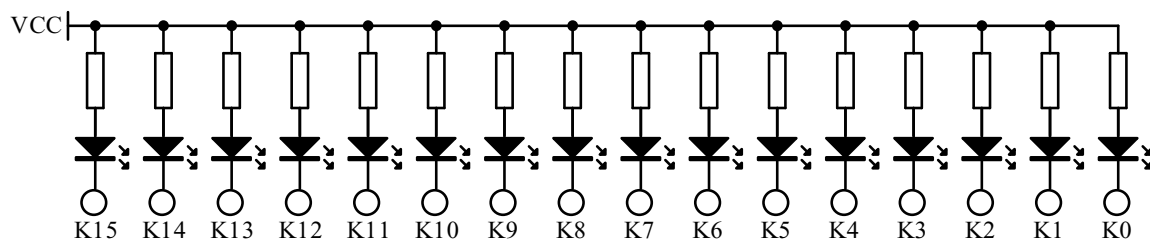
1.8.1 逻辑电平开关电路

在实验系统的“逻辑电平开关”单元提供 16 位逻辑电平开关，对应的 K15~K0 插孔为相应的逻辑电平输出端，开关向上拨输出逻辑“0”，向下拨输出逻辑“1”。



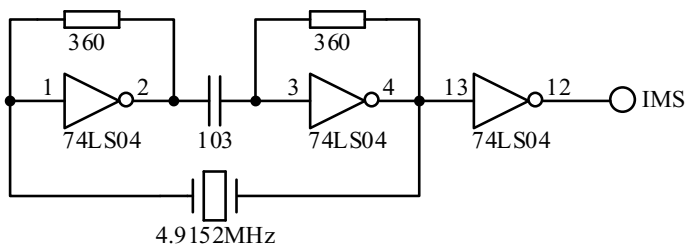
1.8.2 发光二极管显示电路

在实验系统的“发光二极管显示”单元提供 16 位发光二极管，对应的 L15~L0 插孔为相应的发光二极管信号输入端，该输入端为低电平“0”时点亮发光二极管。



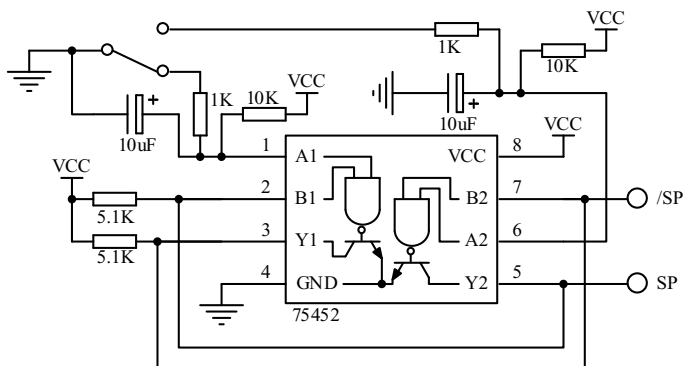
1.8.3 脉冲信号发生器电路

在实验系统的“PACK 扩展区”单元提供脉冲源输出插孔 IMS，其输出频率为 4.9152MHz。



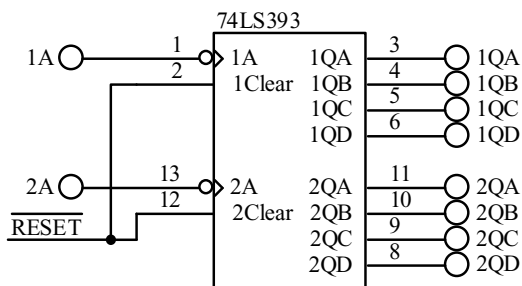
1.8.4 单脉冲电路

在实验系统的“单脉冲”单元提供一个单脉冲按钮，每按一次按钮，即可从 SP、/SP 两个插孔上分别输出一次正脉冲和一次负脉冲，供中断控制器、定时器 / 计数器等实验使用。



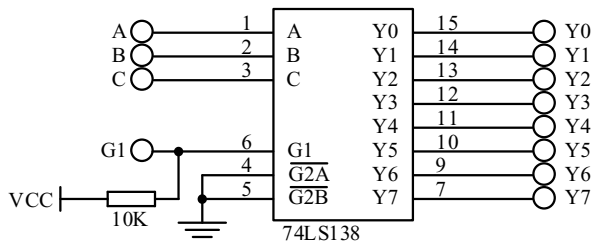
1.8.5 计数器电路

在实验系统的“393 分频”单元提供一个双 4 位计数器，该电路由 1 片 74LS393 组成，其 1A、2A 为脉冲输入端，其输出端分别为 1QA,1QB,1QC,1QD 和 2QA,2QB,2QC,2QD。该计数器在上电时由系统复位信号清零。



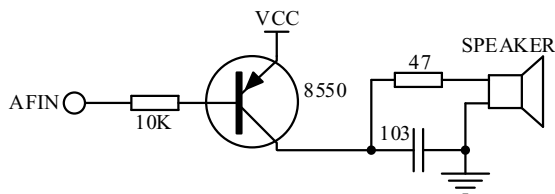
1.8.6 译码器电路

实验系统提供 74LS138 一片，其输入、输出引脚用插孔全部引出，可根据实验需求自行定义 Y0~Y7 的寻址范围。



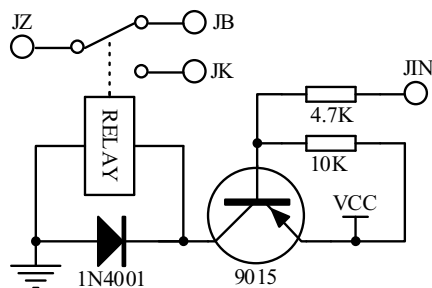
1.8.7 蜂鸣器及其驱动电路

实验系统提供一个蜂鸣器及其的驱动电路，可进行音频驱动实验，AFIN 为信号输入端。其电路如右图所示。



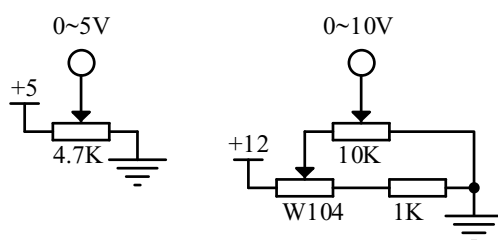
1.8.8 继电器及其驱动电路

在实验系统的“继电器”单元提供一个+5V 直流继电器及相应的驱动电路，当其开关量输入端“JIN”插孔输入数字电平“0”时，继电器动作，常开触点闭合、常闭触点断开。其电路如右图所示。



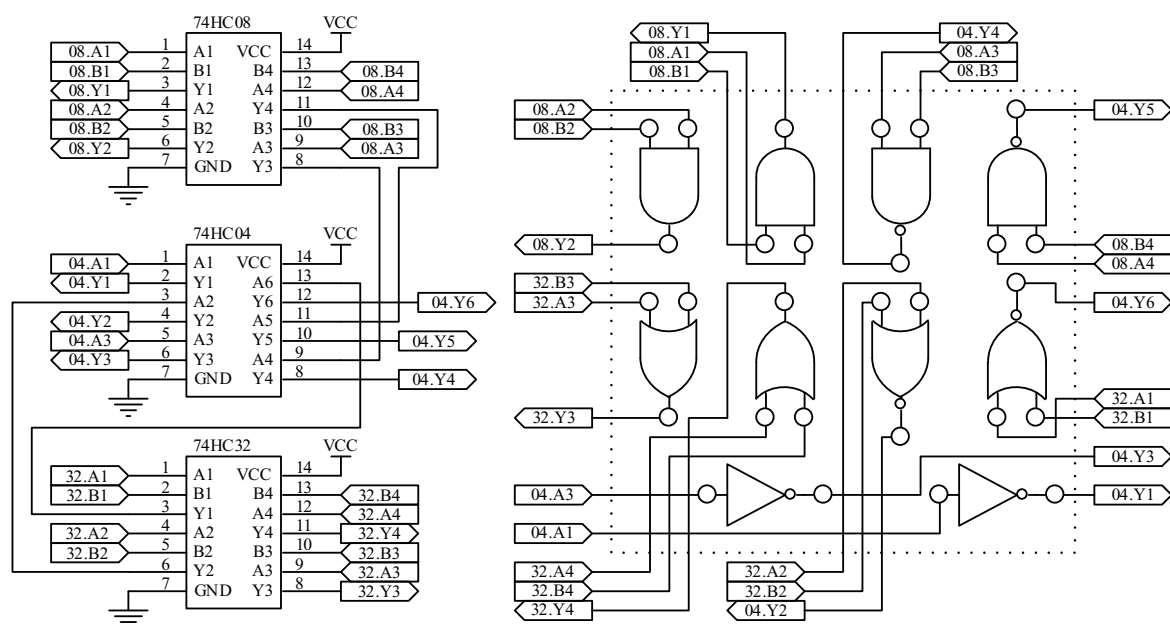
1.8.9 模拟电压调节电路

实验系统提供 0~5V、0~10V 可调模拟电压输出端，供 A/D 转换实验的模拟量输入使用。其中 0~5V 供系统的 ADC0809 使用；0~10V 供外扩的 A/D 转换器（如 AD574）使用。为防电路误接使高电压对器件造成损坏，0~10V 组的出厂设置最高为 5V，用户可自行通过 W104 基准调节电位器使其输出 10V。其电路如右图所示。



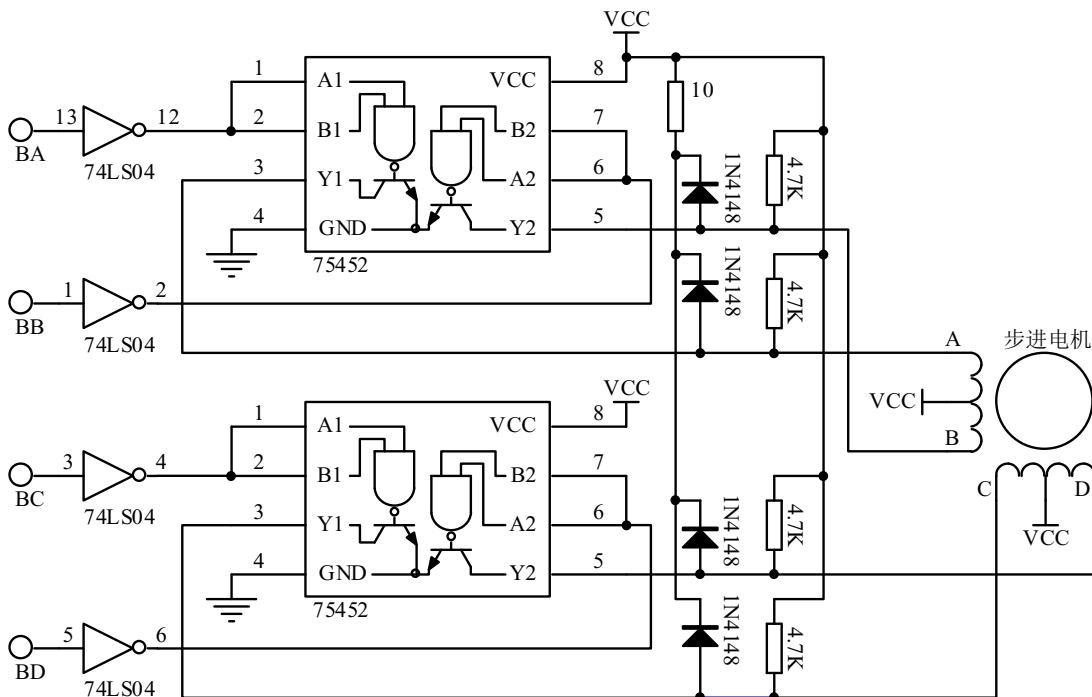
1.8.10 逻辑门电路

实验系统提供与门、与非门、或门、或非门、非门各 2 组，便于接口电路的搭接。



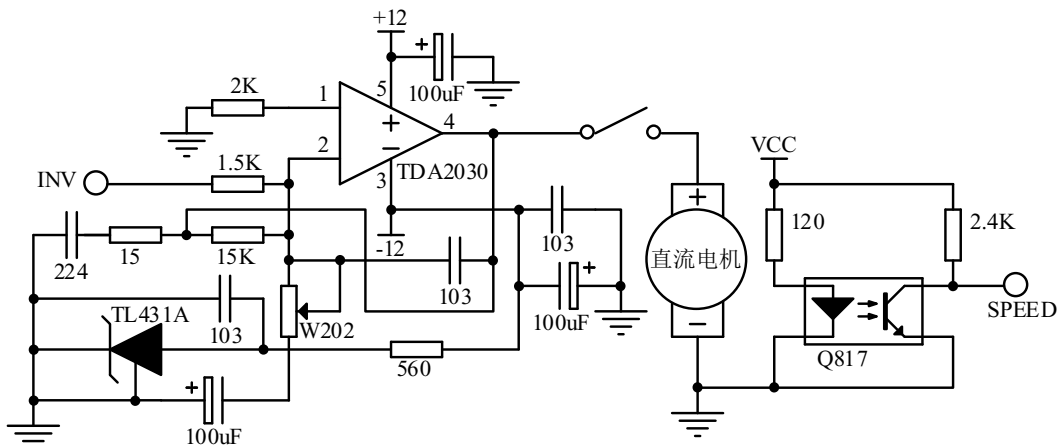
1.8.11 步进电机及其驱动电路

实验系统提供一个步进电机及其驱动电路，BA、BB、BC、BD 四个脉冲信号输入端通过驱动器，输出 A、B、C、D 控制步进电机。



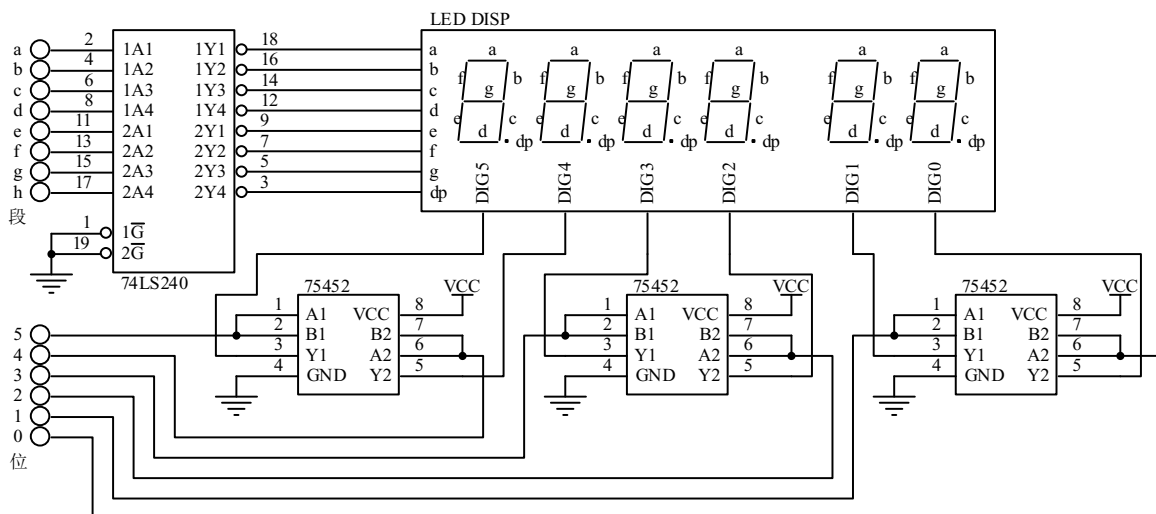
1.8.12 直流电机及其驱动电路

实验系统提供一个直流电机及其驱动电路、光耦隔离测速电路。直流电机的转速是由加到其输入端“INV”的脉冲电平及占空比来决定的，正向占空比越大转速越快，反之越慢；直流电机转速反馈到“SPEED”输出端，该输出端的频率与直流电机的转速成正比。



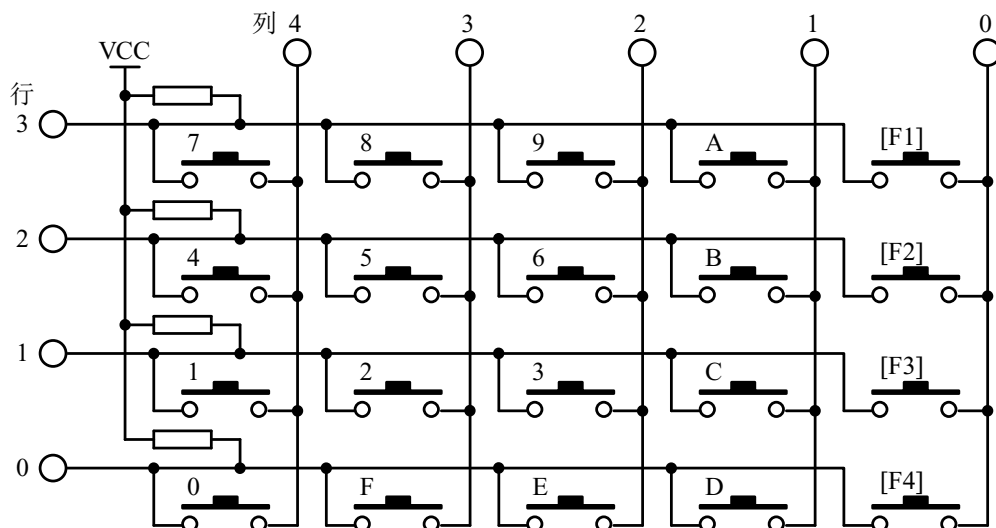
1.8.13 6 位七段码显示电路

实验系统提供 6 位七段码显示器，其字形口 a~h、字位口 5~0 全部引出，可使用 8279、8255 等器件对其控制，作为人机交互的输出设备。



1.8.14 4×5 阵列式键盘电路

实验系统提供 4×5 阵列式键盘，其键入口 3~0、键扫口 4~0 全部引出，可使用 8279、8255 等器件对其控制，作为人机交互的输入设备。



1.9 实验环境的认识（重点阅读）

1.9.1 电缆连接须知

- 1) 在 PC 机或实验系统**断电**的情况下,用随机所配的通信电缆正确连接实验系统与 PC 机;
- 2) 用随机所配的电源电缆向实验系统提供 AC220V 电源。

1.9.2 系统状态指示

实验系统设有双色系统状态指示灯（位于面板中上方的系统核心单元的右侧），需要强调的是该指示灯并非简单的电源指示，它映射着系统的设置和系统的当前工作状态。

1) 初始待令状态

- 在上电或复位后，状态灯呈绿色，表示实验系统初始化成功并进入联机待令状态；
- 在上电或复位后，状态灯呈红色，表示实验系统初始化失败，应立即关机报修；
- 在上电或复位后，状态灯不发光，表示实验系统供电有误或初始化失败，应核实后重启或报修。

2) 程序运行状态

在装载用户程序并开始全速运行时，状态指示灯呈**红色**，表示系统已进入运行状态。

- 在全速运行后，状态灯呈红色闪动，表示用户程序已开始运行；
- 在全速运行后，状态灯呈红色常亮，表示用户程序在执行 INT 21h 功能调用并等待用户键盘输入。

第2章 微机原理及其程序设计实验

本章主要介绍汇编语言程序设计，通过实验来学习 8086 的指令系统、寻址方式以及程序的设计方法，同时掌握 MKStudio 软件的使用。

实验一 系统认识实验

一、实验目的

掌握 Dais 系列微机原理与接口实验系统的操作，熟悉 MKStudio 集成开发环境的操作。

二、实验设备

PC 计算机一台，Dais-86PRO+或 Dais-PRO163C 实验系统一套。

三、实验内容

编写程序，将 80h~8Fh 共 16 个数据写入数据段 0000h 偏移地址开始的 16 个内存单元中。

四、实验步骤

1. 运行 MKStudio 软件，进入集成开发环境，在首次运行或检测不到实验系统时弹出设置通信端口对话框（如图 2-1-1），请确保实验装置与 PC 正确连接，并已打开实验装置电源，使其进入在待命状态。这里选择“硬件联机调试”，软件开始检测设备，若联机成功则提示“设备检测成功”，单击“完成”进入 MKStudio 主界面，状态栏显示“联机调试”；当实验系统硬件或状态异常时，会弹出“通信出错”的信息框（如图 2-1-2），请检查系统工作状态及通信电缆连接情况后重试。

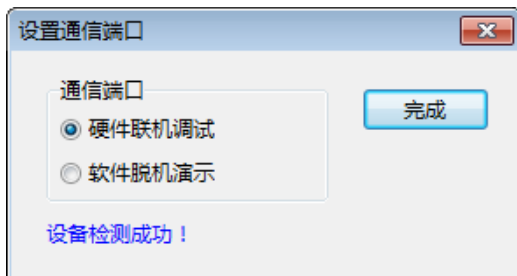


图 2-1-1 设置通信端口对话框

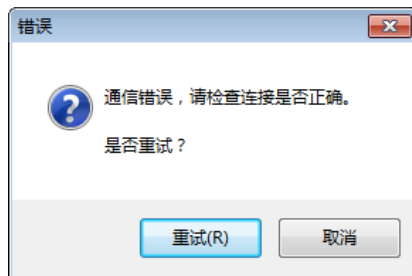



图 2-1-2 通信错误信息框

2. 根据程序设计使用指令的不同，单击菜单栏“设置”→“设置工作方式”项打开对话框，选择需要使用的目标 CPU 型号，因本实验例程仅使用 16 位指令编程，所以将目标 CPU 型号设置为“8086/8088”，以使寄存器窗口采用 16 位方式显示（如图 2-1-3），最后单击“确定”按钮保存设置。



图 2-1-3 设置工作方式对话框


3. 工作方式设置完毕后，单击菜单栏“文件”→“新建”项或按 Ctrl+N 组合键（建议单击工具栏“”按钮）来新建一个文件，软件会出现一个空白的文件编辑窗口。
4. 在新窗口中输入程序代码（A86\2-1.ASM）：

```

DATA    SEGMENT
        DB 16 DUP(?)    ;申请缓冲区
DATA    ENDS

CODE    SEGMENT
        ASSUME CS:CODE, DS:DATA
START   PROC NEAR
        MOV AX, DATA
        MOV DS, AX
        MOV AL, 80H
        MOV SI, 0000H    ;建立数据起始地址
        MOV CX, 16       ;循环次数
MLOOP:  MOV [SI], AL
        INC SI            ;地址自加 1
        INC AL            ;数据自加 1
        LOOP MLOOP
        MOV AH, 4CH
        INT 21H          ;程序退出
START   ENDP
CODE    ENDS
        END START

```

5. 单击菜单栏“文件”→“保存”项或按 Ctrl+S 组合键（建议单击工具栏“”按钮）保存文件。若是新建的文件尚未命名，系统会弹出文件保存对话框（如图 2-1-4），提示用户选择文件保存的路径和文件名，再单击“保存”按钮。

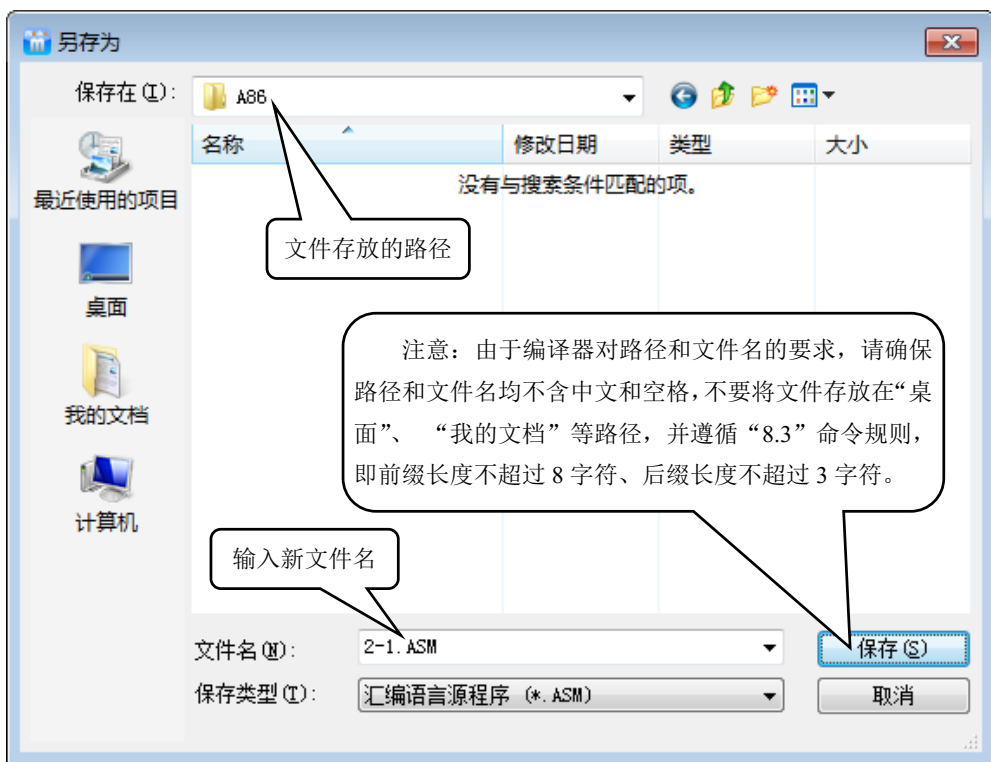



图 2-1-4 文件保存对话框

6. 单击菜单栏“编译”→“编译并调试”项或按 Ctrl+F9 组合键（建议单击工具栏“ 编译”按钮），对当前文件进行编译和链接，若程序无语法错误，则开始自动装载，信息窗口的编译框（如图 2-1-5）显示输出信息。

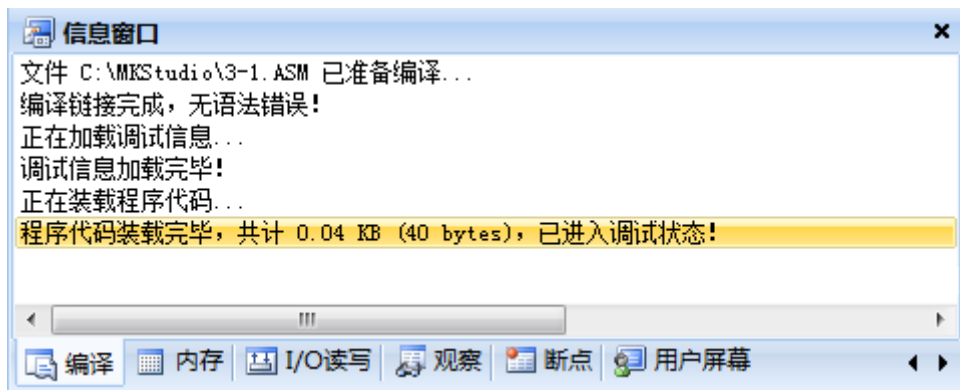


图 2-1-5 编译信息框

7. 当编译无误且下载成功后,即进入调试状态,软件自动将工作窗口切换到寄存器,当前 IP 行高亮突出显示,源程序窗口左侧显示小方块以标识可执行语句行,如图 2-1-6 所示。

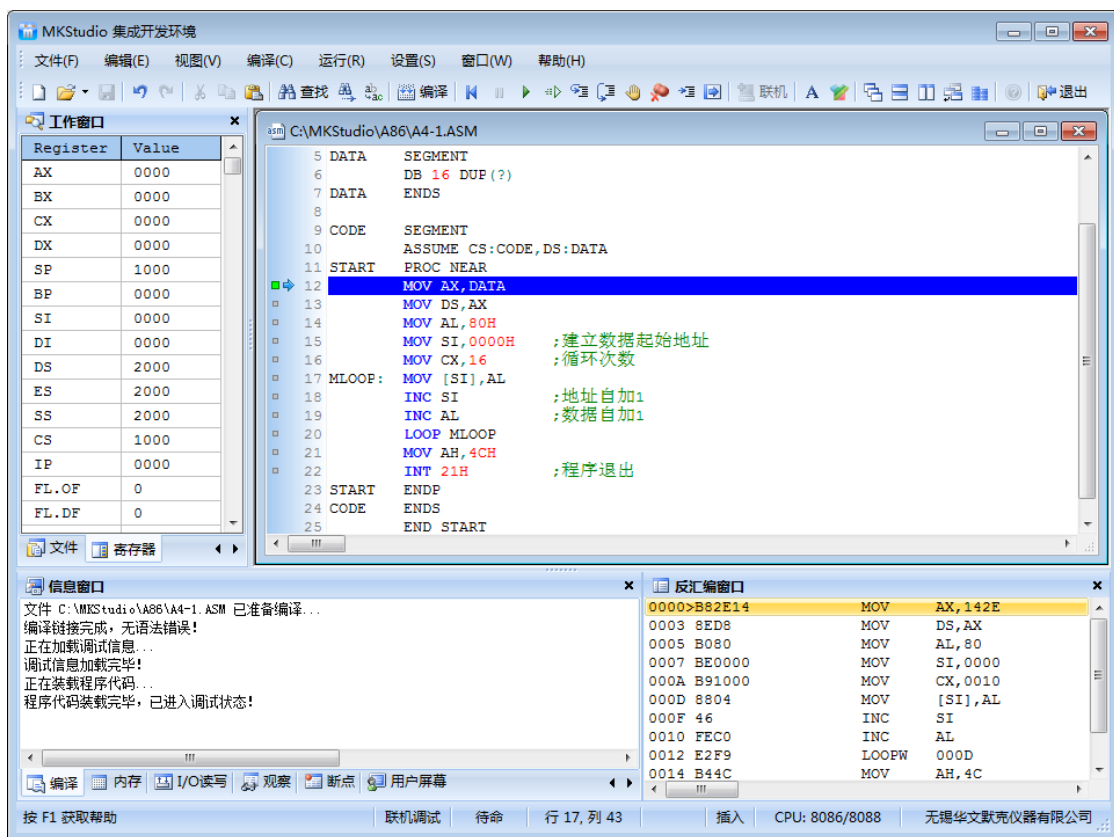


图 2-1-6 进入调试状态的主界面

8. 将信息窗口切换到“内存”,在内存区单击鼠标右键切换到“DS 数据段”,除了程序申请的缓冲区 0000h~000Fh 为“00”以外,其余为随机数(如图 2-1-7)。

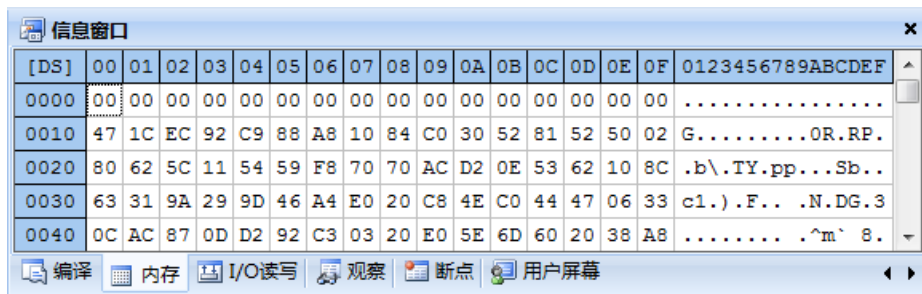




图 2-1-7 初始时的内存单元数据显示

9. 单击菜单栏“运行”→“单步跟踪”项或按 F7 快捷键(建议单击工具栏“”按钮),单步运行程序,观察寄存器和内存单元的数据变化;

10. 也可单击菜单栏“运行”→“全速运行”项或按 F9 快捷键（建议单击工具栏“”按钮），使用全速运行方式，当程序结束时，弹出信息框（如图 2-1-8 所示），提示用户是否复位。若需要再次运行，可单击“是”进行复位。

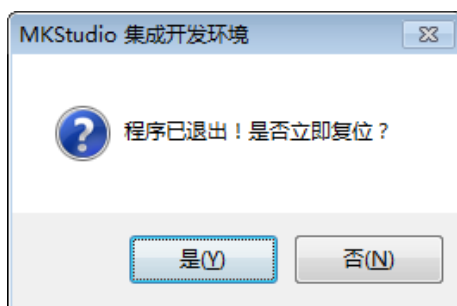


图 2-1-8 程序结束时的信息提示框

11. 再观察 DS:0000 内存单元，验证程序运行结果（如图 2-1-9 所示）。




图 2-1-9 程序运行后的内存单元数据显示

12. 可以用鼠标单击选中某个内存单元，按键盘的 0~9 或 A~F，直接写入数据以覆盖该内存单元原有数据。如图 2-1-10 所示。



图 2-1-10 修改后的内存单元数据显示

13. 除了单步或全速运行，还可以使用断点手段来调试程序。单击菜单栏“运行”→“复位”项或按 Ctrl+F2 组合键（建议单击工具栏“”按钮）进行复位，以便重新运行程序。

14. 单击源程序编辑窗口左侧的行号即可快速设置断点，断点行为红色高亮显示（如图 2-1-11），若要删除该断点只需再次单击断点行的行号即可清除断点。

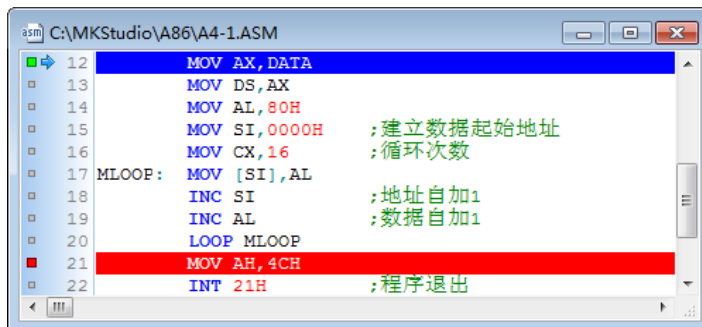


图 2-1-11 设置断点后的源程序窗口

15. 设置断点后，单击菜单栏“运行”→“全速运行”项或按 F9 快捷键（建议单击工具栏“▶”按钮），使用全速运行方式，待程序运行到断点行时自动停下（如图 2-1-12）。

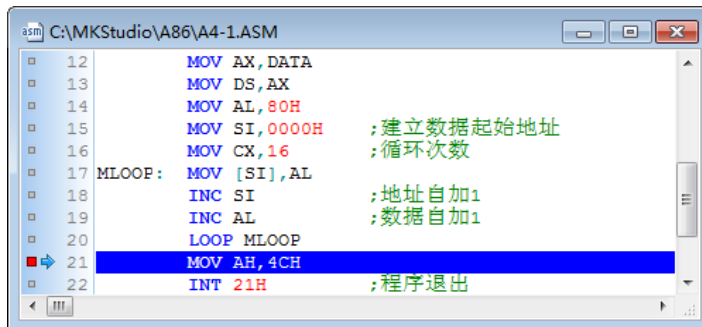


图 2-1-12 全速运行后遇断点暂停的源程序窗口

五、操作练习

编写程序，将数据段内存 0000h~000Fh 单元的 16 个数据复制到 0010h~001F 单元，通过单步、断点等调试手段验证程序功能，查看 0010h~001Fh 单元的数据是否传送正确。

实验二 数制转换实验

一、实验目的

1. 掌握不同进制数及编码相互转换的程序设计方法，加深对数制转换的理解；
2. 熟悉和了解计算机的数制体系。

二、实验设备

PC 计算机一台，Dais-86PRO+或 Dais-PRO163C 实验系统一套。

三、实验内容及步骤

二进制、十进制、十六进制这三种数制的计数系统是计算机最重要的数值代码，其中二进制数只有两个独立的 0 和 1，是计算机唯一能够识别的机器代码；而十六进制数是程序设计中常用的基本数制；之于十进制数则是人类通用的标准数制。因此它们之间的换算在应用程序的设计中必不可少。三者之间的相互关系如表 2-2-1 所列。从表中可以看出二进制与十六进制之间仅存在表示方法上的差异，一个用“8421 码”表示，另一个则用“符号”表示，但“逢二进一”是它们共同的计数规则，实质上十六进制数是二进制数的简写，亦可视为二进制数的“符号”表示法，从数制角度上讲它们之间存在直截了当的转换关系，例如二进制数“1010”等于十六进制数“A”，而“A”的位码又等于二进制数的“1010”，由此可见它们之间的转换毫无实际意义。

1. 十六进制数转换为十进制数

参考程序 (A86\2-2-1.ASM):

```
DATA    SEGMENT
DBUF    DW 3039H ;16 进制数 3039H 为 10 进制数 12345
DVAL    DB 5 DUP(?) ;存放转换后的数据
DLEN    = $-DBUF
DATA    ENDS

CODE    SEGMENT
        ASSUME CS:CODE,DS:DATA
START   PROC NEAR
        MOV AX,DATA
        MOV DS,AX
        MOV SI,OFFSET DBUF ;源数据地址
        MOV DX,[SI]
        MOV SI,OFFSET DLEN ;目标数据地址
A1:     DEC SI
```

表 2-2-1 常用数制关系

十进制	二进制	十六进制
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

```

MOV AX,DX
MOV DX,0
MOV CX,10          ;除数 10
DIV CX              ;商送 AX, 余数送 DX
XCHG AX,DX
MOV [SI],AL         ;存入目标地址
CMP DX,0000H
JNE A1 ;判断转换结束否, 未结束则转 A1
A2:  CMP SI,OFFSET DVAL ;与目标地址的首址比较
     JZ A3 ;等于首地址则转 A3, 否则将剩余地址
     DEC SI ;中填 00H
     MOV AL,00H
     MOV [SI],AL
     JMP A2
A3:  MOV AH,4CH
     INT 21H ;程序终止
START ENDP
CODE ENDS
END START

```

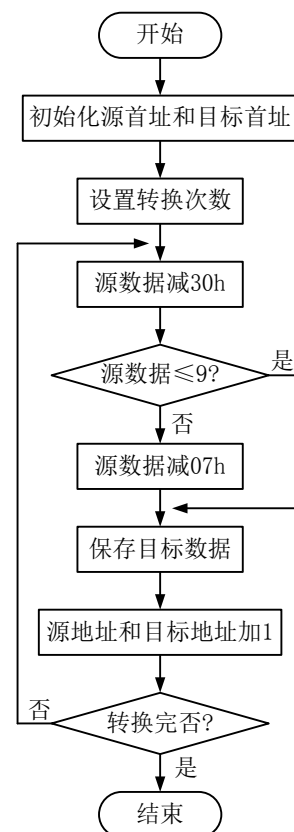


图2-2-1 十六进制转十进制流程图

实验步骤:

- 1) 编写实验程序, 经编译、链接无误后装载到实验系统;
- 2) 待转换数据存放于数据段 0000h~0001h 单元, 可根据需求另行输入, 默认为 3039h;
- 3) 运行程序, 等待程序结束;
- 4) 查看数据段 0002h~0006h 单元, 即为转换结果, 应为: 01h, 02h, 03h, 04h, 05h;
- 5) 反复测试几组数据, 验证程序的正确性。

2. 十进制数转换为十六进制数

参考程序 (A86\2-2-2.ASM):

```

DATA SEGMENT
NUMS DB 3,2,7,6,7 ;十进制数:32767
NUMO DW ?
DATA ENDS

CODE SEGMENT
ASSUME CS:CODE,DS:DATA
START: MOV AX,DSEG ;DS 指向数据段

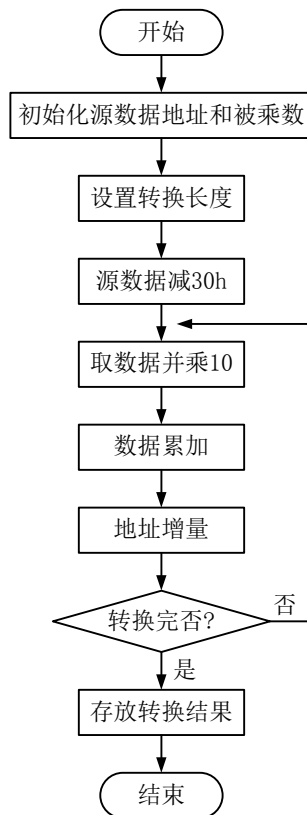
```



```

MOV DS,AX
MOV SI,OFFSET NUMS ;源数据地址
MOV BX,10           ;被乘数
MOV CX,4            ;转换长度
MOV DH,0            ;屏蔽源数据高 8 位
MOV AH,0
MOV AL,[SI]         ;取首个数据
A1:  IMUL BX         ;乘 10
     MOV DL,[SI+1]   ;取当前数据
     ADC AX,DX       ;源数据累加
     INC SI         ;源地址增量
     LOOP A1        ;未结果转 A1 继续
     MOV NUMO,AX     ;存放转换结果
     MOV AH,4CH     ;停止程序运行
     INT 21H
CODE  ENDS
      END START

```



实验步骤:

- 1) 编写实验程序, 经编译、链接无误后装载到实验系统;
- 2) 待转换数据存放于数据段 0000h~0004h 单元, 可根据需求另行输入, 默认为 32767;
- 3) 运行程序, 等待程序结束;
- 4) 查看数据段 0005h~0006h 单元, 即为转换结果, 应为: 7FFFh;
- 5) 反复测试几组数据, 验证程序的正确性。

图2-2-2 十进制转十六进制流程图

实验三 码制转换实验

一、实验目的

1. 掌握不同类型码相互转换程序的设计方法，加深对码制之间转换的理解；
2. 熟悉和了解计算机操作中的编码定义及与数制码的关系。

二、实验设备

PC 计算机一台，Dais-86PRO+或 Dais-PRO163C 实验系统一套。

三、实验内容及步骤

计算机的操作有其独特的专门编码，例如数制就代表一种类型的编码，其它特殊编码包括 ASCII、Gray、Excess-3、BCD 等。在程序设计中，经常有需要将一种码转换成另一种码。码的转换使用查表法比较容易实现，但在本例程中将采用简单的数字操作来完成转换。常用的 ASCII 码与十六进制的对应关系如表 2-3-1 所列。

1. ASCII 码（数字符）转换为十六进制数

参考程序（A86\2-3-1.ASM）：

```
DATA    SEGMENT
NUMS    DB  30H,31H,02H,41H,42h,43h,44H,45H
DLEN     = $-NUMS           ;声明 NUMS 长度
NUMO     DB  7 DUP(?)
DATA    ENDS

CODE    SEGMENT
        ASSUME CS:CODE,DS:DATA
START   PROC NEAR
        MOV AX,DATA        ;DS 指向数据段
        MOV DS,AX
        MOV CX,DLEN        ;转换数
        MOV SI,OFFSET NUMS ;ASCII 码首地址
        MOV DI,OFFSET NUMO ;十六进制数首地址
A1:     MOV AL,[SI]
        SUB AL,30H
        JC  A4              ;跳过非数值 ASCII 码
        JMP A5
A4:     DEC CX              ;计数减 1
```

表 2-3-1 ASCII 与十六进制对应表

ASCII	十六进制
30H	00H
31H	01H
32H	02H
33H	03H
34H	04H
35H	05H
36H	06H
37H	07H
38H	08H
39H	09H
41H	0AH
42H	0BH
43H	0CH
44H	0DH
45H	0EH
46H	0FH

```

        JMP A3          ;继续转换
A5:     CMP AL, 9
        JNG A2          ;为 30h~39h 转 A2
        SUB AL, 7
        CMP AL, 0FH
        JNG A2          ;为 41h~46h 转 A3
A3:     INC SI          ;ASCII 码地址加 1
        JMP A1          ;转换下一个
A2:     MOV [DI], AL    ;结果存入目标地址
        INC DI          ;目标地址加 1
        INC SI          ;源地址加 1
        LOOP A1         ;若转换未结束则继续
        MOV AH, 4CH     ;程序结束
        INT 21H
START  ENDP
CODE   ENDS
      END START

```

实验步骤:

- 1) 编写实验程序, 经编译、链接无误后装载到实验系统;
- 2) 待转换数据存放于数据段 0000h~0007h 单元, 可根据需求另行输入;
- 3) 运行程序, 等待程序结束;
- 4) 查看数据段 0008h 起始的单元, 即为转换结果;
- 5) 程序遇到非数值 ASCII 码, 则自动跳过, 继续转换下一个单元;
- 6) 反复测试几组数据, 验证程序的正确性。

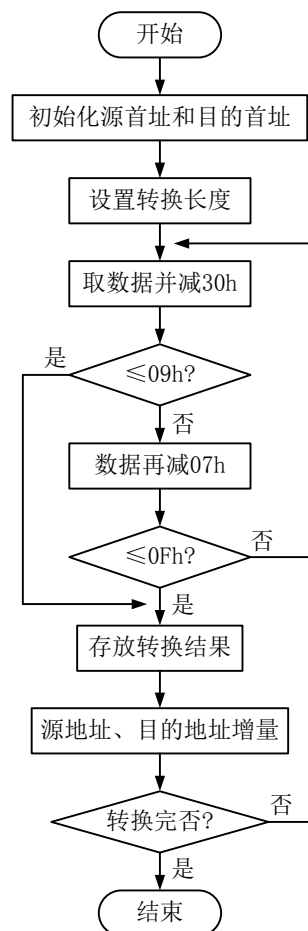


图2-3-1 ASCII转十六进制流程图

2. 十六进制数转换为 ASCII 码

参考程序 (A86\2-3-2.ASM):

```

DATA    SEGMENT
NUMS    DW 12ABH

DLEN    = ($ - NUMS) * 2 ;声明 NUMS 长度
NUMO    DD ?
CLEN    = $ - NUMO + 1
DATA    ENDS

CODE    SEGMENT
        ASSUME CS:CODE, DS:DATA

```

```

START    PROC NEAR
          MOV AX, DATA      ;DS 指向数据段
          MOV DS, AX
          MOV CX, DLEN       ;转换长度
          MOV DI, OFFSET NUMS ;十六进制数源地址
          MOV DX, [DI]
A1:       MOV AX, DX
          AND AX, 000FH       ;取低 4 位
          CMP AL, 0AH
          JB A2               ;小于 0AH 则转 A2
          ADD AL, 07H         ;在 0Ah~0Fh 之间, 需加 07h
A2:       ADD AL, 30H         ;转换为相应 ASCII 码
          MOV [DI+CLEN], AL ;结果存入目标地址
          DEC DI
          PUSH CX
          MOV CL, 04H
          SHR DX, CL         ;将十六进制数右移 4 位
          POP CX
          LOOP A1
          MOV AH, 4CH        ;程序结束
          INT 21H
START    ENDP
CODE     ENDS
        END START

```

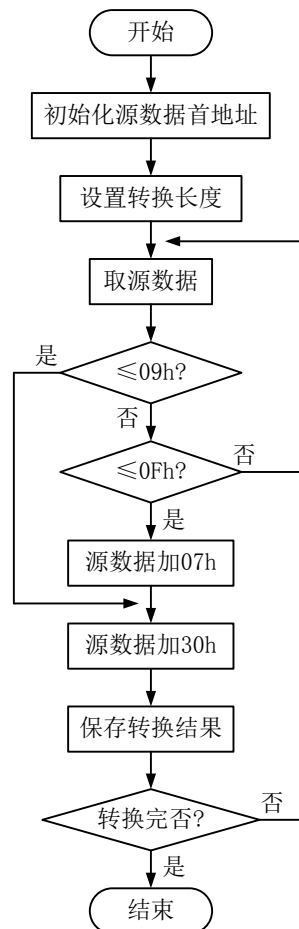


图2-3-2 十六进制转ASCII流程图

实验步骤:

- 1) 编写实验程序, 经编译、链接无误后装载到实验系统;
- 2) 待转换数据存放于数据段 0000h~0001h 单元, 默认为 12ABh, 可根据需求另行输入;
- 3) 运行程序, 等待程序结束;
- 4) 查看数据段 0002h~0005h 起始的单元, 即为转换结果: 31h, 32h, 41h, 42h;
- 5) 程序遇到非数值 ASCII 码, 则自动跳过, 继续转换下一个单元;
- 6) 反复测试几组数据, 验证程序的正确性。

3. ASCII 码 (数字符) 转换为十进制数

参考程序 (A86\2-3-3.ASM):

```

DATA     SEGMENT
NUMS     DB 30H, 31H, 41H, 38H, 39h, 32h, 33H, 36H
NUMO     DB 7 DUP(?)
DLEN     = $ - NUMO
DATA     ENDS

```

```

CODE    SEGMENT
        ASSUME CS:CODE,DS:DATA
START    PROC NEAR
        MOV AX,DATA          ;DS 指向数据段
        MOV DS,AX
        MOV CX,DLEN          ;转换位数
        MOV SI,OFFSET NUMS   ;ASCII 码首地址
        MOV DI,OFFSET NUMO   ;十进制数首地址
A1:      MOV AL,[SI]
        SUB AL,30H
        JC A3                ;非数值 ASCII 码转
        CMP AL,9
        JNG A2               ;为 30h~39h 则转 A2
A3:      INC SI               ;ASCII 码地址加 1
        JMP A1               ;转换下一个
A2:      MOV [DI],AL          ;结果存入目标地址
        INC DI               ;目标地址加 1
        INC SI               ;源地址加 1
        LOOP A1              ;尚未结束转
        MOV AH,4CH           ;程序结束
        INT 21H
START    ENDP
CODE     ENDS
        END START

```

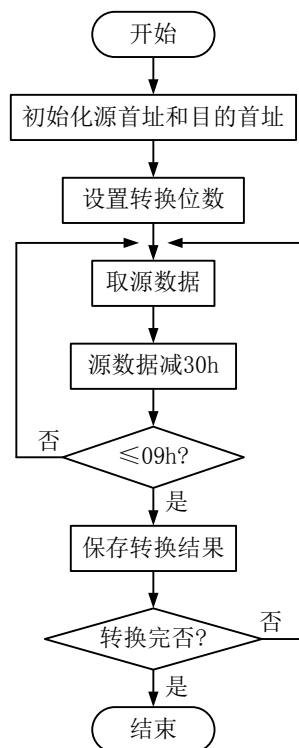


图2-3-3 ASCII转十进制流程图

实验步骤:

- 1) 编写实验程序，经编译、链接无误后装载到实验系统；
- 2) 待转换数据存放于数据段 0000h~0007h 单元，可根据需求另行输入；
- 3) 运行程序，等待程序结束；
- 4) 查看数据段 0008h 起始的单元，即为转换结果：00h, 01h, 08h, 09h, 02h, 03h, 06h；
- 5) 程序遇到非“0”~“9”的 ASCII 码，则自动跳过，继续转换下一个单元；
- 6) 反复测试几组数据，验证程序的正确性。

4. 十进制数转换为 ASCII 码

参考程序 (A86\2-3-4.ASM):

```

DATA    SEGMENT
NUMS    DB 01H,02H,03H,0AH,05H,06H,07H,08H
DLEN    = $ - NUMS
NUMO    DB 7 DUP(?)
DATA     ENDS

```

```

CODE    SEGMENT
        ASSUME CS:CODE,DS:DATA
START   PROC NEAR
        MOV AX,DATA           ;DS 指向数据段
        MOV DS,AX
        MOV CX,DLEN
        MOV SI,OFFSET NUMS    ;十进制数源地址
        MOV DI,OFFSET NUMO    ;ASCII 目标地址
A1:      MOV AL,[SI]
        AND AL,0FH            ;取低 4 位
        ADD AL,30H            ;转换为相应 ASCII 码
        CMP AL,3AH
        JB A2                 ;小于 0AH 则转 A2
        INC SI                 ;大于 09H 地址跳过
        DEC CX                 ;大于 09H 计数减 1
        JMP A1
A2:      MOV [DI],AL           ;结果存入目标地址
        INC DI
        INC SI
        LOOP A1
        MOV AH,4CH            ;程序结束
        INT 21H
START   ENDP
CODE    ENDS
        END START

```

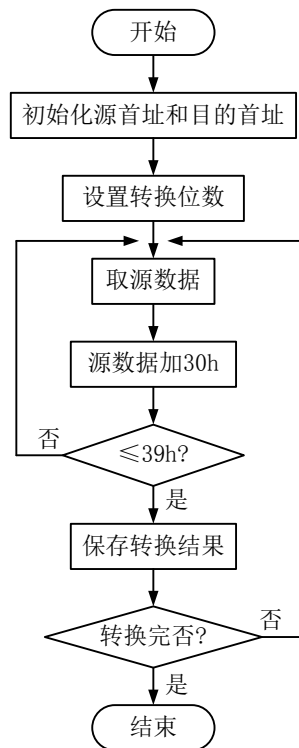


图2-3-4 十进制转ASCII流程图

实验步骤:

- 1) 编写实验程序，经编译、链接无误后装载到实验系统；
- 2) 待转换数据存放于数据段 0000h~0007h 单元，可根据需求另行输入；
- 3) 运行程序，等待程序结束；
- 4) 查看数据段 0008h 起始的单元，即为转换结果：31h, 32h, 33h, 35h, 36h, 37h, 38h；
- 5) 程序遇到非“0”~“9”数值的 ASCII 码，则自动跳过，继续转换下一个单元；
- 6) 反复测试几组数据，验证程序的正确性。

5. 十进制数的 ASCII 码转换为 BCD 码

参考程序 (A862-3-5.ASM):

```

DATA    SEGMENT
NUMS     DB 30H,31H,41H,38H,39h,32h,33H,36H
NUMO     DB 7 DUP(?)
DLEN     = $ - NUMO

```

```

DATA    ENDS

CODE    SEGMENT
        ASSUME CS:CODE, DS:DATA
START   PROC NEAR
        MOV AX, DATA          ;DS 指向数据段
        MOV DS, AX
        MOV CX, DLEN           ;转换位数
        MOV SI, OFFSET NUMS    ;ASCII 码首地址
        MOV DI, OFFSET NUMO    ;BCD 码首地址
A1:      MOV AL, [SI]
        SUB AL, 30H
        JC A3                  ;低于 30H 则转 A3
        CMP AL, 9
        JNG A2                 ;为"30~39"则转 A2
A3:      INC SI                 ;ASCII 码地址加一
        JMP A1                 ;转换下一个
A2:      MOV [DI], AL          ;结果存入目标地址
        INC DI
        INC SI
        LOOP A1
        MOV AH, 4CH
        INT 21H
START   ENDP
CODE    ENDS
        END START

```

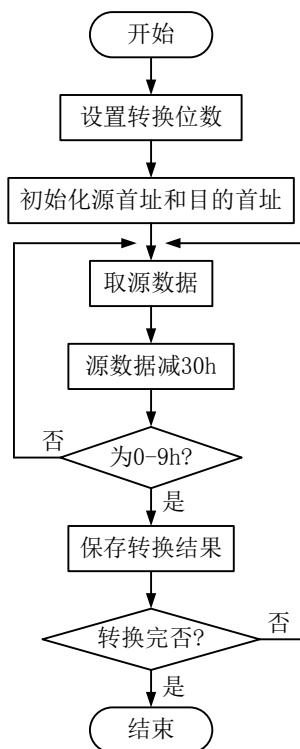


图2-3-5 十进制ASCII码转BCD流程图

实验步骤:

- 1) 编写实验程序，经编译、链接无误后装载到实验系统；
- 2) 待转换数据存放于数据段 0000h~0007h 单元，可根据需求另行输入；
- 3) 运行程序，等待程序结束；
- 4) 查看数据段 0008h 起始的单元，即为转换结果：00h, 01h, 08h, 09h, 02h, 03h, 06h；
- 5) 程序遇到非“0”~“9”数值的 ASCII 码，则自动跳过，继续转换下一个单元；
- 6) 反复测试几组数据，验证程序的正确性。

6. 十进制 BCD 码转换为二进制数

参考程序 (A86\2-3-6.ASM):

```

DATA    SEGMENT
NUMS    DB 08H, 07H, 06H, 05H, 04H, 03H, 02H, 01H
NUMO    DB 4 DUP(?)

```

```

DLEN      = $ - NUMO
DATA      ENDS

CODE      SEGMENT
          ASSUME CS:CODE, DS:DATA
START     PROC NEAR
          MOV AX, DATA          ; DS 指向数据段
          MOV DS, AX
          XOR AX, AX
          MOV CX, DLEN          ; 转换位数
          MOV SI, OFFSET NUMS    ; BCD 码首地址
          MOV DI, OFFSET NUMO    ; 二进制数首地址
A1:        MOV AL, [SI]          ; 取当前 BCD 码
          ADD AL, AL
          MOV BL, AL
          ADD AL, AL
          ADD AL, AL
          ADD AL, BL
          INC SI
          ADD AL, [SI]
          MOV [DI], AL
          INC SI
          INC DI
          LOOP A1
          MOV AH, 4CH            ; 程序结束
          INT 21H

START     ENDP
CODE      ENDS
          END START

```

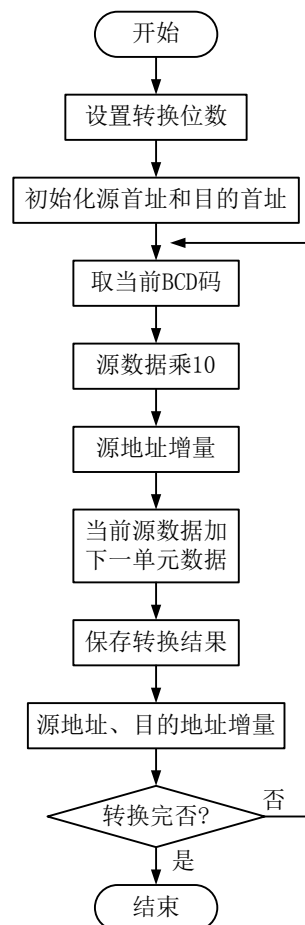


图2-3-6 十进制BCD码转换为二进制数

实验步骤:

- 1) 编写实验程序，经编译、链接无误后装载到实验系统；
- 2) 待转换数据存放于数据段 0000h~0007h 单元，可根据需求另行输入；
- 3) 运行程序，等待程序结束；
- 4) 查看数据段 0008h 起始的单元，即为转换结果：57h，41h，2Bh，15h；
- 5) 反复测试几组数据，验证程序的正确性。

实验四 运算类编程实验

一、实验目的

1. 掌握使用运算类指令编程及调试方法；
2. 掌握运算类指令对各状态标志位的影响及其测试方法；
3. 学习使用软件观察变量的方法。

二、实验设备

PC 计算机一台，Dais-86PRO+或 Dais-PRO163C 实验系统一套。

三、实验内容及步骤

80X86 指令系统提供了实现加、减、乘、除运算的基本指令，可对表 2-4-1 所示的数据类型进行算术运算。

表 2-4-1 数据类型算术运算表

数制	二进制		BCD 码	
	有符号	无符号	组合	非组合
运算符	+、-、×、÷		+	+
操作数	字节、字、多精度		字节（2 位数字）	字节（1 位数字）

1. 二进制双精度加法运算

计算 $X+Y=Z$ ，将结果 Z 存入某存储单元。实验程序参考如下。

本实验是双精度（2 个 16 位，即 32 位）加法运算，编程时可利用累加器 AX ，先求低 16 位的和，并将运算结果存入低地址存储单元，然后求高 16 位的和，将结果存入高地址存储单元中。由于低 16 运算后可能向高位产生进位，因此高 16 位运算时使用 ADC 指令，这样在低 16 位相加运算有进位时，高位相加会加上 CF 中的 1。

参考程序（A86\2-4-1.ASM）：

```
DATA    SEGMENT
XH      DW 0015H    ;X 低位
XL      DW 65A0H    ;X 高位
YH      DW 0021H    ;Y 低位
YL      DW 0B79EH   ;Y 高位
ZH      DW ?        ;Z 低位
ZL      DW ?        ;Z 高位
DATA    ENDS
```

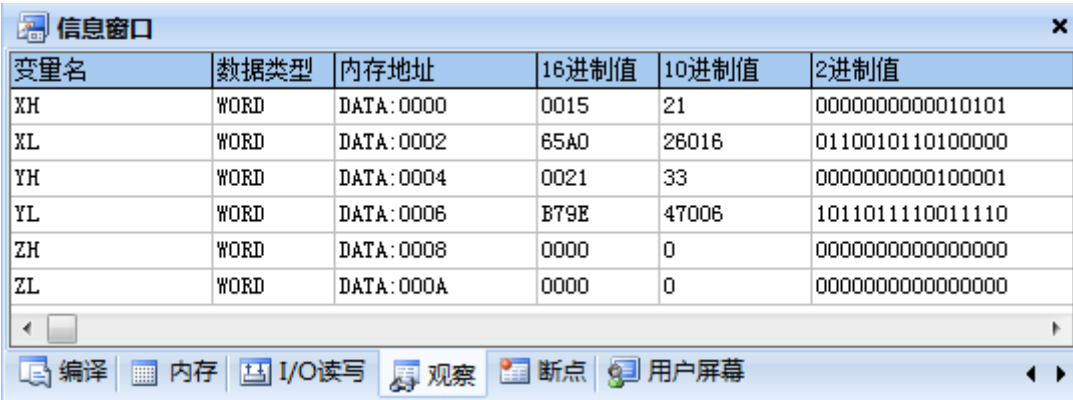
```

CODE    SEGMENT
        ASSUME CS:CODE,DS:DATA
START   PROC NEAR
        MOV AX,DATA
        MOV DS,AX
        MOV AX,XL
        ADD AX,YL    ;X 低位加 Y 低位
        MOV ZL,AX    ;低位和存到 z 的低位
        MOV AX,XH
        ADC AX,YH    ;高位带进位加
        MOV ZH,AX    ;存高位结果
        MOV AH,4CH
        INT 21H
START   ENDP
CODE    ENDS
        END START

```

实验步骤:

- 1) 编写实验程序，经编译、链接无误后装载到实验系统；
- 2) 程序装载完成后，点击“信息窗口”的“观察”标签，切换到变量观察窗口，此时 XH=0015h, XL=65A0h, YH=0021h, YL=B79Eh（如图 2-4-1 所示）；



变量名	数据类型	内存地址	16进制值	10进制值	2进制值
XH	WORD	DATA:0000	0015	21	00000000000010101
XL	WORD	DATA:0002	65A0	26016	0110010110100000
YH	WORD	DATA:0004	0021	33	00000000000100001
YL	WORD	DATA:0006	B79E	47006	1011011110011110
ZH	WORD	DATA:0008	0000	0	00000000000000000
ZL	WORD	DATA:000A	0000	0	00000000000000000

图 2-4-1 程序装载后的观察框信息

- 3) 运行程序，等待程序结束；
- 4) 当程序停止运行后，查看变量观察窗口，计算结果 ZH=0037h, ZL=1D3Eh（如图 2-4-2 所示）；

变量名	数据类型	内存地址	16进制值	10进制值	2进制值
XH	WORD	DATA:0000	0015	21	00000000000010101
XL	WORD	DATA:0002	65A0	26016	0110010110100000
YH	WORD	DATA:0004	0021	33	00000000000100001
YL	WORD	DATA:0006	B79E	47006	1011011110011110
ZH	WORD	DATA:0008	0037	55	00000000000110111
ZL	WORD	DATA:000A	1D3E	7486	0001110100111110

图 2-4-2 程序运行后的观察框信息

- 5) 修改 XH、XL、YH、YL 的值，重复实验步骤 1~4，观察实验结果。反复测试几组数据，验证程序的正确性。

2. 十进制的 BCD 码减法运算

计算 $X - Y = Z$ ，其中 X、Y、Z 为 BCD 码。

参考程序 (A86\2-4-2.ASM):

```

DATA    SEGMENT
X        DW 0400H      ;40
Y        DW 0102H      ;12
Z        DW ?
DATA    ENDS

CODE    SEGMENT
        ASSUME CS:CODE,DS:DATA
START   PROC NEAR
        MOV AX,DATA
        MOV DS,AX
        MOV AH,00H
        SAHF
        MOV CX,0002H
        MOV SI,OFFSET X
        MOV DI,OFFSET Z
A1:     MOV AL,[SI]
        SBB AL,[SI+02H]
        DAS
        PUSHF
        AND AL,0FH

```

```
        POPF
        MOV [DI],AL
        INC DI
        INC SI
        LOOP A1
        MOV AH,4CH
        INT 21H
START    ENDP
CODE     ENDS
        END START
```

实验步骤:

- 1) 编写实验程序, 经编译、链接无误后装载到实验系统;
- 2) 程序装载完成后, 点击“信息窗口”的“观察”标签, 切换到变量观察窗口, 此时 X=0400h, Y=0102h (如图 2-4-3 所示);



变量名	数据类型	内存地址	16进制值	10进制值	2进制值
X	WORD	DATA:0000	0400	1024	0000010000000000
Y	WORD	DATA:0002	0102	258	0000000100000010
Z	WORD	DATA:0004	0000	0	0000000000000000

图 2-4-3 程序装载后的观察框信息

- 3) 运行程序, 等待程序结束;
- 4) 当程序停止运行后, 查看变量观察窗口, 计算结果 Z=0208h (如图 2-4-2 所示);



变量名	数据类型	内存地址	16进制值	10进制值	2进制值
X	WORD	DATA:0000	0400	1024	0000010000000000
Y	WORD	DATA:0002	0102	258	0000000100000010
Z	WORD	DATA:0004	0208	520	0000001000001000

图 2-4-4 程序运行后的观察框信息

- 5) 修改 X、Y 的值，重复实验步骤 1~4，观察实验结果。反复测试几组数据，验证程序的正确性。

3. 乘法运算

实现十进制数的乘法运算，被乘数与乘数均以 BCD 码的形式存放在内存中，乘数为 1 位，被乘数为 5 位，结果为 6 位。

参考程序 (A86\2-4-3.ASM):

```
DATA    SEGMENT
DATA1   DB 1,2,3,4,5      ;被乘数
DATA2   DB 2              ;乘数
RESULT  DB 6 DUP(?)       ;计算结果
DATA    ENDS

CODE    SEGMENT
        ASSUME CS:CODE,DS:DATA
START   PROC NEAR
        MOV AX,DATA
        MOV DS,AX
        CALL INIT          ;初始化目标地址单元为 0
        MOV SI,OFFSET DATA2
        MOV BL,[SI]
        AND BL,0FH         ;得到乘数
        CMP BL,09H
        JNC ERROR
        MOV SI,OFFSET DATA1
        MOV DI,OFFSET RESULT
        MOV CX,0005H
A1:     MOV AL,[SI+04H]
        AND AL,0FH
        CMP AL,09H
        JNC ERROR
        DEC SI
        MUL BL
        AAM                ;乘法调整指令
        ADD AL,[DI+05H]
        AAA
        MOV [DI+05H],AL
        DEC DI
        MOV [DI+05H],AH
```

```
        LOOP A1
A2:      MOV AH,4CH
        INT 21H          ;程序终止
;错误处理
ERROR:   MOV SI,OFFSET RESULT ;若输入数据不符合要求
        MOV CX,0003H      ;则 RESULT 所指向内存单元
        MOV AX,0EEEEH     ;元全部写入 E
A4:      MOV [SI],AX
        INC SI
        INC SI
        LOOP A4
        JMP A2
START    ENDP

;将 RESULT 所指内存单元清零
INIT     PROC NEAR
        MOV SI,OFFSET RESULT
        MOV CX,0003H
        MOV AX,0000H
A3:      MOV [SI],AX
        INC SI
        INC SI
        LOOP A3
        RET
INIT     ENDP

CODE     ENDS
        END START
```

实验步骤:

- 1) 编写实验程序，经编译、链接无误后装载到实验系统；
- 2) 乘数存放于数据段 0000h~0004h 单元，程序中默认为 01h, 02h, 03h, 04h, 05h；
- 3) 被乘数存放于数据段 0005h 单元，程序中默认为 02h；
- 4) 运行程序，等待程序结束；
- 5) 查看数据段 0006h~000Bh 单元，应为 00h, 02h, 04h, 06h, 09h, 00h；在为被乘数和乘数赋值时，如果一个数的低 4 位大于 9，则数据段 0006~000Bh 单元均为 EAh。
- 6) 反复测试几组数据，验证程序的正确性。

实验五 分支程序设计实验

一、实验目的

1. 掌握分支程序的结构；
2. 掌握分支程序的设计、调试方法。

二、实验设备

PC 计算机一台，Dais-86PRO+或 Dais-PRO163C 实验系统一套。

三、实验内容及步骤

设计一数据块间的搬移程序。设计思想：程序要求把内存中一数据区（称为源数据块）传送到另一存储区（称为目的数据块）。源数据块和目的数据块在内存中可能有如下三种情况，如图 2-5-1 所示。

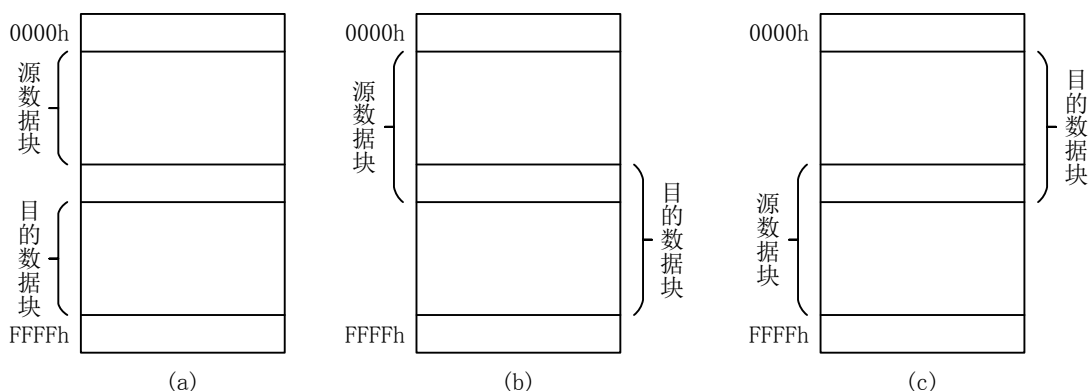


图2-5-1 源数据块与目的数据块在存储中的位置情况

对于两个数据块分离的情况，如图 2.5-1(a)，数据的传送从数据块的首地址开始，或从数据块的末地址开始均可。但是对于有重叠的情况，则要加以分析，否则重叠部分会因“搬移”而遭到破坏，可有如下结论：

当源数据块首地址 < 目的块首地址时，从数据块末地址开始传送数据，如图 2-5-1(b)所示。

当源数据块首地址 > 目的块首地址时，从数据块首地址开始传送数据，如图 2-5-1(c)所示。

实验程序流程图如图 2-5-2 所示。

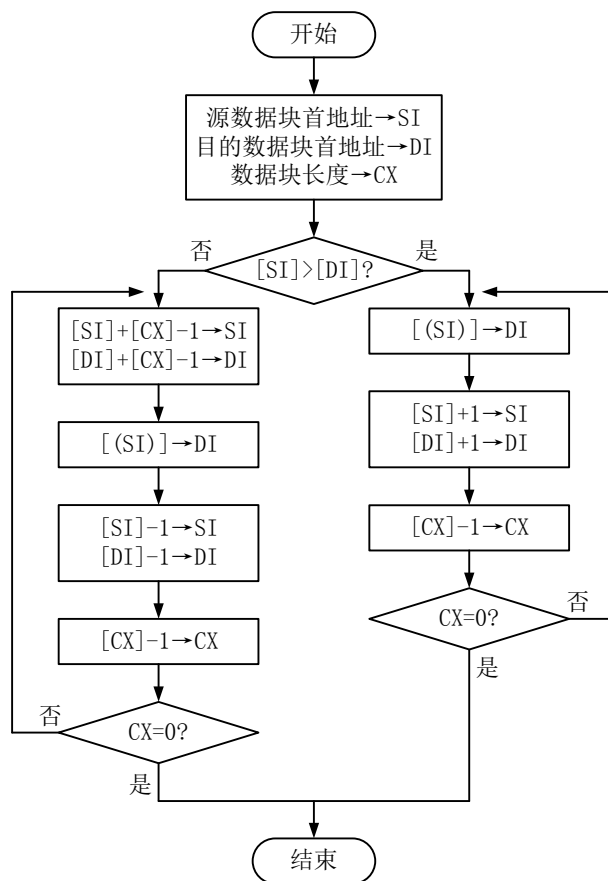


图2-5-2 分支程序流程图

参考程序 (A86\2-5.ASM):

```

DATA    SEGMENT
BUFS     DB  "www.MERKE.com.cn"
DLEN     = $ - BUFS
BUFD     DB  16 DUP (?)
DATA     ENDS

CODE     SEGMENT
        ASSUME CS:CODE, DS:DATA
START    PROC NEAR
        MOV AX, DATA
        MOV DS, AX
        MOV SI, OFFSET BUFS ;源地址
        MOV DI, OFFSET BUFD ;目的地址
        MOV CX, DLEN
  
```



```
        CMP SI,DI
        JA  A2
        ADD SI,CX
        ADD DI,CX
        DEC SI
        DEC DI
A1:      MOV AL,[SI]
        MOV [DI],AL
        DEC SI
        DEC DI
        DEC CX
        JNE A1
        JMP A3
A2:      MOV AL,[SI]
        MOV [DI],AL
        INC SI
        INC DI
        DEC CX
        JNE A2
A3:      MOV AH,4CH
        INT 21H           ;程序终止
START    ENDP
CODE     ENDS
        END START
```

实验步骤:

- 1) 按流程图编写实验程序,经编译、链接无误后装载到实验系统;
- 2) 源数据块存放于数据段 0000h~000Fh 单元,可根据需求另行输入;
- 3) 目标数据块位于数据段 0010h~001Fh 单元,初始时内容为空;
- 4) 运行程序,待程序运行停止;
- 5) 查看数据段 0010h~001Fh 单元,应与 0000h~000Fh 单元一致;
- 6) 通过改变 SI、DI 的值,观察在三种不同的数据块情况下程序的运行情况,并验证程序的正确性。

实验六 循环程序设计实验

一、实验目的

1. 加深对循环结构的理解;
2. 掌握循环结构程序设计的方法以及调试方法。

二、实验设备

PC 计算机一台, Dais-86PRO+或 Dais-PRO163C 实验系统一套。

三、实验内容及步骤

1. 计算 $S=1+2\times 3+3\times 4+4\times 5+\dots+N(N+1)$

编写实验, 计算上式的结果, 直到 $N(N+1)$ 项大于 200 为止, 参考流程图如图 2-6-1 所示。

参考程序 (A86\2-6-1.ASM):

```
DATA    SEGMENT
RESULT  DW ?           ;计算结果
DATA    ENDS
CODE    SEGMENT
        ASSUME CS:CODE, DS:DATA
START   PROC NEAR
        MOV DX, 0001H
        MOV BL, 02H
A1:     MOV AL, BL
        INC BL
        MUL BL
        ADD DX, AX      ;结果存于 DX 中
        CMP AX, 00C8H   ;判断 N(N+1) 与 200 的大小
        JNA A1
        MOV AX, DATA   ;保存计算结果
        MOV DS, AX
        MOV RESULT, DX
        MOV AH, 4CH
        INT 21H         ;程序终止
START   ENDP
CODE    ENDS
        END START
```

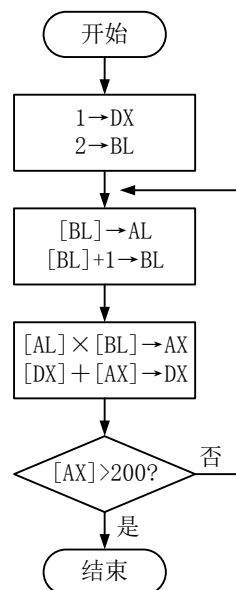


图2-6-1 计算 $S=1+2\times 3+3\times 4+4\times 5+\dots+N(N+1)$ 程序流程图

实验步骤:

- 1) 编写实验程序, 编译、链接无误后装载到实验系统;
- 2) 运行程序, 等待程序结束;
- 3) 运算结果存放在数据段 0000h~0001h 单元, 可使用内存或观察窗口查看结果是否正确;
- 4) 可以改变 N (N+1) 的条件来验证程序功能是否正确, 但要注意, 结果若大于 0FFFFH 将产生数据溢出。

2. 求某数据区内负数的个数

设数据区的第一单元存放区内单元数据的个数, 从第二单元开始存放数据, 在区内最后一个单元存放结果。为统计数据区内负数的个数, 需要逐个判断区内的每一个数据, 然后将所有数据中凡是符号位为 1 的数据的个数累加起来, 即得到区内所包含负数的个数。

实验程序流程图如图 2-6-2 所示。

参考程序 (A86\2-6-2.ASM):

```

DATA    SEGMENT
BUF      DB 06H,12H,88H,82H,90H,22H,33H;首单元为总数
RESULT   DB ?                ;统计后的负数个数
DATA     ENDS
CODE     SEGMENT
        ASSUME CS:CODE,DS:DATA
START    PROC NEAR
        MOV AX,DATA
        MOV DS,AX
        MOV DI,OFFSET BUF ;数据区首地址
        MOV CL,[DI]       ;取数据个数
        XOR CH,CH
        MOV BL,CH
        INC DI             ;指向第一个数据
A1:      MOV AL,[DI]
        TEST AL,80H       ;检查数据首位是否为 1
        JE A2
        INC BL             ;负数个数加 1
A2:      INC DI
        LOOP A1
        MOV [DI],BL       ;保存结果
        MOV AH,4CH
        INT 21H           ;程序终止
START    ENDP
CODE     ENDS
        END START

```

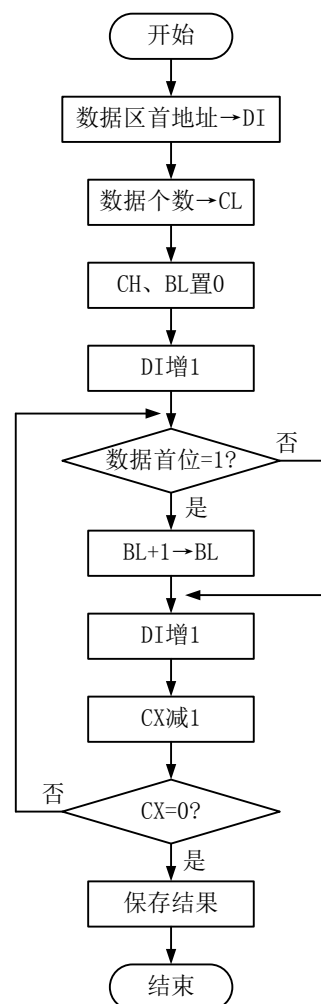


图2-6-2 求某数据区内负数的个数
程序流程图

实验步骤:

- 1) 编写实验程序, 编译、链接无误后装载到实验系统;
- 2) 统计的数据个数位于数据段 0000h 单元, 被统计的数据位于数据段 0001h~0006h 单元, 可根据需求另行输入;
- 3) 运行程序, 等待程序结束;
- 4) 统计结果存放在数据段 0007h 单元, 可使用内存或观察窗口查看结果是否正确, 本例程中应为 03h;
- 5) 反复测试几组数据, 验证程序的正确性。

实验七 排序程序设计实验

一、实验目的

1. 掌握分支、循环、子程序调用等基本的程序结构；
2. 学习综合程序的设计、编写及调试。

二、实验设备

PC 计算机一台，Dais-86PRO+或 Dais-PRO163C 实验系统一套。

三、实验内容及步骤

1. 气泡排序法

在数据区中存放着一组数，数据的个数就是数据缓冲区的长度，要求采用气泡法对该数据区中的数据按递增关系排序。

设计思想：

- 1) 从最后一个数（或第一个数）开始，依次把相邻的两个数进行比较，即第 N 个数与第 $N-1$ 个数比较，第 $N-1$ 个数与第 $N-2$ 个数比较等等；若第 $N-1$ 个数大于第 N 个数，则两者交换，否则不交换，直到 N 个数的相邻两个数都比较完为止。此时， N 个数中的最小数将被排在 N 个数的最前列。
- 2) 对剩下的 $N-1$ 个数重复 1) 这一步，找到 $N-1$ 个数中的最小数。
- 3) 再重复 2)，直到 N 个数全部排列好为止。

参考程序（A86\2-7-1.ASM）：

```
DATA    SEGMENT
BUF      DB  89H,20H,12H,64H,88H,06H,66H,78H,99H,01H
DLEN     = $ - BUF
DATA    ENDS

CODE    SEGMENT
        ASSUME CS:CODE,DS:DATA
START   PROC NEAR
        MOV AX,DATA
        MOV DS,AX
        MOV CX,DLEN
        MOV SI,OFFSET BUF+DLEN
        MOV BL,0FFH
A1:     CMP BL,0FFH
```

```
JNZ A4
MOV BL,00H
DEC CX
JZ A4
PUSH SI
PUSH CX
A2: DEC SI
    MOV AL,[SI]
    DEC SI
    CMP AL,[SI]
    JA A3
    XCHG AL,[SI]
    MOV [SI+01H],AL
    MOV BL,0FFH
A3: INC SI
    LOOP A2
    POP CX
    POP SI
    JMP A1
A4: MOV AH,4CH
    INT 21H ;程序终止
START ENDP
CODE ENDS
END START
```

实验步骤:

- 1) 分析参考程序, 绘制流程图并编写实验程序;
- 2) 编写实验程序, 经编译、链接无误后装载到实验系统;
- 3) 待排序的数据存放于数据段 0000h~000Ah 单元, 可根据需求另行输入;
- 4) 运行程序, 等待程序结束;
- 5) 查看数据段 0000h~000Ah 单元, 数据应从小到大排列;
- 6) 可以反复测试几组数据, 观察结果, 验证程序的正确性。

2. 学生成绩名次表

将分数在 1~100 之间的 10 个成绩存入数据段首地址为 0000h 的单元中, 0000h+I 表示学号为 I 的学生成绩。编写程序, 将排出的名次表放在 0100h 开始的数据区, 0100h+I 中存放的为学号为 I 的学生名次。

参考程序 (A86\2-7-2.ASM):

```

DATA    SEGMENT
        ORG 0000H

SCORES  DB 86H,90H,99H,70H,66H ;学生成绩
        DB 40H,92H,78H,98H,82H

DLEN    = $ - SCORES

        ORG 0010H

RANK    DB 10 DUP(?)
        ORG 0020H

BUF     DB 10 DUP(?)

DATA    ENDS


CODE    SEGMENT
        ASSUME CS:CODE,DS:DATA

START   PROC NEAR
        MOV AX,DATA
        MOV DS,AX
        MOV CX,DLEN
        MOV SI,OFFSET SCORES
        MOV DI,OFFSET BUF
        CALL BACKUP

        MOV SI,OFFSET SCORES ;存放学生成绩
        MOV CX,DLEN          ;共 10 个成绩
        MOV DI,OFFSET RANK  ;名次表首地址
A1:     CALL BRANCH          ;调用子程序
        MOV AL,DLEN
        SUB AL,CL
        INC AL
        MOV BX,DX
        MOV [BX+DI],AL
        LOOP A1
        MOV CX,DLEN
        MOV SI,OFFSET BUF
        MOV DI,OFFSET SCORES
        CALL BACKUP
        MOV AH,4CH

        INT 21H              ;程序终止

START   ENDP

```

;扫描成绩表，得到最高成绩者的学号

```
BRANCH PROC NEAR
    PUSH CX
    MOV CX,DLEN
    MOV AL,00H
    MOV BX,OFFSET SCORES
    MOV SI,BX
A2:    CMP AL,[SI]
    JAE A3
    MOV AL,[SI]
    MOV DX,SI
    SUB DX,BX
A3:    INC SI
    LOOP A2
    ADD BX,DX
    MOV AL,00H
    MOV [BX],AL
    POP CX
    RET
BRANCH ENDP

BACKUP PROC NEAR
B1:    MOV AL,[SI]
    MOV [DI],AL
    INC SI
    INC DI
    LOOP B1
    RET
BACKUP ENDP

CODE    ENDS
        END START
```

实验步骤:

- 1) 分析参考程序, 绘制流程图并编写实验程序;
- 2) 编写实验程序, 经编译、链接无误后装载到实验系统;
- 3) 10 个学生成绩存放于数据段 0000h~000Ah 的内存单元中;
- 4) 运行程序, 等待程序结束;
- 5) 检查数据段 0010h~001Ah 单元中的名次表是否正确。

实验八 子程序设计实验

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、子程序的嵌套、递归子程序的结构；
3. 掌握子程序的程序设计及调试方法。

二、实验设备

PC 计算机一台，Dais-86PRO+或 Dais-PRO163C 实验系统一套。

三、实验内容及步骤

1. 求无符号字节序列中的最大值和最小值

设有一字节序列，存放在数据段 0000h~0007h，长度为 08h。利用子程序的方法编程求出该序列中的最大值和最小值。程序流程图如图 2-8-1 所示。

参考程序（A86\2-8-1.ASM）：

```
DATA    SEGMENT
BUF      DB  8FH,90H,45H,0F9H,20H,23H,11H,01H
DLEN     = $ - BUF
MAX      DB  ?           ;存放最大值
MIN      DB  ?           ;存放最小值
DATA     ENDS

CODE     SEGMENT
        ASSUME CS:CODE,DS:DATA
START    PROC NEAR
        MOV AX,DATA
        MOV DS,AX
        MOV SI,OFFSET BUF ;数据区首址
        MOV CX,DLEN
        CALL BRANCH       ;调用子程序
        MOV [MAX],AH
        MOV [MIN],AL
        MOV AH,4CH
        INT 21H
START    ENDP
```

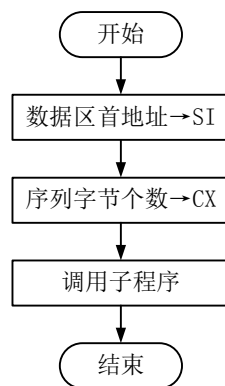


图2-8-1(a) 主程序流程图

```

BRANCH PROC NEAR    ;子程序，出口参数在 AX 中
    JCXZ A4
    PUSH SI
    PUSH CX
    PUSH BX
    MOV BH, [SI]
    MOV BL, BH
    CLD
A1:    LODSB
    CMP AL, BH
    JBE A2
    MOV BH, AL
    JMP A3
A2:    CMP AL, BL
    JAE A3
    MOV BL, AL
A3:    LOOP A1
    MOV AX, BX
    POP BX
    POP CX
    POP SI
A4:    RET
BRANCH ENDP
CODE   ENDS
      END START

```

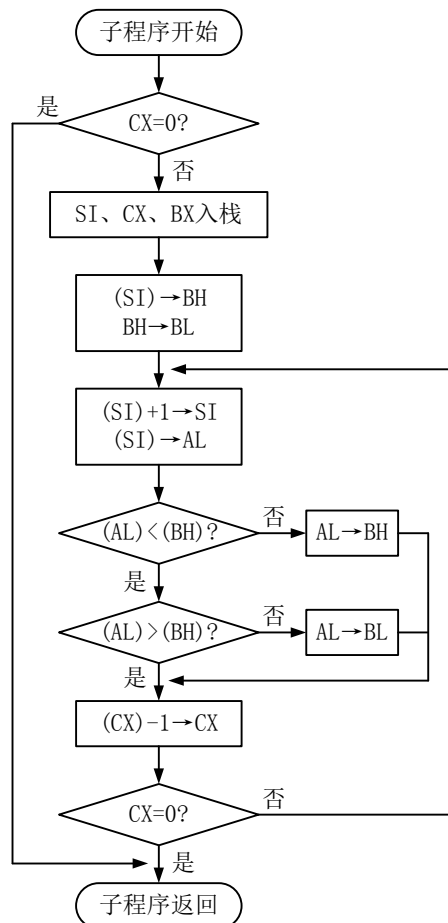


图2-8-1(b) 子程序流程图

实验步骤:

- 1) 分析参考程序，绘制流程图并编写实验程序；
- 2) 编写实验程序，经编译、链接无误后装载到实验系统；
- 3) 字节序列存放于数据段 0000h~0007h，可根据需求另行输入；
- 4) 运行程序，等待程序结束；
- 5) 最大值存放于数据段 0008h 单元，最小值存放于数据段 0009h 单元，可使用内存或观察窗口查看结果是否正确；
- 6) 反复测试几组数据，检验程序的正确性。

程序说明：该程序使用 BH 和 BL 暂存现行的最大值和最小值，开始时初始化成首字节的内容，然后进入循环操作，从字节序列中逐个取出一个字节的的内容与 BH 和 BL 相比较，若取出的字节内容比 BH 的内容大或比 BL 的内容小，则修改之。当循环操作结束时，将 BH 送 AH，将 BL 送 AL，作为返回值，同时恢复 BX 原先的内容。

2. 求 N!

利用子程序的嵌套和子程序的递归调用，实现 N! 的运算。根据阶乘运算法则，可以得：

$$N! = N(N-1)(N-2)\cdots 1$$

$$0! = 1$$

由此可知，欲求 N 的阶乘，可以用一递归子程序来实现，每次递归调用时应将调用参数减 1，即求 (N-1) 的阶乘，并且当调用参数为 0 时应停止递归调用，且有 0! = 1，最后将每次调用的参数相乘得到最后结果。因每次递归调用时参数都送入堆栈，当 N 为 0 而程序开始返回时，应按嵌套的方式逐层取出相应的调用参数。

定义两个变量 N 及 RESULT，RESULT 中存放 N! 的计算结果，N 在 00h~08h 之间取值。

参考程序 (A862-8-2.ASM)：

```
DATA    SEGMENT
N        DB 4          ;N 的范围在 1~8 之间
RESULT   DW ?          ;N! 的结果存于该变量中
DATA     ENDS

CODE     SEGMENT
        ASSUME CS:CODE, DS:DATA
START    PROC NEAR
        MOV AX, DATA
        MOV DS, AX
        MOV AX, OFFSET RESULT
        PUSH AX
        MOV AL, N
        MOV AH, 00H
        PUSH AX
        MOV DI, 0000H
        CALL BRANCH
        MOV AH, 4CH
        INT 21H
START    ENDP

BRANCH   PROC NEAR
        PUSH BP
        MOV BP, SP
        PUSH BX
        PUSH AX
        MOV BX, [BP+DI+06H]
        MOV AX, [BP+DI+04H]
        CMP AX, 0000H
```

```
JZ A1
PUSH BX
DEC AX
PUSH AX
CALL BRANCH ;递归调用
MOV BX, [BP+DI+06H]
MOV AX, [BX]
PUSH BX
MOV BX, [BP+DI+04H]
MUL BX
POP BX
JMP A2
A1: MOV AX, 0001H
A2: MOV RESULT, AX ;保存结果
POP AX
POP BX
POP BP
RET 0004H
BRANCH ENDP
CODE ENDS
END START
```

实验步骤:

- 1) 分析参考程序, 绘制流程图并编写实验程序;
- 2) 编写实验程序, 经编译、链接无误后装载到实验系统;
- 3) 变量 N 位于数据段 0000h 单元, N 在 0~8 之间取值, 本例程设 N=4;
- 4) 运行程序, 等待程序结束;
- 5) 变量 RESULT 为计算结果, 位于数据段 0001h~0002h 单元, 可使用内存或观察窗口查看结果是否正确, 本例程中应为 18h (十进制值 24);
- 6) 修改变量 N 的值, 重复实验步骤 2~4, 观察实验结果。

表 2-8 阶乘表

N!	0	1	2	3	4	5	6	7	8
16 进制	0001h	0001h	0002h	0006h	0018h	0078h	02D0h	13B0h	9D80h
10 进制	1	1	2	6	24	120	720	5040	40320

实验九 查表程序设计实验

一、实验目的

学习查表程序的设计方法。

二、实验设备

PC 计算机一台, Dais-86PRO+或 Dais-PRO163C 实验系统一套。

三、实验内容及步骤

所谓查表,就是根据某个值,在数据表格中寻找与之对应的一个数据,在很多情况下,通过查表比通过计算要使程序更简单,更容易编制。

通过查表的方法实现十六进制数转换为 ASCII 码。根据本章实验三中的表 2-3-1 可知, 0~9 的 ASCII 码为 30h~39h, 而 A~F 的 ASCII 码为 41h~46h, 这样就可以将 0~9 与 A~F 对应的 ASCII 码保存在一个数据表格中。当给定一个需要转换的十六进制数时, 就可以快速的在表格中找出相应的 ASCII 码值。

参考程序 (A86\2-9.ASM):

```
DATA    SEGMENT
TAB      DB  30H,31H,32H,33H,34H,35H,36H,37H,38H,39H ;0~9
          DB  41H,42H,43H,44H,45H,46H ;A~F
HEX      DB  9AH ;要转换的数
ASCH     DB  ? ;高4位的ASCII码
ASCL     DB  ? ;低4位的ASCII码
DATA     ENDS

CODE     SEGMENT
          ASSUME CS:CODE,DS:DATA
START    PROC NEAR
          MOV AX,DATA
          MOV DS,AX
AA1:     MOV AL,HEX ;需转换的十六进制数
          MOV AH,AL
          AND AL,0F0H
          MOV CL,04H
          SHR AL,CL
          MOV BX,OFFSET TAB ;表首地址存放于BX中
          XLAT
          MOV ASCH,AL ;存放十六进制数高4位的BCD码
```

```
        MOV AL, AH
        AND AL, 0FH
        XLAT
        MOV ASCL, AL      ;存放十六进制数低 4 位的 BCD 码
        MOV AH, 4CH
        INT 21H
START    ENDP
CODE     ENDS
        END START
```

实验步骤:

- 1) 分析参考程序，绘制流程图并编写实验程序；
- 2) 编写实验程序，经编译、链接无误后装载到实验系统；
- 3) 变量 HEX 位于数据段 0010h 单元，本例程设为 9Ah；
- 4) 运行程序，等待程序结束；
- 5) 变量 ASCH、ASCL 分别位于数据段 0011h、0012 单元；
- 6) 可使用内存或观察窗口查看 ASCH、ASCL 结果是否正确，本例程中 ASCH=39h，ASCL=41h；
- 7) 反复修改 HEX 变量，观察 ASCH 与 ASCL 的值，验证程序的正确性。

实验十 输入输出程序设计实验

一、实验目的

1. 了解常用的 INT 21H 功能调用的用途及用法；
2. 掌握 MKStudio 软件界面下数据输入和输出的方法。

二、实验设备

PC 计算机一台，Dais-86PRO+或 Dais-PRO163C 实验系统一套。

三、实验内容及步骤

INT 21H 功能调用使用说明如下：

- 1) 入口：AH=00H 或 AH=4CH
功能：程序终止
- 2) 入口：AH=01H
功能：读键盘输入到 AL 中并回显
- 3) 入口：AH=02H, DL=数据
功能：写 DL 中的数据到显示屏
- 4) 入口：AH=08H
功能：读键盘输入到 AL 中无回显
- 5) 入口：AH=09H, DS:DX=字符串首地址，字符串以“\$”结束
功能：显示字符串，直到遇到“\$”为止
- 6) 入口：AH=0AH, DS:DX=缓冲区首地址，(DS:DX)=缓冲区最大字符数，
(DS:DX+1)=实际输入字符数，(DS:DX+2)=输入字符串起始地址
功能：读键盘输入的字符串到 DS:DX 指定缓冲区中并以回车结束

1. 显示 A~Z 共 26 个大写英文字母

汇编语言参考程序（A86\2-10-1.ASM）：

```
CODE    SEGMENT
        ASSUME CS:CODE
START   PROC NEAR
        MOV CX,26           ;26 个字母
        MOV DL,41H          ;"A"的 ASCII 码值
        MOV AL,DL
A1:     MOV AH,02H
        INT 21H             ;功能调用
        INC DL              ;切换下一个字母
```

```
        DEC CX
        JNZ A1
        MOV AH, 4CH
        INT 21H          ;程序终止
START    ENDP
CODE     ENDS
        END START
```

C 语言参考程序 (A86\2-10-1.C):

```
void main()
{
    unsigned char count;
    for(count='A'; count<='Z'; count++)
    {
        _DL = count;
        _AH = 0x02;
        asm INT 0x21
    }
    _AH = 0x4C;
    asm INT 0x21
}
```

实验步骤:

- 1) 编写实验程序, 经编译、链接无误后装载到实验系统;
- 2) 运行程序, 观察实验结果;
- 3) 修改程序, 使用 AH=09h 功能 (显示字符串) 显示 “Hello World!”。

2. INT 21H 功能调用示例程序实验

参考程序 (A86\2-10-2.ASM):

```
DATA1    SEGMENT
MES1     DB "This is MERKE INT 21H!$"
DATA1    ENDS
DATA2    SEGMENT
MES2     DB 255 DUP(?)
DATA2    ENDS
CODE     SEGMENT
        ASSUME CS:CODE
START    PROC NEAR
        MOV AH, 08H
```



```
INT 21H          ;读键盘输入到 AL 中无回显
MOV AH,01H
INT 21H          ;读键盘输入到 AL 中并回显
CALL ENTERR      ;显示回车换行
MOV CX,04H       ;显示"ABCD" 4 个字母
MOV DL,41H       ;字母"A"的 ASCII 码值
AA: MOV AH,02H
INT 21H
INC DL
LOOP AA          ;将 DL 中的数据显示出来
CALL ENTERR      ;显示回车换行
MOV AX,DATA1     ;显示"This is MERKE INT 21H!"
MOV DS,AX
MOV DX,OFFSET MES1
MOV AH,09H
INT 21H
CALL ENTERR      ;显示回车换行
MOV AX,DATA2
MOV DS,AX
MOV DX,OFFSET MES2
MOV AH,0AH
INT 21H          ;读入字符串放到数据段 DATA2 中,以回车结束
ADD DX,02H
MOV AH,09H
INT 21H          ;将数据段 DATA2 中的字符串显示出来
MOV AH,4CH
INT 21H          ;程序终止
START ENDP
ENTERR PROC NEAR
MOV AH,02H
MOV DL,0DH
INT 21H          ;回车
MOV AH,02H
MOV DL,0AH
INT 21H          ;换行
RET
ENTERR ENDP
CODE ENDS
END START
```

实验步骤:

- 1) 参考实验内容中的 INT 21H 功能调用使用说明, 编写实验程序, 经编译、链接无误后装载到实验系统;
- 2) 全速运行实验程序, 观察实验结果:
 - a) 等待键盘输入 (无回显);
 - b) 等待键盘输入 (有回显);
 - c) 以字符显示方式输出 “ABCD” 字母, 并显示换行、回车;
 - d) 显示 “This is MERKE INT 21H!” 字符串;
 - e) 等待键盘输入一字符串, 以回车键结束;
 - f) 显示刚输入的字符串;
 - g) 程序结束。
- 3) 仔细分析实验内容, 理解常用的 INT 21H 功能调用的用法。

3. PC 键盘输入、实验系统数码管输出

参考程序 (A862-10-3.ASM):

```
CS8279D EQU 02F0H
```

```
CS8279C EQU 02F1H
```

```
DATA SEGMENT
```

```
BWF DW 4 DUP (?) ; 闪烁指针
```

```
BUF DB 6 DUP (?) ; 显示缓冲区
```

```
KUF DW START, START, START, START ; F2, F1, F4, F3 功能键定义
```

```
LEDMAP DB 0CH, 9FH, 4AH, 0BH, 99H, 29H, 28H, 8FH ; 01234567 字形代码
```

```
DB 08H, 09H, 88H, 38H, 6CH, 1AH, 68H, 0E8H ; 89ABCDEF 字形代码
```

```
DB 0FFH, 0C0H ; 关闭 P.
```

```
LAD DB 30H, 31H, 32H, 33H, 34H, 35H, 36H, 37H ; 01234567 ASCII 码
```

```
DB 38H, 39H, 41H, 42H, 43H, 44H, 45H, 46H ; 89ABCDEF ASCII 码
```

```
ACI DB 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H, 10H, 20H, 20H
```

```
DB 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H
```

```
DB 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H
```

```
DB 00H, 01H, 02H, 03H, 04H, 05H, 06H, 07H, 08H, 09H, 20H, 20H, 20H, 20H, 20H, 20H
```

```
DB 20H, 0AH, 0BH, 0CH, 0DH, 0EH, 0FH, 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H
```

```
DB 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H
```

```
DB 20H, 0AH, 0BH, 0CH, 0DH, 0EH, 0FH, 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H
```

```
DB 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H
```

```
DATA ENDS
```

```

CODE      SEGMENT
          ASSUME CS:CODE,DS:DATA
START     PROC NEAR
START:    MOV BUF+0,11H    ;数码管显示初始化
          MOV BUF+1,10H
          MOV BUF+2,10H
          MOV BUF+3,10H
          MOV BUF+4,10H
          MOV BUF+5,10H
          LEA BX,BUF      ;指向显示缓冲区首址
          MOV BWF,BX
XMON:     CALL DISP      ;"P."态待令入口
XMOS:     MOV AH,08H
          INT 21H
          AND AL,7FH      ;ASCII 码转换
          LEA BX,ACI      ;指字符表首
          XLAT            ;索字符代码
          CMP AL,20H      ;为 20H 视无键
          JZ XMOS         ;返无键入口
          CMP AL,10H      ;大于 10H 吗?
          JNC START       ;是命令键转
          MOV BX,BWF      ;数字键送显缓区
          MOV [BX],AL
          INC BX          ;显示缓冲区调正
          CMP BX,OFFSET BUF+6
          JC STBX
          LEA BX,BUF
STBX:     MOV BWF,BX
          JMP XMON
START     ENDP

```

;8279 显示子程序

```

DISP      PROC NEAR
          PUSH AX
          PUSH BX
          PUSH DX
          LEA DI,BUF
          MOV AH,85H
DILEX:    MOV DX,CS8279C  ;指向字位口

```

```

MOV AL,AH      ;索字位代码
OUT DX,AL      ;送字位代码
MOV BX,DI      ;指显缓地址
MOV AL,[BX]    ;取显示内容
LEA BX,LEDMAP  ;指字型表首
XLAT           ;索字形代码
MOV DX,CS8279D ;指向字形口
OUT DX,AL      ;送字型代码
INC DI
DEC AH
CMP AH,7FH
JNZ DILEX
POP DX
POP BX
POP AX
DSEND: RET
DISP ENDP

CODE ENDS
      END START

```

实验步骤:

- 1) 编写实验程序，经编译、链接无误后装载到实验系统；
- 2) 全速运行程序，切换到“用户屏幕”（人机交互窗口），鼠标点击用户屏幕的空白区，在 PC 键盘按 0~9、A~F，实验系统数码管显示对应的数字，按 Enter 清除数码管显示；
- 3) 实验完毕后，应使用暂停命令中止程序的运行。

4. 实验系统键盘输入、PC 屏幕输出

参考程序（A86\2-10-4.ASM）:

```

CS8279D EQU 02F0H
CS8279C EQU 02F1H

DATA SEGMENT
KEYCODE DB 13H,12H,11H,10H,'D','C','B','A','E','3' ;按键代码
        DB '6','9','F','2','5','8','0','1','4','7'
DATA ENDS

CODE SEGMENT
ASSUME CS:CODE,DS:DATA

```

```
START    PROC NEAR
          MOV DX,CS8279C ;指 8279 命令口
          MOV AX,0
          OUT DX,AL      ;写 8279 命令字
          MOV AL,32H
          OUT DX,AL      ;写 8279 分频字
          MOV AL,0DFH
          OUT DX,AL      ;清显示缓冲区
CLRBUF:  IN AL,DX
          TEST AL,80H
          JNZ CLRBUF     ;遇 8279 忙则转
MLOOP:   CALL INKEY
          CMP AL,20H
          JE MLOOP       ;无键继续
          JB CR          ;按下 F1~F4 功能键则回车
          MOV DL,AL      ;显示 0~F 数字键
MDISP:   CALL DISP
          JMP MLOOP
CR:      MOV DL,13
          CALL DISP
          MOV DL,10
          CALL DISP
          JMP MLOOP
START    ENDP

DISP     PROC NEAR      ;INT 21H 显示子程序
          MOV AH,02H
          INT 21H
          RET
DISP     ENDP

INKEY    PROC NEAR     ;键盘扫描子程序
          MOV DX,CS8279C ;指向状态口
          IN AL,DX      ;获取键盘标志
          AND AL,07H    ;屏蔽无效位
          JNZ KEYS      ;有键按下转
          MOV AL,20H    ;定义无键码
          RET           ;返回
KEYS:    MOV DX,CS8279D ;指向键入口
```

```
        IN AL,DX          ;读键盘接口
        MOV AH,AL
        AND AL,03H
        XCHG AL,AH
        AND AL,38H
        ROR AL,1
        OR AL,AH
        LEA BX,KEYCODE    ;键值表首
        XLAT              ;查表取键值
        RET
INKEY    ENDP

CODE     ENDS
        END START
```

实验步骤:

- 1) 编写实验程序，经编译、链接无误后装载到实验系统；
- 2) 全速运行程序，在实验系统键盘上按 0~F 数字键，PC 的“用户屏幕”（人机交互窗口）显示对应的数字，按实验系统键盘上的 F1~F4 功能键，用户屏幕插入回车换行符；
- 3) 实验完毕后，应使用暂停命令中止程序的运行。

第3章 微机接口技术及其应用实验

微机接口技术是将由微处理器、存储器等组成的基本系统与外部设备连接起来，从而实现CPU与外部设备通信的一门技术。微机的应用是随着外部设备的不断更新和接口技术的不断发展而深入到各行各业，任何微机应用开发工作都离不开接口的设计、选用及连接。微机应用系统需要设计的硬件是一些接口电路，所要编写的软件是控制这些接口电路按要求工作的驱动程序。因此，接口技术是微机应用中必不可少的基本技能。

实验一 存储器扩展实验

一、实验目的

熟悉存储器的电路构成，掌握存储器读写的编程方法。

二、实验设备

PC 计算机一台，Dais-86PRO+或 Dais-PRO163C 实验系统一套。

三、实验内容及步骤

本实验用2片6116构成 $2K \times 16$ 位的存储器单元。

编写程序，将2000~20FFh共256个存储器单元写入特定的数据，通过内存窗口查看该存储空间，检测写入数据是否正确。

实验电路：

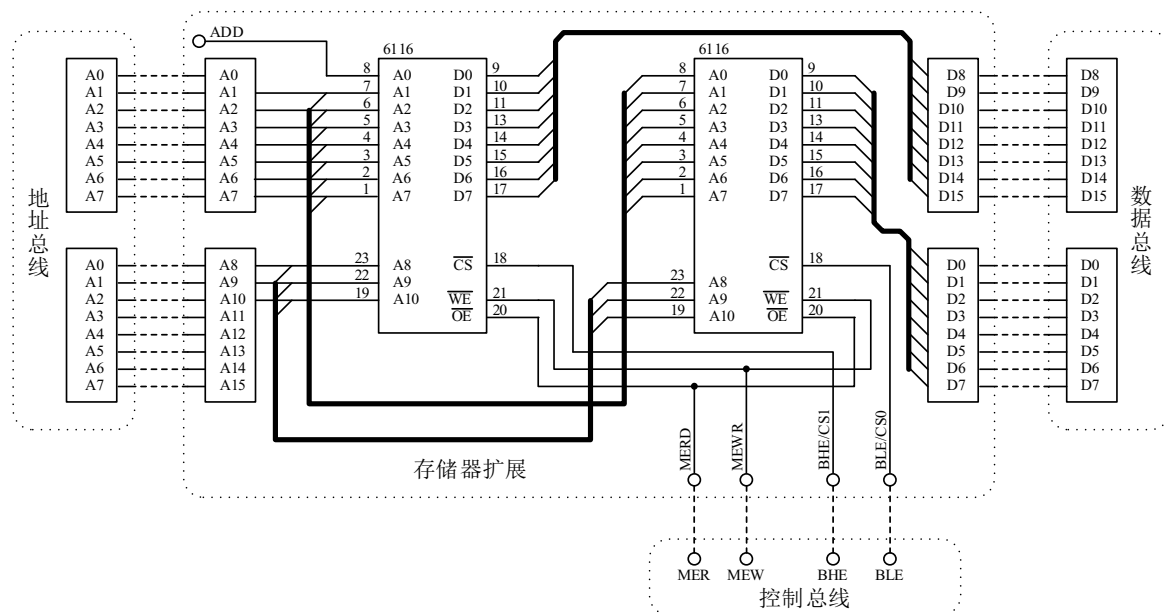


图 3-1 存储器扩展实验电路图

实验步骤:

- 1) 按图 3-1 连接实验电路, 参考程序: A86\RAM16.ASM;
- 2) 编写实验程序, 经编译、链接无语法错误后装载到实验系统;
- 3) 打开内存窗口, 并在内存窗口点鼠标右键选择“转到指定地址”, 在对话框中输入“2000”再点击确定;
- 4) 运行程序, 等待程序结束;
- 5) 检查内存窗口 2000~20FFh 共 256 个内存单元数据, 是否与程序写入的数据一致;
- 6) 修改程序, 变换写入的数据和长度, 观察实验结果。

实验二 I/O 扩展实验

一、实验目的

学习在微机接口系统中扩展简单 I/O 设备的基本方法。

二、实验设备

PC 计算机一台, Dais-86PRO+或 Dais-PRO163C 实验系统一套。

三、实验内容及步骤

1. 8 位 I/O 扩展

本实验用 74LS244 作为缓冲输入接口、74LS273 作为锁存输出接口扩展构成 8 位 I/O 接口。

编写程序, 读取 74LS244 缓冲输入接口所连接的 K7~K0 开关数据, 写入到 74LS273 锁存输出接口, 控制 L7~L0 八个发光二极管的亮灭。

实验电路:

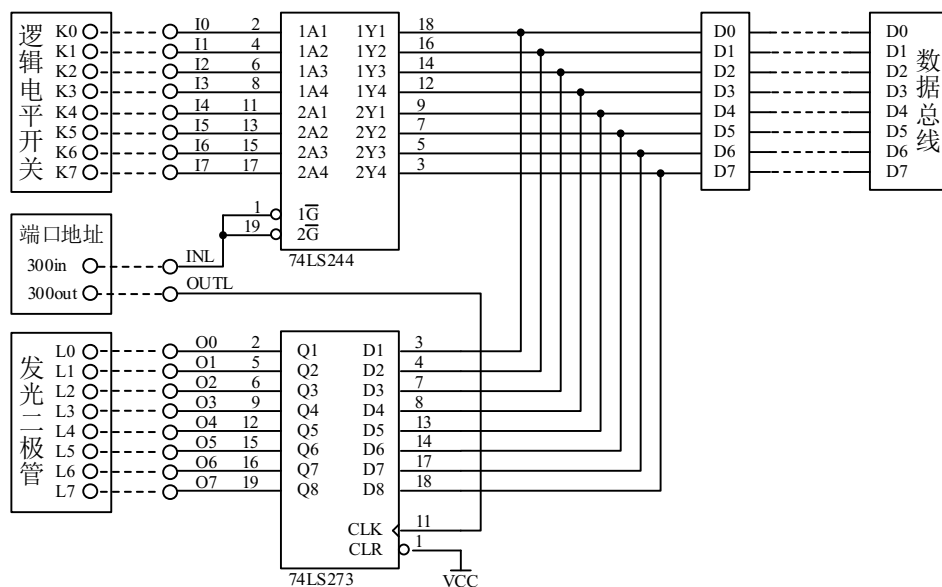


图 3-2-1 8 位 I/O 扩展实验电路图

实验步骤:

- 1) 按图 3-1 连接实验电路, 参考程序: A86\IO8.ASM;
- 2) 编写实验程序, 经编译、链接无语法错误后装载到实验系统;
- 3) 全速运行程序, 拨动开关 K7~K0, 观察发光二极管 L7~L0 状态;
- 4) 实验完毕后, 应使用暂停命令中止程序的运行。

2. 16 位 I/O 扩展

本实验用 2 片 74LS244（缓冲输入）、2 片 74LS273（锁存输出）构成 16 位 I/O 接口。

编写程序，读取 2 片 74LS244 缓冲输入接口所连接的 K15~K0 开关数据，写入到 2 片 74LS273 锁存输出接口，控制 L15~L0 八个发光二极管的亮灭。

实验电路：

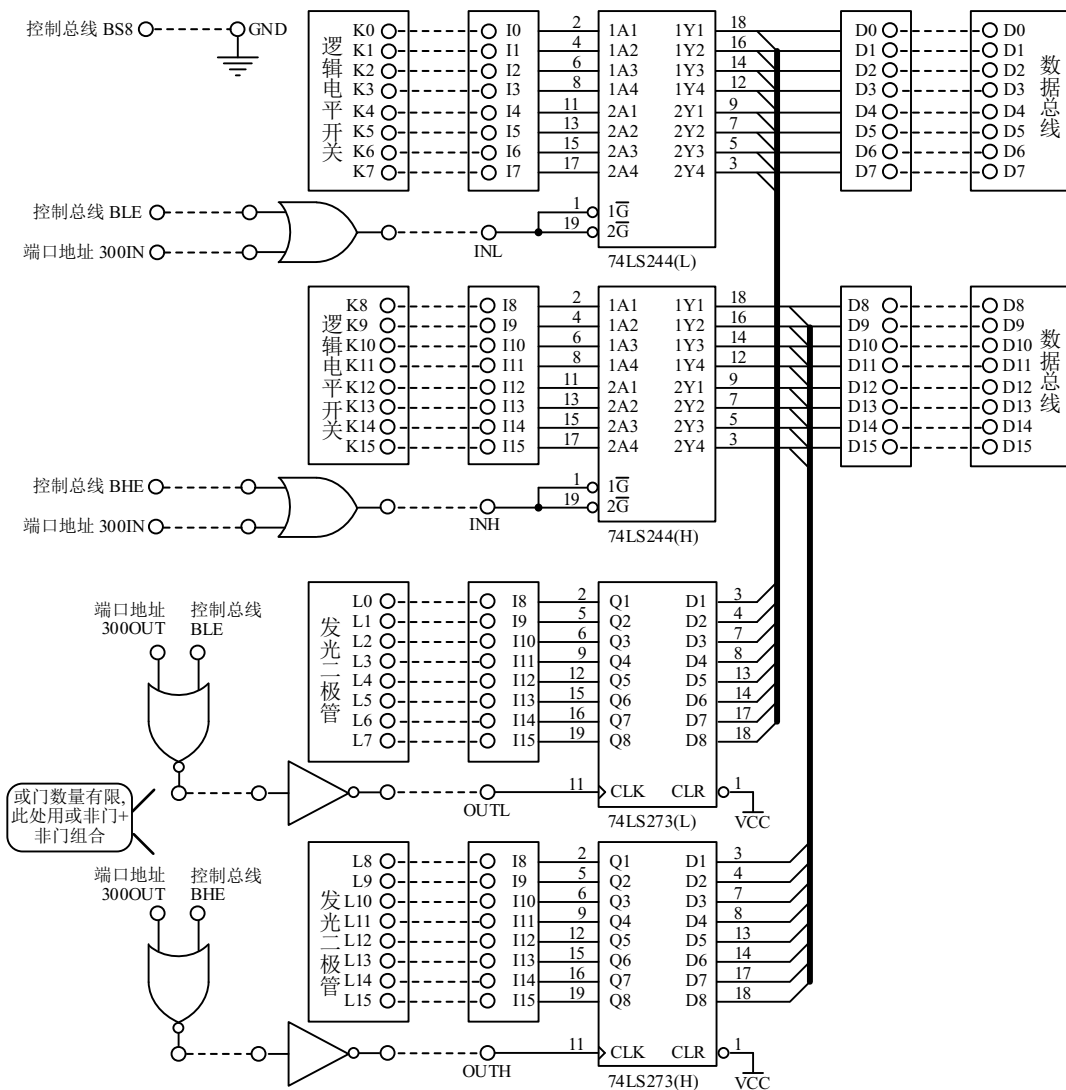


图 3-2-2 16 位 I/O 扩展实验电路图

实验步骤：

- 1) 按图 3-2-2 连接实验电路，参考程序：A86IO16.ASM；
- 2) 编写实验程序，经编译、链接无语法错误后装载到实验系统；
- 3) 全速运行程序，拨动开关 K15~K0，观察发光二极管 L15~L0 状态；
- 4) 实验完毕后，应使用暂停命令中止程序的运行。

实验三 8255 并行口实验

一、实验目的

学习 8255 的工作方式，掌握 8255 典型应用电路和输入 / 输出程序的设计方法。

二、实验设备

PC 计算机一台，Dais-86PRO+或 Dais-PRO163C 实验系统一套。

三、实验内容及步骤

1. 8255 A/B/C 口输出方波

编写程序，使 8255 的 PA、PB、PC 口每一位能循环输出高低电平。

实验电路：

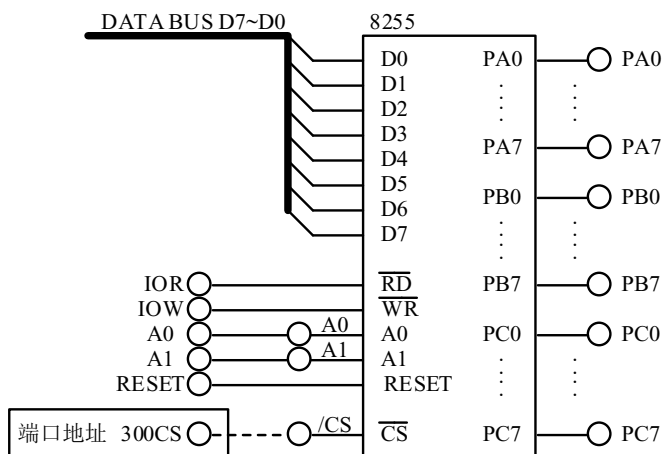


图 3-3-1 8255 输出方波实验电路图

实验步骤：

- 1) 按图 3-3-1 连接实验电路，参考程序：A86\8255_1.ASM；
- 2) 编写实验程序，经编译、链接无语法错误后装载到实验系统；
- 3) 全速运行程序，将一发光二极管接入 8255 PA、PB、PC 任意一位端口，应有循环亮灭；也可用示波器测量，应输出方波；
- 4) 实验完毕后，应使用暂停命令中止程序的运行。

2. 8255 PA 输入 / PB 输出

编写程序，读取 PA7~PA0 所连接的 K7~K0 开关数据，写入到 PB7~PB0，控制 L7~L0 八个发光二极管的亮灭。

实验电路：

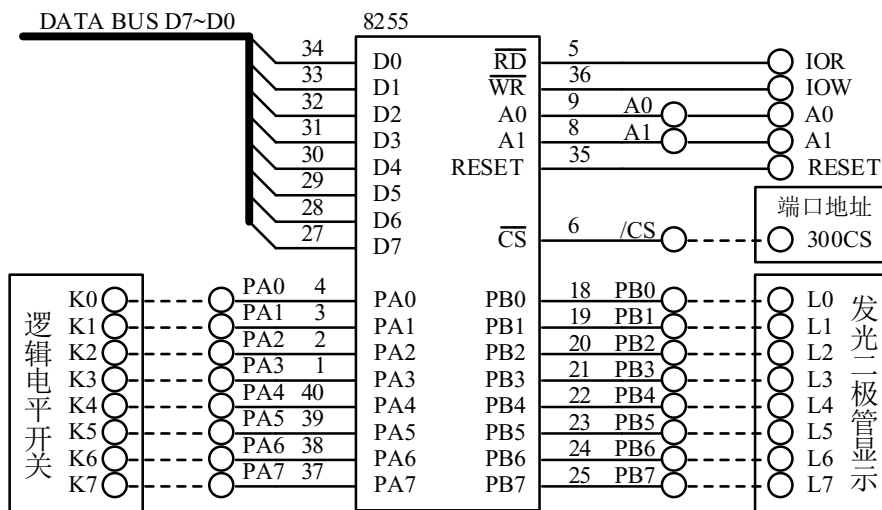


图 3-3-2 8255 输入输出实验电路图

实验步骤：

- 1) 按图 3-3-2 连接实验电路，参考程序：A86\8255_2.ASM；
- 2) 编写实验程序，经编译、链接无语法错误后装载到实验系统；
- 3) 全速运行程序，拨动开关 K7~K0，观察发光二极管 L7~L0 状态；
- 4) 实验完毕后，应使用暂停命令中止程序的运行。

3. 8255 控制交通灯

用 8255 作输出口，控制 12 个发光二极管亮灭，模拟交通灯管理：

- 程序初始时为 A 路口绿灯亮、B 路口红灯亮；
- 延迟一段时间后，A 路口由绿灯亮变为黄灯闪烁；
- 接着 A 路口红灯亮、B 路口绿灯亮；
- 延迟一段时间后，B 路口由绿灯亮变为黄灯闪烁；
- 最后循环至初始时继续。

实验电路：

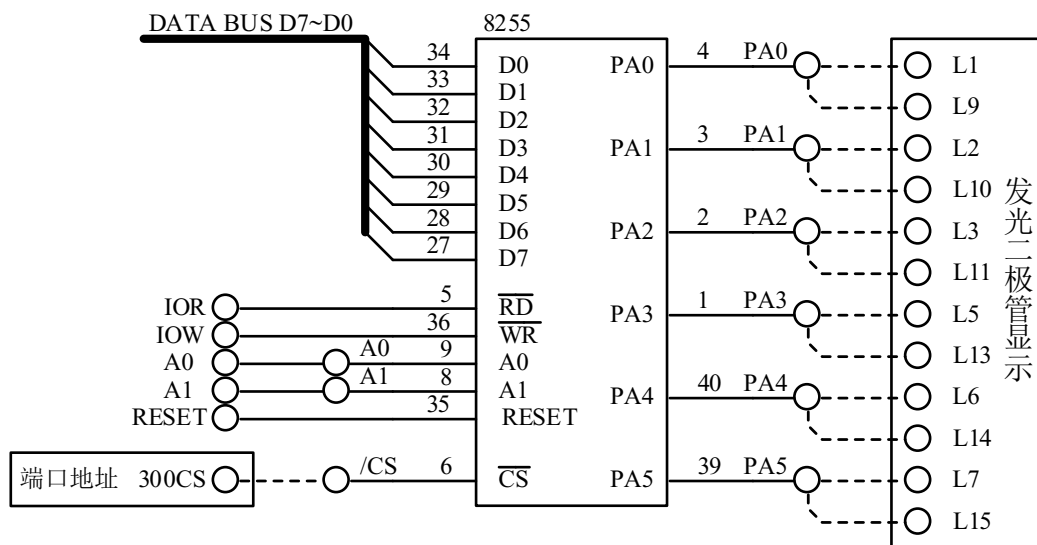


图 3-3-3 8255 控制交通灯实验电路图

实验步骤：

- 1) 按图 3-3-3 连接实验电路，参考程序：A86\8255_3.ASM；
- 2) 编写实验程序，经编译、链接无语法错误后装载到实验系统；
- 3) 全速运行程序，观察发光二极管显示，应能循环模拟交通灯显示；
- 4) 实验完毕后，应使用暂停命令中止程序的运行。

实验四 A/D 0809 模数转换实验

一、实验目的

了解模 / 数转换基本原理，掌握 ADC0809 的使用方法。

二、实验设备

PC 计算机一台，Dais-86PRO+或 Dais-PRO163C 实验系统一套。

三、实验内容及步骤

利用实验系统上的 ADC0809 作为 A/D 转换器，实验系统的电位器提供模拟量输入，编制程序，将模拟量转换成数字量并显示。

实验电路：

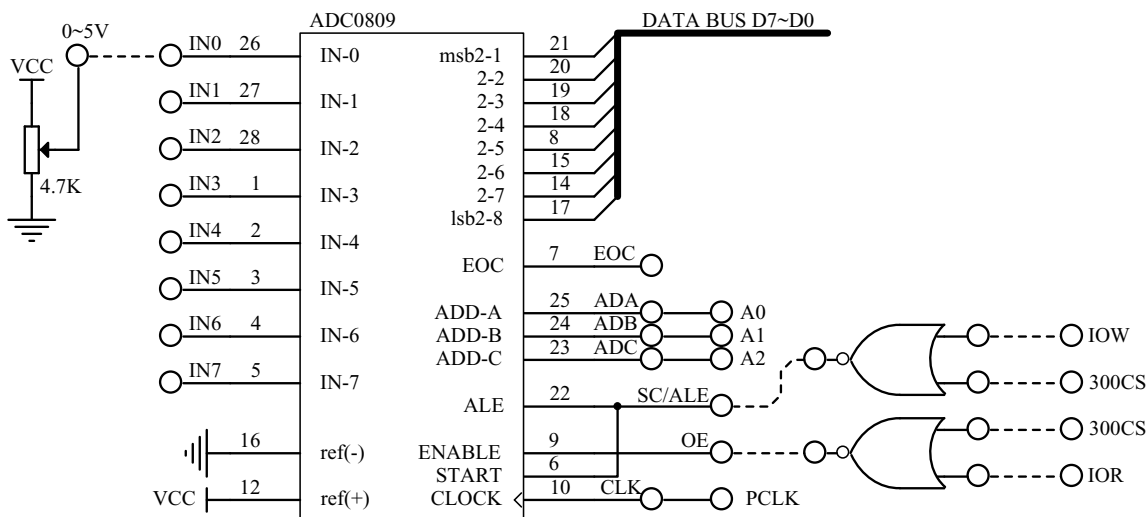


图 3-4 A/D 0809 模数转换实验电路图

实验步骤：

- 1) 按图 3-4 连接实验电路，参考程序：A86\ADC0809.ASM；
- 2) 编写实验程序，经编译、链接无语法错误后装载到实验系统；
- 3) 全速运行程序，调节 0~5V 模拟电压，观察数码管显示的 A/D 转换值；
- 4) 实验完毕后，应使用暂停命令中止程序的运行。

实验五 D/A 0832 数模转换实验

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、实验设备

PC 计算机一台，Dais-86PRO+或 Dais-PRO163C 实验系统一套。

三、实验内容及步骤

编制程序，利用 0832 芯片输出锯齿波。

实验电路：

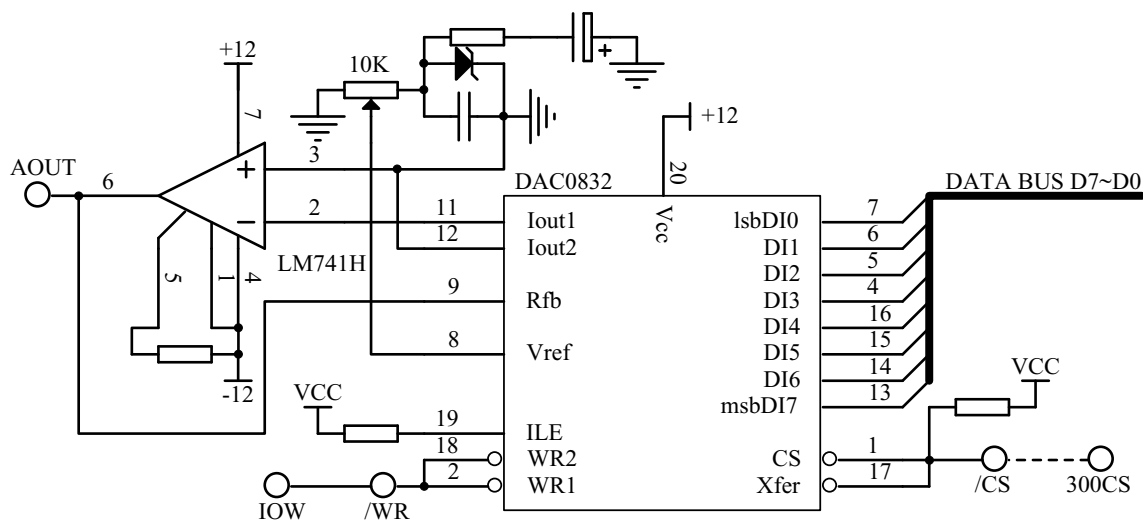


图 3-5 D/A 0832 数模转换实验电路图

实验步骤：

- 1) 按图 3-5 连接实验电路，参考程序：A86\DAC0832.ASM；
- 2) 编写实验程序，经编译、链接无语法错误后装载到实验系统；
- 3) 全速运行程序，用示波器测量 D/A 输出端 AOUT，应有锯齿波输出；
- 4) 实验完毕后，应使用暂停命令中止程序的运行；
- 5) 修改程序，变换 D/A 转换数据，输出三角波、方波等。

实验六 键盘与显示设计实验

一、实验目的

了解键盘扫描、数码显示的基本原理，掌握接口电路的设计与编程方法。

二、实验设备

PC 计算机一台，Dais-86PRO+或 Dais-PRO163C 实验系统一套。

三、实验内容及步骤

1. 8279 键盘与显示设计

编制程序，利用 8279 对 4×5 键盘进行扫描和键值读取，将键值显示到 6 位数码管上。

实验电路：

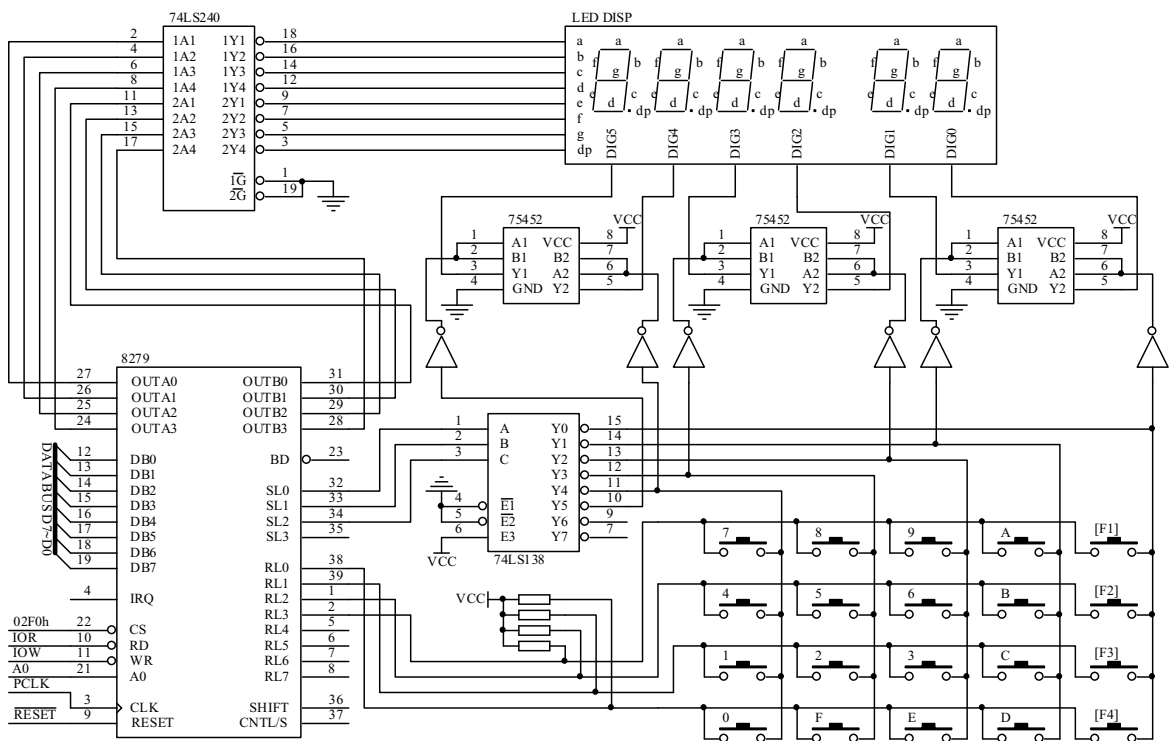


图 3-6-1 8279 键盘与显示设计实验电路图

实验步骤:

- 1) 实验电路如图 3-6-1 所示, 参考程序: A86\8279.ASM;
- 2) 用 14 芯扁平电缆连接“键盘动态扫描电路”单元的行列接口(该组接口位于 4×5 键盘的右上方, 出厂时默认已连接);
- 3) 用 14 芯扁平电缆连接“七段显示器动态扫描电路”单元的段位接口(该组接口位于 6 位数码管的右下方, 出厂时默认已连接);
- 4) 编写实验程序, 经编译、链接无语法错误后装载到实验系统;
- 5) 全速运行程序, 按实验系统键盘上的 0~F 数字键, 数码管显示对应数字; 按 F1~F4 功能键, 清除数码管显示;
- 6) 实验完毕后, 应使用暂停命令中止程序的运行。

2. 8255 键盘与显示设计

本实验使用 8255 的 PA7~PA0 控制数码管字形口、PB5~PB0 控制数码管字位口, 同时 PB4~PB0 作为键盘扫描口、PC3~PC0 作为键盘读入口。

利用 CPU 控制 8255, 对 4×5 键盘进行扫描和键值读取, 将键值显示到 6 位数码管上。

实验电路:

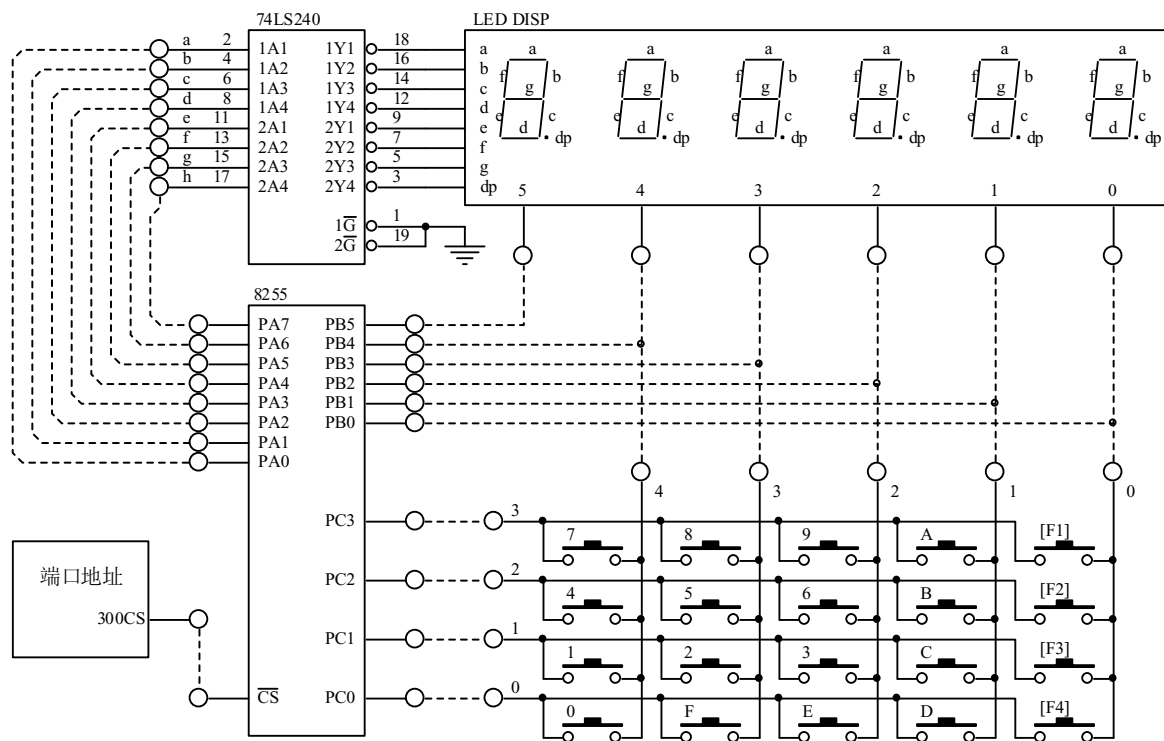


图 3-6-2 8255 键盘与显示设计实验电路图

实验步骤:

- 1) 拆除“键盘动态扫描电路”单元的行列接口（位于 4×5 键盘的右上方）和“七段显示器动态扫描电路”单元的段位接口（位于 6 位数码管的右下方）的 14 芯扁平电缆；
- 2) 按图 3-6-2 连接实验电路，参考程序：A86\8255DK.ASM；
- 3) 编写实验程序，经编译、链接无语法错误后装载到实验系统；
- 4) 全速运行程序，按实验系统键盘上的 0~F 数字键，数码管显示对应数字；按 F1~F4 功能键，清除数码管显示；
- 5) 实验完毕后，应使用暂停命令中止程序的运行；
- 6) 当拆除本实验所连接的导线后，应及时用 14 芯扁平电缆对步骤 1) 中提及的 2 组接口的重新连接，以方便其它实验使用。

实验七 8237 可编程 DMA 控制器实验

一、实验目的

了解串行通信的实现原理，掌握 8251 的工作方式和编程方法。

二、实验设备

PC 计算机一台，Dais-86PRO+或 Dais-PRO163C 实验系统一套。

三、实验内容及步骤

直接存储器访问（Direct Memory Access，简称 DMA），是指外部设备不经过 CPU 的干涉，直接实现对存储器的访问。DMA 传送方式可用来实现存储器到存储器、存储器到 I/O 接口、I/O 接口到存储器之间的高速数据传送。

实验电路：

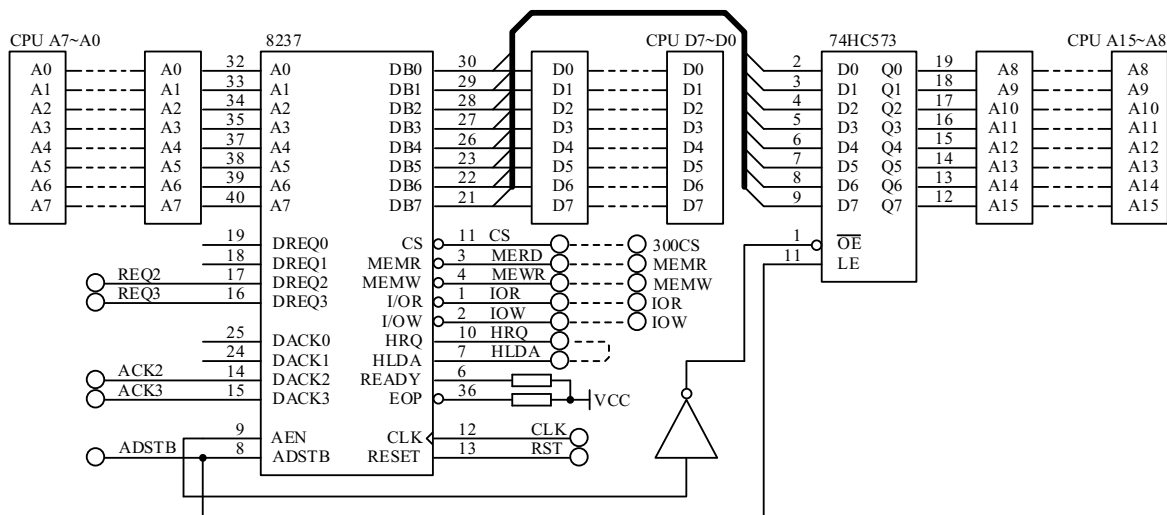


图 3-7 8237 可编程 DMA 控制器实验电路图

实验步骤：

- 1) 按图 3-7、图 3-1（存储器扩展）连接实验电路，参考程序：A86/8237.ASM；
- 2) 编写实验程序，经编译、链接无语法错误后装载到实验系统；
- 3) 打开内存窗口，并在内存窗口点鼠标右键选择“转到指定地址”，在对话框中输入“2000”再点击确定；
- 4) 运行程序，等待程序结束；
- 5) 检查内存窗口 2010h~201Fh 共 16 个内存单元数据，是否与 2000h~200Fh 单元一致；
- 6) 修改程序，变换 DMA 地址范围和数据长度，观察实验结果。

实验九 8253 定时 / 计数器应用实验

一、实验目的

学习 8253 芯片和微机接口的方法，掌握 8253 定时 / 计数器的工作方式和编程原理。

二、实验设备

PC 计算机一台，Dais-86PRO+或 Dais-PRO163C 实验系统一套。

三、实验内容及步骤

本实验置 8253 的 1 通道、2 通道工作在方式 3，通过级联的方法产生一个周期为 1s 的方波。

实验电路：

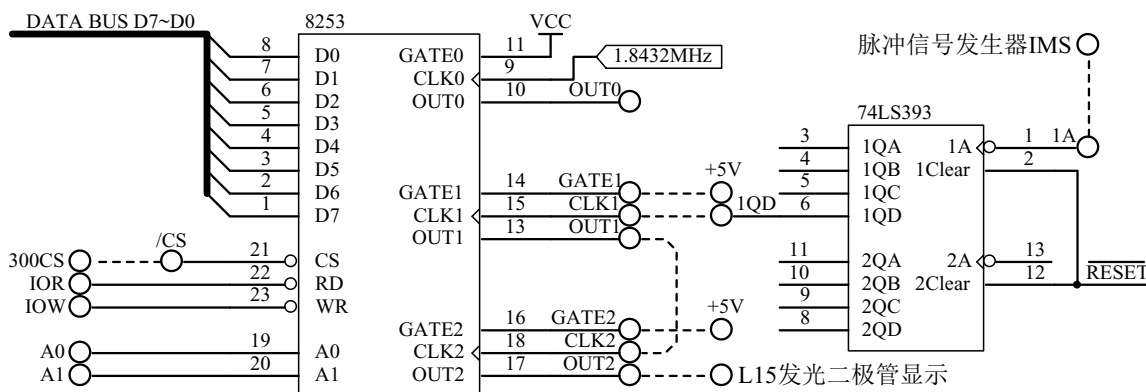


图 3-9 8253 定时 / 计数器实验电路图

实验步骤：

- 1) 按图 3-9 连接实验电路，参考程序：A86\8253.ASM；
- 2) 编写实验程序，经编译、链接无语法错误后装载到实验系统；
- 3) 全速运行程序，观察发光二极管 L15，应有周期为 1s 的点亮、熄灭；
- 4) 实验完毕后，应使用暂停命令中止程序的运行。

实验十 8259 中断控制器实验

一、实验目的

掌握 8259 中断控制器的工作原理，学习 8259 的应用编程方法。

二、实验设备

PC 计算机一台，Dais-86PRO+或 Dais-PRO163C 实验系统一套。

三、实验内容及步骤

在实验系统中设有 8259 中断控制器，将 IRQ0~IRQ7 作为中断源，以电平触发产生中断，并在数码管显示相应的中断号。

实验电路：

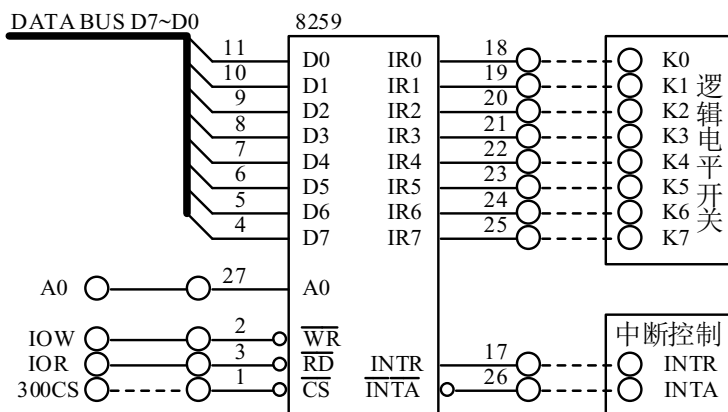


图 3-10 8259 中断控制器实验电路图

实验步骤：

- 1) 按图 3-10 连接实验电路，参考程序：A86\8259.ASM；
- 2) 编写实验程序，经编译、链接无语法错误后装载到实验系统；
- 3) 全速运行程序，当 K0~K7 任意一位开关置高电平（向下拨）时，进入相应的中断服务程序，显示相应中断号；当多位开关置高电平（向下拨）时，进入优先级高的中断服务程序；当 K0~K7 均为低电平（向上拨）时，退出中断服务程序，返回主程序；
- 4) 实验完毕后，应使用暂停命令中止程序的运行。

实验十一 LCD 128×64 液晶显示实验

一、实验目的

掌握图形液晶模块的控制方法，学习液晶驱动程序及高级接口函数的编写。

二、实验设备

PC 计算机一台，Dais-86PRO+或 Dais-PRO163C 实验系统一套。

三、实验内容及步骤

控制字符型液晶模块，在屏幕上显示字符串。

实验电路：

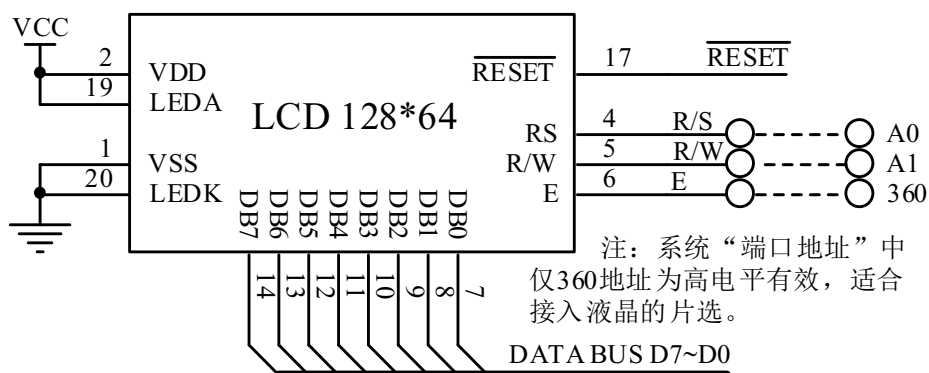


图 3-11 LCD 128×64 液晶显示实验电路图

实验步骤：

- 1) 按图 3-11 连接实验电路，参考程序：A86\LCD12864.ASM；
- 2) 编写实验程序，经编译、链接无语法错误后装载到实验系统；
- 3) 全速运行程序，观察液晶模块，应能显示中英文字符串；
- 4) 实验完毕后，应使用暂停命令中止程序的运行。

实验十二 LED 16×16 点阵显示实验

一、实验目的

学习如何利用扩展锁存器的方式控制点阵显示，掌握单片机与 LED 16×16 点阵块之间的接口电路设计及程序设计方法。

二、实验设备

PC 计算机一台，Dais-86PRO+或 Dais-PRO163C 实验系统一套。

三、实验内容及步骤

编写程序，在 16×16 点阵模块上显示汉字或图形。

实验电路：

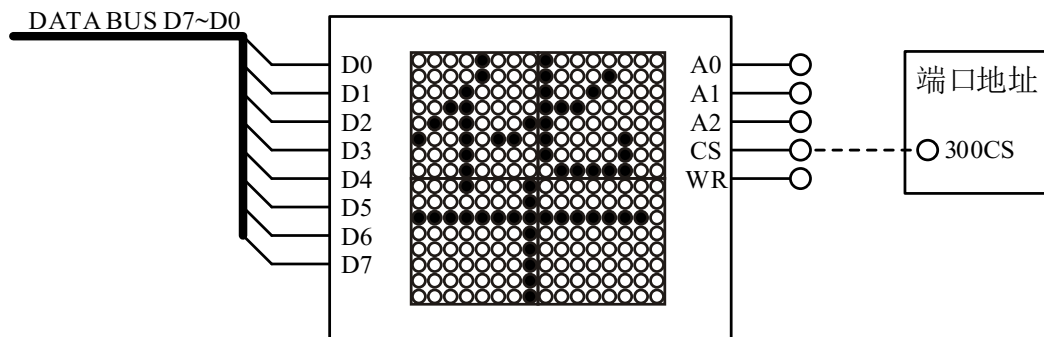


图 3-12 LED 16×16 点阵显示实验电路图

实验步骤：

- 1) 按图 3-12 连接实验电路，参考程序：A86\LED1616.ASM；
- 2) 编写实验程序，经编译、链接无语法错误后装载到实验系统；
- 3) 全速运行程序，点阵模块应能循环显示汉字或图形；
- 4) 实验完毕后，应使用暂停命令中止程序的运行。

实验十三 音频驱动实验

一、实验目的

学习用 8253 定时 / 计数器输出信号使蜂鸣器发声的方法。

二、实验设备

PC 计算机一台, Dais-86PRO+或 Dais-PRO163C 实验系统一套。

三、实验内容及步骤

编写程序, 控制 8253, 使其输出连接到蜂鸣器上能发出相应的乐曲。

实验电路:

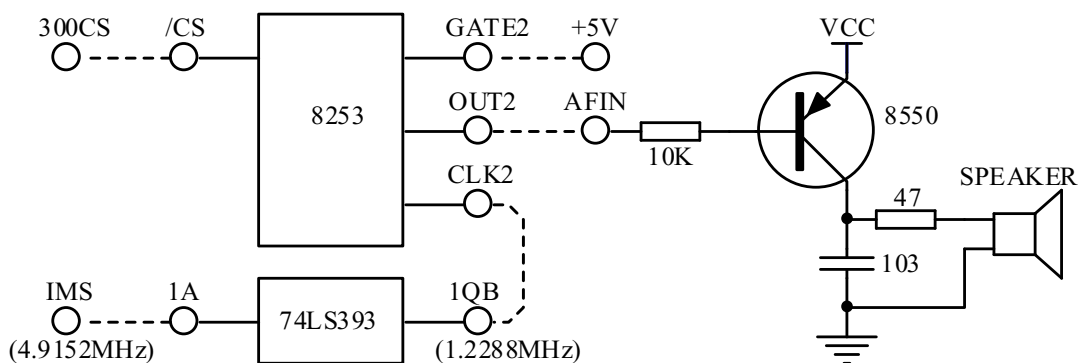


图 3-13 音频驱动实验电路图

实验步骤:

- 1) 按图 3-13 连接实验电路, 参考程序: A86\AUDIO.ASM;
- 2) 编写实验程序, 经编译、链接无语法错误后装载到实验系统;
- 3) 全速运行程序, 蜂鸣器开始演奏音乐;
- 4) 实验完毕后, 应使用暂停命令中止程序的运行。

实验十四 继电器控制

一、实验目的

掌握继电器控制的基本方法和编程。

二、实验设备

PC 计算机一台，Dais-86PRO+或 Dais-PRO163C 实验系统一套。

三、实验内容及步骤

在自动化控制设备中都存在一个电子与电气电路的互相连接问题，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对象（电机、电磁铁、电灯等），另一方面又要为电子提供良好的电隔离，以保护电子电路和人身的安全，使用继电器便达到这一目的。

利用 8255 PA0 输出高低电平来控制继电器的吸合与断开，以实现对外部装置的控制。

实验电路：

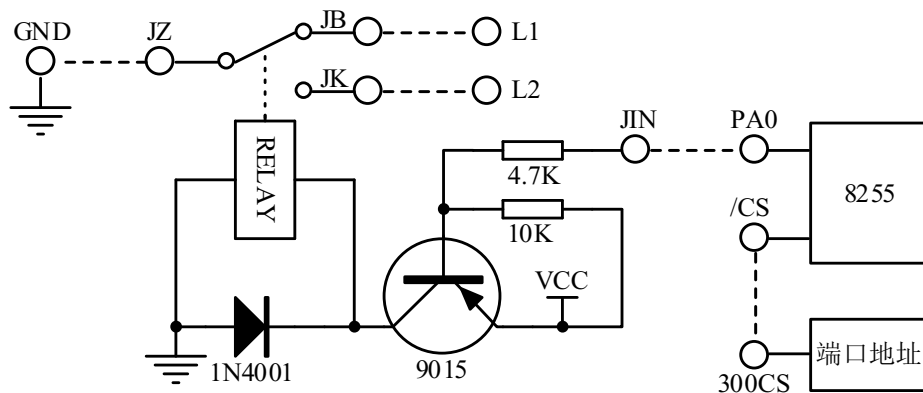


图 3-14 继电器控制实验电路图

实验步骤：

- 1) 按图 3-14 连接实验电路，参考程序：A86\RELAY.ASM；
- 2) 编写实验程序，经编译、链接无语法错误后装载到实验系统；
- 3) 全速运行程序，观察发光二极管，L1 为继电器常闭指示、L2 为继电器常开指示；
- 4) 实验完毕后，应使用暂停命令中止程序的运行。

实验十五 步进电机控制

一、实验目的

了解步进电机控制的基本原理，掌握步进电机转动编程方法。

二、实验设备

PC 计算机一台，Dais-86PRO+或 Dais-PRO163C 实验系统一套。

三、实验内容及步骤

步进电机驱动原理是通过对它每组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为控制，所以调节脉冲信号的频率便可改变步进电机的转速。

利用 8255 的 PA0~PA3 输出脉冲信号，驱动步进电机转动。

实验电路：

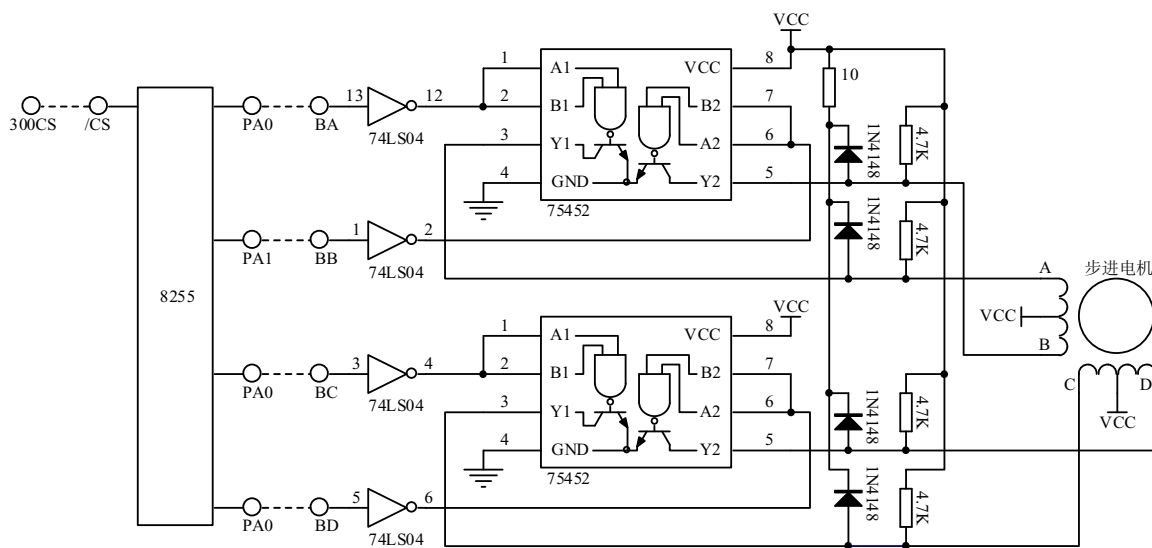


图 3-15 步进电机控制实验电路图

实验步骤：

- 1) 按图 3-15 连接实验电路，参考程序：A86\STMOTOR.ASM；
- 2) 编写实验程序，经编译、链接无语法错误后装载到实验系统；
- 3) 全速运行程序，观察步进电机转动情况；
- 4) 实验完毕后，应使用暂停命令中止程序的运行。

实验十六 直流电机控制

一、实验目的

学习直流电机的驱动原理，掌握使用 D/A 转换器对直流电机的控制方法。

二、实验设备

PC 计算机一台，Dais-86PRO+或 Dais-PRO163C 实验系统一套。

三、实验内容及步骤

改变 D/A 转换器的输出，经放大后控制直流电机的转速、转向。

实验电路：

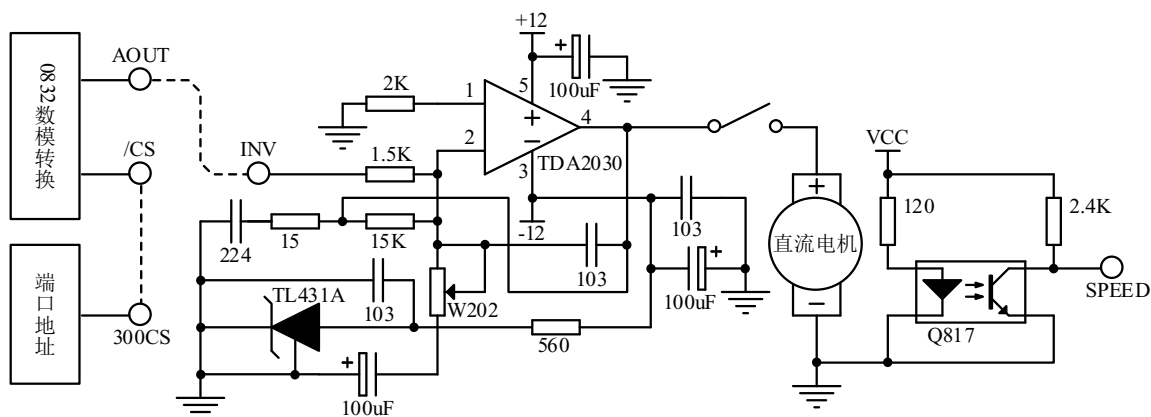


图 3-16 直流电机控制实验电路图

实验步骤：

- 1) 按图 3-16 连接实验电路，参考程序：A86\DCMOTOR.ASM；
- 2) 编写实验程序，经编译、链接无语法错误后装载到实验系统；
- 3) 全速运行程序，直流电机开始循环反转、停转、正转、停转；
- 4) 实验完毕后，应使用暂停命令中止程序的运行。

实验十七 数字温度传感器实验

一、实验目的

学习 DS18B20 数字温度传感器的编程方法。

二、实验设备

PC 计算机一台，Dais-86PRO+或 Dais-PRO163C 实验系统一套。

三、实验内容及步骤

使用 8255 的 PC0 连接 DS18B20 的 DQ 管脚，完成对 DS18B20 的初始化及温度的读取。

实验电路：

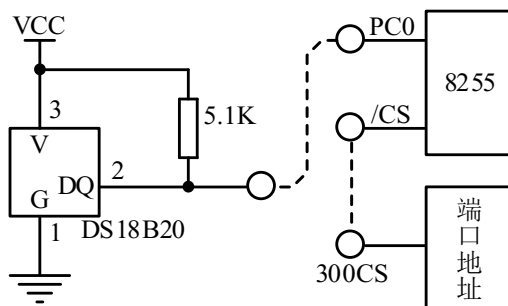


图 3-17 温度测量实验电路图

实验步骤：

- 1) 按图 3-17 连接实验电路，参考程序：A86\DS18B20.ASM；
- 2) 编写实验程序，经编译、链接无语法错误后装载到实验系统；
- 3) 全速运行程序，显示当前环境温度值；
- 4) 实验完毕后，应使用暂停命令中止程序的运行。

实验十八 V/F 转换实验

一、实验目的

1. 了解 LM331 器件的工作原理及外部电路的连接；
2. 掌握利用 LM331 器件实现 V/F 转换的基本方法；
3. 熟悉简易低频频率计的设计方法。

二、实验设备

PC 计算机一台，Dais-86PRO+或 Dais-PRO163C 实验系统一套。

三、实验内容及步骤

利用 LM331 器件实现 V/F 转换，将 0~5V 的模拟电压转换成与模拟量电压变化成线性关系的频率值，用 8253 设计一个频率计程序，并把所测频率通过数码管显示出来。

实验电路：

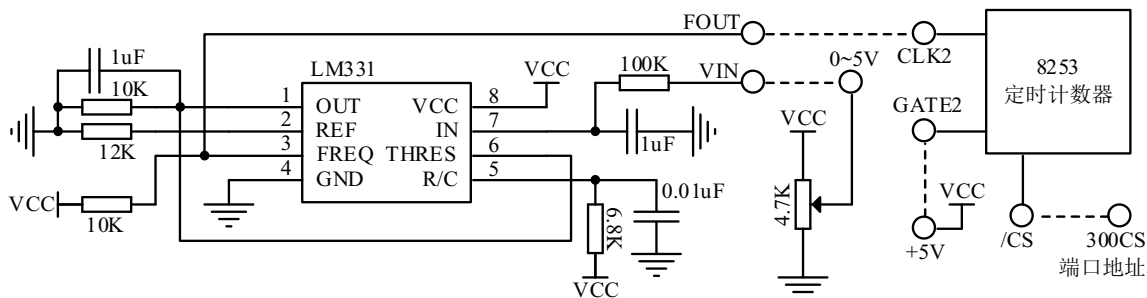


图 3-18 V/F 转换实验电路图

实验步骤：

- 1) 按图 3-18 连接实验电路，参考程序：A86\VF.ASM；
- 2) 编写实验程序，经编译、链接无语法错误后装载到实验系统；
- 3) 全速运行程序，调节 0~5V 模拟电压输入，数码管显示输出的频率；
- 4) 实验完毕后，应使用暂停命令中止程序的运行。

实验十九 PWM 输出实验

一、实验目的

了解脉宽调制（PWM）的原理，学习使用 PWM 输出模拟量。

二、实验设备

PC 计算机一台，Dais-86PRO+或 Dais-PRO163C 实验系统一套。

三、实验内容及步骤

固定周期内，改变脉宽（即修改其占空比），再经积分电路形成直流电压，从而实现对电机等设备的速度控制。

用 74LS273 锁存输出端口 O0 输出不同占空比的脉冲，通过 PWM 转换成电压输出。

实验电路：

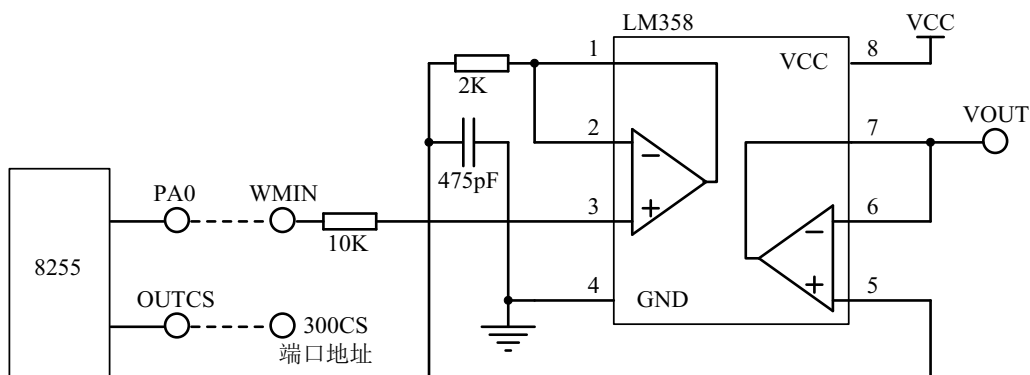


图 3-19 PWM 输出实验电路图

实验步骤：

- 1) 按图 3-19 连接实验电路，参考程序：A86\PWM.ASM；
- 2) 编写实验程序，经编译、链接无语法错误后装载到实验系统；
- 3) 全速运行程序，程序中占空比为 50%，因此 PWM 输出模拟量 VOUT 约为 2.5V；
- 4) 实验完毕后，应使用暂停命令中止程序的运行。