



# 微机原理与接口技术 实验指导

长安大学

自动化与交通控制工程实验教学中心

# 实验要求

## 一、预习实验内容

- 明确本次实验目的及任务，了解和掌握实验所需的理论知识及相关接口芯片的工作原理。
- 通过阅读示例程序，掌握编程方法及相关技巧。
- 对每次实验，选择其中一个项目自己设计编写程序。

## 二、实验过程

- 带上理论课教材、实验指导书及准备的实验程序。
- 若为接口电路，请关闭电源搭接线路，检查无误后，再开电源。
- 调试程序，记录结果。
- 实验结束后，请关闭电源，将各实验器材归位，清洁自己的桌面。

# 实验要求

## 三、编写实验报告

- ❧ 实验题目。
- ❧ 分析设计思想，绘制实验原理图、流程图。
- ❧ 程序清单及相关注释。
- ❧ 分析实验结果。
- ❧ 说明在实验过程中遇到了哪些问题，是如何处理的。
- ❧ 收获体会，不足之处和今后应注意的问题等。

## 四、注意事项

- ❧ 实验以前，应确保**PCI**总线扩展卡与实验箱间扁平电缆连接的正确性。
- ❧ 实验前后应仔细检查实验箱，防止导线、元件等物品落入装置内，导致线路短路、元件损坏。
- ❧ 爱护实验设施，插接、拔取排线时，手握白色插头，**不得**从线中间拉取。
- ❧ 实验箱电源关闭后，不能立即重新开启。关闭与重新开启之间至少应有**30秒**间隔

# 目录

## 第一部分 汇编程序设计

实验一、系统认识实验

实验二、**I/O**程序设计

实验三、分支程序设计

实验四、循环程序设计

实验五、代码转换程序设计

实验六、运算类程序设计

实验七、字符串处理

实验八、子程序及软中断程序设计

## 第二部分 接口部分实验

实验九、中断特性及**8259**中断应用

实验十 **PCI**总线中断应用

实验十一、**8254**定时/计数器应用设计

实验十二、**8255**并行接口应用设计

# 实验一 认识Tddebug 集成操作软件

## 一、实验目的

- 熟悉汇编程序的开发过程。
- 认识Tddebug集成操作软件。
- 掌握在Tddebug集成环境中编辑、编译、连接汇编语言程序方法。
- 掌握INT 21H软件中断来调用DOS内部子程序的方法

## 二、实验设备

- PC微机一台

# 实验一 认识Tdddebug

## 集成操作软件

### 三、汇编语言程序的开发过程

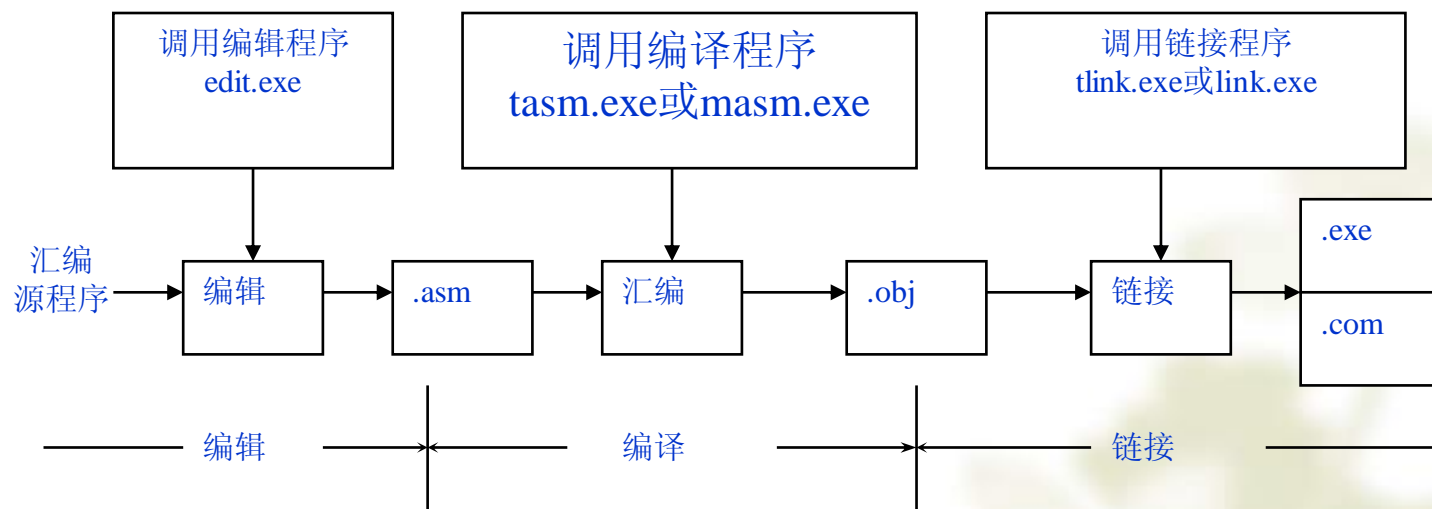


图1.1 汇编语言程序开发过程

# 实验一 认识Tddebug

## 集成操作软件

### 四、Tddebug集成操作软件使用说明

- ① 该软件是集编辑（**Edit.exe**）、编译（**Tasm.exe**）、连接（**Link.exe**）和调试（**TD.exe**）等多个功能于一体，可在**Windows、DOS**环境下运行，为用户提供了一个学习微机原理的实验平台。
- ② 启动Tddebug **D: \wjyl> Tddebug**
- ③ 若启动成功，进入主界面。

# 实验一 认识Tddebug 集成操作软件

## ④ Tddebug的菜单结构

主菜单

子菜单

菜单说明

<u>E</u> dit			编辑源文件 (.asm)
<u>C</u> ompile	——	Compile	编译源文件 (.asm)
	——	Link	连接目标文件 (.obj)
	——	Build All	编译和连接
<u>P</u> mrun			进入保护模式调试状态
<u>R</u> mrun	——	Run	运行实模式程序 (.exe)
	——	Debug	进入实模式调试状态 (.exe)
<u>H</u> elp			版本信息
<u>Q</u> uit			退出Tddebug



# 实验一 认识Tdddebug 集成操作软件

## 五、实验内容及说明

### (1) 实验内容：数据传送实验

- 编程将数据段中的一个字符串传送到附加段中，并输出附加段中的目标字符串到屏幕上。参见示例程序。
- 修改此程序，采用字符串传送指令完成。

### (2) 实验中使用DOS功能调用（INT 21H），参见教材《32位微型计算机原理与接口技术》89页

#### ❖ 显示字符串

- 入口：AH=09H
- 调用参数：DS:DX=串首地址，‘\$’为结束字符。

#### ❖ 返回DOS系统

- 入口：AH=4CH
- 调用参数：AL=返回码

# 实验一 认识Tddebug 集成操作软件

- (3) 有关字符串的操作指令，参见教材《32位微型计算机原理与接口技术》62页
- (4) 实验目的：通过对该程序进行调试，查看程序段、数据段、附加段装入内存后的分配情况。单步执行数据传送指令后，观察各个寄存器及数据区的内容。

## 六、实验步骤

- (1) 运行Tddebug软件，选择Edit菜单编写实验程序
- (2) 使用Compile菜单中的Compile和Link对实验程序进行汇编、连接，生成执行文件。
- (3) 使用Rmrun菜单中的Run运行程序，观察运行结果。

# 实验一 认识Tdddebug

## 集成操作软件

(4) 使用Rmrun菜单中的Debug调试程序，查看程序段、数据段、附加段装入内存后的分配情况。单步执行数据传送指令后，观察各寄存器及数据区的内容。过程如下：

- a. 按F7单步执行，在代码区中有一个三角，表示正在执行的指令。每一条指令的执行一定会使目标寄存器和状态寄存器发生变化，从相关窗口看结果。
- b. 检查内存数据区的内容，关键是找出用户程序的数据段和附加段：
  - ❖ 方法1：在CPU窗口按Tab键使内存数据显示区成为活动区，按Ctrl+G键，输入：“DS或ES寄存器的值：偏移地址”，即可显示用户指定的数据区
  - ❖ 方法2：选择菜单View| Dump，弹出内存数据显示窗口。
- c. 查看执行结果：按Alt+F5，切换到用户窗口。
- d. 更改数据区中的数据，考察、调试程序的正确性。

# 实验二、I/O程序设计

## 一、实验目的

- 理解软中断的调用方法及中断过程。
- 掌握INT 21H软件中断来调用DOS系统I/O子程序的方法。
- 掌握将存储在内存单元中的数据在屏幕上显示的方法。
- 掌握从键盘输入信息到内存的方法。

## 二、实验设备

- PC微机一台

# 实验二、I/O程序设计

## 三、实验预习要求

- ❧ 1) . 复习DOS功能调用 (INT 21H) 中用于字符输入 (功能号为01H)、字符输出 (功能号为02H)、字符串输入 (功能号为0AH) 以及字符串输出 (功能号为09H) 的调用方法。参见教材《32位微型计算机原理与接口技术》89页。
- ❧ 2) . 阅读示例程序, 掌握输入/输出方法。
- ❧ 3) . 从实验内容中任选一道题目, 仔细阅读相关的实验要求及说明, 编写程序, 以便上机调试。

# 实验二、I/O程序设计

## 四、实验内容

- 1) 【示例】：将指定数据区中一个字节的数数据以十六进制数形式显示在屏幕上。
  - ❖ 设计思路：首先将该数以十六进制数形式分解，然后将每一位分别转换为ASCII码送屏幕显示。其中0H~9H之间的数加30H，即可得0H~9H的ASCII码，而AH~FH的ASCII码，则需再加7H，参见表2-1。参考程序流程如图2-1所示。示例程序参见参考程序清单。
- 2) 编程由键盘输入任意一个字符，将该字符的ASCII码值显示在屏幕上。
- 3) 从键盘输入两个1位十进制数，计算二者之和，并将结果显示在屏幕上。
- 4) 编程由键盘输入任意一位数，将该数以二进制数形式显示在屏幕上，例：5=00000101B

表 数码转换对应关系

十六进制数	BCD 码	二进制机器码	ASCII 码	七段码	
				共阳	共阴
0	0000	0000	30H	40H	3FH
1	0001	0001	31H	79H	06H
2	0010	0010	32H	24H	5BH
3	0011	0011	33H	30H	4FH
4	0100	0100	34H	19H	66H
5	0101	0101	35H	12H	6DH
6	0110	0110	36H	02H	7DH
7	0111	0111	37H	78H	07H
8	1000	1000	38H	00H	7FH
9	1001	1001	39H	18H	67H
A		1010	41H	08H	77H
B		1011	42H	03H	7CH
C		1100	43H	46H	39H
D		1101	44H	21H	5EH
E		1110	45H	06H	79H
F		1111	46H	0EH	71H

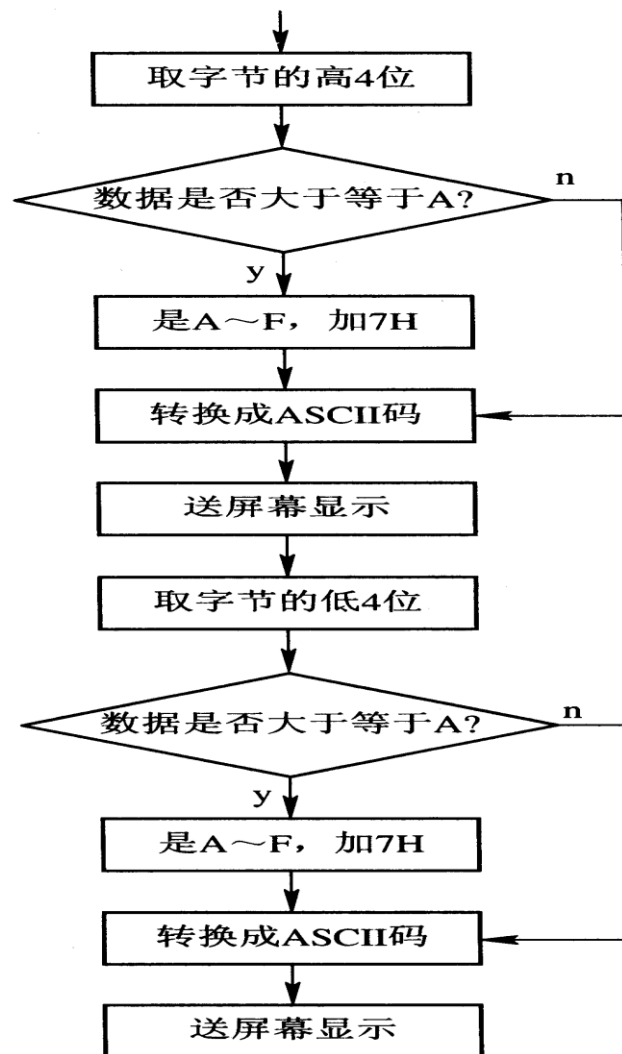


图2-1 一个字节数以十六进制数形式显示处理流程

# 实验二、I/O程序设计

## 五、实验说明

- ❖ 存储在内存单元中的数据要显示在屏幕上，必须首先将一个数据区中的数分解为各位不同进制的数（参见表2-1），然后转换为字符，通过DOS功能调用（INT 21H）完成显示。例：

### ☞ 显示单个字符

- ❖ MOV AH, 02H
- ❖ MOV DL, 待显示字符的ASCII码
- ❖ INT 21H

### ☞ 键盘输入并回显

- ❖ MOV AH, 01H
- ❖ INT 21H
- ❖ 得到按键的ASCII码，存放于AL中



# 实验二、I/O程序设计

## 六、实验步骤

- ❧ 画出实验流程图。
- ❧ 运行Tddebug软件，选择Edit菜单编写实验程序
- ❧ 使用Compile菜单中的Compile和Link对实验程序进行汇编、连接，生成执行文件。
- ❧ 使用Rmrun菜单中的Run运行程序，观察运行结果。
- ❧ 使用Rmrun菜单中的Debug调试程序。单步执行指令，观察各寄存器及数据区的内容。
- ❧ 更改数据区中的数据，考察程序的正确性。

# 实验三、分支程序设计

## 一、实验目的

- ☞ 掌握比较指令、条件转移指令的用法
- ☞ 掌握分支结构程序的组成
- ☞ 掌握分支程序的设计、调试方法

## 二、实验设备

- ☞ PC微机一台

## 三、实验预习要求

- 1) 复习比较指令、条件转移指令的用法。
- 2) 阅读示例程序，掌握分支编程方法及相关技巧。
- 3) 从实验内容中任选一道题目，仔细阅读相关的实验要求及说明，编写程序，以便上机调试。

# 实验三、分支程序设计

## 四、实验内容

- 1) 【示例】判断X的大小，并根据判断结果以不同方式分别显示。
- 2) 键入一个数，判断数是否在 $5 < X < 24$ 范围内，是输出标志0，不是输出-1。
- 3) 编写程序，在屏幕上显示一组菜单，根据键盘输入的菜单编号，提示你选择的菜单名。
- 4) 从键盘输入一个字符，判断该字符是小写字母、大写字母、数字或其他字符，并给出相应的提示。

# 实验三、分支程序设计

## 五、实验要求及说明

1) 判断X的大小，并根据判断结果以不同方式分别显示

☞ 设NUMBER单元的数X以及数值N1,N2均为单字节无符号数，请判断X的大小，并根据判断结果分别显示： $N1 \leq X \leq N2$ ，或 $X < N1$ ，或 $X > N2$ 。参见程序清单

2) 键入一个数，判断数是否在 $5 < X < 24$ 范围内，是输出标志0，不是输出-1

3) 编写程序，在屏幕上显示一组菜单，根据键盘输入的菜单编号，提示你选择的菜单名。

☞ 假设在屏幕上显示的菜单如下：

☞ 1.....Open

☞ 2.....Save

☞ 3.....Copy

☞ 4.....Exit

☞ (Please choose 1, 2, 3, OR 5)——

☞ 若输入1，则显示“I choose Open”后程序结束；若输入2，则显示“I choose Save”后程序结束；若输入3，则显示“I choose Copy”后程序结束；若输入4，则显示“I choose Exit”后程序结束；若输入其它字符，则显示“You press an error key”后程序结束；

☞ 注意可使用宏指令简化程序。

4) 从键盘输入一个字符，判断该字符是小写字母、大写字母、数字或其他字符，并给出相应的提示。

☞ 数字0~9的ASCII码为30H~39H；大写字母的ASCII码为41H~5AH；小写字母的ASCII码为61H~7AH；

# 实验三、分支程序设计

## 六、实验步骤

- ❖ 画出实验流程图。
- ❖ 运行Tddebug软件，选择Edit菜单编写实验程序
- ❖ 使用Compile菜单中的Compile和Link对实验程序进行汇编、连接，生成执行文件。
- ❖ 使用Rmrun菜单中的Run运行程序，观察运行结果。
- ❖ 使用Rmrun菜单中的Debug调试程序。观察构成分支条件的各标志的变化情况及相关寄存器内容。
- ❖ 更改数据区中的数据，考察程序的正确性

# 实验四、循环程序设计

## 一、实验目的

- 1) 掌握比较指令、转移指令和循环指令的使用方法
- 2) 掌握循环结构程序的组成
- 3) 掌握循环程序的设计、调试方法

## 二、实验设备

PC微机一台

## 三、实验预习要求

- 1) 复习比较指令、条件转移指令和循环指令。
- 2) 阅读示例程序，掌握循环程序的结构、循环控制方法及相关技巧。
- 3) 从实验内容中任选一道题目，仔细阅读相关的实验要求及说明，编写程序，以便上机调试。

# 实验四、循环程序设计

## 四、实验内容

- 1) 【示例】求某数据区内负数的个数
- 2) 试编程统计数据区中正数、零和负数的个数。
- 3) 编程求无符号数字字节序列中最大值和最小值。
- 4) 从键盘输入一字符串，搜索该字符串中是否有字符“A”，若有请输出其在串中的位置。

# 实验四、循环程序设计

## 五、实验说明

### 1) 求某数据区内负数的个数

为统计数据区内负数的个数，需逐个判别区内的每一个数据，然后将所有数据中凡是符号位为1的数据个数累加起来，即得到区内包含负数的个数。参见程序清单。

### 2) 试编程统计数据区中正数、零和负数的个数。

### 3) 编程求无符号数字字节序列中最大值和最小值

实验中可使用BH和BL作为暂存现行最大值和最小值，且在初始时将BH和BL初始化为首字节的内容，循环依次比较每个字节的内容，求得最大值（存于BH中）和最小值（存于BL中），送屏幕显示。

### 4) 从键盘输入一字符串，搜索该字符串中是否有字符“A”，若有请输出其在串中的位置。



# 实验四、循环程序设计

## 六、实验步骤

- ❧ 画出实验流程图。
- ❧ 运行Tdebug软件，选择Edit菜单编写实验程序
- ❧ 使用Compile菜单中的Compile和Link对实验程序进行汇编、连接，生成执行文件。
- ❧ 使用Rmrun菜单中的Run运行程序，观察运行结果。
- ❧ 使用Rmrun菜单中的Debug调试程序。观察构成分支、循环条件的各标志的变化情况及相关寄存器内容。
- ❧ 更改数据区中的数据，考察程序的正确性。

# 实验五、运算类程序设计

## 一、实验目的

- 1) 掌握运算类指令对各状态标志位的影响及其测试方法
- 2) 掌握运用运算类指令编程及调试的方法
- 3) 掌握子程序设计及调用方法

## 二、实验设备

PC机一台

## 三、实验预习要求

- 1) 复习运算类指令的用法。
- 2) 复习子程序设计及调用方法。
- 3) 阅读示例程序，掌握子程序的编程方法及相关技巧。
- 4) 从实验内容中任选一道题目，仔细阅读相关的实验要求及说明，用子程序处理方式编写程序，以便上机调试。

# 实验五、运算类程序设计

## 四、实验内容

- 1) 示例：二进制双精度加法运算
- 2) 十进制数的BCD码加减法运算
- 3) 乘法运算
- 4) 除法运算
- 5) 编程用减奇数法开平方

# 实验五、运算类程序设计

## 五、实验要求及说明

80x86指令系统提供了实现加、减、乘、除运算的基本指令，可对表4-1所示的数据类型进行算术运算。

表4-1 数据类型算术运算表

数制	二进制		BCD码	
	带符号	无符号	组合	非组合
运算符	+、-、×、÷		+、-	+、-、×、÷
操作符	字节、字、多精度		表字节（二位数字）	字节（一位数字）

# 实验五、运算类程序设计

## 1) 二进制双精度加法运算

- ❖ 计算 $X+Y=Z$ ，并将结果输出到屏幕上。其中 $X=001565A0H$ ， $Y=0021B79EH$ ，运算的结果 $Z=00371D3EH$ 。
- ❖ (1) 本实验是双精度（2个16位，即32位）运算，利用累加器AX，先求低16位和，并存储于低地址存储单元，后求高16位和，再存入高地址存储单元。由于低位和可能向高位有进位，此时 $CF=1$ ，因而高位字相加语句需用ADC指令，即同时加上CF中的1。参见参考程序清单。
- ❖ (2) 画出X、Y、Z三个数在内存中的分布示意图。
- ❖ (3) 修改示例程序，直接使用32位寄存器和32位加法指令完成本实验功能。

## 2) 十进制数的BCD码加减法运算

- ❖ 计算 $X-Y=Z$ ，其中，X、Y、Z为非组合的BCD码， $X=0400H$ ， $Y=0102H$ ，运算结果 $Z=0208H$ 。
- ❖ (1) 编写两个数值长度不等的BCD码相加程序。
- ❖ 思路：对应位用ADC带进位加法求和（考虑CF标志），再进行DAA调整（只对AL处理）。

# 实验五、运算类程序设计

## 3) 乘法运算

- ❖ 本实验要求实现十进制数乘法，被乘数和乘数均以BCD码形式存放在内存中，被乘数为02345，乘数为3，运算结果7035，显示在屏幕上为：00 00 07 00 03 05。
- ❖ (1) 调试程序观察加、减、乘、除及相关调整指令，对标志位的影响。

## 4) 除法运算

- ❖ 本实验要求在内存中存储5个无符号数，将其用101H除，在进行BCD码转换后，若奇偶位为偶（PF=1），则显示“P”；为负（SF=1），则显示“S”；为零（ZF=1），则显示“Z”；若PF、SF、ZF标志皆为0，则显示“；”。
- ❖ 除法指令DIV、IDIV执行后，不影响任何标志位，而BCD码调整指令AAD会影响PF、SF、ZF标志。通过本实验考察这些指令对标志的影响情况。

# 实验五、运算类程序设计

## 5) 编程用减奇数法开平方

- ❖ 80x86指令系统中没有开平方指令，因此，开平方运算是通过程序来实现的。用减奇数法可求得近似平方根，获得平方根的整数部分。因为， $N$ 个自然数中奇数之和等于 $N^2$ ，展开：
- ❖  $1+3+5=9=3^2$ ； $1+3+5+7+9+11+13+15=64=8^2$ ；所以，若要对 $S$ 做开平方的运算，就可以从 $S$ 中逐次减去自然数中的奇数1, 3, 5, 7, ……，一直到被减数为0或不够减下一个自然数的奇数为止，然后统计减去自然数的奇数的个数，它就是 $S$ 的近似平方根。
- ❖ (1) 本实验要求编程实现计算0040H的开平方值，并将运算结果显示在屏幕上。
- ❖ (2) 试编程，由键盘输入一个十六进制数，将其转换为十进制数，并进行开平方运算。

# 实验五、运算类程序设计

## 六、实验步骤

- ④ 画出实验流程图。
- ④ 画出内存数据分布示意图
- ④ 运行Tdebug软件，选择Edit菜单编写实验程序
- ④ 使用Compile菜单中的Compile和Link对实验程序进行汇编、连接，生成执行文件。
- ④ 使用Rmrun菜单中的Run运行程序，观察运行结果。
- ④ 使用Rmrun菜单中的Debug调试程序。单步执行加、减、乘、除及相关调整指令时，观察各标志的变化情况及寄存器、数据区的内容。
- ④ 更改数据区中的数据，考察程序的正确性。



# 实验六、代码转换程序设计

## 一、实验目的

- 1) 了解微机中所使用的各类数制及编码方法。
- 2) 掌握不同进制数及编码相互转换的程序设计方法，加深对数码转换的理解。
- 3) 掌握将存储在内存单元中的数据以十六进制数形式、十进制数形式和二进制数形式显示在屏幕上的方法。

## 二、实验设备

PC微机一台

# 实验六、代码转换程序设计

## 三、实验预习要求

- 1) 复习运算类指令的用法。
- 2) 复习子程序设计及调用方法。
- 3) 阅读示例程序，掌握子程序的编程方法及相关技巧。
- 4) 从实验内容中任选一道题目，仔细阅读相关的实验要求及说明，用子程序处理方式编写程序，以便上机调试。

## 四、实验内容

- 1) 【示例】将5位十进制数的数字串转换为一个字的二进制数
- 2) 编写程序将内存中一个字的数转换为十进制数的数字串显示。
- 3) 编程将十进制的数字串转换为BCD码
- 4) 编程将BCD码转换为二进制数
- 5) 编程将二进制数转换为十进制数显示

# 实验六、代码转换程序设计

## 五、实验原理及说明

- 计算机输入设备输入的信息一般是由ASCII码或BCD码表示的数据或字符，CPU一般均用二进制数进行计算或用其他信息进行处理，处理的结果又必须依照外设的要求变为ASCII码、或BCD码或七段显示码等。因此，在应用软件中，各类数制的转换和代码的转换是必不可少的。计算机与外设间的数码转换关系如图6-1所示，数码对应关系如表6-1所示。

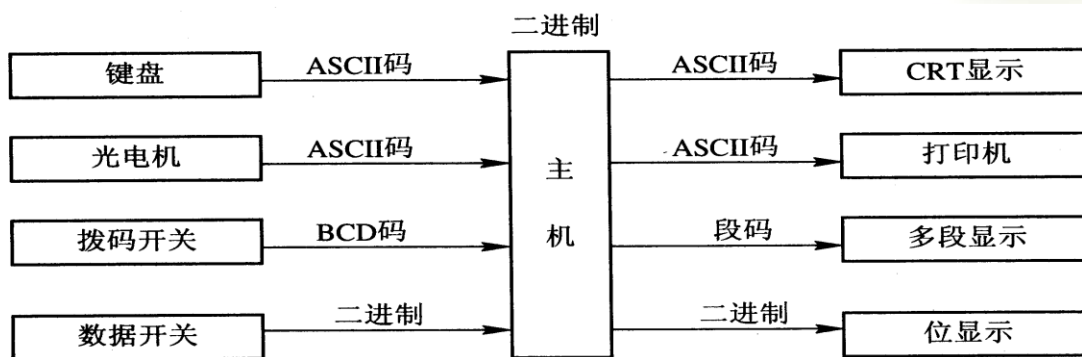


图 计算机与外设间的数码转换关系

# 实验六、代码转换程序设计

表 6-1 数码转换对应关系

十六进制数	BCD 码	二进制机器码	ASC II 码	七段码	
				共阳	共阴
0	0000	0000	30H	40H	3FH
1	0001	0001	31H	79H	06H
2	0010	0010	32H	24H	5BH
3	0011	0011	33H	30H	4FH
4	0100	0100	34H	19H	66H
5	0101	0101	35H	12H	6DH
6	0110	0110	36H	02H	7DH
7	0111	0111	37H	78H	07H
8	1000	1000	38H	00H	7FH
9	1001	1001	39H	18H	67H
A		1010	41H	08H	77H
B		1011	42H	03H	7CH
C		1100	43H	46H	39H
D		1101	44H	21H	5EH
E		1110	45H	06H	79H
F		1111	46H	0EH	71H

# 实验六、代码转换程序设计

## 1、将十进制数的数字串转换为二进制数

- ❖ 十进制数可以表示为：

$$D_n \times 10^n + D_{n-1} \times 10^{n-1} + \dots + D_0 \times 10^0 = D_i \times 10^i$$

其中代表十进制数1、2、3、...、9、0。

- ❖ 上式可以转换为：

$$\sum D_i \times 10^i = (((D_n \times 10 + D_{n-1}) \times 10 + D_{n-2}) \times 10 + \dots + D_1) \times 10 + D_0$$

由上式可归纳出十进制数转换为二进制数的方法：从十进制数的最高位开始做乘10加次位的操作，依次类推，则可求出二进制数结果。

- ❖ 本实验要求将缓冲区中的一个5位十进制数00012的ASCII码转换成二进制数，并将转换结果以十六进制数形式000C按位显示在屏幕上。转换过程的参考程序清单见后参考程序清单1，参考流程如图6-2所示。
- ❖ 修改该示例程序，从键盘任意输入5个数，实现转换并显示。

# 实验六、代码转换程序设计

## 5、实验原理及说明

### 2、将内存中一个字的数转换为十进制数的数字串。

- ❖ 十六位二进制数（一个字）的值域为0~65535，最大可转换为5位十进制数。五位十进制数可表示为：

$$N_D = D_4 \times 10^4 + D_3 \times 10^3 + D_2 \times 10^2 + D_1 \times 10^1 + D_0$$

- ❖ 因此，将十六位二进制数转换为5位ASCII码表示的十进制数数字串，即采用除10取余方法，分别求，并将它们转换为ASCII码。以字符串方式显示在屏幕上。转换部分的参考流程参见图6-4。
- ❖ 假设缓冲区中存放的数是000CH转换成十进制数的数字串后，在屏幕显示“00012”。

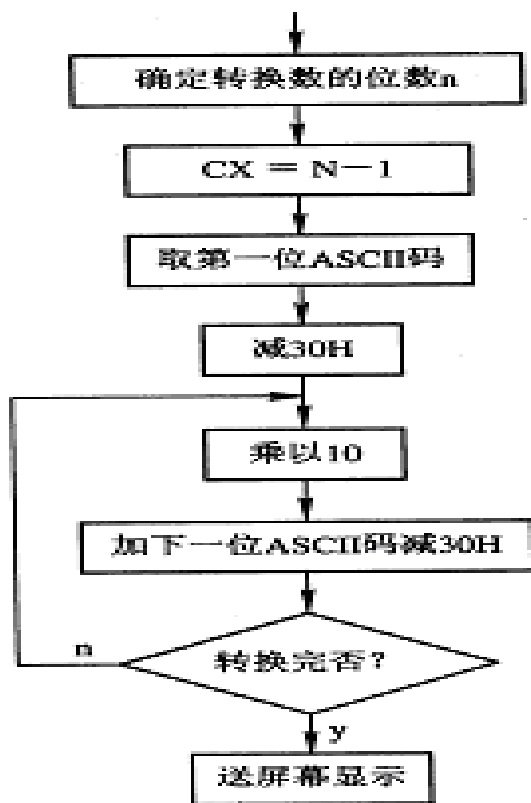


图6-2 将5位十进制数的数字串转换为二进制数参考流程

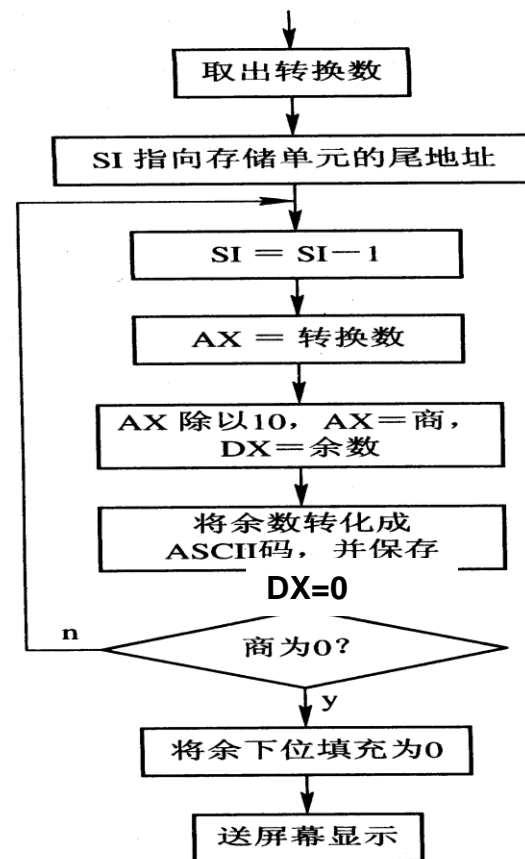


图6-4 将内存中一个字的数转换为十进制数的数字串参考流程

# 实验六、代码转换程序设计

## 六、实验步骤

- ❧ 画出实验流程图。
- ❧ 运行Tddebug软件，选择Edit菜单编写实验程序
- ❧ 使用Compile菜单中的Compile和Link对实验程序进行汇编、连接，生成执行文件。
- ❧ 使用Rmrun菜单中的Run运行程序，观察运行结果。
- ❧ 使用Rmrun菜单中的Debug调试程序。单步执行指令，观察各寄存器及数据区的内容。
- ❧ 更改数据区中的数据，考察程序的正确性。



# 实验七、字符串处理

## 一、实验目的

- 1) 掌握字符串操作指令。
- 2) 掌握实现字符串的搜索、排序等操作。

## 二、实验设备

PC微机一台

## 三、实验预习要求

- 1) 复习字符串操作指令。
- 2) 阅读示例程序，掌握编程方法及相关技巧。
- 3) 从实验内容中任选一道题目，仔细阅读相关的实验要求及说明，编写程序，以便上机调试。

## 四、实验内容

- 1) 【示例】 为标准ASCII设置校验码。
- 2) 从源串字符中搜索一个字串。
- 3) 从键盘输入一系列字符串，对他们按从小到大的顺序排序输出在屏幕上。
- 4) 输入一条英语句子，将其规范化：即将每个单词的第一个字符变为大写，其余为小写，并将规范化后的句子显示在屏幕上，同时统计出该句中单词个数，显示： **Number=?** 。

# 实验七、字符串处理

## 五、实验要求及说明

### 1) 【示例】为标准ASCII设置校验码

- ❖ 标准ASCII码的最高位（D7位）为0。微机系统与I/O设备之间在字符传送过程中，为了使接收方能够判断出接收自负的正确性，最简易的方法是定义标准ASCII码的最高位为奇偶校验位。包括校验位在内一个字节中“1”的个数为奇数个，则称为奇校验ASCII码，反之，一个字节中“1”的个数为偶数个则称为偶校验ASCII码。校验位的值由发送方设置，接收方负责校验，如果双方约定是偶校验传送，而接收方收到的一个字符编码中，有奇数个“1”，那么这个字符肯定是错误的。
- ❖ 本示例程序要求把STRING单元中的一串标准ASCII码转换成奇校验ASCII码，存入BUF单元开始的缓冲区。
- ❖ 技巧：通过AND AL, AL指令对P标志设置，判断P标志，决定是否设置校验位D7，由指令OR AL, 80H完成。参见程序清单。

# 实验七、字符串处理

## 2) 从源串字符中搜索一个子串。

- ❖ 假设从STRING单元开始有一串自负，程序执行开始采用人机对话方式，从键盘输入一个任意长度的子串，清查找源串中是否蕴含着键入的子串，并给出结果显示。
- ❖ 思路：键盘输入子串：用INT 21H的0AH功能，将子串存入内存。
- ❖ 搜索次数=源串长-子串长+1
- ❖ 字符串的比较：REPE CMPSB，再判断Z标确定是否找到

# 实验八、子程序及软中断 程序设计

## 一、实验目的

- ❖ 掌握子程序的定义和调用方法
- ❖ 掌握系统功能调用程序（软中断程序）的使用和编写方法
- ❖ 了解子程序与软中断之间的差异。
- ❖ 认识和理解中断特性。

## 二、实验设备

- ❖ PC微机一台

# 实验八、子程序及软中断 程序设计

## 三、实验预习要求

- 1) 复习子程序的定义与调用方法。
- 2) 复习中断的概念，了解PC机系统中断向量表的占用情况（见表8-1）。
- 3) 复习子程序的调用过程与软中断调用过程之间异同。
- 4) 阅读示例程序，掌握编写软中断程序的方法及相关技巧。
- 5) 从实验内容中任选一道题目，仔细阅读相关的实验要求及说明，编写程序，以便上机调试。

# 实验八、子程序及软中断 程序设计

## 四、实验内容

从PC机系统中断向量表的占用（[见表8-1](#)）情况可见，42H~4FH是系统未使用的中断向量，我们可利用它们开发用户自己软中断程序。

- 1) 【示例】实验要求利用47H号中断将一组字符转换成16进制数码，并在屏幕上显示出来。参见参考程序清单。
- 2) 编程求出一系列数据中的最大值、最小值，并送屏幕显示。要求：
  - ①将求最大值和最小值程序的程序段编写为46H号软中断，利用46H号中断调用实现求最大值和最小值功能；
  - ②将送屏幕显示用子程序实现。

# 实验八、子程序及软中断 程序设计

## 五、实验说明

- ❖ 用户可将常用的具有特定功能的程序段编写成子程序使用。一般过程定义伪操作的格式为：

过程名 PROC Attribute

.....

过程名 ENDP

- ❖ **Attribute**是指类型属性，可以是**NEAR**或**FAR**，调用程序和过程在同一个代码段中使用**NEAR**，不在同一个代码段中，使用**FAR**。



表 8.1 PC 机系统中中断向量表的占用情况

中断 向量号	功 能	内存地址	中断 向量号	功 能	内存地址
00H	被零除	000~003H	20H	程序终止退出	080~083H
01H	单步	004~007H	21H	系统功能调用	084~087H
02H	不可屏蔽中断	008~00BH	22H	程序结束地址	088~08BH
03H	断点中断	00C~00FH	23H	Ctrl+C 处理	08C~08FH
04H	溢出	010~003H	24H	严重错误处理	090~093H
05H	屏幕打印	014~017H	25H	绝对磁盘读	094~097H
06H~07H	保留	018~01FH	26H	绝对磁盘写	098~09BH
08H	日时钟中断	020~023H	27H	结束但驻留内存	09C~09FH
09H	键盘中断	024~027H	28H~3FH	保留	0A0~0FFH
0AH	接从片 8259A	028~02BH	40H	新软盘中断向量	100~103H
0BH	COM2	02C~02FH	41H	硬盘参数块	104~107H
0CH	COM1	030~033H	42H~4FH	未使用	108~13CH
0DH	硬盘(XT)/并口 2 中断	034~037H	50H	警报功能	140~143H
0EH	软盘中断	038~03BH	51H	鼠标功能	144~147H
0FH	打印机/并口 1 中断	03C~03FH	52H~59H	未使用	148~167H
10H	视频显示 I/O 调用	040~043H	5AH	功能	168~16BH
11H	设备配置监测	044~047H	5BH	Revector in 19H	16C~16FH
12H	内存容量监测	048~04BH	5CH	网络应用	170~173H
13H	磁盘 I/O 调用	04C~04FH	5DH~6FH	未使用	174~1BFH
14H	串行通信 I/O	050~053H	70H	实时钟	1C0~1C3H
15H	盒带/多功能	054~057H	71H	保留	1C4~1C7H
16H	键盘 I/O	058~05BH	72H	保留	1C8~1CBH
17H	打印机 I/O	05C~05FH	73H	保留	1CC~1CFH
18H	ROM~BASIC	060~063H	74H	保留	1D0~1D3H
19H	自举程序	064~067H	75H	协处理器中断	1D4~1D7H
1AH	时钟 I/O	068~06BH	76H	硬盘控制器	1D8~1DBH
1BH	Ctrl+Break 入口	06C~06FH	77H	保留	1DC~1DFH
1CH	时钟控制	070~073H	78H~7FH	未使用	1E0~1FFH
1DH	视频初始化参数表	074~077H	80H~85H	为 BASIC 保留	200~217H
1EH	软盘参数表	078~07BH	86H~F0H	BASIC 使用区	218~3C3H
1FH	图形显示扩展字符表	07C~07FH	F1H~FFH	保留	3C4~3FFH

# 实验八、子程序及软中断 程序设计

## 1)、子程序段内调用与返回

(1) 调用格式:      **CALL** 过程名

(2) **CPU**执行**CALL**指令

☞ 首先将断口地址压入堆栈，为返回做准备，然后把子程序入口的有效地址→**IP**，从而转入子程序。

(3) 返回指令:      **RET**

☞ 在具有**NEAR**属性的子程序中，**RET**指令从栈顶弹出2个字节→**IP**，然后**SP+2**→**SP**。从而回到断点。

# 实验八、子程序及软中断 程序设计

## 2)、软中断程序设计

### (1) 中断:

中断就是当系统运行或者程序运行期间在遇到某些特殊情况时,需CPU暂停执行现行程序,自动去处理随机事件,处理完毕后再返回被中断的程序,这一全过程称为中断。处理随机事件的程序,就称为中断服务子程序。中断分为CPU中断(像除法错等)、软件中断(INT n指令而引发的中断)和硬件中断(由CPU以外的器件发出的中断请求信号而引发的中断)。

### (2) 中断向量

中断向量就是中断子程序的入口地址(由段基址CS和有效地址IP组成,占4个字节)存储在中断向量表中。微机系统中可以使用0—255共256个中断。当80x86系统工作在实模式时,内存的000H~3FFH被用作于中断向量表,向量表中包含了256个中断入口,参见表8-1,且在系统启动时由BIOS或DOS负责初始化。

# 实验八、子程序及软中断 程序设计

## (3) 软中断程序设计

- ❖ 用户可自己开发中断服务程序，用它取代系统原有的服务程序，过程如下：
- ❖ 编写中断服务子程序，用IRET指令返回
- ❖ 确定使用的中断号n，修改中断向量表中 $4 \times n \sim 4 \times n + 3$ 单元的对应中断服务程序入口
- ❖ INT n指令调用中断，此时CPU保护现场：将F、CS、IP寄存器入栈，然后从中断向量表中 $4 \times n \sim 4 \times n + 3$ 单元取出n型中断向量写入IP，CS中，CPU根据CS: IP的值转向服务子程序。
- ❖ 当执行IRET时，恢复现场：从堆栈中弹出6个字节传送给IP、CS、F寄存器，回到断点继续执行。
- ❖ 程序结束返回DOS之前，要恢复系统的中断向量设置。

# 实验八、子程序及软中断 程序设计

## 六、实验步骤

- 运行程序，纪录运行结果。
- 使用Rmrun菜单中的Debug调试程序。观察在执行CALL指令及INT N指令时堆栈段中数据的变化及程序的流向。

# 实验九、PC机内部8259中断应用

## 一、实验目的

- ❖ 进一步认识和理解中断特性。
- ❖ 掌握8259中断控制器的工作原理、编程方法以及PC机如何通过8259A实现对外部可屏蔽硬件中断源的管理。
- ❖ 熟悉实验中涉及到的中断屏蔽寄存器IMR和中断服务寄存器ISR等的使用方法。
- ❖ 进一步掌握中断服务程序的设计方法。

## 二、实验设备

PC微机一台

### 三、实验预习要求

- 1) 复习中断的概念，了解PC机系统中断向量表的占用情况（见表8-1）。
- 2) 复习8259中断控制器的工作原理，了解PC机中外部可屏蔽硬件中断的处理过程。
- 3) 复习PC机如何通过8259A实现对外部可屏蔽硬件中断源的管理。
- 4) 仔细阅读实验原理、示例程序，掌握中断服务程序的编写方法及相关技巧。
- 5) 从实验内容中任选一道题目，仔细阅读相关的实验要求及说明，编写程序，以便上机调试。

# 实验九、PC机内部8259中断应用

## 四、实验内容

### 1) 【示例】编写一键盘中断处理程序:

- ❖ 计算键盘中断次数，并用该程序替换系统键盘中断处理程序，使当按键8次（键盘中断产生16次）后，显示中断次数并结束应用。参见参考程序清单。
  - ❖ 键盘中断处理的说明
  - ❖ 当从键盘上键入一个键时，键盘上的处理器首先向微机产生硬件中断请求（IRQ1），然后将该键的扫描码传送给主机。而PC主机在IRQ1中断的作用下，调用09H型中断服务程序从键盘接口电路（口地址60H）读入扫描码，并转换成ASCII码，存入键盘缓冲区。
  - ❖ 本实验置换系统的09H型中断服务程序，完成对键盘中断次数的统计。在该中断服务程序中，必须对键盘控制器（口地址61H）复位。复位键盘控制器只需读出端口61H的内容，并将最高位置1，再写入该端口。此外，按键按下和抬起均会引起中断，所以实际中断次数等于按键次数乘2。
  - ❖ 修改上述程序，将按键的扫描码输出到屏幕上，并统计中断次数。
- ### 2) 利用系统的1CH型中断（又称外扩的日时钟中断），编写程序，实现每隔2S在屏幕上显示一串字符“TIME TO!”，按任意键停止。
- ❖ 参见《32位微型计算机原理与接口技术》教程8.9章节。



# 实验九、PC机内部8259中断应用

## 五、实验原理

- ❖ 硬件中断是由CPU以外的器件发出的中断请求信号而引发的中断。80x86CPU只有两个引脚（INTR和NMI）可以接受外部的中断脉冲，为了管理众多的外部中断源，INTER公司设计了专用的配套芯片——8259A中断控制器

### 1、8259A中断控制器简介

8259A中断控制器将中断源优先级排队、辨别中断源以及提供中断向量的电路于一片中，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置。同时，在不需增加其他电路的情况下，通过多片8259A的级连，能构成多达64级的矢量中断系统。8259A的内部结构和管脚如图9-1所示

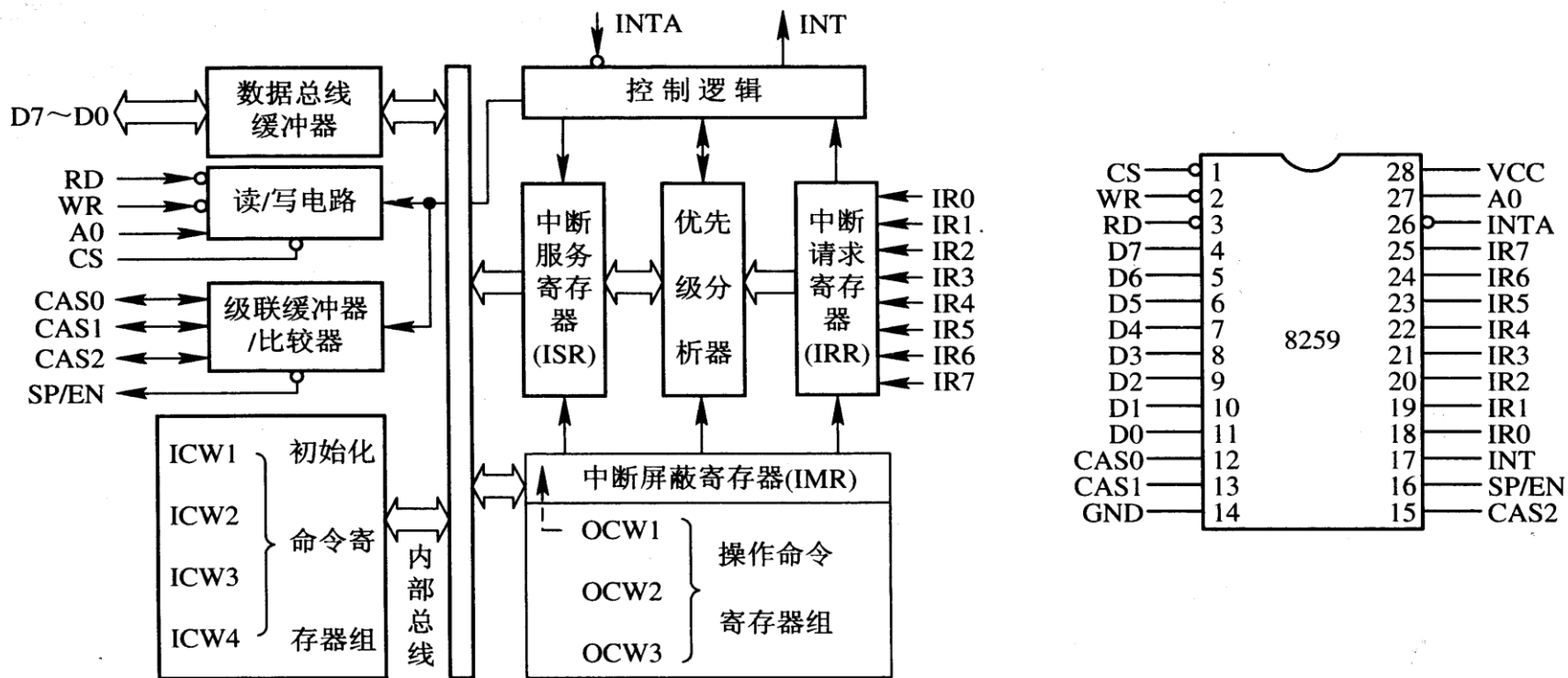


图 9.1 8259 内部结构和管脚图

# 实验九、PC机内部8259中断应用

## 2、8259A的中断过程，即微机系统响应可屏蔽中断的过程

- (1) 将加到引脚IR0~IR7上的中断请求寄存到中断请求寄存器中。
- (2) 在中断屏蔽寄存器的管理下，没有被屏蔽的中断请求被送到优先权电路判优。
- (3) 选中当前级别最高的中断源，然后从引脚INT向CPU发出中断请求信号。
- (4) CPU满足一定的条件后，向8259A发出两个中断响应信号（负脉冲）：
  - 1) 8259A从引脚INTA收到第1个中断响应信号之后，立即使中断服务寄存器中与被选中的中断源对应的那一位置1，同时把中断请求寄存器中的相应位清0。
  - 2) 从引脚INTA收到第2个中断响应信号之后，8259A把选中的中断源类型码n通过数据线送往CPU。
- (5) 在实模式下，CPU从 $4 \times n \sim 4 \times n + 3$ 单元取出该中断源的中断向量 $\rightarrow IP, CS$ ，从而引导CPU执行该中断源的中断服务程序。

# 实验九、PC机内部8259中断应用

## 3、8259A编程

### (1) 初始化编程

- ✎ 提供了4个（ICW1~ICW4）初始化命令字，写入命令寄存器组后，就建立了8259A的基本工作方式。系统8259A的初始化编程在微机启动时，由BIOS自动完成。用户不需再对其初始化，更改它的初始化设置。
- ✎ BIOS对系统8259A初始化为：
  - ❖ 中断触发方式采用边沿触发。
  - ❖ 中断屏蔽方式采用常规屏蔽方式。
  - ❖ 中断优先级的管理采用完全嵌套即固定优先级方式。IR0的请求级别最高，IR7的请求级别最低。
  - ❖ 中断结束，采用常规结束方式。

# 实验九、PC机内部8259中断应用

## (2) 操作方式编程

- ❖ 将操作命令字OCW1~OCW3写入操作命令寄存器组，对中断处理过程实现动态控制。OCW1~OCW3各命令格式如图9-2所示。
  - ↪ OCW1——写中断屏蔽字（对奇地址操作）
    - ❖ 某位 $M_i$ 为1，表示对应的中断源 $IRQ_i$ 被屏蔽； $M_i$ 为0， $IRQ_i$ 被开放。
  - ↪ OCW2——写中断方式命令字（对偶地址操作）
    - ❖ 设置优先级是否进行循环，循环的方式及中断结束的方式。
  - ↪ OCW3——（对偶地址操作）
    - ❖ 用来设置特殊屏蔽方式、查询方式
    - ❖ 用来读8259A的中断请求寄存器IRR，中断服务寄存器ISR、中断屏蔽寄存器IMR的当前状态。

# 实验九、PC机内部8259中断应用

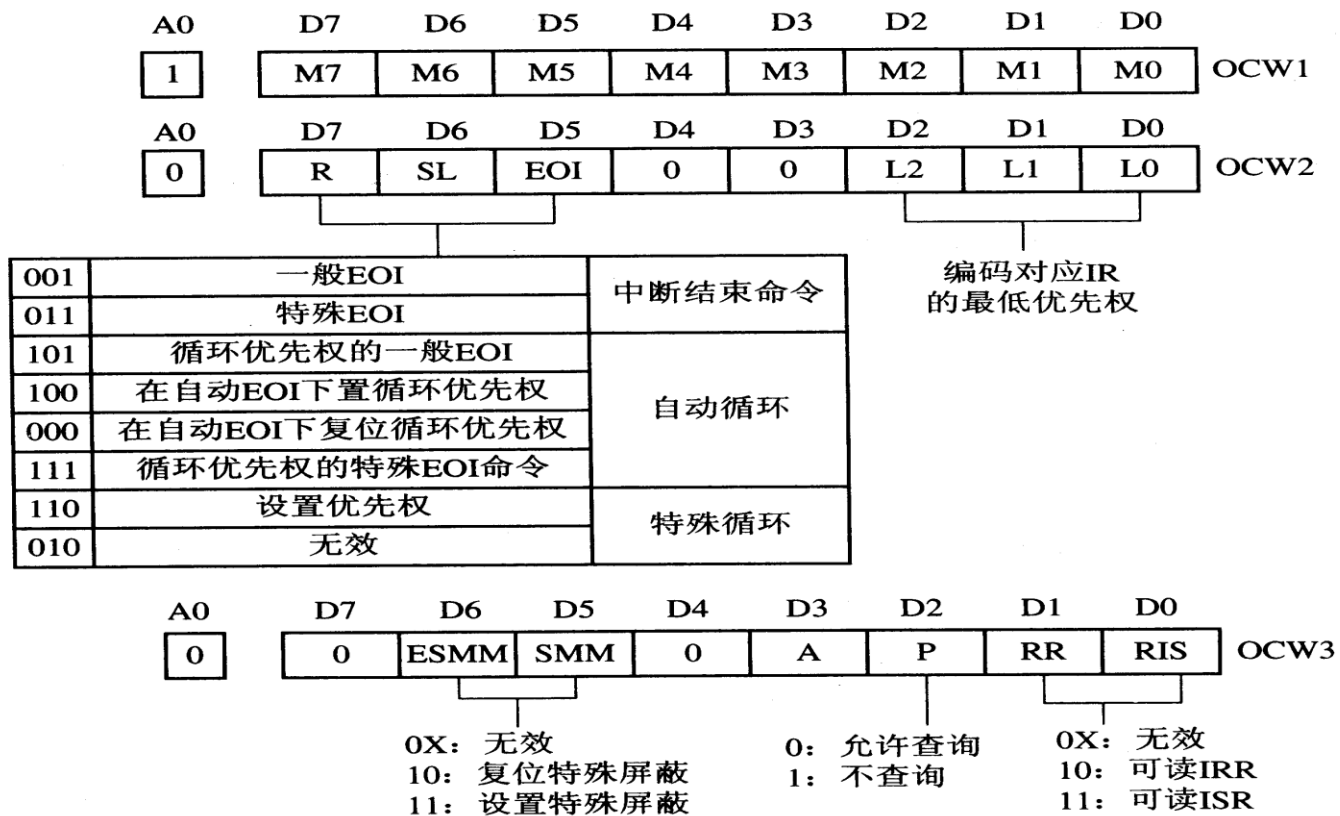


图 9.2 OCW 命令字格式

# 实验九、PC机内部8259中断应用

## 五、实验原理

### 4、PC机系统中8259A的应用

- ❖ 现代PC机系统中包含了两片8259A中断控制器，经级连可以管理15级硬件中断，但其中部分中断源已经被系统硬件占用，具体情况如表9-1所示。中断向量表的占用情况见表8-1。
- ❖ 两片8259A的端口地址为：主片在020H~03FH，实际使用020H和021H两个端口；从片在0A0H~0BFH范围，实际使用0A0H和0A1H两个端口。

表 9.1 PC 机系统中的硬件中断

中断源	功 能	中断向量号	中断源	功 能	中断向量号
主 8259A IRQ0	定时中断	08H	从 8259A IRQ8	实时钟	70H
主 8259A IRQ1	键盘中断	09H	从 8259A IRQ9	保留	71H
主 8259A IRQ2	接从 8259A	0AH	从 8259A IRQ10	保留	72H
主 8259A IRQ3	COM2	0BH	从 8259A IRQ11	保留	73H
主 8259A IRQ4	COM1	0CH	从 8259A IRQ12	保留	74H
主 8259A IRQ5	硬盘/并口 2 中断	0DH	从 8259A IRQ13	写处理器中断	75H
主 8259A IRQ6	软盘	0EH	从 8259A IRQ14	硬盘控制器	76H
主 8259A IRQ7	打印机	0FH	从 8259A IRQ15	保留	77H

# 实验九、PC机内部8259中断应用

## 五、实验原理

5、8259A的应用编程流程，如图9-3所示。

开发用户中断时，做到以下几点：

(1) 硬件方面：将外扩中断源的中断请求（由低电平到高电平的跃变）接入选定的中断源IRQ<sub>i</sub>上。

(2) 软件方面：

- ④ 置换相应的中断向量，即把用户中断子程序的入口地址写入 $4 \times n \sim 4 \times n + 3$ 单元。
- ④ 分别向主从8259A写入屏蔽字，使主片IMR的D2位置0；使从片IMR的D<sub>i</sub>位置0，开放用户中断。(即填写OCW1)
- ④ 每一次中断服务结束，即执行IRET之前，向从片、主片8259A送中断结束命令。(即填写OCW2)
- ④ 返回DOS之前，写中断屏蔽字OCW1，使主片IMR的D2位置1；使从片IMR的D<sub>i</sub>位置1，屏蔽用户中断。恢复系统中断屏蔽字、恢复系统中断向量。



## 6、实验步骤

运行示例程序，按  
键8次，观察运  
行结果。

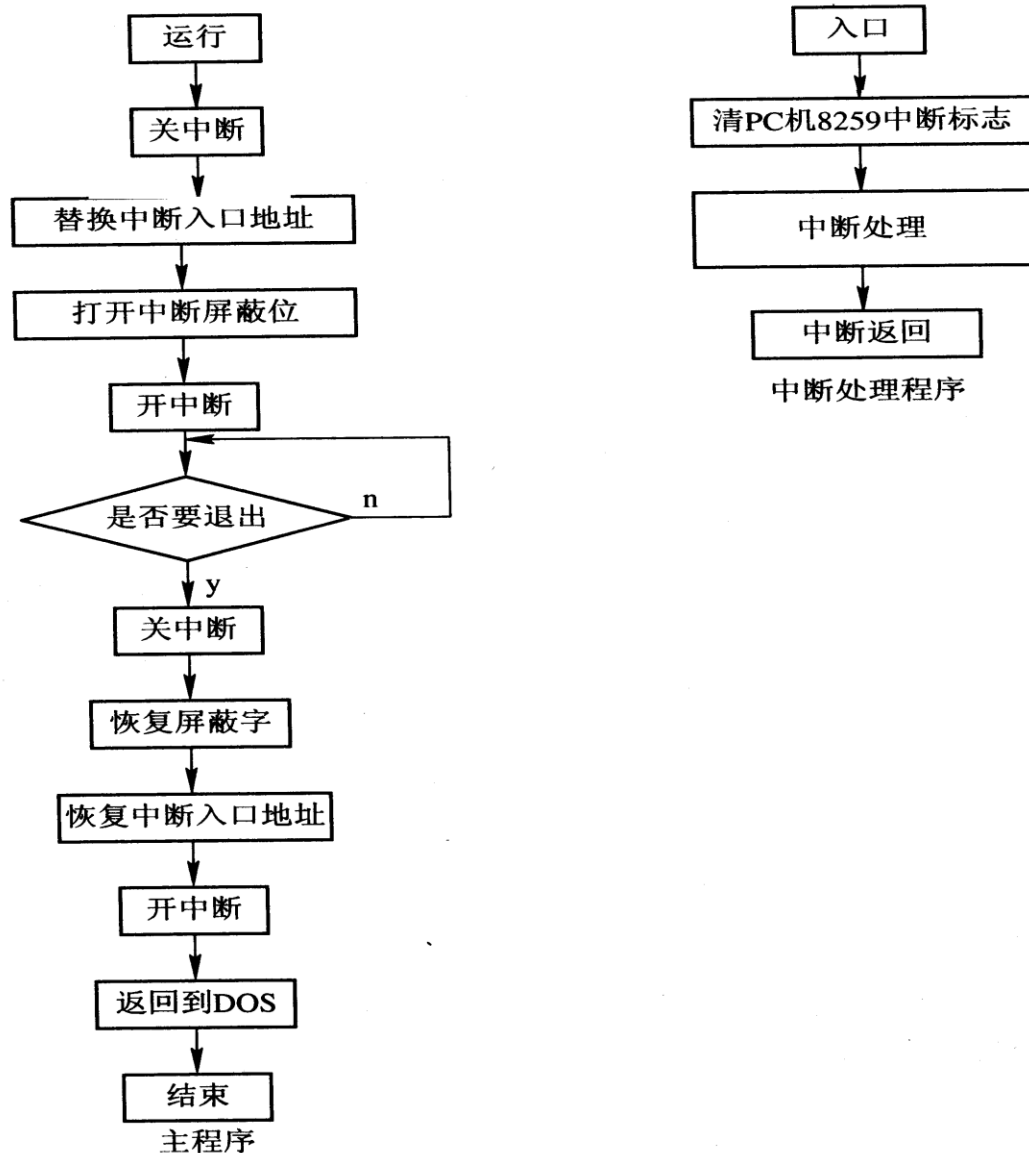


图 9.3 中断应用流程图

# 实验十、 PCI总线中断应用

## 一、实验目的

- ❖ 掌握使用PCI中断的方法。
- ❖ 了解操作PCI设备I/O端口的的方法。
- ❖ 进一步掌握8259中断控制器的工作原理、编程方法以及PC机如何通过8259A实现对外部可屏蔽硬件中断源的管理。
- ❖ 进一步掌握中断服务程序的设计方法。

## 二、实验设备

- ❖ PC机一台， TD-PIT-B实验装置一套。

# 实验十、 PCI总线中断应用

## 三、实验预习要求

- ❖ 1. 复习中断的概念，了解PC机系统中断向量表的占用情况（见表8-1）。
- ❖ 2. 复习8259中断控制器的工作原理，了解PC机中外部可屏蔽硬件中断的处理过程。
- ❖ 3. 复习PC机如何通过8259A实现对外部可屏蔽硬件中断源的管理。
- ❖ 4. 仔细阅读实验说明、示例程序，掌握PCI总线中断服务程序的编写方法。
- ❖ 5. 根据实验内容，仔细阅读相关的实验要求及说明，编写程序，以便上机调试。

## 四、实验内容

- ❖ 1. 【示例】编写一实验程序，利用实验装置提供的中断源，完成每按动一次脉冲开关，产生一次中断，向显示器输出字符“7”。
- ❖ 2. 修改示例程序，控制中断响应次数为10次，即当第11次以后按动脉冲开关，屏幕上不再显示“7”。

# 实验十、 PCI总线中断应用

## 五、实验说明

### 1. TD-PIT-B实验系统编程信息简要说明

- (1) . 本实验的接口电路是在PD-PIT-B实验装置上搭接完成，而PC机作为主机控制该实验装置。二者之间通过PCI总线扩展卡及转接逻辑在PD-PIT-B实验装置上提供了一个仿真ISA的总线接口，用户可以基于该接口对常用接口芯片进行应用编程。TD-PIT-B实验系统的硬件环境，参见附录。
- (2) . 要利用仿真ISA的总线接口资源，扩展接口电路，首先必须掌握实验装置获取的系统配置资源，即PCI总线资源配置的有关内容。

执行PCI\_BIOS.EXE，获取实验用PCI总线扩展卡分配的地址空间及中断请求线。

# 实验十、PCI总线中断应用

```
*****PCI CONFIG INFO*****
Vendor Identification:      10E8
Device Identification:     5933
PCI Command Register:     0007
PCI Status Register:      0000
Revision Identification Register: 00
Interrupt Line:           09
Base Address Register0:    0000B801
Base Address Register1:    0000BC01
Base Address Register2:    0000C001
Base Address Register3:    05000000
Base Address Register4:    0000C401
```

- ❖ 从以上获取的信息可知道，本实验用的PCI总线扩展卡申请了：
- ❖ 4个I/O空间（最后一位为1来表示），其首地址分别是：0B800H，0BC00H，0C000H，0C400H。
- ❖ 一个存储器空间（最后一位为0来表示），其首地址是：5000000H。
- ❖ 一个中断请求IRQ9

# 实验十、 PCI总线中断应用

## 五、实验说明

### 1. TD-PIT-B实验系统编程信息简要说明

(3) . 确定TD-PIT-B实验装置中各接口电路所使用的端口地址

- ❖ 本系统中PCI配置空间的首地址是PCI总线扩展卡申请的第一个I/O空间（即BASE 0）。所以PCI各控制寄存器的地址定义为：

BASE 0的首地址0B800H + 各控制寄存器的偏移地址

- ❖ 而I/O部分实验使用的是PCI总线扩展卡申请的第三个I/O空间（即BASE 2）。各I/O接口芯片的端口地址定义为：

BASE 2的首地址0C000H + 各端口的偏移地址

# 实验十、 PCI总线中断应用

## 2. 实验说明及处理流程

- ❖ 本实验使用实验装置提供的中断请求信号IRQ，利用KK1的负脉冲作为中断源，每按一次KK1，在中断处理中完成字符“7”的显示。
- ❖ 使用实验装置提供的IRQ信号，除了要操作PC机的8259寄存器，还需要操作TD-PIT-PCI总线扩展卡上的控制寄存器“INTCSR”和“IMB4”，才可以实现中断的初始化、清除等。
- ❖ PCI\_BIOS.EXE实验装置获取的系统配置资源，确定控制寄存器INTCSR（偏移38H~ 3BH）和IMB4寄存器（偏移1FH）的端口地址：

BASE 0的首地址0B800H + 各控制寄存器的偏移地址  
及中断号IRQ9。

# 实验十、PCI总线中断应用

## ❖ 初始化PCI中断

- ① 初始化INTCSR: 向INTCSR的端口写入003F1F00H, 即0B838H写00H, 0B839H写1FH, 0B83AH写3FH, 0B83BH写00H。
- ② 初始化中断向量: 在修改中断入口地址时, 建议先保存原来的入口地址。查表9-1获得IRQ9对应的中断向量号为71H, 中断入口地址即为01C4H, 并将该地址保存, 替换为用户自己的中断服务程序入口地址。
- ③ 设置PC机8259中断屏蔽寄存器, 对应位为0, 允许中断。21H中设置0~7号屏蔽位, A1H中设置8~15号屏蔽位。同样保存原先的屏蔽位。IRQ9号中断须将A1H中的位1置0。

## ❖ 清除实验用中断的中断源

- ① 清PCI板卡的中断标志: 对IMB4寄存器的第3字节 (IMB\_BYTE3) 做一读写操作, 端口地址为0B81FH, 然后向INTCSR第2字节 (0B83AH) 写入3FH。
- ② 清PC机8259中断标志: 设置8259的OCW2, 复位中断标志。(IRQ9中断向A0H写入61H, 向20H写入62H。)
- ③ 退出程序返回DOS时恢复系统的初始化的设置: 首先将保存的屏蔽命令字恢复到屏蔽寄存器中, 中断服务程序入口的段地址和偏移地址恢复到中断向量表中, 然后返回DOS。

## ❖ 参考实验流程如图10-1所示。



# 实验十、 PCI总线中断 应用

## 五、实验说明

### 2. 实验说明及处理流 程

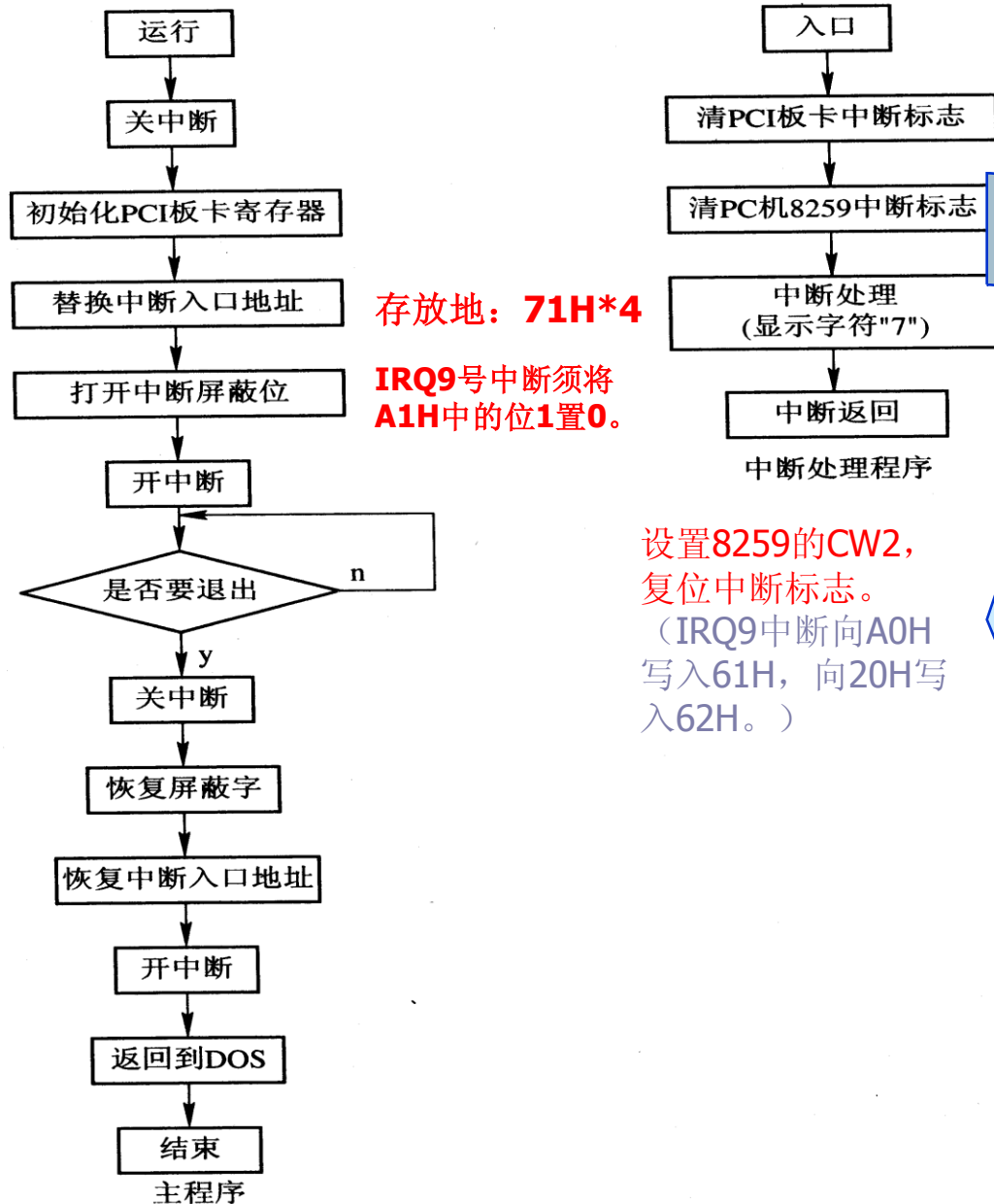


图 10.1 PCI 中断应用实验流程图

# 实验十、 PCI总线中断应用

## 六、实验步骤

- ❖ 分析程序，画出实验流程图。
- ❖ 用排线将KK UNIT中的KK1脉冲开关与仿真ISA总线中的IRQ信号连接。打开实验箱电源。
- ❖ 运行程序，按动KK1，观察运行结果。
- ❖ 注意保留该程序，以便后续程序使用。

# 实验十一、8254定时/计数器应用

## 一、实验目的

- ❖ 掌握8254的工作方式及应用编程。
- ❖ 掌握8254的典型应用电路的接法。
- ❖ 学习8254在PC系统中的典型应用方法。

## 二、实验设备

- ❖ PC机一台，TD-PIT-B实验装置一套。

## 三、实验预习要求

- 1) . 复习8254的工作原理、六种工作方式各自的特点。
- 2) . 复习8254的控制字格式、读/写操作方式以及如何如何进行初始化编程等。
- 3) . 仔细阅读实验原理、说明、示例程序，掌握8254多个方向的应用设计方法及电路连接方式。
- 4) . 根据自选实验内容，仔细阅读相关的实验要求及说明，编写程序，以便上机调试。

# 实验十一、8254定时/计数器应用

## 四、实验内容

1. 【示例】计数应用实验：编写程序，应用8254的计数功能，用开关模拟计数。使每当按动开关KK1五次后，产生中断，并在屏幕上显示一个字符“M”。
2. 定时应用：自己编写程序，应用8254的定时功能，将屏幕设计为一个秒表。
3. 电子发声：自己编写程序让实验装置上的扬声器和PC机上的扬声器唱歌。

# 实验十一、8254定时/计数器应用

## 五、实验原理

1、8254是Intel公司生产的可编程间隔定时器。是8253的改进型，比8253具有更优良的性能。8254具有以下基本功能：

- (1) 有3个独立的16位计数通道；
- (2) 每个计数器可按二进制或十进制（BCD）计数；
- (3) 每个计数器可编程工作在6种不同工作方式；
- (4) 每个计数器允许的最高计数频率为10MNZ（8253为2MHZ）；
- (5) 有读回命令（8253没有），可以读出当前计数单元的内容和状态寄存器内容；

# 实验十一、8254定时/计数器应用

## 五、实验原理

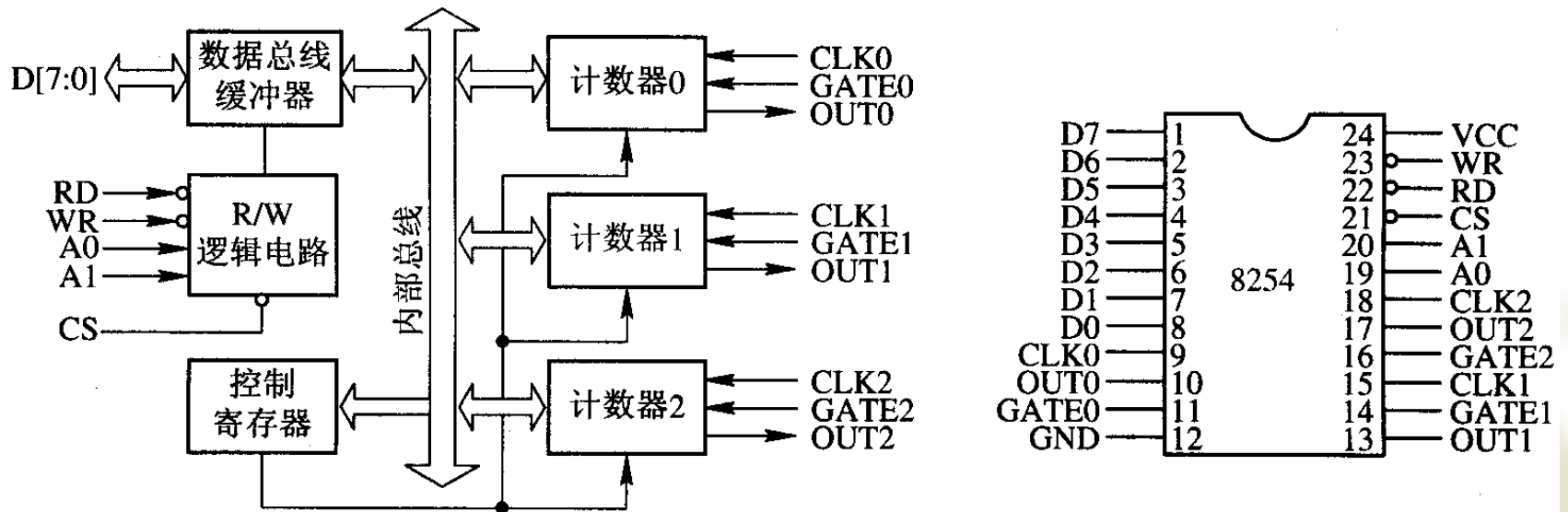


图 8254 的内部接口和管脚

# 实验十一、8254定时/计数器应用

## 五、实验原理

### 2、8254端口地址

❖ 当CS=0时，地址总线的A1 A0组合（00~11），确定选择0#、1#、2#计数器及控制寄存器

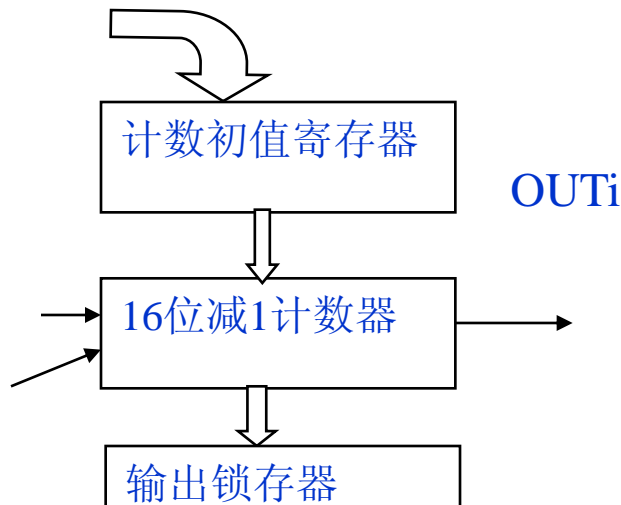
↻ $A_1A_0=00$ ，选中0# 计数器	对应端口的偏移地址是：	40H
↻ $A_1A_0=01$ ，选中1# 计数器		41H
↻ $A_1A_0=10$ ，选中2# 计数器		42H
↻ $A_1A_0=11$ ，选中控制寄存器		43H

# 8254计数器的结构

数据总线

3个引脚：**GATE<sub>i</sub>**——门控信号输入端  
**CLK<sub>i</sub>**——计数脉冲输入端  
**OUT<sub>i</sub>**——信号输出端

**GATE<sub>i</sub>**  
**CLK<sub>i</sub>**



工作过程：

1. 初始化时，程序员将计数初值写入计数初值寄存器，自动送入16位减1计数器。  
计数初值  $N = f_{CLKi} \div f_{OUTi}$
2. 当 **GATE<sub>i</sub> = 1** 时，每一个 **CLK<sub>i</sub>** 信号的下降沿使减1计数器减1，锁存器随之变化。
3. 当计数值减到规定值时，**OUT<sub>i</sub>** 端产生输出信号。

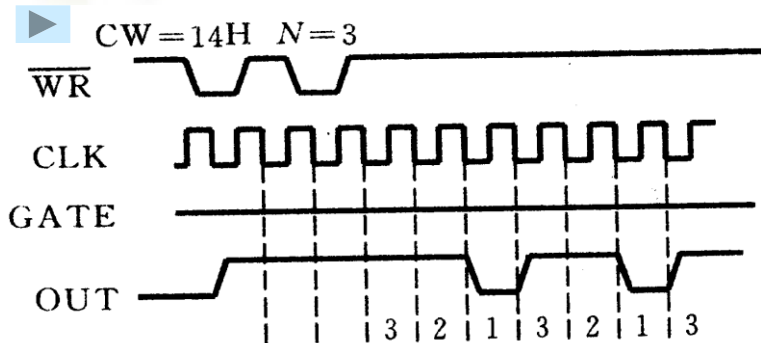


# 8254的工作方式:

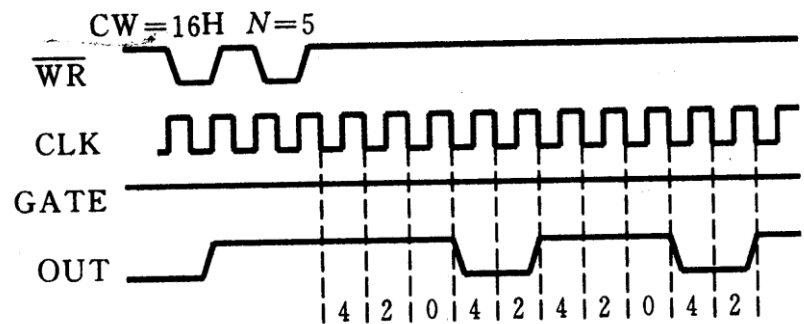
方式0——计数结束输出正跃变信号（可作为中断请求信号）

方式1——单脉冲发生器（形成负脉冲，宽度= $N \times \text{CLK}$ 脉冲周期）

方式2——分频器（输出固定频率的脉冲）



方式2波形图



方式3计数值为奇数时的波形

方式3——方波发生器

方式4——软件触发的单脉冲发生器（输出负脉冲，宽度为一个CLK周期）

方式5——硬件触发的单脉冲发生器（输出负脉冲，宽度为一个CLK周期）

# 8254的控制字:

表 8254 的方式控制字格式

D7	D6	D5	D4	D3	D2	D1	D0
计数器选择		读/写格式选择		工作方式选择			计数码制选择
00—计数器 0		00—锁存计数值		000—方式 0			0—二进制数
01—计数器 1		01—读/写低 8 位		001—方式 1			1—十进制数
10—计数器 2		10—读/写高 8 位		010—方式 2			
11—读出控制 字标志		11—先读/写低 8 位 再读/写高 8 位		011—方式 3			
				100—方式 4			
				101—方式 5			

# 8254编程步骤

- ❖ 一步：向控制寄存器写入控制字，确定使用的计数器及工作方式。
- ❖ 二步：向使用的计数器写入计数初值。

```
MOV DX,MY8254_MODE      ;CLK=10KHZ 实验箱信号源  
MOV AL,15H  
OUT DX,AL
```

```
MOV DX,MY8254_COUNT0    ;N=10000  
MOV AL,0  
OUT DX,A
```

# 实验十一、8254定时/计数器应用

## 六、实验说明及步骤

### 1、确定8254端口地址

- ❖ 执行PCI\_BIOS.EXE，获取实验用PCI总线扩展卡分配的地址空间及中断请求线。确定8254芯片所使用的端口地址：

本系统中I/O部分实验使用的是PCI总线扩展卡申请的第三个I/O空间（即BASE 2，假设为0C000H）。各I/O接口芯片的端口地址定义为：

BASE 2的首地址+ 各端口的偏移地址

- ❖ 8254端口的偏移地址定义为：

计数器0：40H    计数器1：41H

计数器2：42H    控制端口：43H

- ❖ CPU访问8254端口地址为：

计数器0：0C040H

计数器1：0C041H

计数器2：0C042H

控制端口：0C043H

# 实验十一、8254定时/计数器应用

## 六、实验说明及步骤

### 2、计数应用实验

- ❖ 编写程序，将8254的计数器0设置为方式3，计数值为十进制5，用微动开关KK1作为CLK0时钟，OUT0连接IRQ，每当KK1按动5次后产生中断请求，在中断服务程序中完成屏幕上显示字符“M”。参考接线图如图所示。

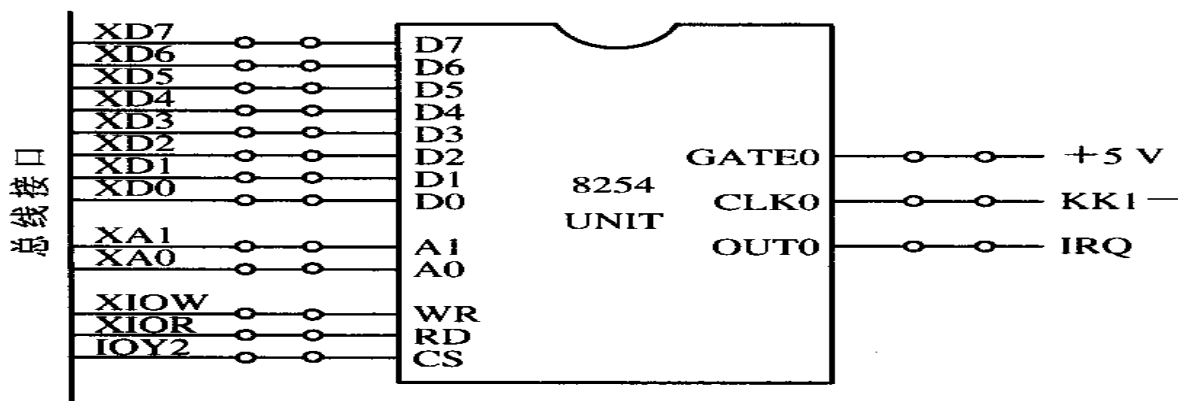


图 8254 计数应用实验参考接线图

# 实验十一、8254定时/计数器应用

## 六、实验说明及步骤

### 2、计数应用实验

❖ 实验步骤如下：

- ① 执行PCI\_BIOS.EXE文件，确定实验装置中8254的各端口地址。按实验内容编写程序（参考程序见后），编译、连接，生成执行文件。
- ② 设计实验线路图，完成线路连接。
- ③ 打开实验箱电源，运行程序，按动KK1微动开关，观察是否按动5次后屏幕显示字符“M”。
- ④ 分析该程序结构，理解如何对8254初始化，如何实现中断服务。
- ⑤ 修改计数初值，从而实现不同要求的计数。

# 实验十一、8254定时/计数器应用

## 六、实验说明及步骤

### 3、定时应用实验——秒表的设计

- ❖ 设计思想：每计时1秒，描述时间的变量就应加1，并且每计时60秒自动归零。因此，可利用8254的计数器0设置为方式2，用实验箱提供的信号源10KHZ作为CLK0时钟，计数值为十进制0（即最大计数值10000），这样每隔1秒就会在OUT0端产生0到1的跃变，该信号接IRQ作为中断请求。在中断服务程序中完成对时钟的修改及显示。
- ❖ 实验步骤如下：
  - 👉 根据设计思想，设计实验线路图，完成线路连接。
  - 👉 画出程序流程图。
  - 👉 执行PCI\_BIOS.EXE文件，确定实验装置中8254的各端口地址。自己编写程序，编译、连接，生成执行文件。
  - 👉 打开实验箱电源，运行程序，观察屏幕的秒表显示是否正常。

# 实验十二、8255并行接口应用

## 一、实验目的

- ❖ 了解8255的基本结构和工作原理
- ❖ 掌握8255的工作方式及应用编程
- ❖ 掌握8255的典型应用电路接法

## 二、实验设备

- ❖ PC微机一台，TD-PIT-B实验系统。

## 三、实验预习要求

1. 复习8255并行可编程芯片的结构及工作原理。
2. 熟悉8255各种工作方式的特点。掌握方式1输入或输出时端口C作联络信号的引脚的定义及功能。
3. 复习8255的工作方式控制字和端口C按位置位复位控制字的格式。
4. 仔细阅读相关的实验原理、要求及说明，预先编写好程序，以便上机调试。



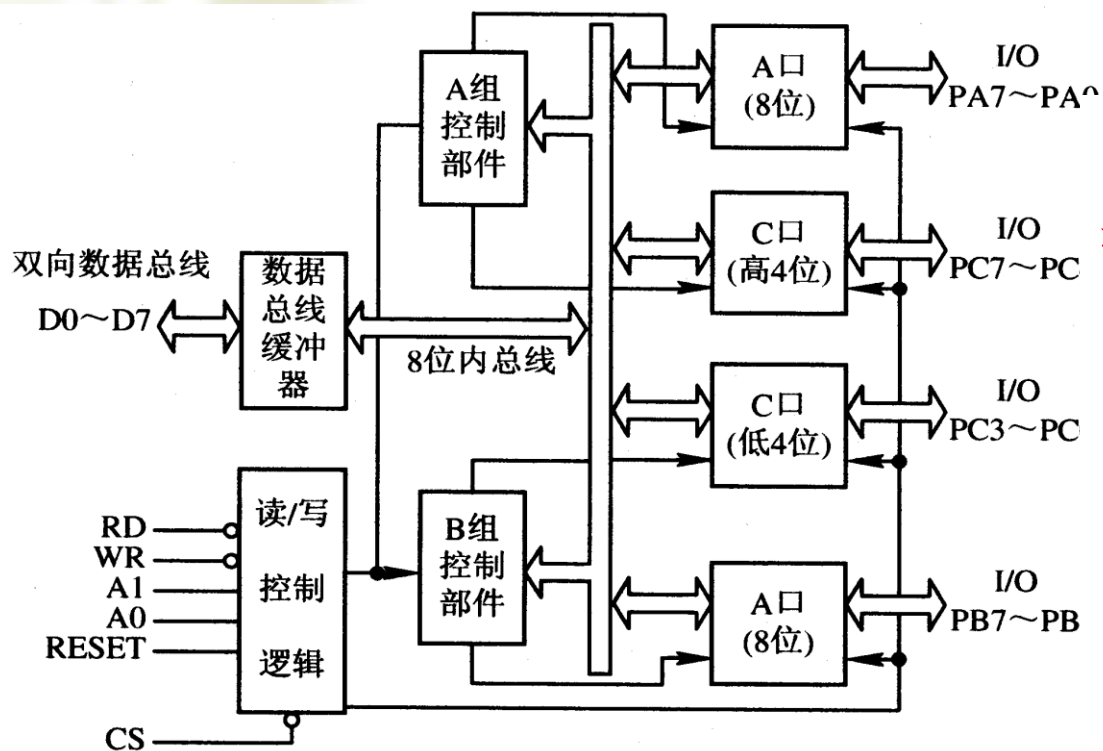
# 实验十二、8255并行接口应用

## 四、实验内容

1. 【示例】基本输入/输出实验：本实验希望实现将一组开关信号通过8255芯片传送到发光二极管上显示。
2. 流水灯显示实验：编写程序，使8255的A口、B口均为输出，实现16位数据灯的相对循环显示。
3. 修改第一个实验，选择8255的A口为基本的输出端接发光二极管，B口为选通输入端接拨动开关。要求当B口数据准备就绪后，通过发STBB信号请求CPU读取B口数据，才送端口A输出显示。分别采用：
  - ❧ 查询方式编程，设计相应的实验电路及程序。
  - ❧ 中断方式编程，设计相应的实验电路及程序。

# 实验十二、8255并行接口应用

## 五、实验原理



当CS=0时，地址总线的A1 A0组合(00~11)，确定选择端口（相对偏移）：

- $A_1A_0 = 00$ ，选中A口， 60H
- $A_1A_0 = 01$ ，选中B口， 61H
- $A_1A_0 = 10$ ，选中C口， 62H
- $A_1A_0 = 11$ ，选中控制寄存器 63H

图 8255的内部结构及管脚

# 实验十二、8255并行接口应用

## 8255三种工作方式：

- ❖ 1) . 方式0——基本输入 / 输出方式。工作特点：
  - ↪ 即为无条件输入 / 输出方式。端口与外设之间不需要联络信号。
  - ↪ A口、B口、C口可由控制字规定为输入 / 输出。
- ❖ 2) . 方式1——选通型输入 / 输出方式。工作特点：
  - ↪ 端口与外设之间需要用联络线，8255A将借用C口的某些端线来完成与外设之间的联络（参见图14-2）。
  - ↪ CPU与8255A之间可以用中断方式或查询方式来交换信息（参见图14-3）。
- ❖ 3) . 方式2——双向选通方式，工作特点：
- ❖ A口为输入输出双向选通
  - ↪ PA7~PA0为双向数据线。
  - ↪ PC4和PC5为一对输入联络线（ $STB_A$ 与 $IBF_A$ ）
  - ↪ PC7和PC6为一对输出联络线（ $OBF_A$ 与 $ACK_A$ ）
  - ↪ PC3为中断请求线
- ❖ B口可以工作在方式0或方式1。

# B口方式1输入

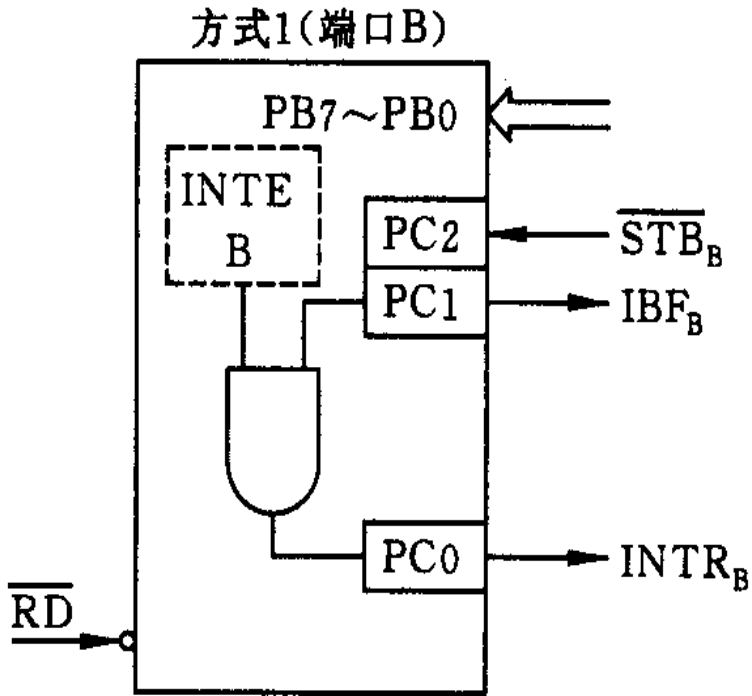


图 B口方式1输入对应的联络信号

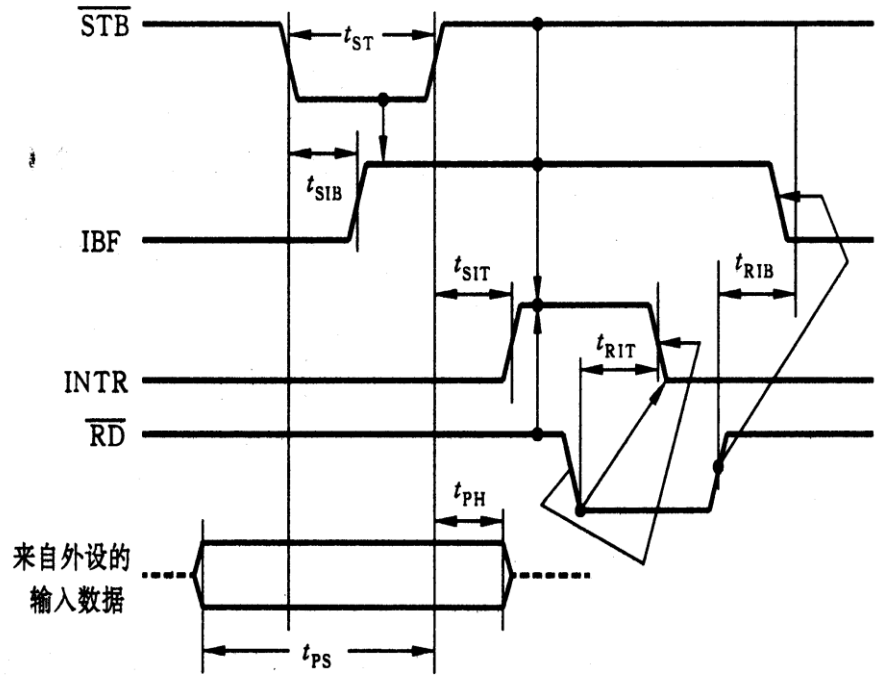
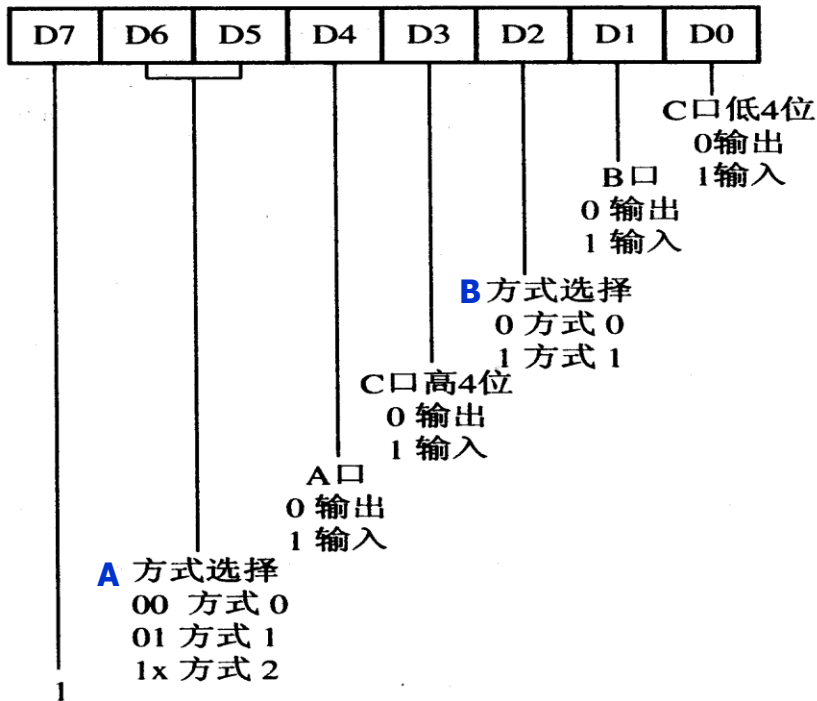
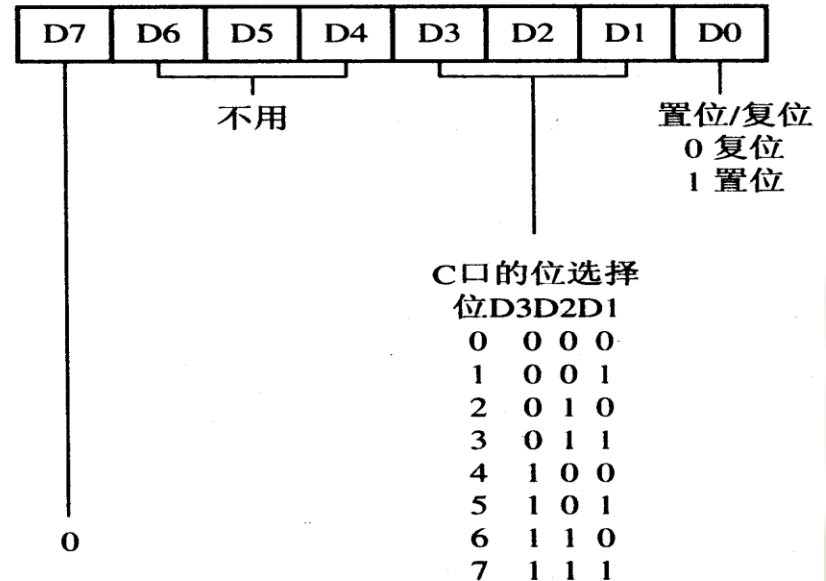


图 方式1的输入时序

# 8255工作方式控制字和 C口按位置位/复位控制字格式



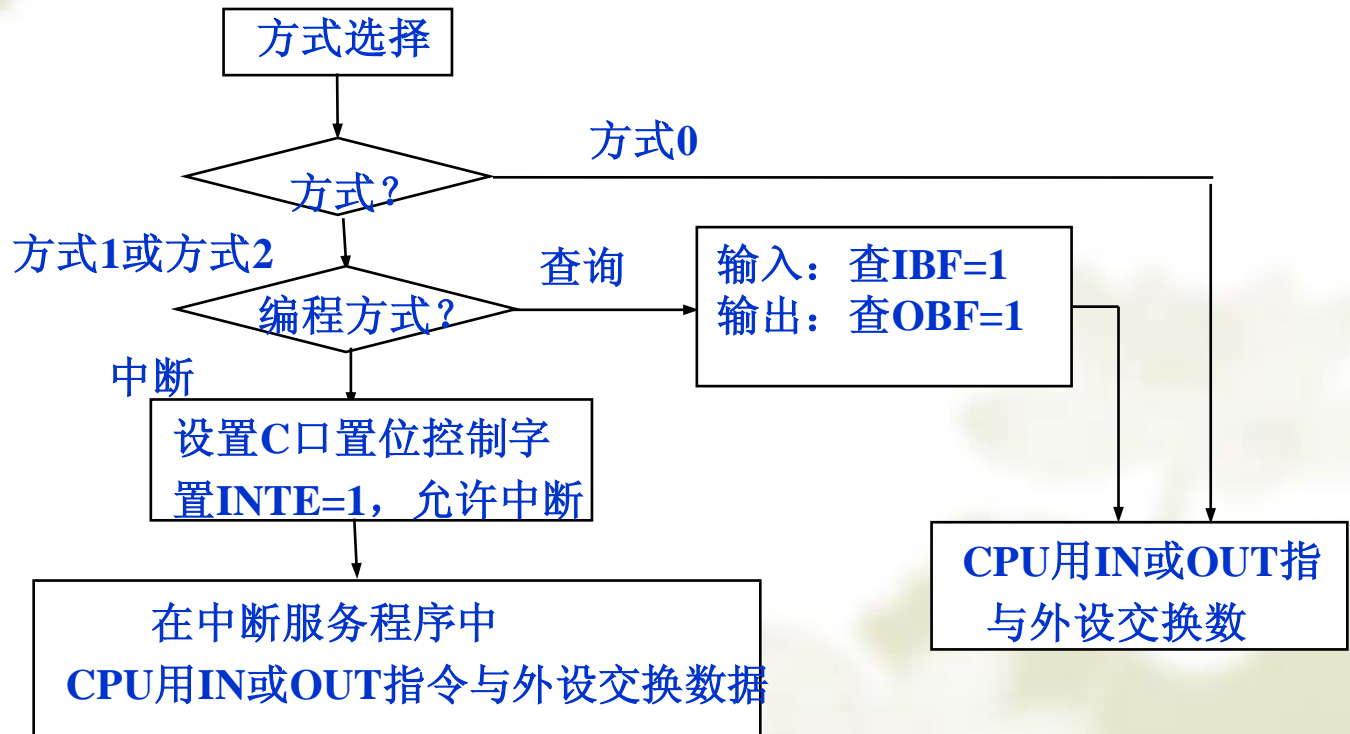
(a) 工作方式控制字



(b) C口按位置位/复位控制字

8255 控制字格式

# 8255应用编程流程



# 实验十二、8255并行接口应用

## 六、实验说明及步骤

- ❖ 执行PCI\_BIOS.EXE，获取实验用PCI总线扩展卡分配的地址空间及中断请求线。确定各接口芯片所使用的端口地址：
  - ✎ 本系统中I/O部分实验使用的是PCI总线扩展卡申请的第三个I/O空间（即BASE 2，假设为0C000H）。各I/O接口芯片的端口地址定义为：  
BASE 2的首地址+ 各端口的偏移地址
- ❖ 8255端口的偏移地址定义为：  
A口：60H    B口：61H    C口：62H    控制端口：63H
- ❖ CPU访问8255端口地址为：  
A口：0C060H    B口：0C061H  
C口：0C062H    控制端口：0C063H

# 实验十二、8255并行接口应用

## 六、实验说明及步骤

### 1. 基本输入/输出实验。

- ❖ 使8255的A口为输出接发光二极管，B口为输入接拨动开关，完成拨动开关到数据灯的数据传送。要求只要开关拨动，数据灯的显示就改变，按任意键结束。实验参考接线图如图所示，程序见参考程序清单。

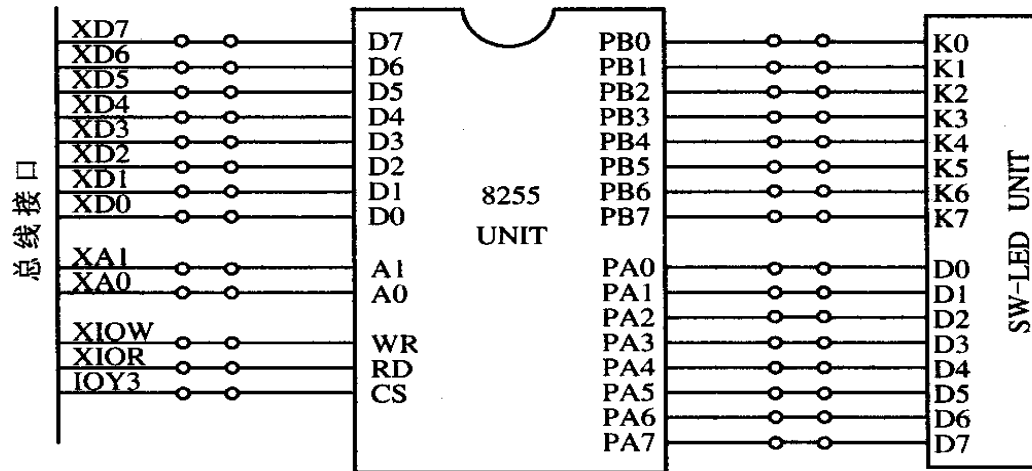


图 8255 基本输入/输出实验参考接线图



# 实验十二、8255并行接口应用

## 六、实验说明及步骤

### 1. 基本输入/输出实验

❖ 具体的实验步骤如下：

- (1) 搭接实验线路。
- (2) 执行PCI\_BIOS.EXE文件，获得BASE 2的首地址，修改参考程序清单1中定义端口地址的伪指令，确定CPU访问8255所使用的地址。
- (3) 编辑、汇编、连接程序。
- (4) 打开实验箱电源，运行程序，只要拨动开关，数据灯的显示就改变。
- (5) 分析程序，画出程序流程。

# 实验十二、8255并行接口应用

## 六、实验说明及步骤

### 2. 流水灯显示实验

使8255的A口、B口均为输出，分别向A口和B口写入7FH和FEH，然后分别将该数定时右移和左移一位，再送端口，这样循环下去，从而实现流水灯的显示，按任意键结束。实验参考接线图如图所示

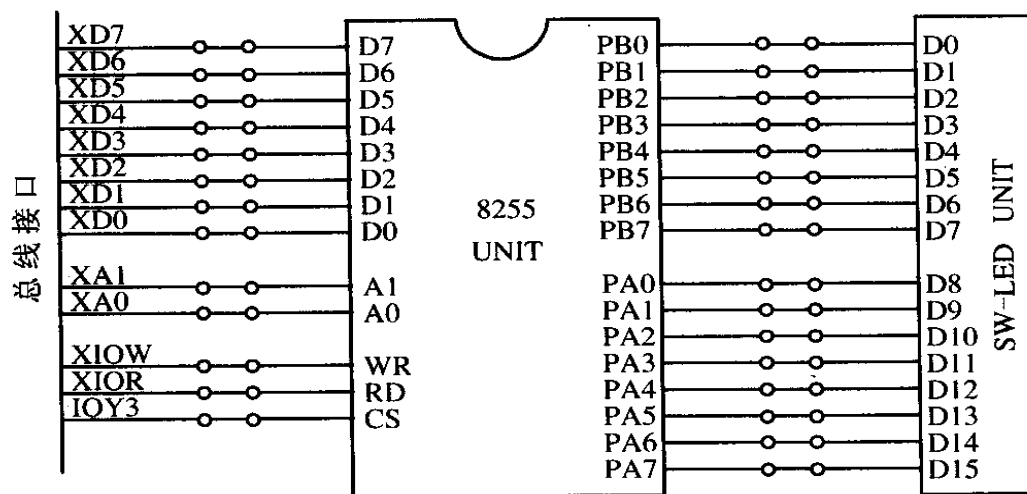


图 8255 流水灯显示实验参考接线图

# 实验十二、8255并行接口应用

## 六、实验说明及步骤

### 2. 流水灯显示实验

❖ 具体的实验步骤如下：

- ① 搭接实验线路。
- ② 执行PCI\_BIOS.EXE文件，确定CPU访问8255所使用的地址。
- ③ 画出程序流程，编写程序。
- ④ 汇编、连接程序。
- ⑤ 打开实验箱电源，运行程序，观察数据灯显示的改变。

结束

谢谢!