第8章 人机交互接口

- 键盘是微机系统上最基本的标准输入设备。
- 键盘按键起开关作用。
- 按键分为：机械式、非机械式两大类。
- 机械式按键常用，靠按键的短暂机械接触动作来使电气触点形成回路，以达到控制开关开启的目的。
- 机械式按键分为四种类型：

1）纯机械式：简单的双金属触点开关。

2）泡沫元件式：在按键所连接的冲杆底部安装了弹性泡沫塑料。

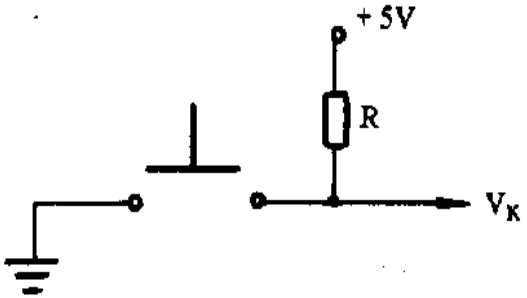
3）橡胶圆顶式：用一个橡胶圆顶来代替弹簧，圆顶下有一个碳材料的按键触点。

4）薄膜式：用一种平坦但有弹性的电路板接收输入并将其传送给键盘的微控制器。可以在没有其他类型的键盘可以胜任的条件下工作。

8.1 键盘接口

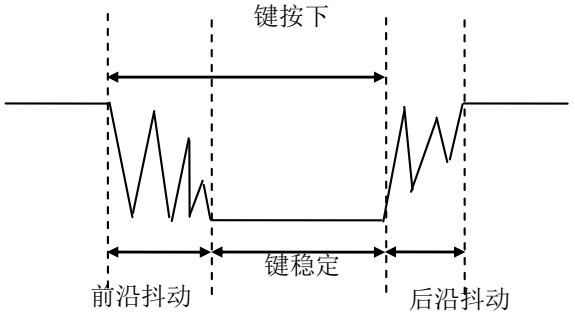
- 按编码提供方式，常用的键盘有两种基本类型：编码键盘、非编码键盘。
- 编码键盘：能够由硬件逻辑自动提供与被按键对应的ASCII码或其它编码。编码键盘中的某一键按下后，能够提供与该键相对应的编码信息。如果是ASCII码键盘，就能提供与该键相对应的ASCII码。编码键盘的缺点是硬件设备随着键数的增加而增加。
- 非编码键盘：仅仅简单地提供被按键行和列的矩阵，其它工作都靠程序实现，这样，非编码键盘就为系统软件在定义键盘的某些操作上提供了更大的灵活性。目前已有一些专用芯片可以完成其中的一些工作。非编码键盘具有价格便宜、配置灵活的特点。
- 在非编码键盘中，为了检测哪个键被按下，必须解决如下问题：

1. 清除键接触时产生的抖动干扰。
2. 防止键盘操作的重键错误。
3. 键盘的结构及被按键的识别。
4. 产生被按键相应的编码。

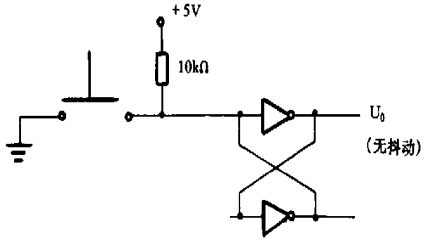


8.1.1 消除抖动及重键处理

- 键盘的按键有机械式、电容式、薄膜式等多种，但就它们的作用而言，都是一个使电路“通”或“断”的开关。在对机械式按键进行键盘输入时，一般存在两个问题，即触点弹跳与同时按下一个以上键的问题，也就是所谓的抖动与重键的问题。
- 1．抖动
- 抖动是开关本身的一个最普遍的问题，它的产生是当机械开关的触点闭合时，在达到稳定之前需要短暂抖动或弹跳几下，即反复闭合、断开几次之后，才能达到可靠地闭合在一起。抖动也存在于开关断开时，其情形与开关闭合时相同。
- 根据所用键的不同质量，键的抖动时间可为10～20ms。键的抖动会引起一次按键被读入多次。



- 解决键的抖动：硬件滤波方法、软件延迟方法。
- 硬件滤波：对每一个键加上R-C滤波电路，或加上RS去抖电路。
- 软件去抖动技术：采用一个产生20ms左右延迟的子程序，以等待键的输出达到完全稳定后才去读取代码。



2. 重键

- 重键：指两个或两个以上的键同时按下，或者一个键按下后还未弹开，另一个键又按下的情况。
- 解决方法3种：
- （1）两键同时按下保护技术，2种：

1）当只有一个键按下时才读取键盘的输出，并且认为最后仍被按下的键是有效的正确按键。这种方法常用于软件扫描键盘场合。

2）当采用硬件技术时，往往采用锁定的方法。锁定保护方法的原理是，当第一个键未松开时，按第二个键不起作用，不产生选通脉冲。这可以通过锁定内部延迟机构来实现，锁定的时间和第一键闭合时间相同。

- （2）n键同时按下保护技术，2种：

1）不理会所有被按下的键，直至只剩下一个键按下时为止；

2）将所有按键的信息存入内部键盘输入缓冲器，逐个处理。这种方法成本较高，在较便宜的系统中很少采用。

- （3）n键连锁技术：当一键被按下时，在此键未完全释放之前，其它的键虽然可被按下或松开，但并不产生任何代码，这种方法实现起来比较简单，因而比较常用。

8.1.2 线性键盘

- 从按键连接方式，键盘分为：线性键盘、矩阵键盘。
- 线性键盘：采用独立式按键，直接用I/O口线构成单个按键电路。每一按键互相独立地各自接通一条输入I/O口线。
- 线性键盘电路配置灵活，软件结构简单。但每个按键必须占用一根I/O口线，在按键数量较多时，I/O口线浪费较大。故在按键数量不多时，常采用这种按键电路。
- 例：
- 假设8255A的A口、B口、C口、控制口的端口地址分别是60H、61H、62H、63H，采用软件消抖技术（只考虑前沿消抖），编程实现对按键K3～K0的识别，假设按键K3～K0的对应编码为3～0，识别按键后，将对应的编码存到AH寄存器中。有D20ms延时子程序可以调用。

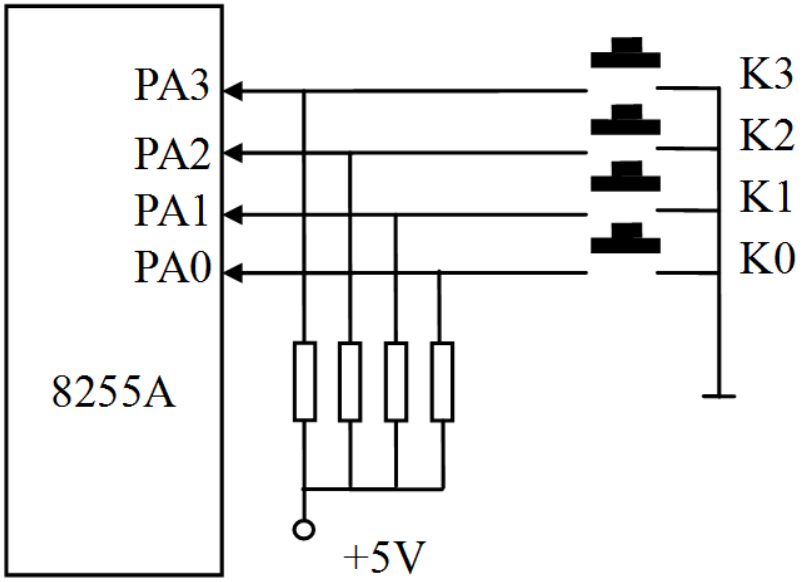


图 8.1.2　线性键盘电路

程序设计

```
CODE    SEGMENT
        ASSUME  CS:CODE
KEY     PROC    FAR
START:  PUSH    DS
        MOV     AX, 0
        PUSH    AX
        MOV     AL, 90H
        OUT     63H, AL     ；设置8255的A口为方式0，输入
X1:     IN      AL, 60H     ；输入A口键盘状态
        AND     AL, 0FH     ；析取K3～K0信号线
        CMP     AL, 0FH
        JZ      X1          ；没有键按下，继续查询
        CALL    D20ms       ；有键按下，延时消抖
        IN      AL, 60H     ；输入A口键盘状态
        AND     AL, 0FH     ；析取K3～K0信号线
        CMP     AL, 0FH
        JZ      X1          ；此时，说明延时消抖前的按键判断是源于干扰，
                            ；或者，延时消抖时间不足，重新查询
```

标准程序段前缀

8255初始化

判断是否有键按下

软件消抖

确认是否有键按下

程序设计

```
        CMP     AL, 00001110B
        JNZ     X2              ；不是单键K0按下，转
        MOV     AH, 0           ；设置K0的编码
        JMP     XEND
X2:     CMP     AL, 00001101B
        JNZ     X3              ；不是单键K1按下，转
        MOV     AH, 1
        JMP     XEND
X3:     CMP     AL, 00001011B
        JNZ     X4              ；不是单键K2按下，转
        MOV     AH, 2
        JMP     XEND
X4:     CMP     AL, 00000111B
        JNZ     X5              ；不是单键K3按下，转
        MOV     AH, 3
        JMP     XEND
```

若单键K0按下，则设置K0编码

若单键K1按下，则设置K1编码

若单键K2按下，则设置K2编码

若单键K3按下，则设置K3编码

程序设计

```
X5:     MOV     AH, 0FFH        ；此时说明有多个按键同时按下，
                                ；用0FFH表示这种状态
XEND:   NOP                     ；在此处可加入其他需要处理的程序
        RET
KEY     ENDP
CODE    ENDS
        END     START
```

设置缺省状态

8.1.3 矩阵键盘

- 矩阵式键盘：又叫行列式键盘，用I/O口线组成行、列结构。按键设置在行列的交点上。
- 在这种矩阵键盘结构中，对按键的识别是对键盘扫描后，通过软件来完成的。
- 键盘扫描方式，两种：行扫描法，线反转法。

行扫描法：是步进扫描方式，每次向键盘的某一行发出扫描信号，同时通过检查列线的输出来确定闭合键的位置。

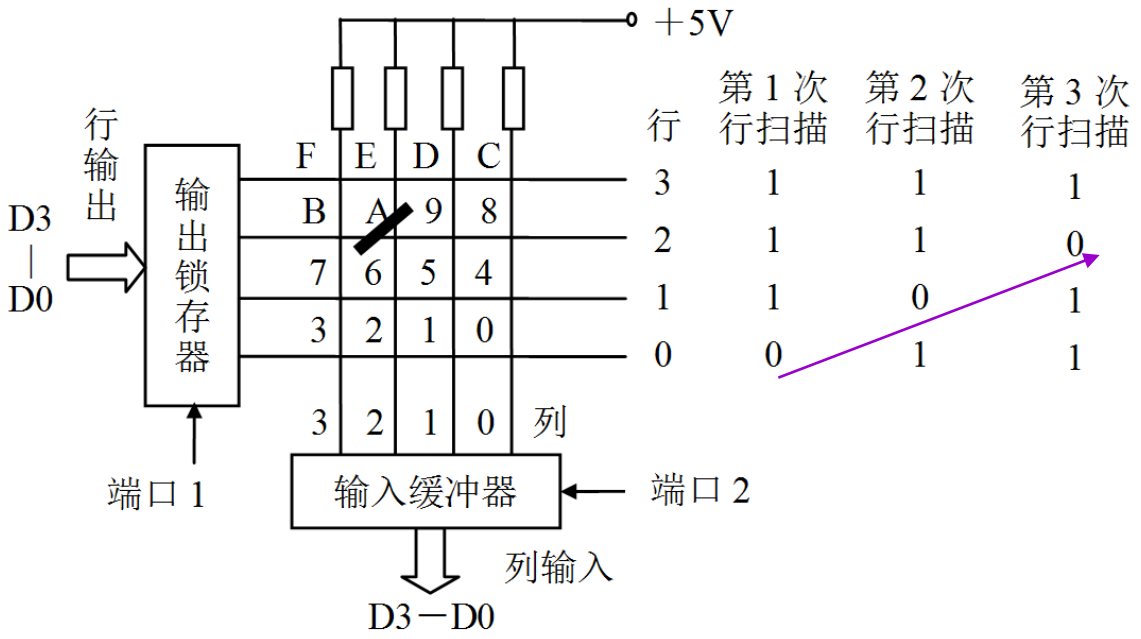


图 8.1.3　行扫描法

例：用行扫描法识别键盘按键

- 假设行输出端口1的地址为200H，列输入端口2的地址为201H，采用软件消抖技术（只考虑前沿消抖），编程实现对0键～F键的识别，识别按键后，将按键的键号（即0～Ｆ）存到AH寄存器中，若为重键，则将0FFH存到AH寄存器中。有D20ms延时子程序可以调用。
- 键的位置码是由行号和列号组合而成的一个字节数据，4位行号占据键位置码的高4位，4位列号占据键位置码的低4位，比如，B键的行号为1011，列号为0111，则B键的位置码为10110111。

键的位置码组成，一个字节

4位行号	4位列号

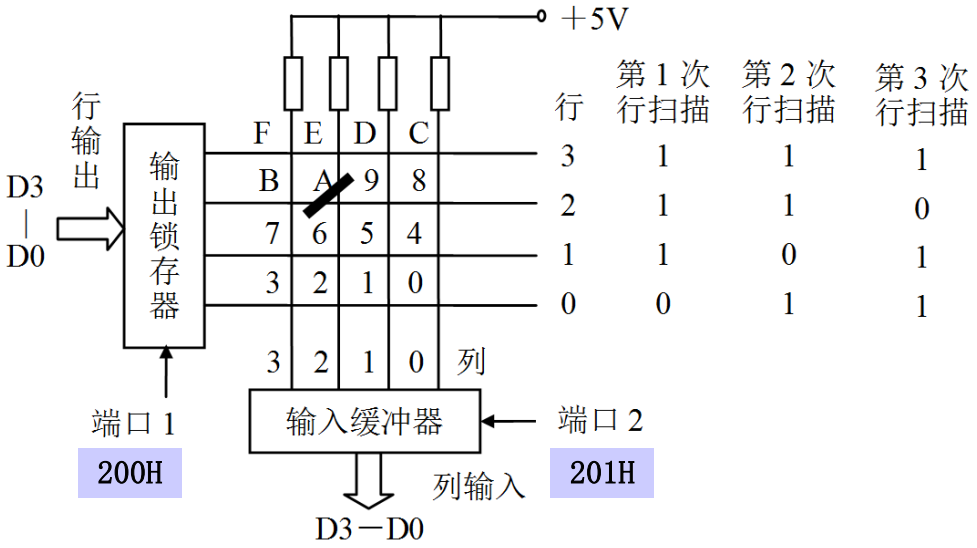


图 8.1.3　行扫描法

程序设计

键值表

```
DATA    SEGMENT
TABLE   DB      11101110B       ；第0行第0列，0键的位置码
        DB      11101101B       ；第0行第1列，1键的位置码
        DB      11101011B       ；第0行第2列，2键的位置码
        DB      11100111B       ；第0行第3列，3键的位置码
        DB      11011110B       ；第1行第0列，4键的位置码
        DB      11011101B       ；第1行第1列，5键的位置码
        DB      11011011B       ；第1行第2列，6键的位置码
        DB      11010111B       ；第1行第3列，7键的位置码
        DB      10111110B       ；第2行第0列，8键的位置码
        DB      10111101B       ；第2行第1列，9键的位置码
        DB      10111011B       ；第2行第2列，A键的位置码
        DB      10110111B       ；第2行第3列，B键的位置码
        DB      01111110B       ；第3行第0列，C键的位置码
        DB      01111101B       ；第3行第1列，D键的位置码
        DB      01111011B       ；第3行第2列，E键的位置码
        DB      01110111B       ；第3行第3列，F键的位置码
DATA    ENDS
```

程序设计

```
CODE    SEGMENT
ASSUME  CS:CODE, DS:DATA
KEY     PROC    FAR
START:  PUSH    DS
        MOV     AX, 0
        PUSH    AX
        MOV     AX, DATA
        MOV     DS, AX
X1:     MOV     DX, 200H        ；设置行输出端口地址
        MOV     AL, 00H
        OUT     DX, AL          ；行输出0000，准备检查是否有任何键按下
        INC     DX              ；设置列输入端口地址201H
        IN      AL, DX          ；输入列线状态
        AND     AL, 0FH         ；析取D3～D0列信号线
        CMP     AL, 0FH
        JZ      X1              ；没有任何键按下，继续查询
        CALL    D20ms           ；有键按下，延时消抖
```

标准程序段前缀

判断是否有键按下

软件消抖

程序设计

确认是否有键按下

```
MOV     DX, 200H ；设置行输出端口地址
MOV     AL, 00H
OUT     DX, AL   ；行输出0000，消抖后确定是否有任何键按下
INC     DX       ；设置列输入端口地址201H
IN      AL, DX   ；输入列线状态
AND     AL, 0FH  ；析取D3～D0列信号线
CMP     AL, 0FH
JZ      X1       ；此时，说明延时消抖前的按键判断是源于干扰，
                 ；或者，延时消抖时间不足，重新查询
```

程序设计

```
        MOV     AH, 11111110B   ；设置行扫描初值，首先扫描第0行
        MOV     CX, 4           ；设置扫描行数计数值，共4行
X2:     MOV     DX, 200H        ；设置行输出端口地址
        MOV     AL, AH          ；传递行扫描值
        OUT     DX, AL          ；行扫描值输出，准备检查键按在哪一列
        INC     DX              ；设置列输入端口地址201H
        IN      AL, DX          ；输入列线状态
        AND     AL, 0FH         ；析取D3～D0列信号线
        CMP     AL, 0FH
        JNZ     X3              ；找到按键所在列号，转，列号保存在AL中
        ROL     AH, 1           ；AH循环左移一位，准备扫描下一行
        LOOP    X2              ；4行未全部扫描完，转
        MOV     AH, 80H         ；4行全部扫描完，却未发现有键按下（可能出现
                                ；了干扰），以80H作为这种情况的标志。
                                ；该指令的设置，主要考虑到程序的完备性，
                                ；即可以使程序在任何情况下都能正确执行。
        JMP     XEND
```

扫描初值

行输出

列输入

确定按键所在列

扫描下一行

缺省处理

程序设计

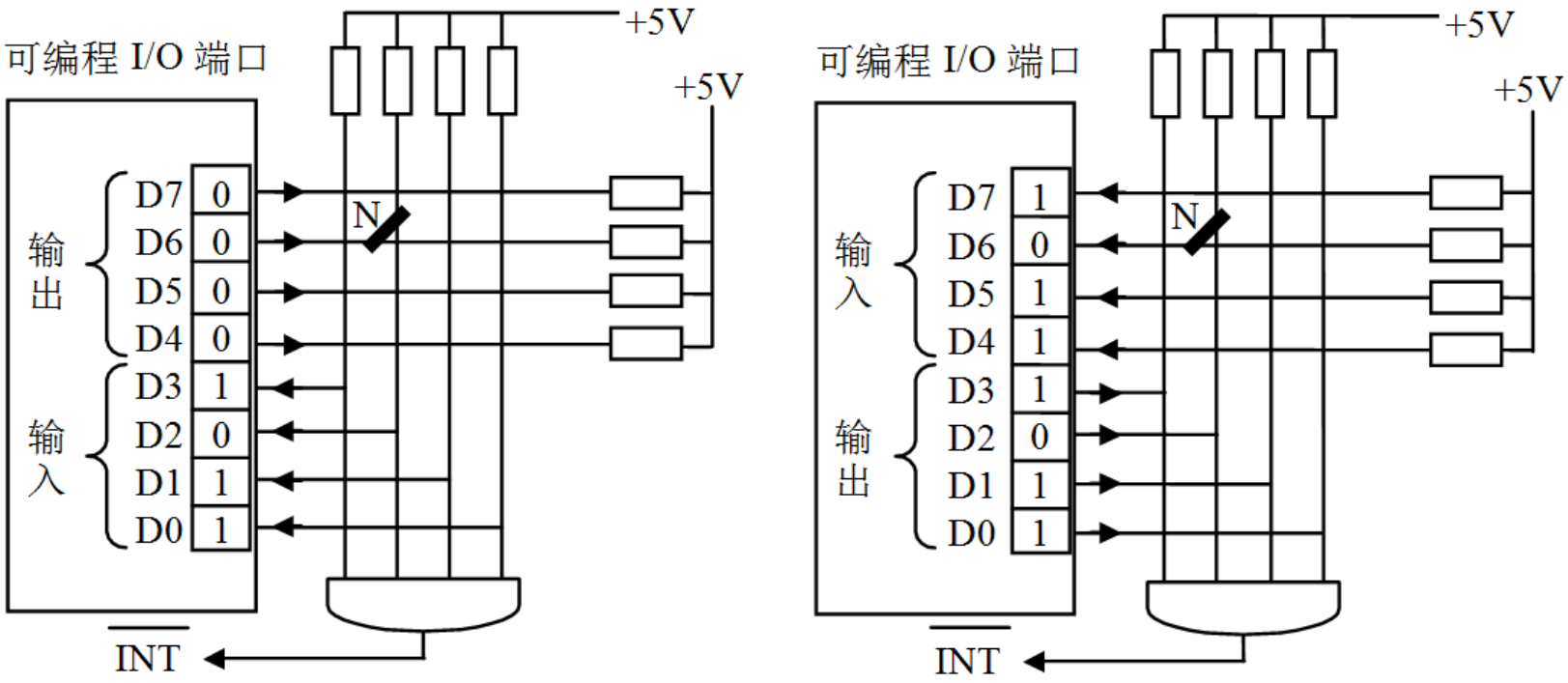
```
X3:     MOV     CL, 4
        SHL     AH, CL      ；AH逻辑左移4位，将低4位的行号移到高4位
        OR      AL, AH      ；行号与列号相“或”，形成键的位置码
        LEA     BX, TABLE   ；设置TABLE位置码表的指针
        MOV     CL, 0       ；设置键号初值为0
X4:     CMP     AL, [BX]    ；在TABLE表中查找本次形成的键位置码
        JZ      X5          ；找到，转，对应的键号就在CL中
        INC     CL          ；未找到，键号加1
        INC     BX          ；指向下一个存储单元保存的键位置码
        CMP     CL, 10H     ；键号等于10H吗？
        JNZ     X4          ；不等，继续查找
        MOV     AH, 0FFH    ；CL等于10H，说明在TABLE表中没有找到
                            ；对应的键位置码，其原因可能是出现了重键
                            ；的情况，以0FFH作为这种情况的标志。
        JMP     XEND
X5:     MOV     AH, CL      ；将CL中保存的键号传到AH中
XEND:   NOP                 ；在此处可加入其他需要处理的程序
        RET
KEY     ENDP
CODE    ENDS
        END     START
```

形成位置码

查表，计算键号，方法不同则键号不同

缺省处理

线反转法



（a）第一步：行输出　　（b）第二步：线反转

图 8.1.5　线反转法原理图

8.1.4 键盘工作方式

- 键盘3种工作方式：程序控制扫描方式，定时扫描方式，中断扫描方式。
- 1．程序控制扫描方式
- 这种方式是利用CPU工作的空余时间，调用键盘扫描子程序，响应键盘的输入请求。
- 2．定时扫描方式
- 这种方式是利用定时器产生定时中断(例如10ms)，CPU响应中断后对键盘进行扫描，并在有键按下时转入键功能处理程序。定时扫描方式在本质上是中断方式，但不是实时响应，而是定时响应。
- 3．中断扫描方式
- 当应用系统工作时，并不经常需要键的输入，因此，无论键盘是工作于程控方式还是定时方式，CPU都经常处于空扫描状态。为了进一步提高CPU效率，可以采用中断扫描方式，当键盘上有键闭合时便产生中断请求，CPU响应中断，执行中断服务程序，对闭合键进行识别，并作相应的处理。

8.1.5 PC机键盘与接口

- PC系列机都采用非编码键盘，其按键排列为矩阵式。不同时期的PC系列机配有物理上各不相同的键盘。
- 主要的键盘类型：

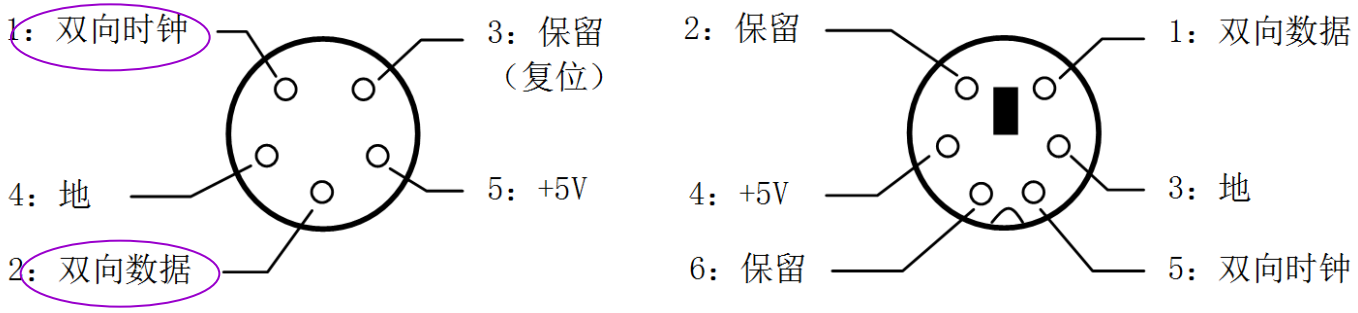
  1）83键PC/XT键盘（已淘汰）（一般称作标准键盘）

  2）84键PC/AT键盘（已淘汰）

  3）101键键盘（386、486机型）

  4）104键Windows键盘（Pentium机型）

  5）108键Windows键盘：在104键盘的基础上又增加了Windows 98功能键Power（关机）、Sleep（休眠）、WakeUp（唤醒）和Fn组合键。
- 早期的PC机、PC/XT机和一些增强型扩展键盘使用的是5针电缆插，后来使用6针微型电缆插头，现在键盘多数使用USB接口。



PC，PC/XT5 针插头　　PC/AT6 针微型插头

图 8.1.6　键盘 5 针插头和 6 针微型插头接线

PC机104键盘布局与位置关系

PC键盘扫描码与按键的对应关系-标准83键

扫描码：键盘输出的数据信号

在系统键盘缓冲区中， ASCII码存放在低字节，扫描码存放在高字节

扫描码是指后面的字节，前面是对应的ASCII码

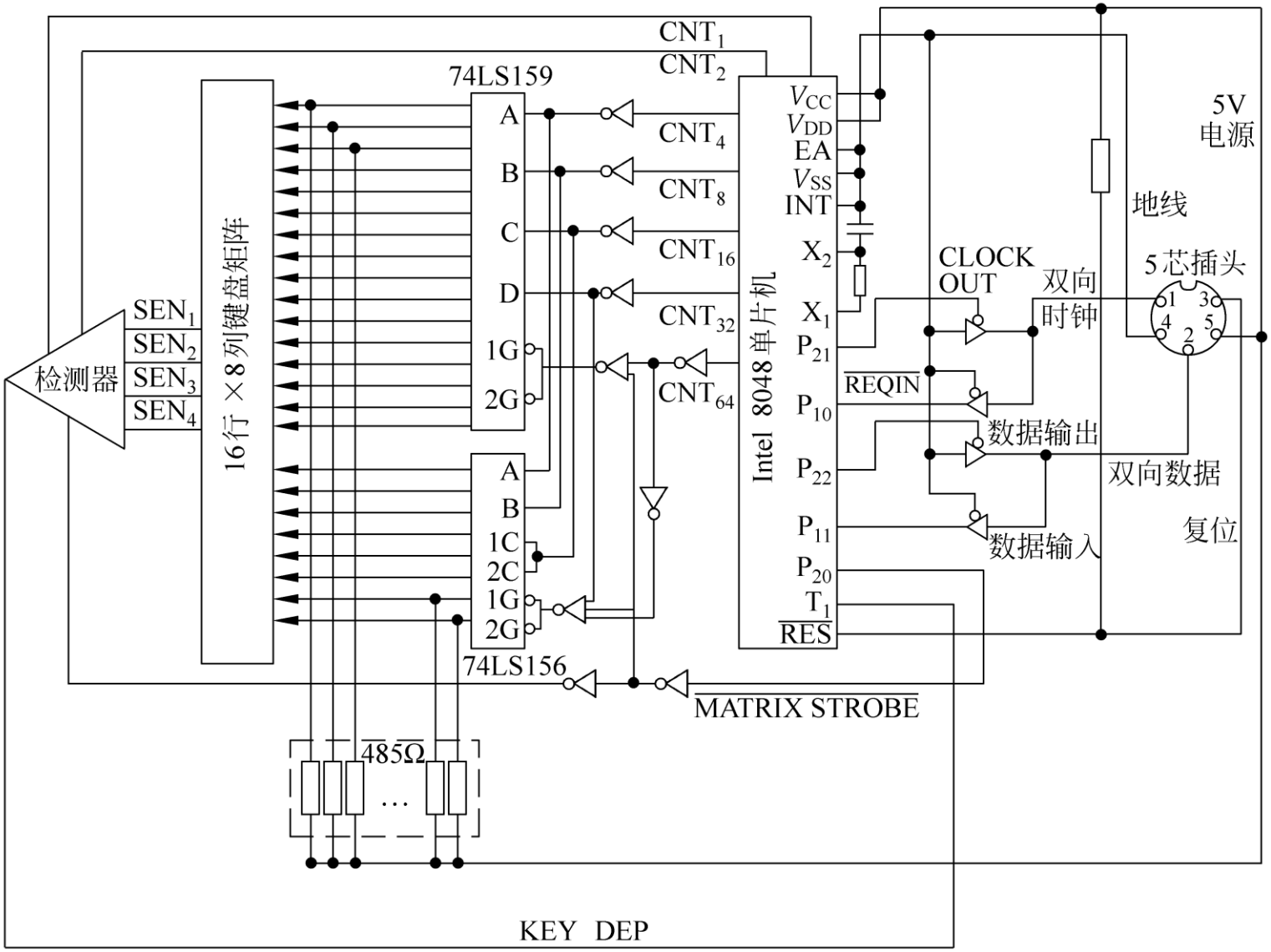
ASCII码　　系统扫描码

扫描码反映键的位置和键的接通或断开状态。1个键的接通与断开分别输出接通扫描码和断开扫描码

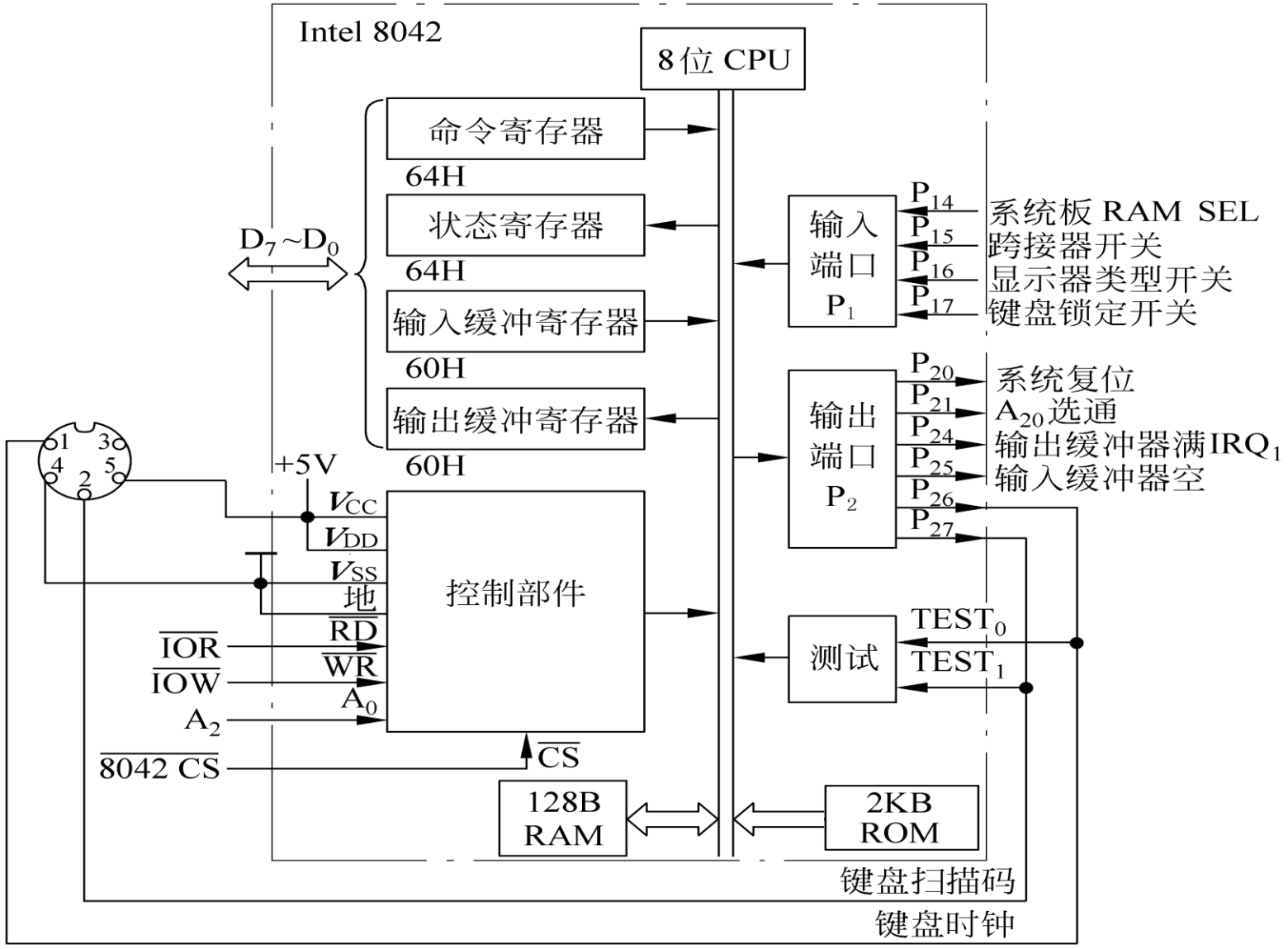
标准键盘的扫描码用1个字节表示。接通扫描码是键号的二进制数，断开扫描码由接通扫描码的最高位置1形成。如f键的接通扫描码为21H，而断开扫描码为21H+80H=A1H。

按键	扫描码	按键	扫描码	按键	扫描码	按键	扫描码
Esc	1B01	u	7516	\	7C2B	F6	0040
1	3102	i	6917	z	7A2C	F7	0041
2	3203	o	6F18	x	782D	F8	0042
3	3304	p	7019	c	632E	F9	0043
4	3405	[	5D1A	v	762F	F10	0044
5	3506	]	5B1B	b	6230	NumLock	0045
6	3607	Enter	0D1C	n	6E31	ScrollLock	0046
7	3708	Ctrl	1D	m	6D32	7/Home	3747
8	3809	a	611E	,	2C33	8/↑	3848
9	390A	s	731F	.	2E34	9/PgUp	3949
0	300B	d	6420	/	2F35	小键盘-	2D4A
-	2D0C	f	6621	Shift(R)	36	4/←	344B
=	3D0D	g	6722	小键盘*	2A37	小键盘5	354C
Backspace	080E	h	6823	Alt(L)	38	6/→	364D
Tab	090F	j	6A24	Space	2039	小键盘+	2B4E
q	7110	k	6B25	CapsLock	3A	1/End	314F
w	7711	l	6C26	F1	003B	2/↓	3250
e	6512	;	3B27	F2	003C	3/PgDn	3351
r	7213	‘	2728	F3	003D	0/Ins	3052
t	7414	`	6029	F4	003E	Del	2E53
y	7915	Shift(L)	2A	F5	003F		

16行×8列的键盘扫描电路



主机的键盘接口电路


Intel 8042
8位 CPU
命令寄存器
64H
状态寄存器
64H
输入缓冲寄存器
60H
输出缓冲寄存器
60H
D7 ~D0
输入端口 P1
P14
P15
P16
P17
系统板 RAM SEL
跨接器开关
显示器类型开关
键盘锁定开关
输出端口 P2
P20
P21
P24
P25
P26
P27
系统复位
A20选通
输出缓冲器满IRQ1
输入缓冲器空
测试
TEST0
TEST1
控制部件
+5V
VCC
VDD
VSS
地
RD
WR
A0
IOR
IOW
A2
8042 CS
CS
128B RAM
2KB ROM
键盘扫描码
键盘时钟

增强型扩展键盘接口逻辑示意图

数据格式变换： 键盘扫描码→ 系统扫描码→键盘缓冲区格式

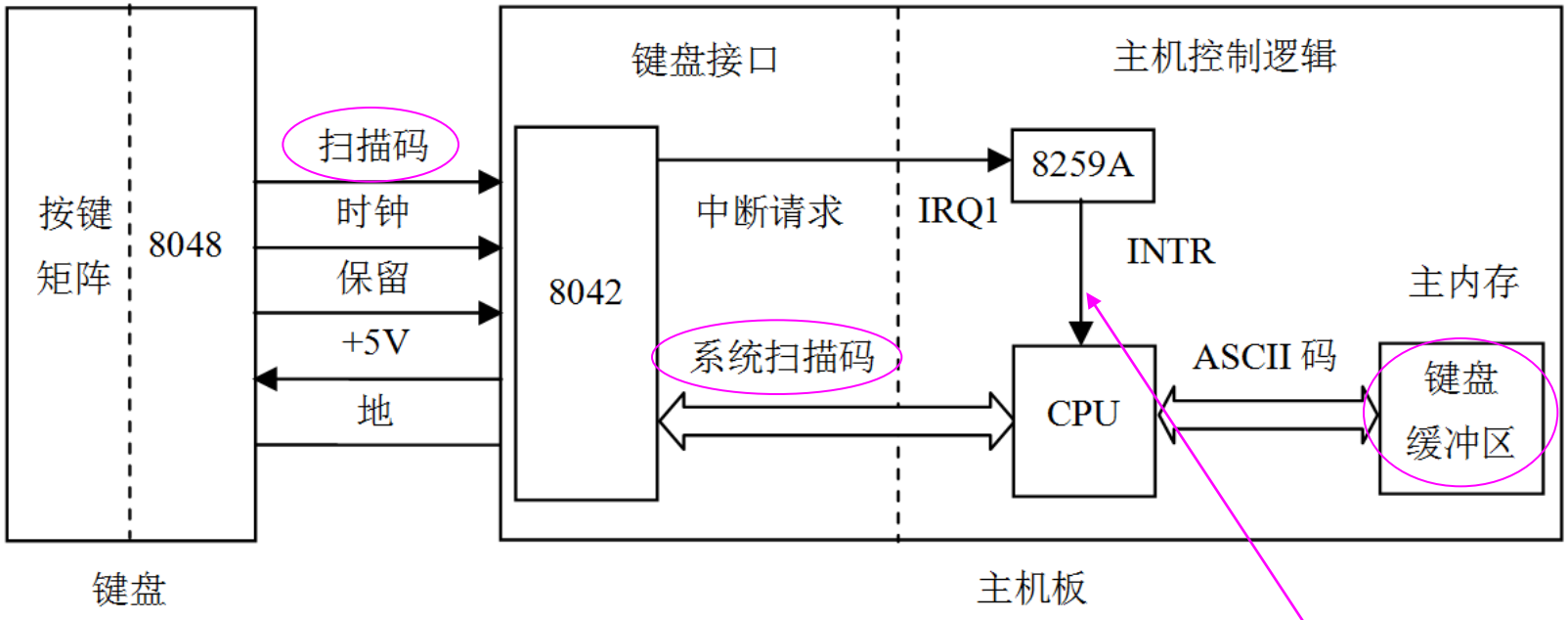


图 8.1.7 增强型扩展键盘接口示意图

响应INT 9中断

8.1.6 BIOS键盘中断及DOS键盘功能调用

- BIOS键盘中断及DOS键盘功能调用有三种方式：
- 中断类型码09H
- 中断类型码16H
- 中断类型码21H

1．中断09H的处理过程

- （1）从键盘接口的输出缓冲寄存器（60H）读取系统扫描码。
- （2）判断该键的分类，并处理：

高位字节	低位字节
系统扫描码	ASCII码
0	ASCII码
扩展码	0

键盘缓冲区

1）ASCII码0～127 的处理方法：

2）ASCII码128～255的处理方法：

3）组合键和功能键的处理方法：

命令键、组合功能键等的编码，称为扩展码，如SHIFT+a。

4）特殊命令组合键的处理方法：不形成代码，直接完成相应操作，如Ctrl+Alt+Del。

5）对特殊键的处理方法：设置“键盘状态字节”的状态。

INT 9H： 向键盘缓冲区写入数据

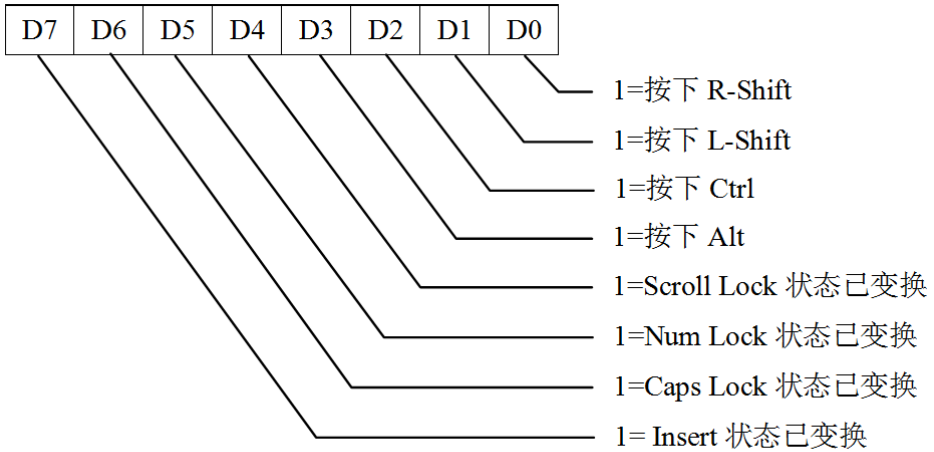
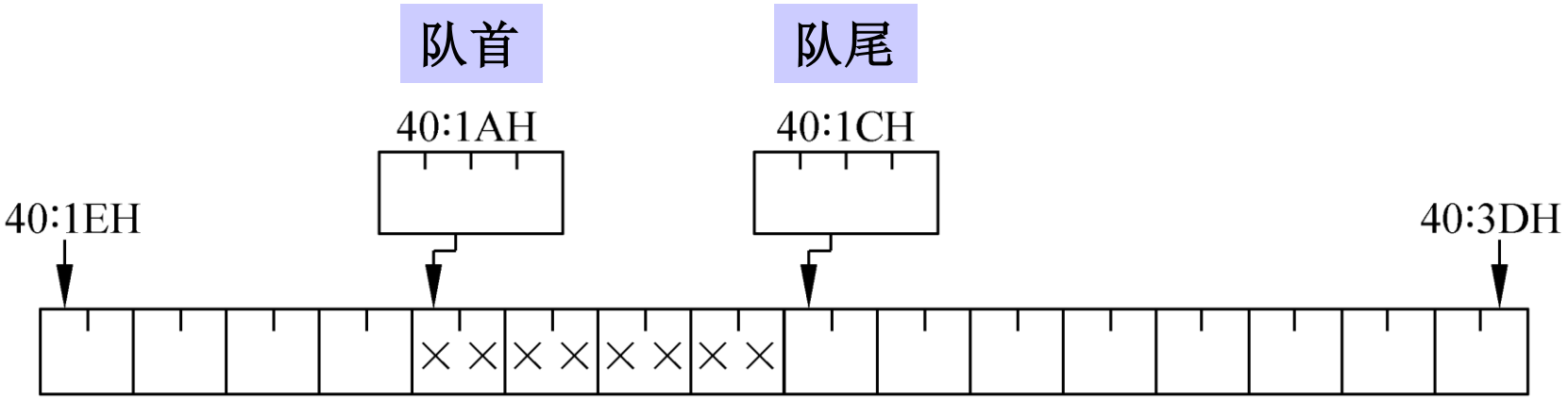


图 8.1.9　键盘状态字节

键盘缓冲区

- 键盘缓冲区：建立在系统主存的BIOS数据区中。
- 占用32个字节，可存放16次击键产生的ASCII码和扫描码。
- 以先进先出的方式工作，输入的键盘代码在其中形成循环队列。
- 中断09H输入的地址指针总指向队尾，从那里写入数据。



2．中断16H的功能

- INT 16H：读取键盘缓冲区的数据。
- 以先进先出的方式工作，INT 16H的输出指针总指向队首。
- INT 16H有3种子功能，由AH=（0、1、2）识别。
- （1）AH=0
- 功能：从键盘读入字符送AL寄存器，当无键按下时，处于等待状态。
- 入口参数：AH=0
- 出口参数：AL中为键盘输入的字符的ASCII码值，AH中为扫描码。
- （2）AH=1
- 功能：从键盘缓冲器中读入字符送给AL，并设置ZF标志，若按过任一键（即键盘缓冲区不空），置ZF=0，否则ZF=1。
- 入口参数：AH=1
- 出口参数：若ZF=0，则AL中为输入的字符的ASCII码。
- 由于该功能是从键盘缓冲区读数，当无键按下时，不等待，常通过检测ZF标志来控制某一程序的执行。
- （3）AH=2
- 功能：读取特殊功能键的状态。
- 入口参数：AH=2
- 出口参数：AL为各特殊功能键的状态。

3．中断21H的功能

- 在DOS功能调用中，也有多个功能调用号用于获得所需要的键盘信息。常用的键盘操作功能如下：
- （1）AH=1
- 功能：从键盘输入一个字符并回显在屏幕上。
- 入口参数：AH=1
- 出口参数：AL=字符
- （2）AH=6
- 功能：读键盘字符（直接控制台I/O）。
- 入口参数：AH=6，DL=0FFH（表示输入）
- 出口参数：若有字符可取，AL=字符，ZF=0。若无字符可取，AL=0，ZF=1
- （3）AH=7
- 功能：从键盘输入一个字符，不回显。
- 入口参数：AH=7
- 出口参数：AL=字符

- （4）AH=8
- 功能：从键盘输入一个字符，不回显。检测Ctrl_Break。
- 入口参数：AH=8
- 出口参数：AL=字符
- （5）AH=0AH
- 功能：输入字符到缓冲区。
- 入口参数：AH=0AH，DS:DX=缓冲区首址
- 出口参数：无
- （6）AH=0BH
- 功能：读键盘状态。
- 入口参数：AH=0BH
- 出口参数：AL=0FFH，有键输入。AL=0，无键输入
- （7）AH=0CH
- 功能：清除键盘缓冲区，并调用一种键盘功能。
- 入口参数：AH=0CH，AL=键盘功能号（1、6、7、8、A）
- 出口参数：与调用的功能有关

8.2 LED显示器接口

- 8.2.1 LED七段显示器结构

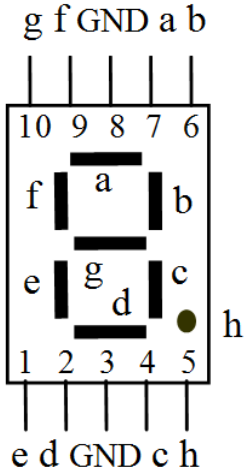
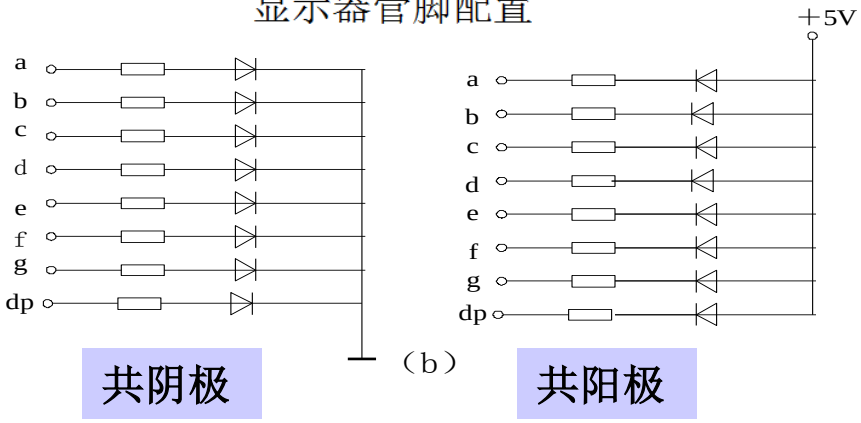


图 8.2.1 LED 七段显示器管脚配置



共阴极段码表　表8.2.1 共阴极LED七段显示器的段选码

字 形	h D7	g D6	f D5	e D4	d D3	c D2	b D1	a D0	段码（H）
0	0	0	1	1	1	1	1	1	3F
1	0	0	0	0	0	1	1	0	06
2	0	1	0	1	1	0	1	1	5B
3	0	1	0	0	1	1	1	1	4F
4	0	1	1	0	0	1	1	0	66
5	0	1	1	0	1	1	0	1	6D
6	0	1	1	1	1	1	0	1	7D
7	0	0	0	0	0	1	1	1	07
8	0	1	1	1	1	1	1	1	7F
9	0	1	1	0	1	1	1	1	6F
A	0	1	1	1	0	1	1	1	77
B	0	1	1	1	1	1	0	0	7C
C	0	0	1	1	1	0	0	1	39
D	0	1	0	1	1	1	1	0	5E
E	0	1	1	1	1	0	0	1	79
F	0	1	1	1	0	0	0	1	71
P	0	1	1	1	0	0	1	1	73
不显示	0	0	0	0	0	0	0	0	00

8.2.2 LED显示器组成与显示方式

- LED显示器显示方式：静态显示、动态显示。

- 静态显示：当显示器显示某一个字符时，相应的发光二极管恒定地导通或截止。LED显示器在静态显示方式下，各显示位的位选线即共阴极点（或共阳极点）连接在一起接地（或接＋5V）；各显示位的段选线（a～h）与一个8位并行口相连。

- 静态显示方式电路每一显示位可独立显示，在同一时刻不同的显示位可以显示不同的字符。

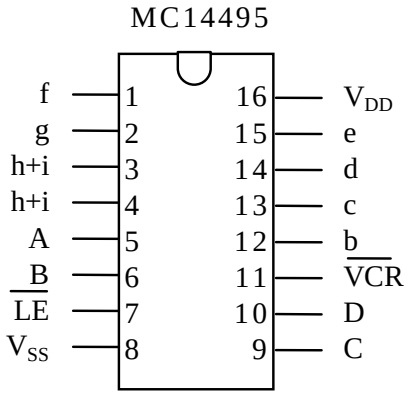
- 动态显示：一位一位地轮流点亮各位显示器（扫描）。对于某一位显示器来说，每隔一段时间点亮一次。

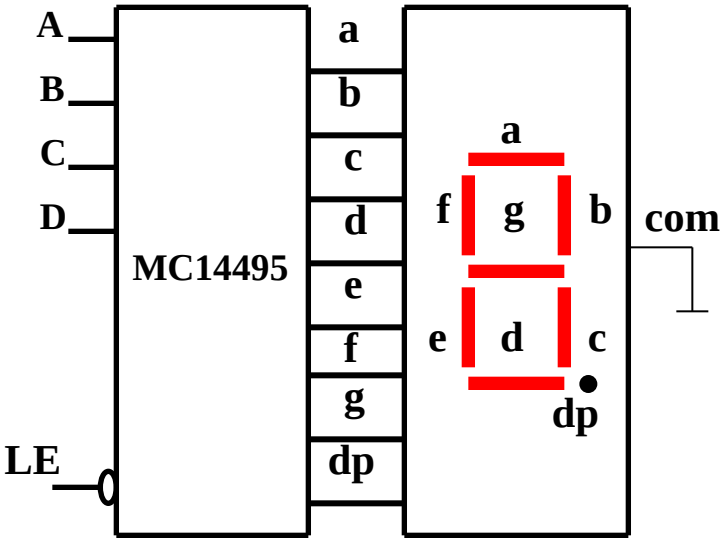
8.2.3 LED显示器接口及应用举例

- 从LED显示器显示原理可知，为显示字母与数字，必须最终转换成相应的段选码。转换方法：硬件译码，软件译码。
- 1．硬件译码显示器接口

MC14495是Motorola公司生产的CMOS BCD-七段十六进制锁存、译码、驱动芯片。MC14495本身不能完成显示小数点的功能。

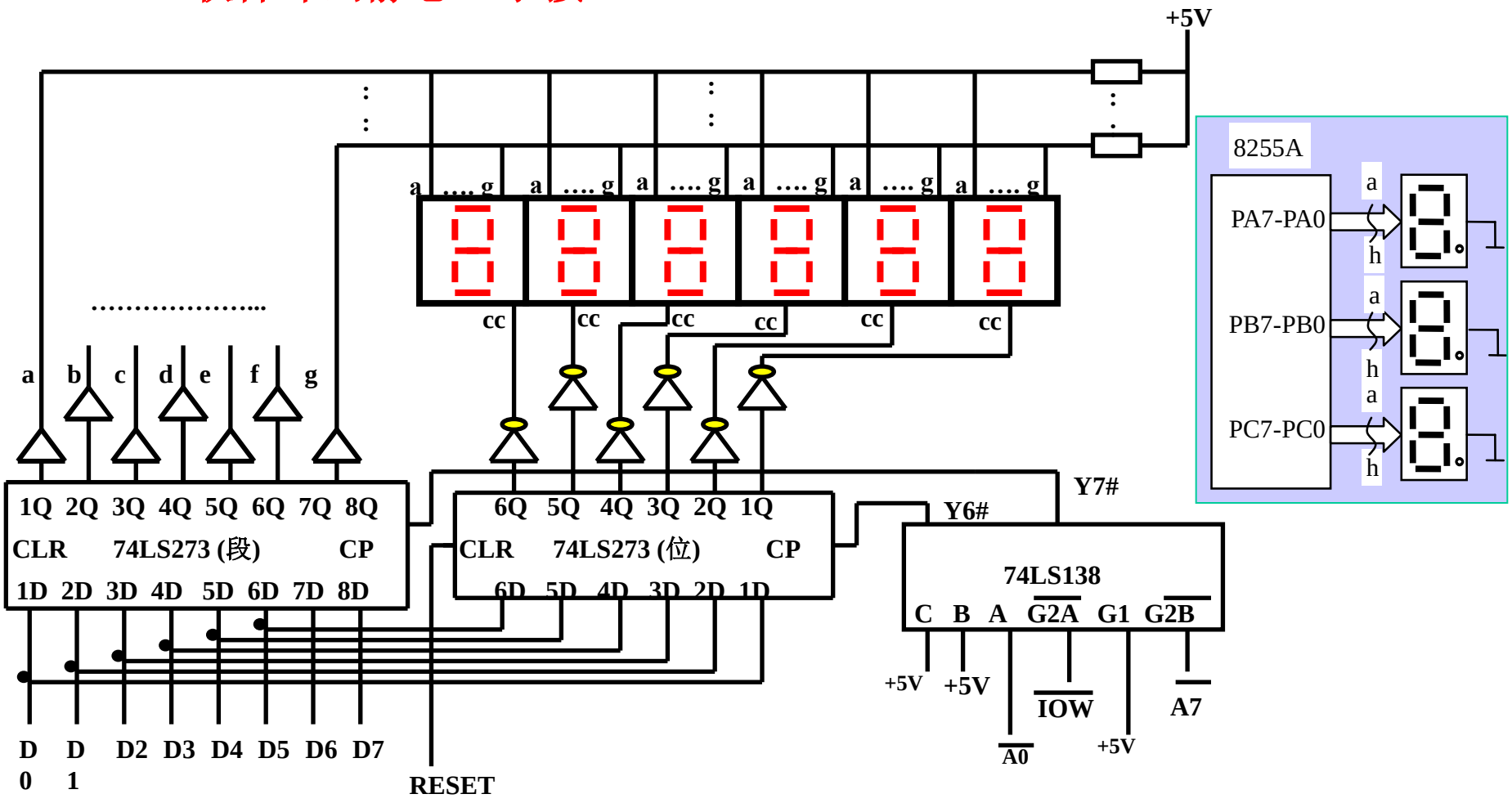
输入 D C B A	输出 h+i	g	f	e	d	c	b	a	显示字形
0 0 0 0	0	0	1	1	1	1	1	1	0
0 0 0 1	0	0	0	0	0	1	1	0	1
0 0 1 0	0	1	0	1	1	0	1	1	2
0 0 1 1	0	1	0	0	1	1	1	1	3
0 1 0 0	0	1	1	0	0	1	1	0	4
0 1 0 1	0	1	1	0	1	1	0	1	5
0 1 1 0	0	1	1	1	1	1	0	1	6
0 1 1 1	0	0	0	0	0	1	1	1	7
1 0 0 0	0	1	1	1	1	1	1	1	8
1 0 0 1	0	1	1	0	1	1	1	1	9
1 0 1 0	1	1	1	1	0	1	1	1	A
1 0 1 1	1	1	1	1	1	1	0	0	B
1 1 0 0	1	0	1	1	1	0	0	1	C
1 1 0 1	1	1	0	1	1	1	1	0	D
1 1 1 0	1	1	1	1	1	0	0	1	E
1 1 1 1	1	1	1	1	0	0	0	1	F



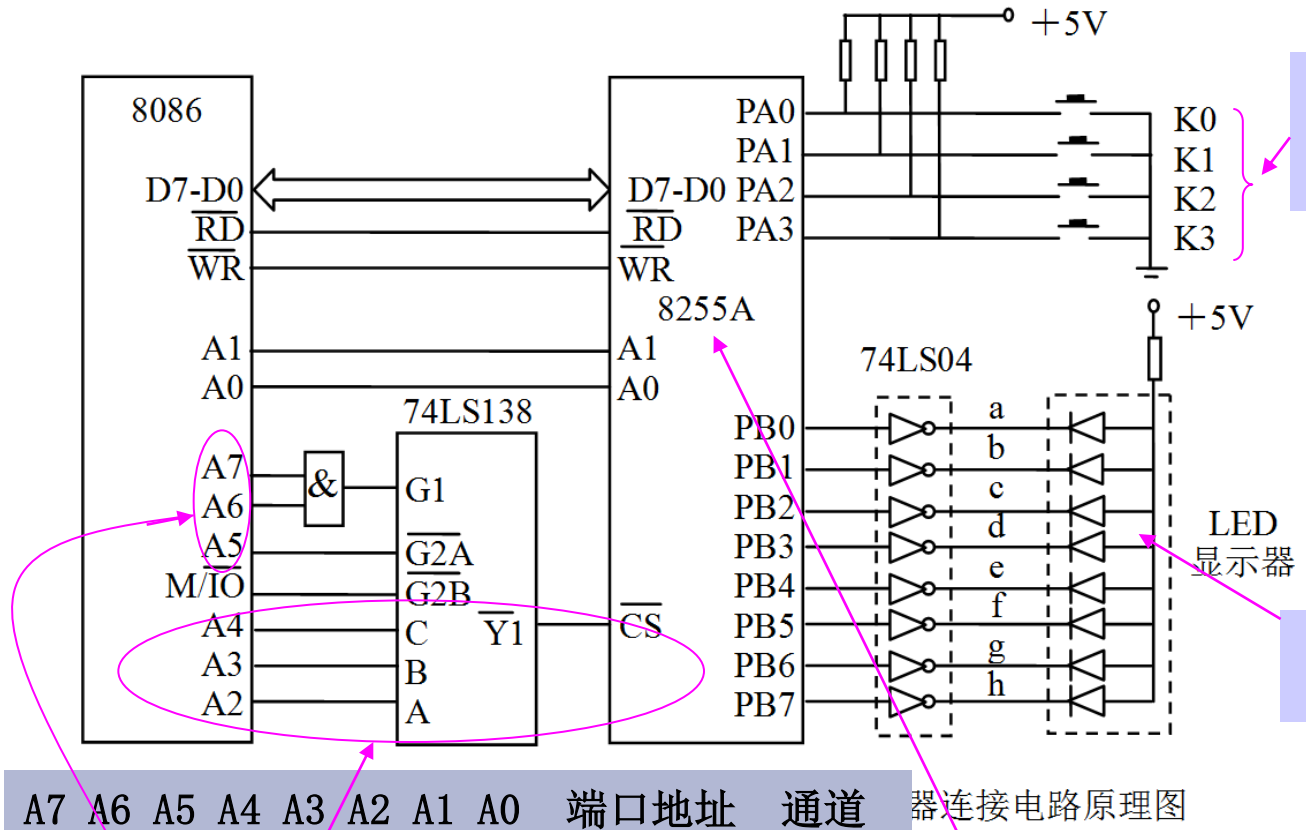


2. 软件译码显示器接口

- 软件译码按静态显示、动态显示两种方式考虑接口设计。
- （1）软件译码静态显示接口



例8.2.1 8255A与按键开关、LED七段显示器等外部设备相连接



器连接电路原理图

4个按键开关K3～K0组成了4位二进制数值，并对应1位十六进制数。

共阳极显示器静态工作方式

A7	A6	A5	A4	A3	A2	A1	A0	端口地址	通道
1	1	0	0	0	1	0	0	C4H	A口
1	1	0	0	0	1	0	1	C5H	B口
1	1	0	0	0	1	1	0	C6H	C口
1	1	0	0	0	1	1	1	C7H	控制口

8255A工作在方式0，端口A输入，端口B输出，控制字为10010000B

程序段

```
START:  MOV   AL, 90H   ；设置方式控制字，口A输入，口B输出
        OUT   0C7H, AL
X1:     IN    AL, 0C4H；输入按键状态
        AND   AL, 0FH   ；屏蔽掉不用的高4位
        MOV   BX, OFFSET LEDTAB；设置段码表指针
        XLAT            ；读取段码
        OUT   0C5H, AL；输出段码到端口B
        MOV   AX, 200H；延时
X2:     DEC   AX
        JNZ   X2
        JMP   X1
        HLT
```

段码表

```
LEDTAB  DB    3FH    ；0的段码
        DB    06H    ；1的段码
        DB    5BH    ；2的段码
        DB    4FH    ；3的段码
        DB    66H    ；4的段码
        DB    6DH    ；5的段码
        DB    7DH    ；6的段码
        DB    07H    ；7的段码
        DB    7FH    ；8的段码
        DB    6FH    ；9的段码
        DB    77H    ；A的段码
        DB    7CH    ；B的段码
        DB    39H    ；C的段码
        DB    5EH    ；D的段码
        DB    79H    ；E的段码
        DB    71H    ；F的段码
```

（2）软件译码动态显示接口

- 动态显示程序设计中显示程序的要点：

1）解决显示译码问题，因为要显示的数字与其对应的段选码并没有有机的联系和转换规律，所以要用查表的方法完成这种译码功能。

2）在进入显示程序之前，为保持显示的数据，专门开辟几个单元作为显示缓冲区，用以存放要显示的数字（十六进制数）。

- 采用软件译码方法一般有两种表格设置方案：

1）顺序表格排列法，即按一定的顺序排列显示段码。通常显示的字形数据就是该段码在段码表中相对表头的偏移量。

2）数据结构法，即按字形和段码的关系，自行设计一组数据结构。该方法设计灵活，但程序运行速度较慢。

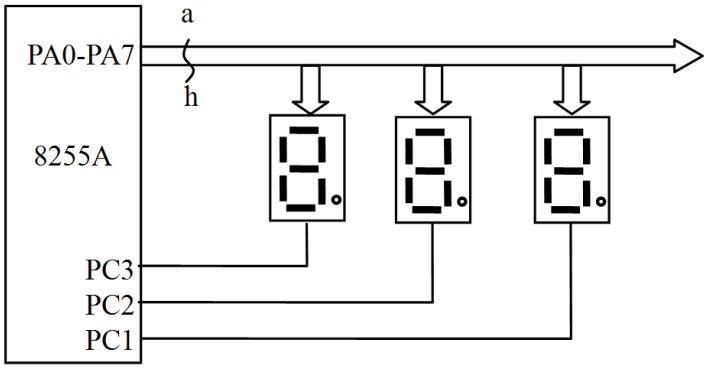


图 8.2.5　通过 8255A 控制的 3 位动态 LED 显示器接口原理图

例8.2.2 LED七段显示器及其接口

设定8255A端口A、端口B、端口C、控制端口的地址分别为60H、61H、62H和63H。

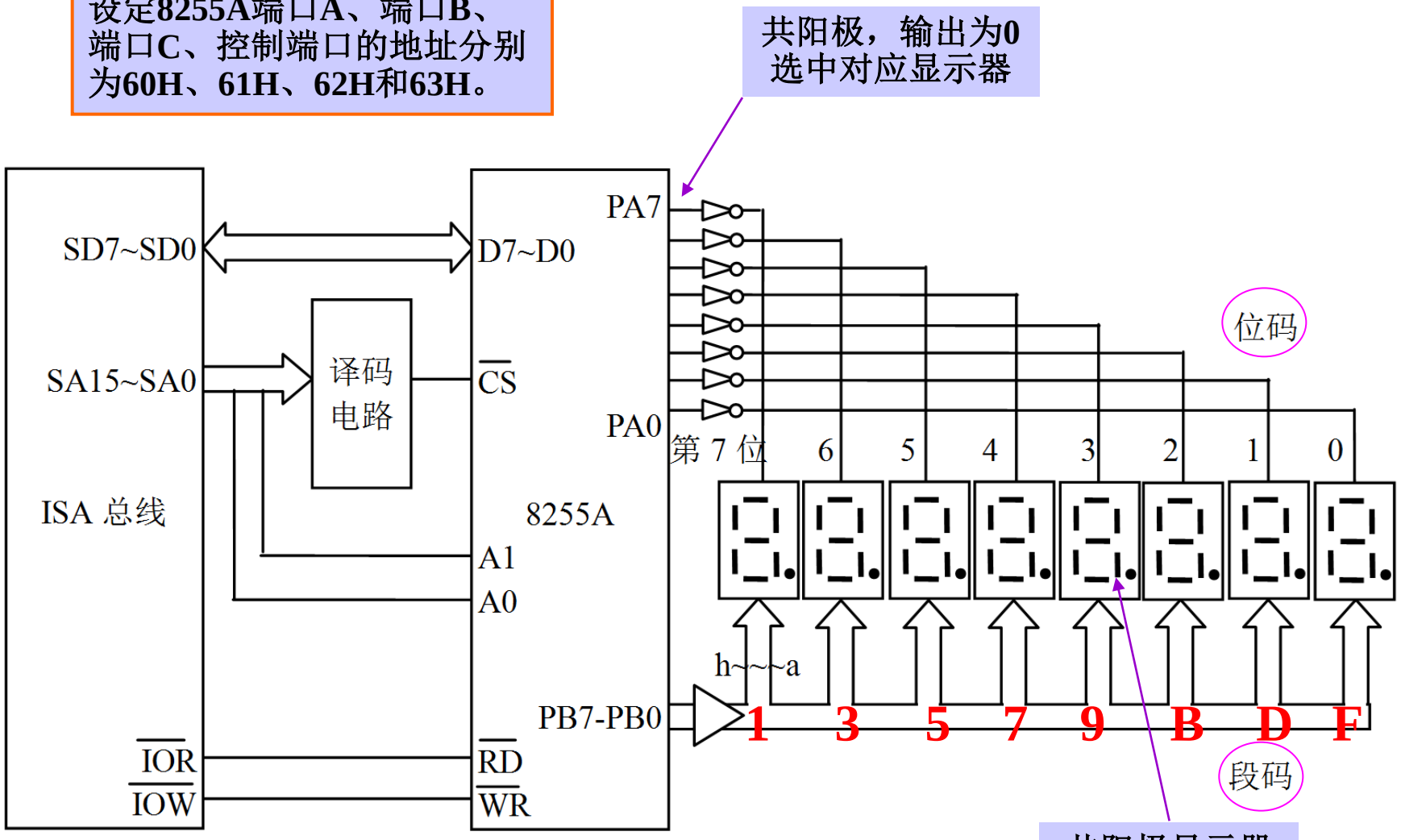


图 8.2.6　8 位 7 段 LED 显示器接口电路

程序设计

- 下面是8个显示器重复显示（50次）8位十六进制数13579BDF的源程序。

```
DATA    SEGMENT
TABLE   DB      0C0H    ；0的段码，开始设置段码表
        DB      0F9H    ；1的段码
        DB      0A4H    ；2的段码
        DB      0B0H    ；3的段码
        DB      99H     ；4的段码
        DB      92H     ；5的段码
        DB      82H     ；6的段码
        DB      0F8H    ；7的段码
        DB      80H     ；8的段码
        DB      98H     ；9的段码
        DB      88H     ；A的段码
        DB      83H     ；B的段码
        DB      0C6H    ；C的段码
        DB      0A1H    ；D的段码
        DB      86H     ；E的段码
        DB      8EH     ；F的段码
DATA    ENDS
```

段码表

程序设计

```
CODE    SEGMENT
        ASSUME  CS:CODE,DS:DATA
START:  MOV     AX, DATA
        MOV     DS, AX
        MOV     AL, 80H
        MOV     63H, AL         ；送各数据端口方式0的输出控制字
        MOV     DL, 50          ；设置重复次数，显示50次
        LEA     SI, TABLE       ；取段码表首址
        MOV     BX, 1           ；欲显示的字形设置为数字“1”，是最左位显示的数
        MOV     AH, 7FH         ；显示位7的位选码，指向最左位（第7显示位）
```

8255初始化

循环参数设置

程序设计

```
X1:     MOV     AL, [BX+SI]     ；取数的段码，首次取1
        OUT     61H, AL         ；送段选码，B端口
        MOV     AL, AH
        OUT     60H, AL         ；送位选码，A端口
        ROR     AH, 1           ；形成下一个位选码
        ADD     BX, 2           ；形成下一个要显示的数（奇数）
        AND     BX, 0FH
        MOV     CX, 30H         ；延迟一定的时间，在实际中应调整该参数
X2:     LOOP    X2
        CMP     AH, 7FH
        JNZ     X1              ；判第7～0显示位是否结束
        DEC     DL
        JNZ     X1              ；判重复显示50次是否结束
        MOV     AH, 4CH
        INT     21H
CODE    ENDS
        END     START
```

取段码，并送段码

取位码，并送位码

修改循环参数

延时

循环结束条件

第8章 结 束