

## I/O接口扩展概述

### MCS-51单片机外部扩展I/O接口的主要原因:

- 51单片机在外扩存储器后，单片机自身实际可用的I/O端口非常有限；
- 外部设备和单片机之间数据交换速度匹配的要求。

### 外部扩展I/O端口的编址与读写方式

- 统一编址方式: 与外部扩展数据存储器同等对待，地址空间统一分配，每一个扩展的I/O端口相当于一个RAM存储单元。
- 扩展I/O端口的读写方式与外部扩展数据存储器的访存方式相同，采用同样的MOVX指令操作。

## I/O接口扩展概述

### 接口和端口

接口一般指计算机系统与外部设备之间的信息传输部件，包括硬件芯片、驱动软件和协议规约。端口通常是指信息传输的物理通道，在硬件中一般指具有地址的寄存器或者缓冲器。

### I/O数据传输方式

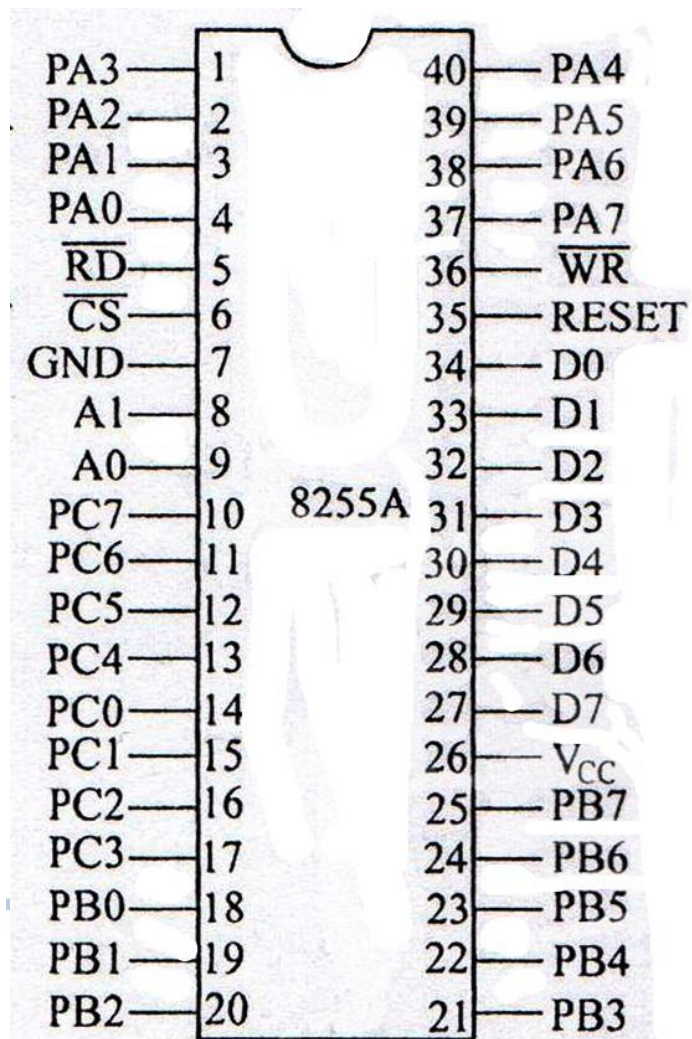
- 同步传输方式：与单片机与外设之间以完全相同的时钟节拍进行数据传输，如外部数据存储器的访存。
- 查询传输方式：也称异步传输方式，双方通过程序查询“握手”信号协调传输过程。这种方式效率较低。
- 中断传输方式：当外设准备好进行数据交换时，发出中断请求信号，单片机通过中断服务完成数据传输。

### MCS-51与并行I/O芯片8255A的接口设计

#### 芯片介绍

8255A是Intel公司生产的可编程并行I/O芯片，具有3个8位的并行I/O口，三种工作方式，可通过编程设置其功能。

- **D7~D0:** 与三态双向数据线;
- **$\overline{CS}$ :** 片选信号，低电平有效;
- **$\overline{RD}$ :** 读出信号，低电平有效;
- **$\overline{WR}$ :** 写入信号，低电平有效;
- **PA7~PA0:** A口输入/输出线;
- **PB7~PB0:** B口输入/输出线;
- **PC7~PC0:** C口输入/输出线;
- **A1~A0:** 地址线，用于选择内部的4个端口。

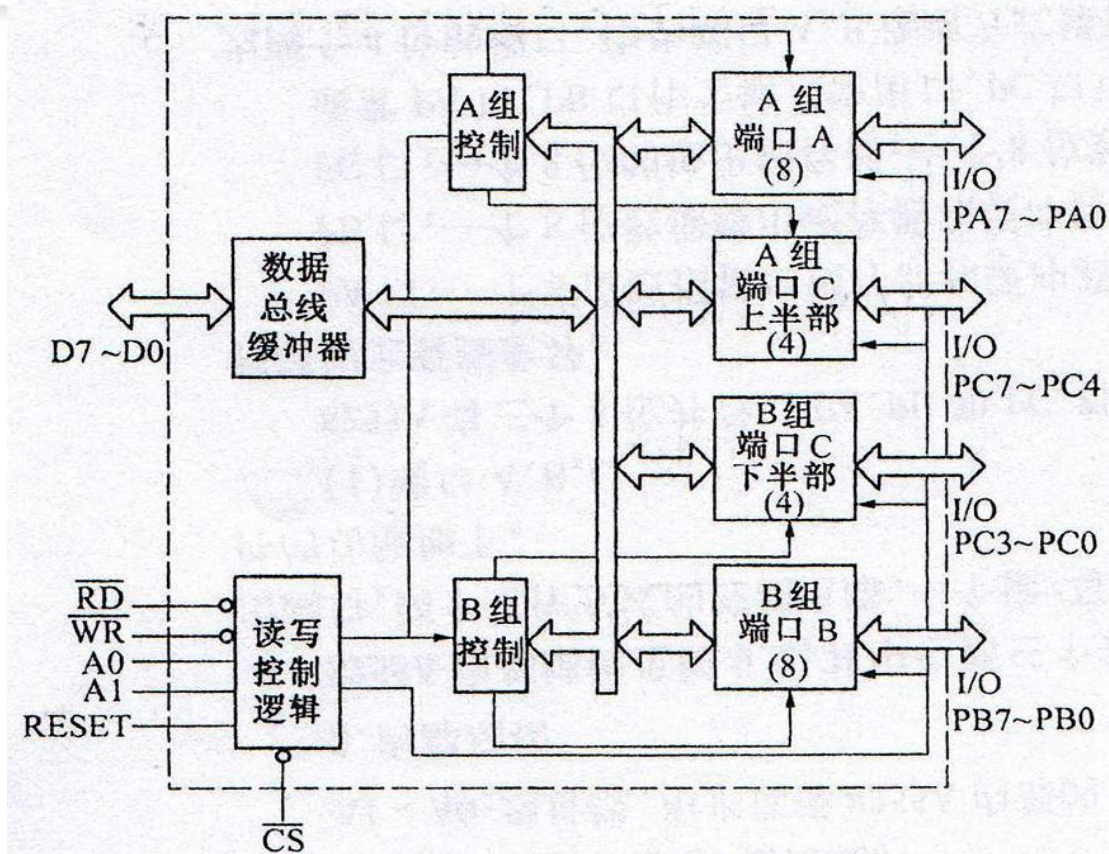


## MCS-51与并行I/O芯片8255A的接口设计

### 芯片介绍

8255A包括三个并行数据输入/输出端口，两个工作方式的控制电路，一个读/写控制逻辑电路和8位数据缓冲器。

- **PA口**：一个8位数据输出锁存器和缓冲器；一个8位数据输入锁存器。
- **PB口**：一个8位数据输出锁存器和缓冲器；一个8位数据输入缓冲器。
- **PC口**：一个8位数据输出锁存器；一个8位数据输入缓冲器。



通常PA口、PB口作为输入输出，PC口可作为输入输出口，也可分为两个4位的端口，作为PA口和PB口选通方式操作时的状态控制信号。



## MCS-51与并行I/O芯片8255A的接口设计

### 芯片介绍

各端口的工作状态与控制信号的关系如下表：

A1	A0	$\overline{\text{RD}}$	$\overline{\text{WR}}$	$\overline{\text{CS}}$	工 作 状 态
0	0	0	1	0	A 口数据→数据总线(读端口 A)
0	1	0	1	0	B 口数据→数据总线(读端口 B)
1	0	0	1	0	C 口数据→数据总线(读端口 C)
0	0	1	0	0	总线数据→A 口(写端口 A)
0	1	1	0	0	总线数据→B 口(写端口 B)
1	0	1	0	0	总线数据→C 口(写端口 C)
1	1	1	0	0	总线数据→控制字寄存器(写控制字)
x	x	x	x	1	数据总线为三态
1	1	0	1	0	非法状态
x	x	1	1	0	数据总线为三态

### MCS-51与并行I/O芯片8255A的接口设计

#### 工作方式选择控制字及PC口位操作控制字

8255A有三种工作方式：方式0：基本输入输出；

方式1：选通输入输出；

方式2：双向传送（仅PA口有此工作方式）

控制字寄存器的内容由两种格式：

工作方式选择控制字；

PC口置位/复位控制字。

# 第9部分 MCS-51扩展I/O接口的设计

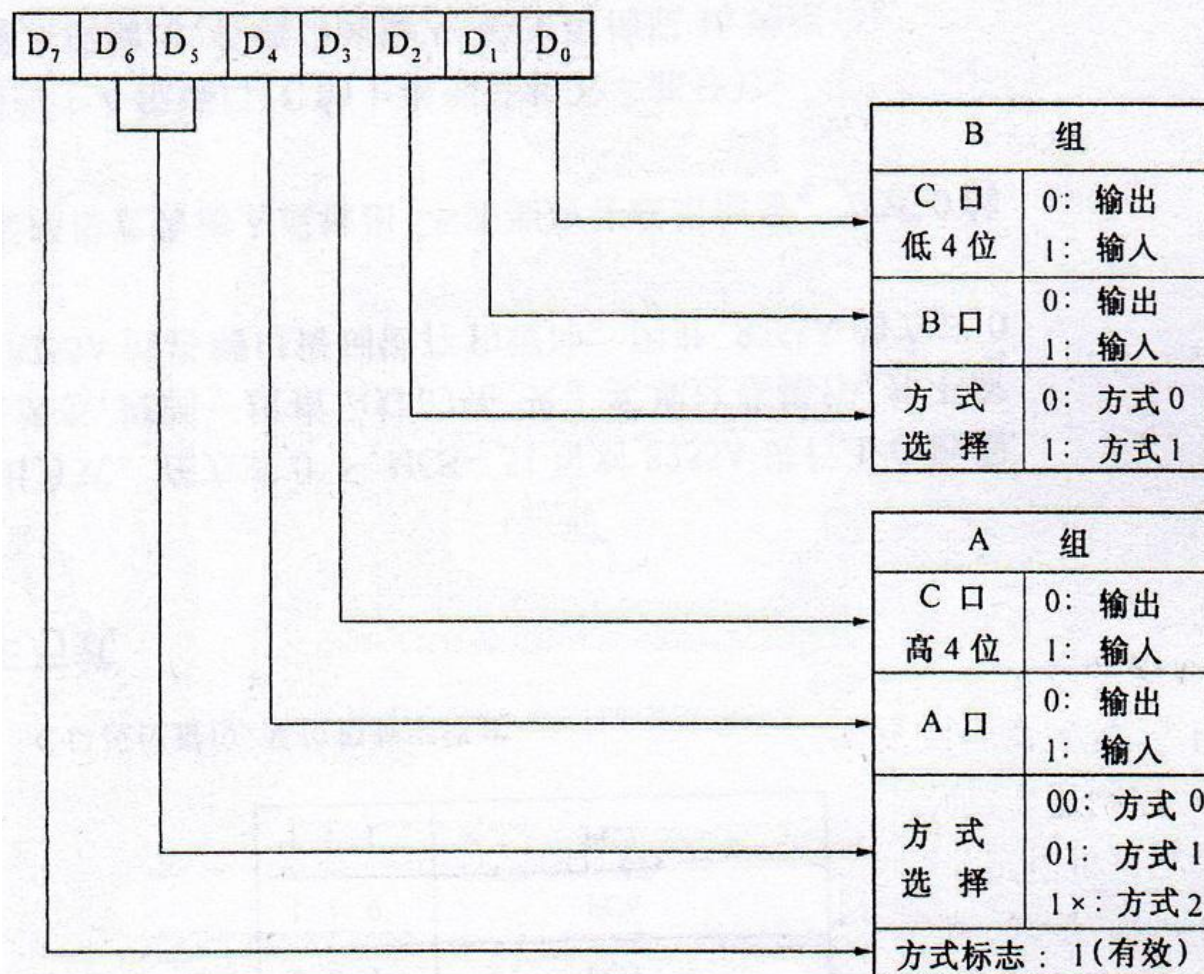
## MCS-51与并行I/O芯片8255A的接口设计

### 工作方式选择控制字

用于决定三种工作方式。

其中PC端口被分为两个部分，上半部分随PA口称为A组，下半部分随PB口称为B组。

- PA口可工作于方式0、方式1和方式2；
- PB口只能工作于方式0和方式1；
- PC口参与方式0、方式1和方式2的工作过程。





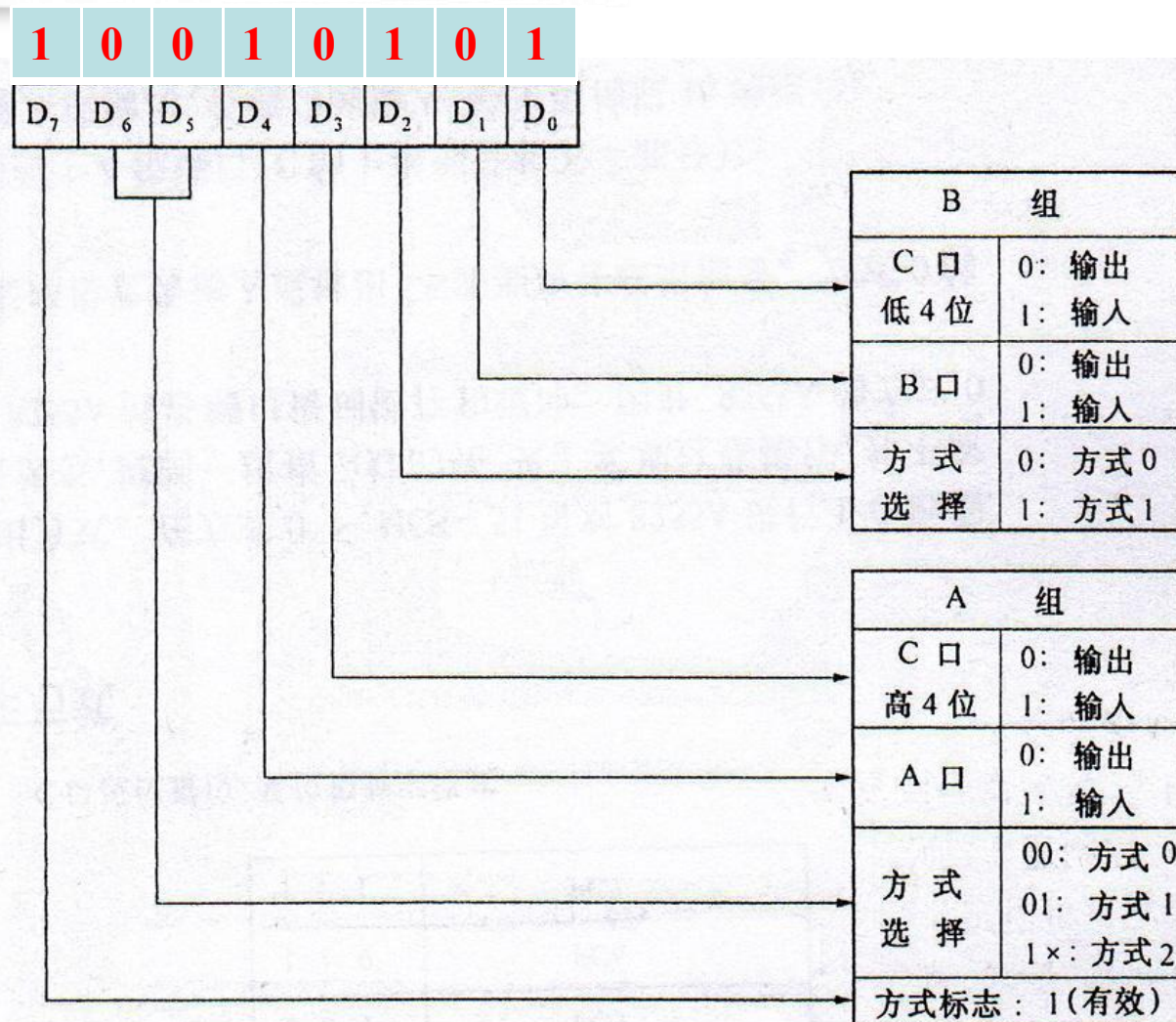
# 第9部分 MCS-51扩展I/O接口的设计

## MCS-51与并行I/O芯片8255A的接口设计

### 工作方式选择控制字

例如：写入工作方式控制字95H，各个端口的工作方式被设置如下：

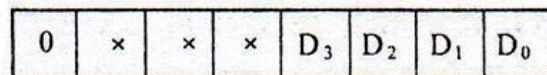
- PA口：方式0输入；
- PB口：方式1输出；
- PC口上半部分（高4位）：输出；
- PC口下半部分（低4位）：输入。





## MCS-51与并行I/O芯片8255A的接口设计

### PC口置位/复位控制字



D <sub>0</sub>	置复位控制
0	复 位
1	置 位

D <sub>3</sub> D <sub>2</sub> D <sub>1</sub>	C 口位选择
0 0 0	PC0
0 0 1	PC1
0 1 0	PC2
0 1 1	PC3
1 0 0	PC4
1 0 1	PC5
1 1 0	PC6
1 1 1	PC7

PC口中的任一位，可通过向控制字寄存器写入一个PC口置位/复位控制字，按位置1或是清0。这个功能主要用于PC口的位控。

### MCS-51与并行I/O芯片8255A的接口设计

#### 8255A的三种工作方式—方式0

方式0是一种基本输入输出方式，MCS-51可对8255A中的任一端口进行无条件的数据I/O传输，无需应答联络信号。

方式0的基本功能：

- 具有两个8位端口（PA、PB）和两个4位端口（PC的上半部分和下半部分）；
- 任一端口都可以设定为输入或输出，各端口的输入输出可构成16种组合。
- 数据输出时锁存，输入时不锁存。

例子：A组端口工作于方式0输出，B组端口工作于方式0输入。

```
MOVX DPTR, #0FF7FH
```

```
MOV A, #83H
```

```
MOVX @DPTR, A
```

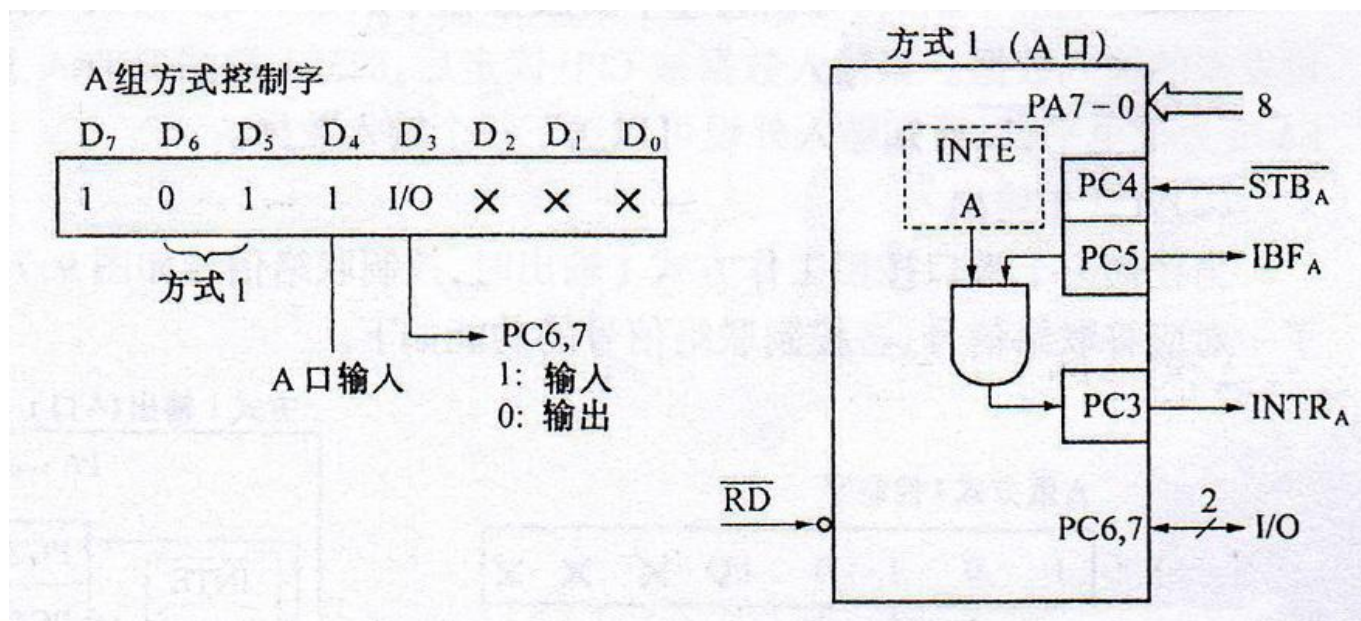
## 第9部分 MCS-51扩展I/O接口的设计

### MCS-51与并行I/O芯片8255A的接口设计

#### 8255A的三种工作方式—方式1

方式1是一种选通输入输出方式，PA口和PB口都可独立设置为这种方式。方式1时，PC口用作PA口和PB口的联络信号。

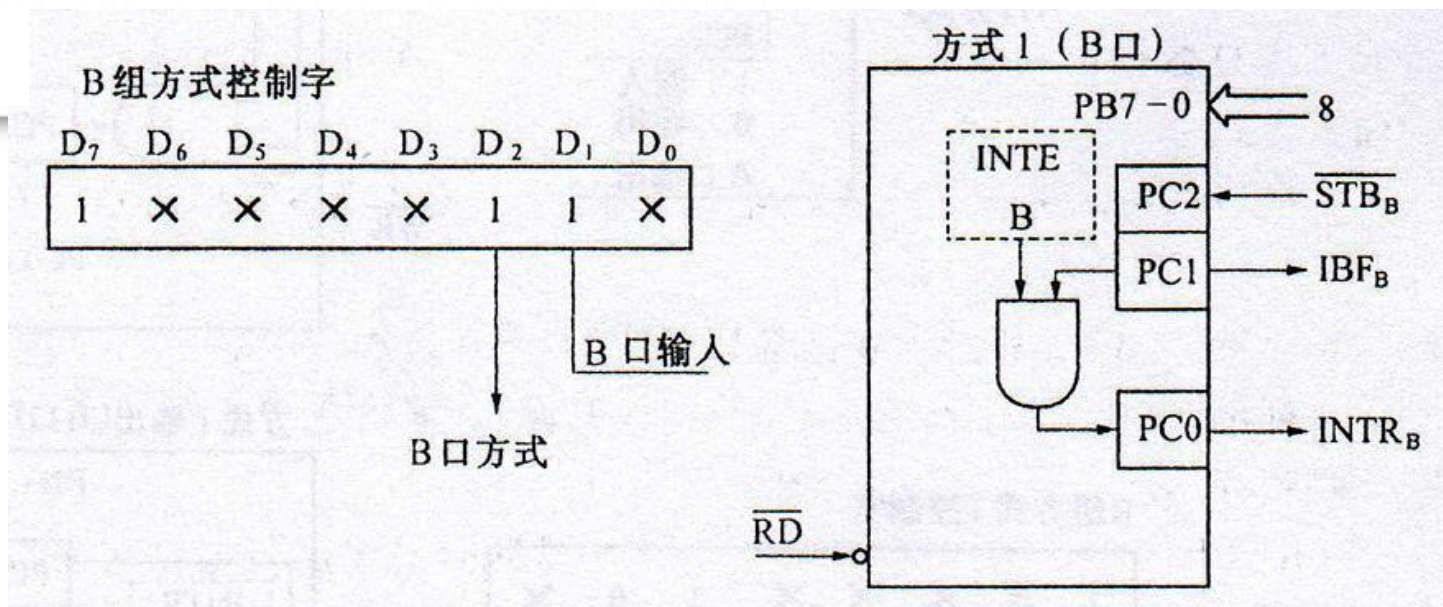
方式1输入： $\overline{\text{STB}}$ 和IBF构成了一对应答联络信号。各控制联络信号如下：





# 第9部分 MCS-51扩展I/O接口的设计

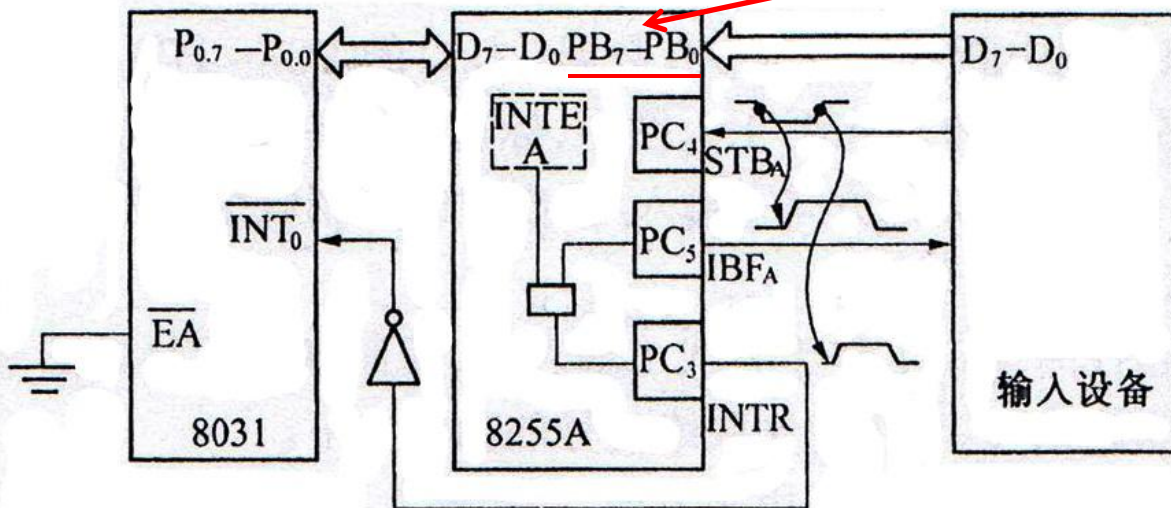
## 方式1输入



- **STB**: 选通输入，低电平有效，由外设送来的输入信号。
- **IBF**: 输入缓冲器满，高电平有效。表示数据已送入8255A的输入锁存器，由 $\overline{STB}$ 信号的下降沿置位，由 $\overline{RD}$ 信号的上升沿复位。
- **INTR**: 中断请求信号，高电平有效。由8255A输出，向CPU发出中断请求。
- **INTE**: 中断允许信号，由PC4（对PA口）或PC2（对PB口）的置位/复位来控制。

## 方式1输入

以PA口为例，方式1  
输入的工作过程如下：



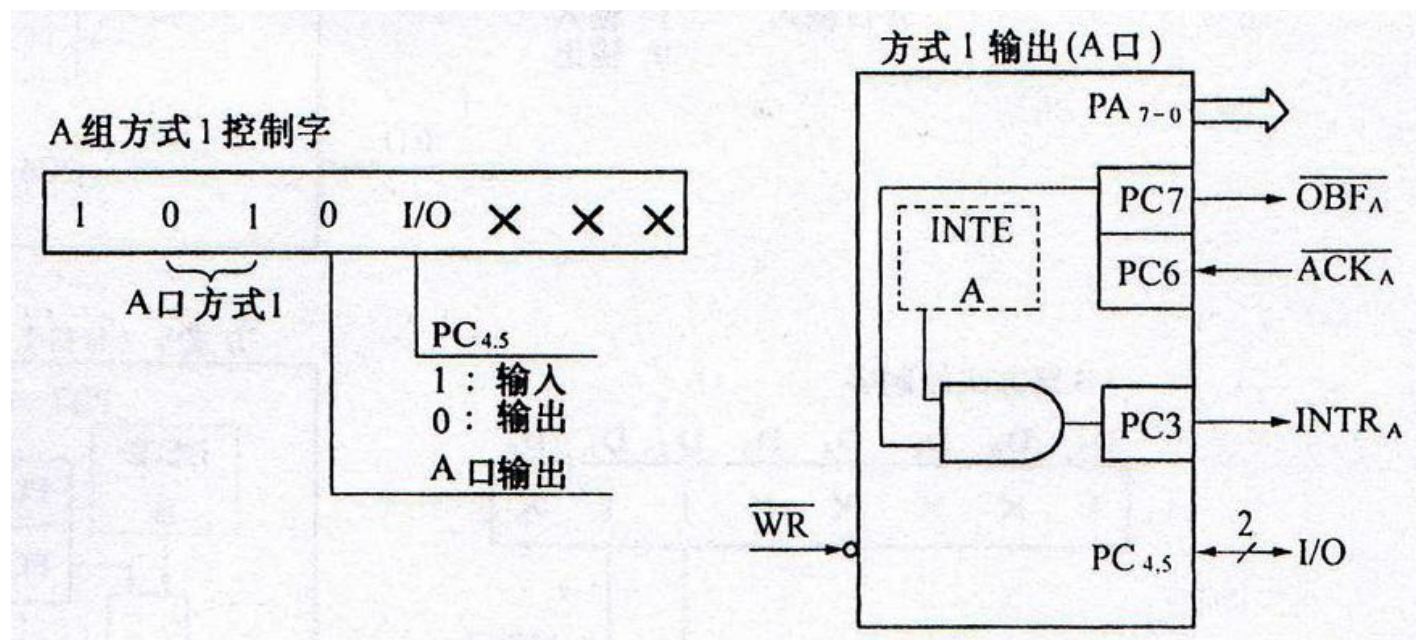
- ① 当外设输入一个数据到PA端口上时，输入设备通过 $\overline{STB}$ 向8255A发送一个低电平选通信号。
- ② 8255A收到选通信号后，首先把PA端口上的输入数据存入输入数据缓冲/锁存器，然后使IBF变为高电平，以通知输入设备已收到数据。
- ③ 8255A检测到 $\overline{STB}$ 由低电平变为高电平、IBF为“1”状态和中断允许触发器INTE为“1”时，使INTR变为高电平，向CPU发出中断请求。
- ④ CPU响应中断后，通过中断服务程序读取PA口的输入数据缓冲/锁存器中的数据。当输入数据被CPU读走后，IBF变为低电平，8255A撤销INTR上的中断请求，以通知外设可以进行下一次输入数据。

## 第9部分 MCS-51扩展I/O接口的设计

### MCS-51与并行I/O芯片8255A的接口设计

#### 8255A的三种工作方式—方式1

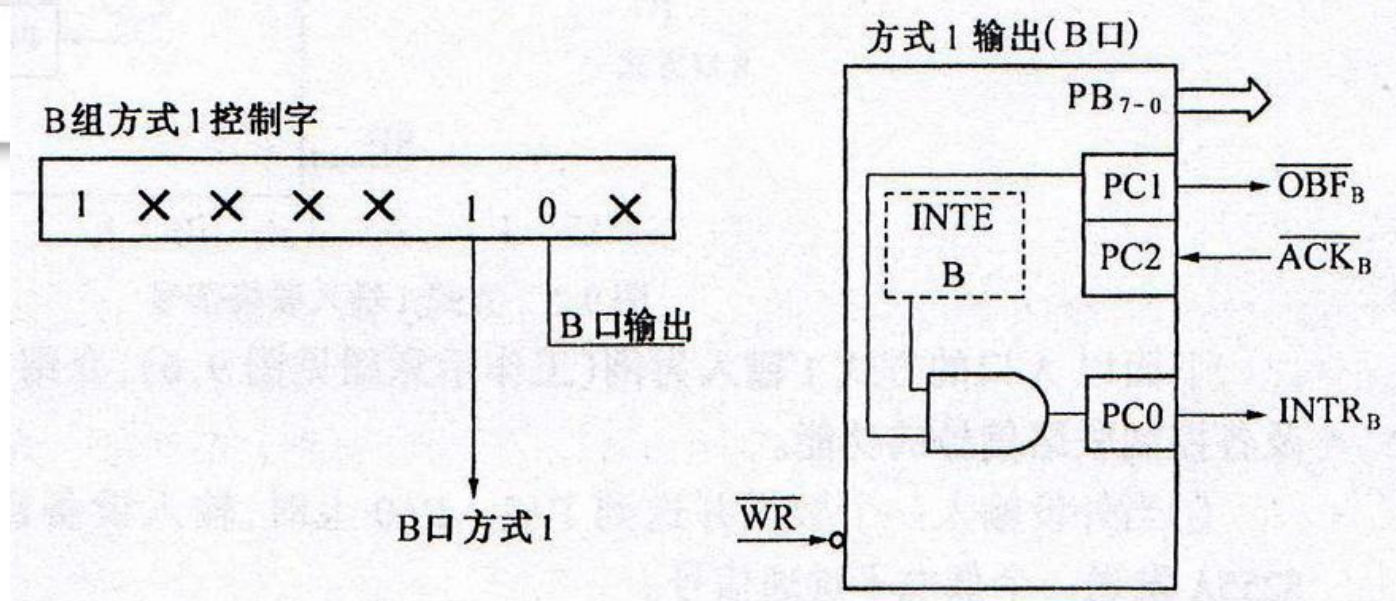
方式1输出:  $\overline{\text{OBF}}$ 和 $\overline{\text{ACK}}$ 构成了一对应答联络信号。各控制联络信号如下:





## 第9部分 MCS-51扩展I/O接口的设计

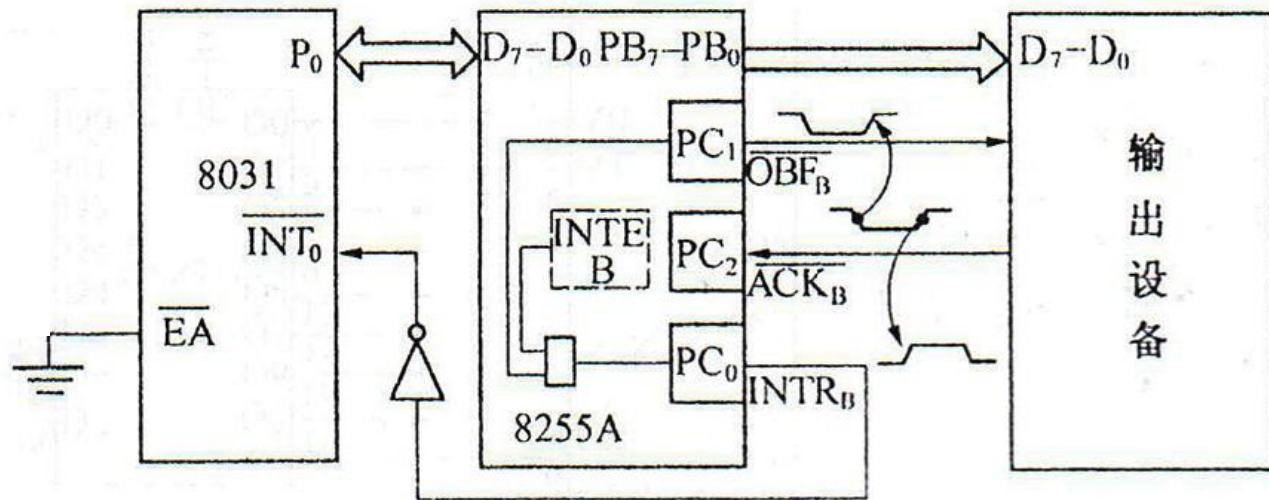
### 方式1输出



- **OBF**: 输出缓冲器满信号，低电平有效，8255A给外设的联络信号，表示CPU已经把数据输出至指定端口，外设可以将数据取走。它由 $\overline{\text{WR}}$ 信号的上升沿置“0”（有效），由 $\overline{\text{ACK}}$ 信号的下降沿置“1”（无效）。
- **ACK**: 外设的响应信号，低电平有效。指示CPU输出给8255A的数据已经被外设取走。
- **INTR**: 中断请求信号，高电平有效。表示数据已被外设取走，请求CPU继续输出下一个数据。中断请求的条件是 $\overline{\text{ACK}}$ 、 $\overline{\text{OBF}}$ 和 $\text{INTE}$ （中断允许）为高电平，中断请求信号由 $\overline{\text{WR}}$ 的下降沿复位。
- **INTE**: 中断允许信号，由PC6（对PA口）或PC2（对PB口）的置位/复位来控制。

## 方式1输出

以PB口为例，方式1  
输出的工作过程如下：



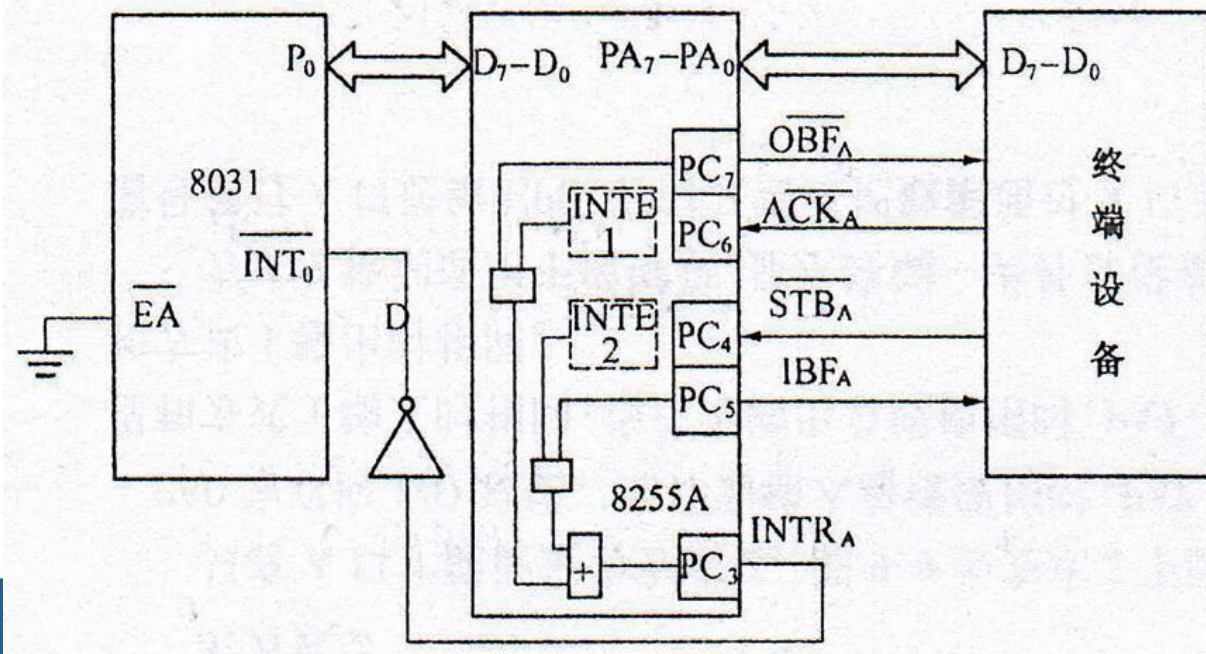
- ① CPU将输出数据送到PB端口的输出数据锁存器，8255A收到后令 $\overline{\text{OBF}}$ 变为低电平，以通知输出设备输出的数据已在PB端口的数据线上。
- ② 输出设备收到 $\overline{\text{OBF}}$ 上的低电平后，先从PB端口取走数据，然后使 $\overline{\text{ACK}}$ 变为低电平，以通知8255A输出设备已收到输出数据。
- ③ 8255A从 $\overline{\text{ACK}}$ 收到低电平后，对 $\overline{\text{OBF}}$ 、 $\overline{\text{ACK}}$ 和中断允许控制位INTE进行检测，若它们皆为高电平，则INTR变为高电平向CPU发出中断请求。
- ④ CPU响应中断后，便可通过中断服务程序把下一个输出数据送到PB口的输出数据锁存器。重复上述过程，完成数据输出。

## 第9部分 MCS-51扩展I/O接口的设计

### MCS-51与并行I/O芯片8255A的接口设计

#### 8255A的三种工作方式—方式2

- 只有PA口才能设定为方式2。
- 方式2下，PA口为双向I/O总线。当作为输入总线使用时，PA口受 $\overline{STB}$ 和 $\overline{IBF}$ 控制，其工作过程和方式1输入时相同；当作为输出总线使用时，PA口受 $\overline{OBF}$ 和 $\overline{ACK}$ 控制，其工作过程和方式1输出时相同。
- 方式2适合于8255A通过同一端口既从外设输入数据，也向外设输出数据的情况。





# 第9部分 MCS-51扩展I/O接口的设计

## MCS-51与并行I/O芯片8255A的接口设计

### MCS-51和8255A的接口

PA口地址:

低8位=7CH;

PB口地址:

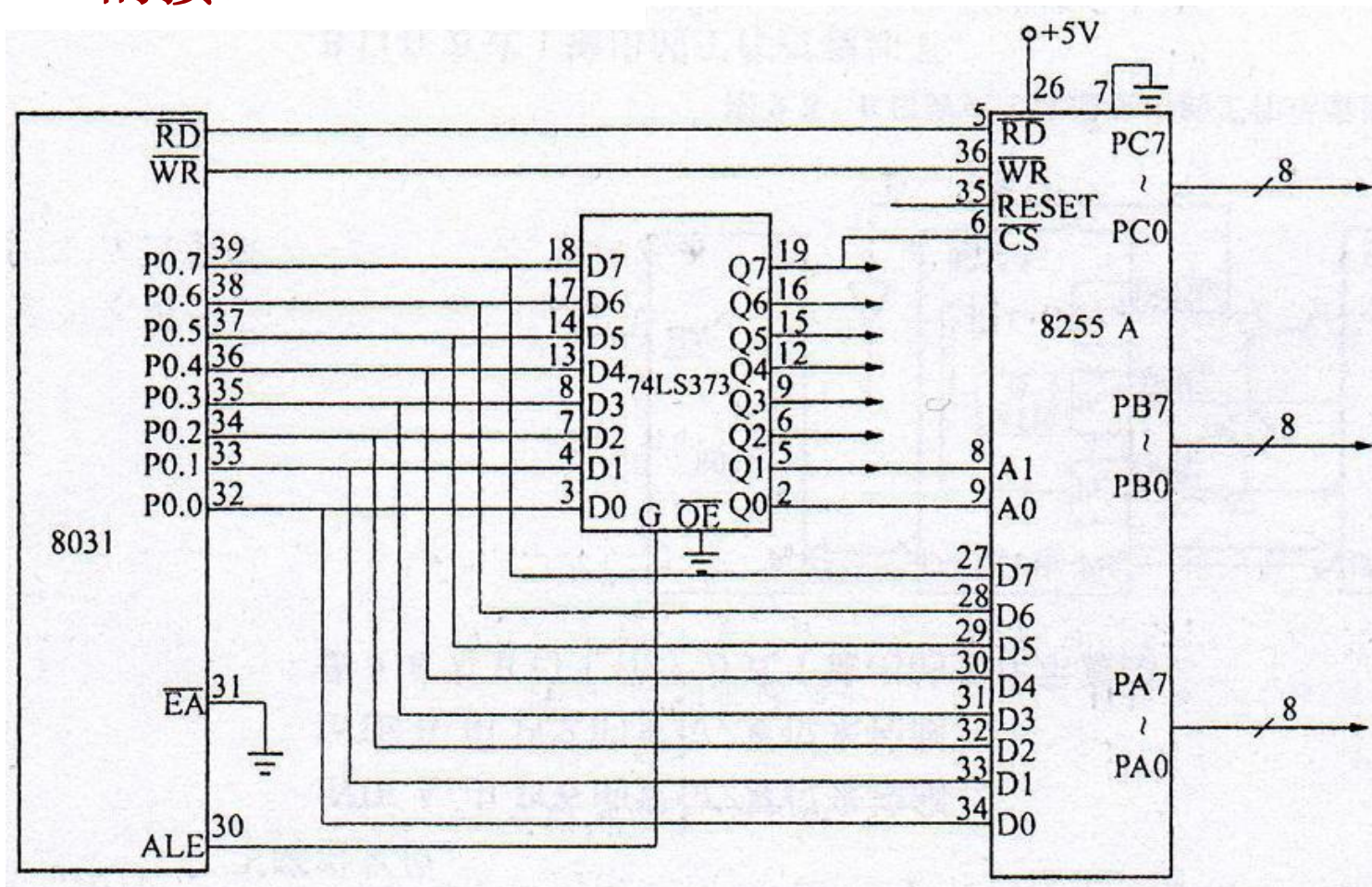
低8位=7DH;

PC口地址:

低8位=7EH;

控制口地址:

低8位=7FH;



## MCS-51与并行I/O芯片8255A的接口设计

### 8255A的接口编程举例

例9.1: 要求8255A工作方式在方式0，且PA口作为输入，PB口、PC口作为输出。

MOV	A, #90H	; 控制字。方式0; PA口输入, PB口、PC口输出
MOV	DPTR, #0FF7FH	; 控制寄存器地址→DPTR
MOVX	@DPTR, A	; 方式控制字→控制寄存器
MOV	DPTR, #0FF7CH	; PA口地址→DPTR
MOVX	A, @DPTR	; 从PA口读数据
MOV	B, A	; 读入数据暂存入寄存器B
MOV	DPTR, #0FF7DH	; PB口地址→DPTR
MOV	A, #DATA1	; 要输出的数据DATA1→A
MOVX	@DPTR, A	; DATA1从PB口输出
MOV	DPTR, #0FF7EH	; PC口地址→DPTR
MOV	A, #DATA2	; 要输出的数据DATA2→A
MOVX	@DPTR, A	; DATA2从PC口输出

## MCS-51与并行I/O芯片8255A的接口设计

### 8255A的接口编程举例

例9.2: 对PC端口的置位/复位。把PC端口的PC5置1（置位），再将PC2清0（复位）。

<b>MOV</b>	<b>R1, #7FH</b>	； 控制寄存器地址→R1
<b>MOV</b>	<b>A, #0BH</b>	； PC5置1的控制字= 00001011B
<b>MOVX</b>	<b>@R1, A</b>	； 控制字→控制寄存器，PC5=1
<b>MOV</b>	<b>R1, #7FH</b>	； 控制寄存器地址→R1，这里可省略
<b>MOV</b>	<b>A, #04H</b>	； PC2清0的控制字= 00000100B
<b>MOVX</b>	<b>@R1, A</b>	； 控制字→控制寄存器，PC2=0



## 外设接口概述

### 输入外设:

- 键盘、拨码开关
- BCD码拨盘

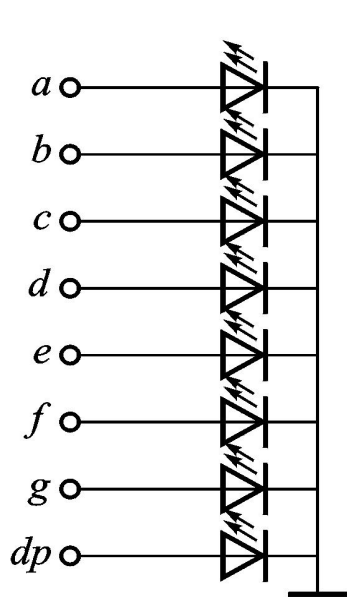
### 输出外设

- LED显示器
- LED数码管
- LCD显示器
- 打印机

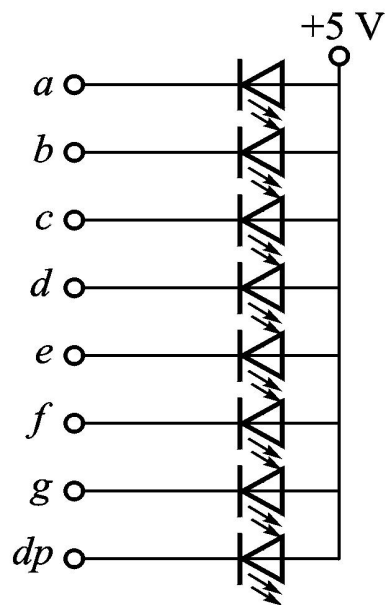
### LED显示器接口原理

- **LED**（Light Emitting Diode）：发光二极管的缩写。磷砷化镓二极管发红光，磷化镓二极管发绿光，碳化硅二极管发黄光，**氮化镓二极管**发蓝光。
- LED显示器，俗称数码管。
- 常用的 LED显示器为8段（或7段，8段比7段多了一个小数点“dp”段）。有**共阳极**和**共阴极**两种。

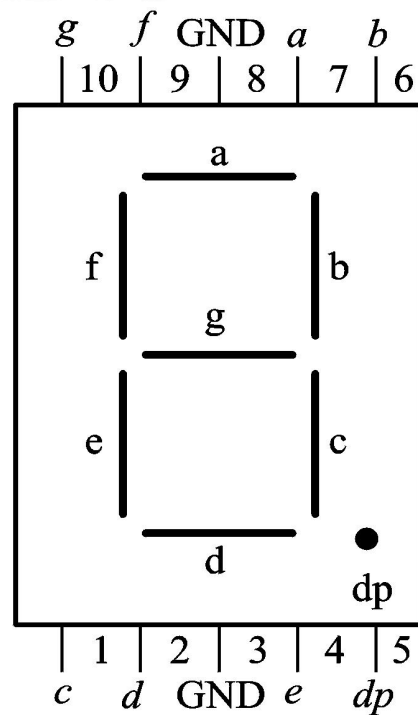
## LED显示器接口原理



(a) 共阴极



(b) 共阳极



(c) 外形及引脚

10-1



## LED显示器接口原理

- LED显示不同的符号或数字，要为LED提供段码（或称字型码）

代码位	D7	D6	D5	D4	D3	D2	D1	D0
显示段	dp	g	f	e	d	c	b	a

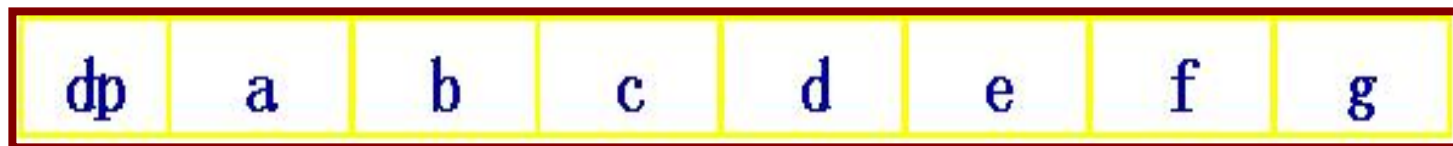
- 若按上述段码表示，则形成段码表。

## 第10部分 MCS-51与键盘、显示器、拨盘、打印机的接口设计

显示字符	共阴极段码	共阳极段码	显示字符	共阴极段码	共阳极段码
0	3FH	C0H	c	39H	C6H
1	06H	F9H	d	5EH	A1H
2	5BH	A4H	E	79H	86H
3	4FH	B0H	F	71H	8EH
4	66H	99H	P	73H	8CH
5	6DH	92H	U	3EH	C1H
6	7DH	82H	T	31H	CEH
7	07H	F8H	y	6EH	91H
8	7FH	80H	H	76H	89H
9	6FH	90H	L	38H	C7H
A	77H	88H	“灭”	00H	FFH
b	7CH	83H	...	...	...

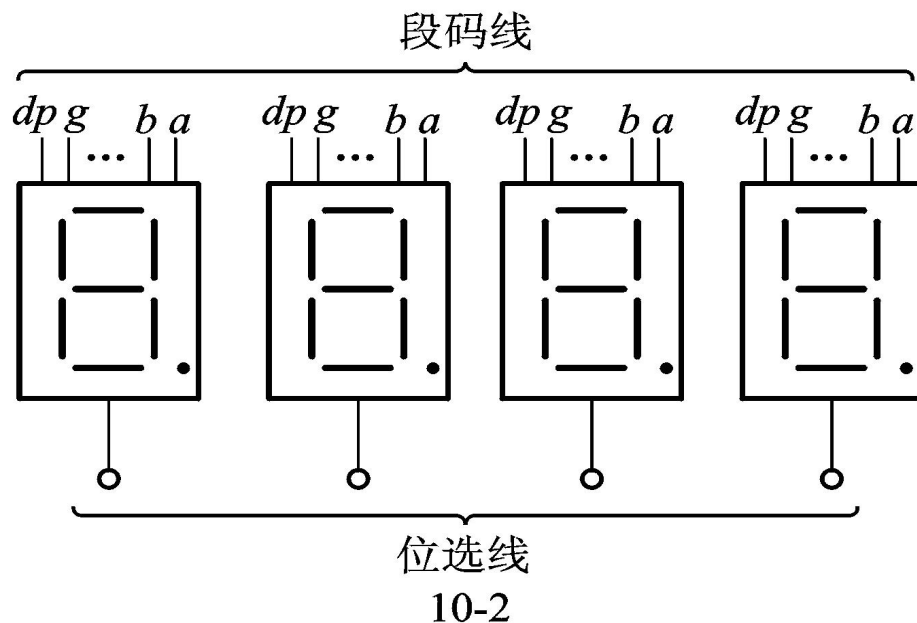
### LED显示器接口原理

- 段码并不固定，习惯上以“a”为对应段码的最低位。若以下图所示形成码段，则“0”的段码为7EH（共阴）。





## LED显示器接口原理

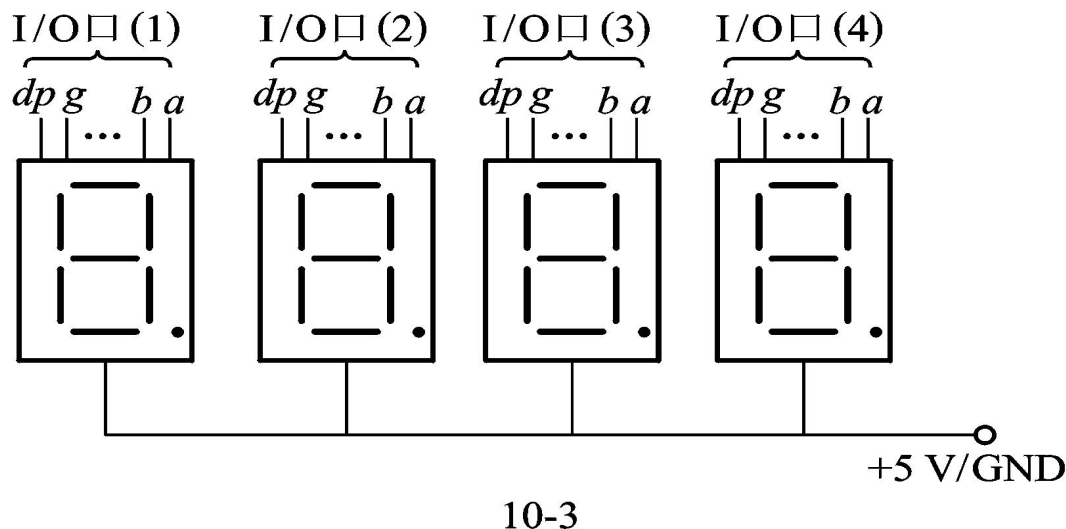


- N个LED显示块有N位位选线和 $8 \times N$ 根段码线。
- 段码线控制显示的字型；位选线控制该显示位的亮或暗。

## LED显示器接口原理

### 静态显示方式







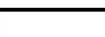

- 各位的公共端连接在一起（接地或+5V）。
- 每位的段码线（a~dp）分别与一个8位的锁存器输出相连。
- 显示字符一确定，相应锁存器的段码输出将维持不变，直到送入另一个段码为止。



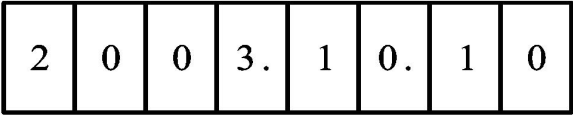
# LED显示器接口原理

## 动态显示方式

- 微观显示过程：某一时刻，只有一位LED被选通显示，其余位则是熄灭的
- 宏观显示结果：人眼看到的是8位稳定的同时显示的字符。

显示字符	段 码	位显码	显示器显示状态(微观)	位选通时序
0	3FH	FEH	<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div>0</div></div>	 $T_1$
1	06H	FDH	<div><div></div><div></div><div></div><div></div><div></div><div></div><div>1</div><div></div></div>	 $T_2$
0	BFH	FBH	<div><div></div><div></div><div></div><div></div><div>0.</div><div></div><div></div><div></div></div>	 $T_3$
1	06H	F7H	<div><div></div><div></div><div></div><div>1</div><div></div><div></div><div></div><div></div></div>	 $T_4$
3	CFH	EFH	<div><div></div><div></div><div>3.</div><div></div><div></div><div></div><div></div><div></div></div>	 $T_5$
0	3FH	DFH	<div><div></div><div>0</div><div></div><div></div><div></div><div></div><div></div><div></div></div>	 $T_6$
0	3FH	BFH	<div><div></div><div>0</div><div></div><div></div><div></div><div></div><div></div><div></div></div>	 $T_7$
2	5BH	7FH	<div><div>2</div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>	 $T_8$

(a) 8 位 LED 动态显示过程



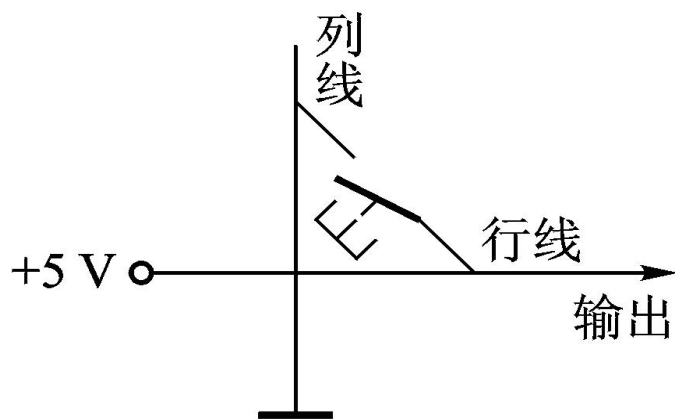
(b) 人眼看到的显示结果



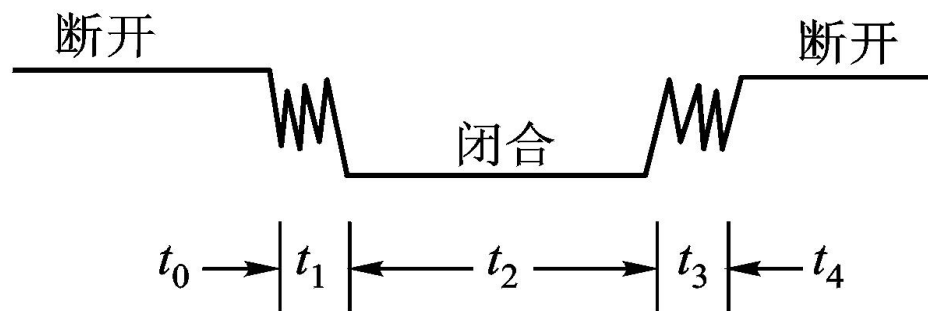
## 键盘接口原理

### 键盘输入特点

- 键盘：一组按键开关的集合。
- 行线电压信号通过键盘开关机械触点的断开、闭合。



(a) 按键开关



(b) 键闭合时行线输出电压波形

10-6

## 键盘接口原理

### 按键的确认

- 检测行线电平。高电平：断开；低电平：闭合。

### 消除按键抖动

- 基本思想：

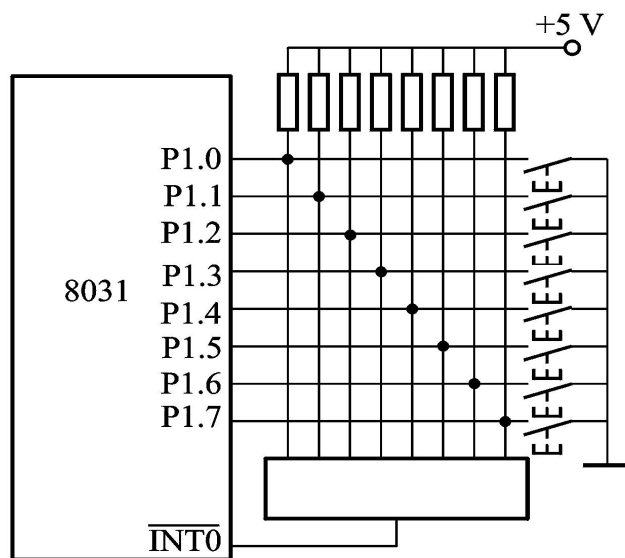
检测到有键按下，键对应的行线为低，软件延时10ms后，行线如仍为低，则确认该行有键按下。

当键松开时，行线变高，软件延时10ms后，行线仍为高，说明按键已松开。

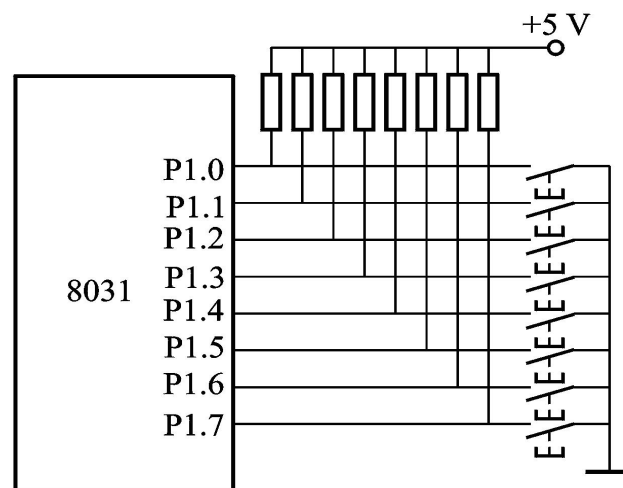
## 键盘接口原理

### 独立式键盘

- 各键相互独立，每个按键各接一根输入线，通过检测输入线的电平状态可很容易判断那个键被按下。



(a) 中断方式



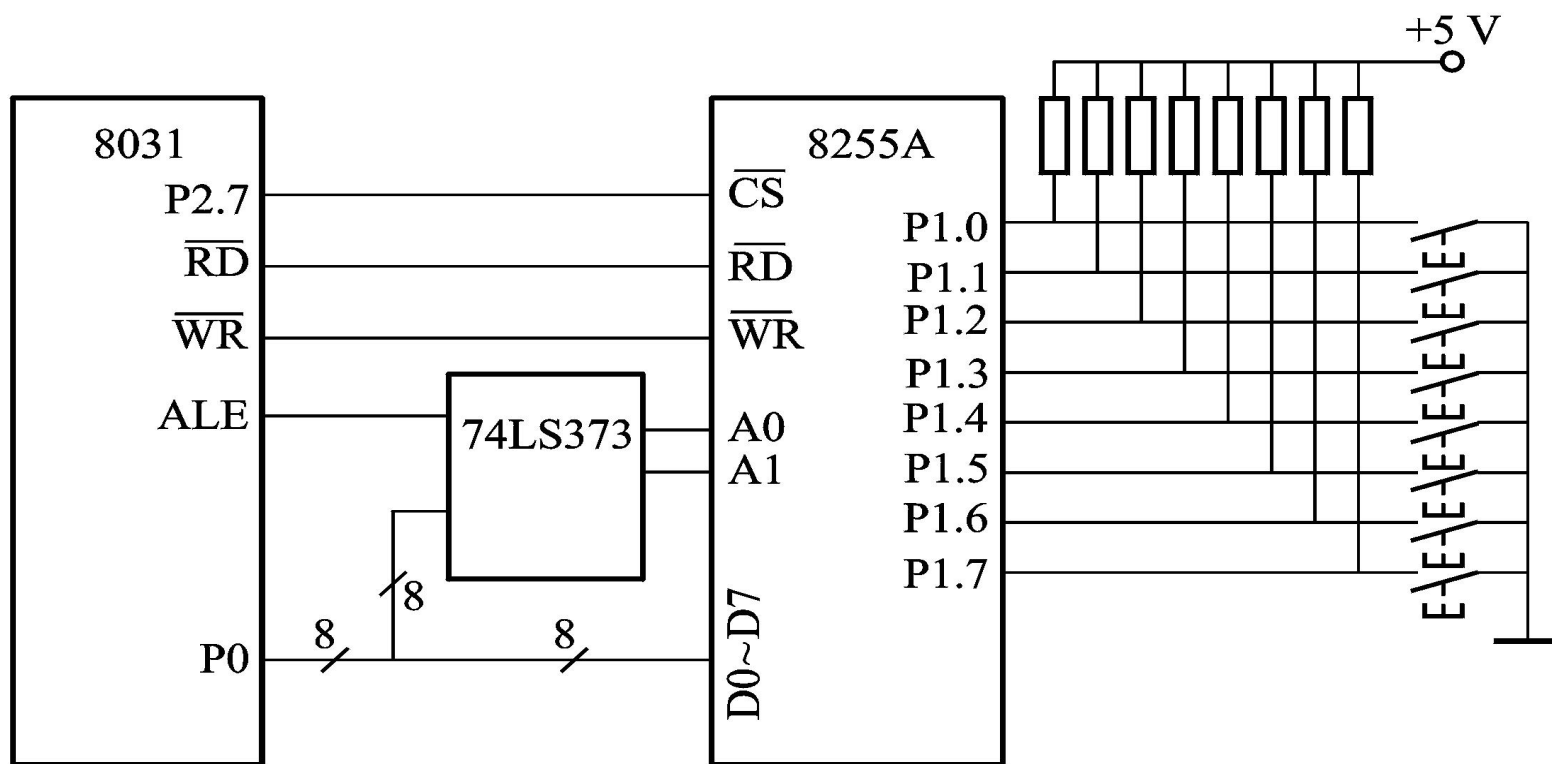
(b) 查询方式

10-7



## 键盘接口原理

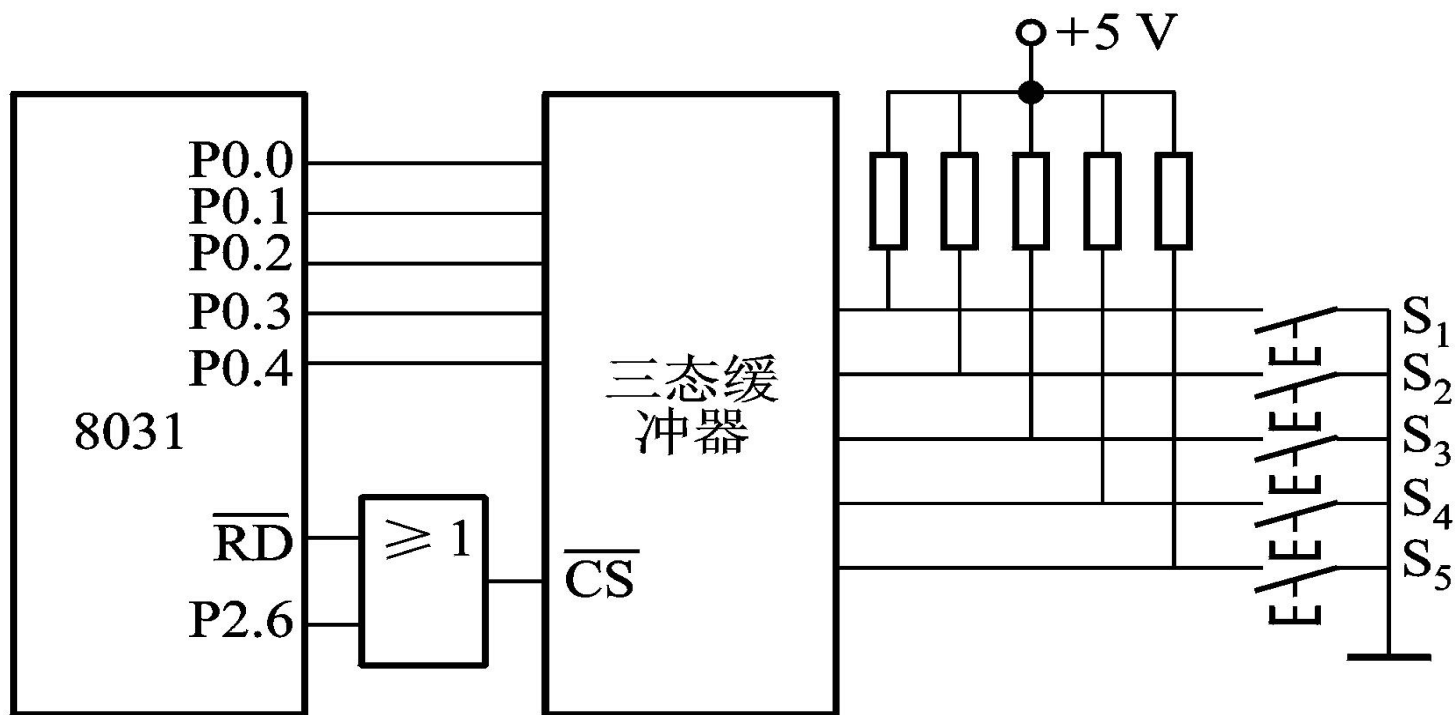
### 8255A扩展I/O口的独立式按键接口电路



10-8

## 键盘接口原理

三态缓冲器扩展的I/O口独立式按键接口电路



10-9

## 键盘接口原理

### 三态缓冲器扩展的I/O口独立式按键接口电路

<b>KEYIN: MOV DPTR, #0BFFFH</b>	<b>; 键盘端口地址BFFFH</b>
<b>MOVX A, @DPTR</b>	<b>; 读键盘状态</b>
<b>ANL A, #1FH</b>	<b>; 屏蔽高三位</b>
<b>MOV R3, A</b>	<b>; 保存键盘状态值</b>
<b>LCALL DELAY10</b>	<b>; 延时10ms去键盘抖动</b>
<b>MOVX A, @DPTR</b>	<b>; 再读键盘状态</b>
<b>ANL A, #1FH</b>	<b>; 屏蔽高三位</b>
<b>CJNE A, R3, RTN</b>	<b>; 两次不同, 抖动引起, 返回</b>
<b>CJNE A, #1EH, KEY2</b>	<b>; 相等, 有键按下, 不等转KEY2</b>
<b>LCALL PKEY1</b>	<b>; 是S1键按下, 转S1键处理</b>
<b>.....</b>	

## 键盘接口原理

三态缓冲器扩展的I/O口独立式按键接口电路

；子程序PKEY1

KEY2:	CJNE A, #1DH, KEY3	； S2键未按下，转KEY3
	LCALL PKEY2	； S2键按下，转PKEY2处理
KEY3:	CJNE A, #1BH, KEY4	； S3未按下，转KEY4
	LCALL PKEY3	； S3按下，转PKEY3处理
KEY4:	CJNE A, #17H, KEY5	； S4键未按下，转KEY5
	LCALL PKEY4	； S4按下，转PKEY4处理
KEY5:	CJNE A, #0FH, PASS	； S5未按下，转RETURN
	LCALL PKEY5	； S5按下，转PKEY5处理
RTN:	RET	； 重键或无键按下，从子程序返回



## 键盘接口原理

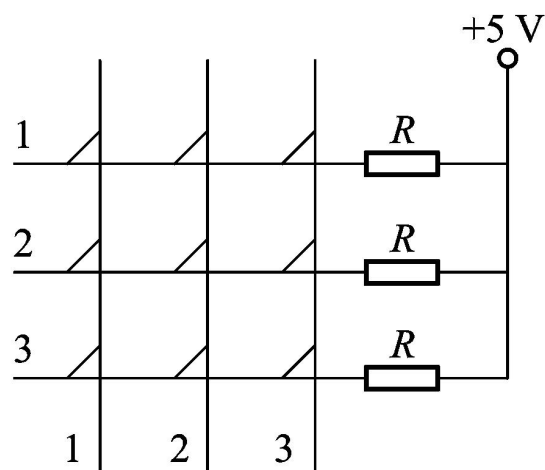
### 行列式（矩阵）键盘

- 无键按下，该行线为高电平，当有键按下时，行线电平由列线的电平来决定。
- 由于行、列线为多键共用，各按键彼此将相互发生影响，必须将行、列线信号配合起来并作适当的处理，才能确定闭合键的位置。

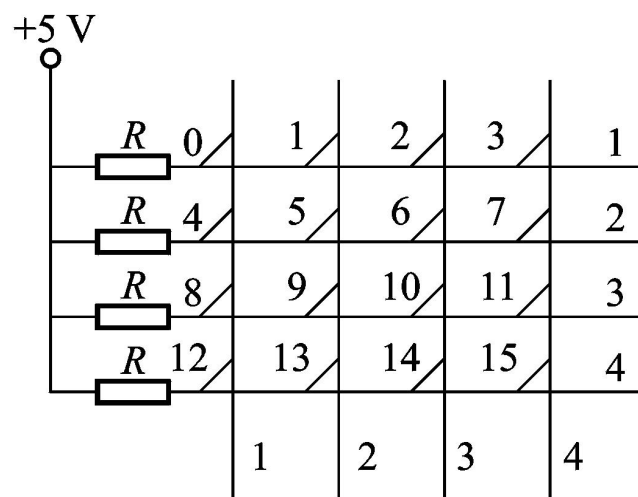
## 键盘接口原理

### 行列式（矩阵）键盘

- 用于按键数目较多的场合，由行线和列线组成，按键位于行、列的交叉点上。



(a) 3×3 键盘



(b) 4×4 键盘

10-10

## 键盘接口原理

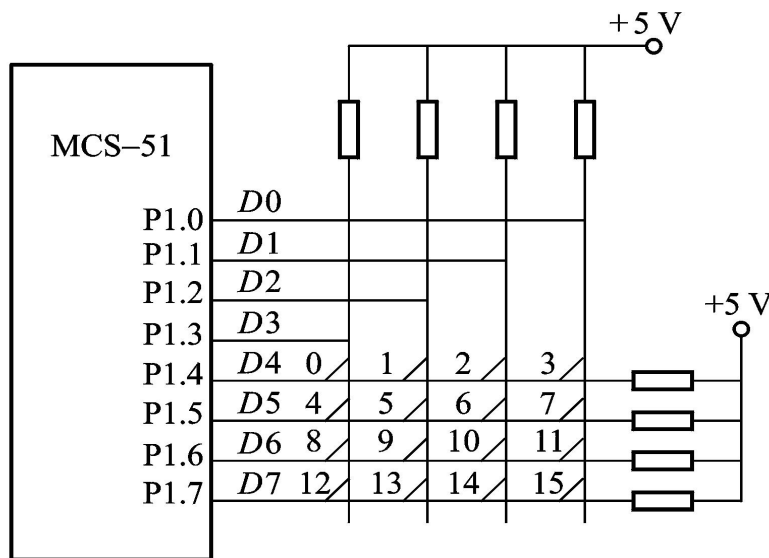
### 按键识别方法——扫描法

- 第1步：识别键盘有无键按下。
- 第2步：如有键被按下，识别出具体的按键。
- 扫描法：先把某一列置低电平，其余各列为高电平，检查各行线电平的变化，如果某行线电平为低，可确定此行列交叉点处的按键被按下。

## 键盘接口原理

### 按键识别方法——线反转法

- 第1步：列线输出为全低电平，则行线中电平由高变低的所在行为按键所在行。
- 第2步：行线输出为全低电平，则列线中电平由高变低所在列为按键所在列。



10-11



### 键盘工作方式

- 原则：即要保证能及时响应按键操作，又不要过多占用CPU的工作时间。
- 键盘工作方式有3种，即编程扫描、定时扫描和中断扫描。

#### 编程扫描方式

- (1) 在键盘扫描子程序中，先判断有无键按下；
- (2) 用软件来消除按键抖动的影响；
- (3) 求按下键的键号；
- (4) 等待按键释放后，再进行按键功能的处理操作。

## 键盘工作方式

### 定时扫描方式

利用单片机内的定时器，产生 10ms 的定时中断，对键盘进行扫描。

### 中断工作方式

只有在键盘有键按下时，才执行键盘扫描程序，如无键按下，单片机将不理睬键盘。

