



微机原理及接口技术

# 第8章 基于总线的I/O接口设计

# 教材中的相关内容:

## 第8章 基于总线的I/O接口设计

### 8.1 基于ISA总线的I/O接口设计

8.1.1 LED接口



8.1.2 键盘接口

8.1.3 光电隔离接口

8.1.4 A/D与D/A变换接口

8.1.5 步进电机接口

### 8.2 基于PCI总线的I/O接口设计

### 8.3 基于USB总线的I/O接口设计

## 第9章 设备驱动程序设计

## 第10章 PC机系统、SOC的概念

## 8.1 基于ISA总线的I/O接口设计（目录）

### 一. LED接口

#### 1. LED数码管

#### 2. 接口电路

### 二. 键盘接口

#### 1. 非编码式键盘

#### 2. 编码式键盘

### 三. 光电隔离I/O接口

#### 1. 光电隔离器件

#### 2. 光隔I/O接口电路

#### 3. 光隔接口应用举例

#### 4. 应注意的问题

## 8.1 基于ISA总线的I/O接口设计（目录）

### 四. A/D、D/A 接口

#### 1. D/A

- ① 基本原理、技术指标
- ② 典型芯片：DAC0832 →  
引线、时序、典型连接、与8088连接

#### 2. A/D

- ① 工作原理、结构
- ② 技术指标
- ③ 芯片及应用：

{	AD574
}	ADC0809

#### 3. 数据监测与控制系统

## 8.1 基于ISA总线的I/O接口设计（目录）

### 五. 步进电机接口

1. 步进电机工作原理
2. { 脉冲分配器  
驱动放大电路
3. 步进电机控制接口实例

微机原理及接口技术

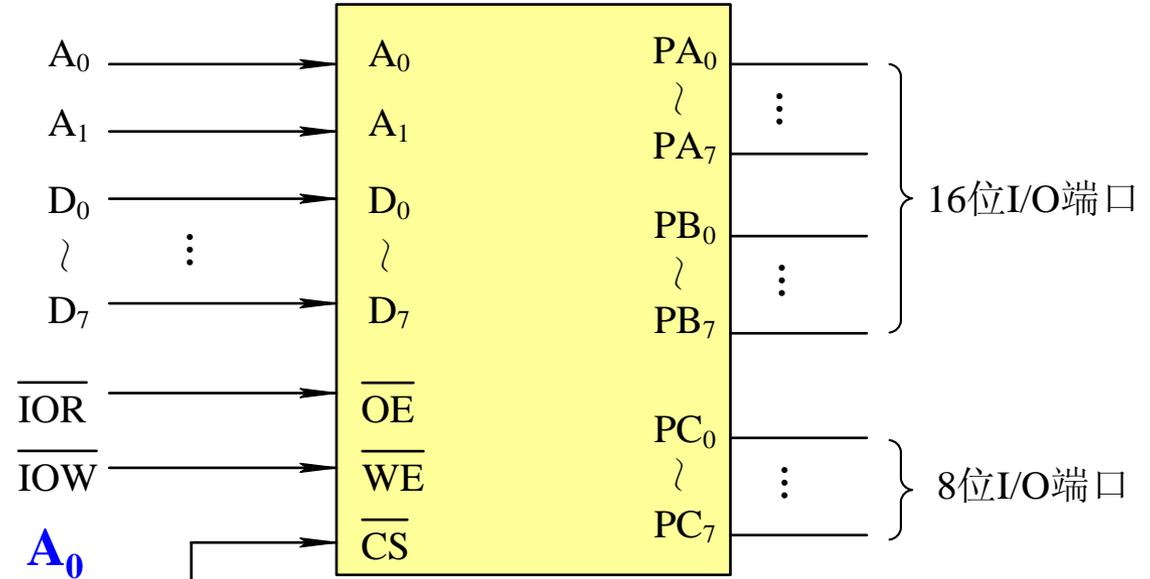
## 第8章 基于总线的I/O接口设计

### 8.1 基于ISA总线的I/O接口设计

# 【例8.1】 8位ISA总线接口

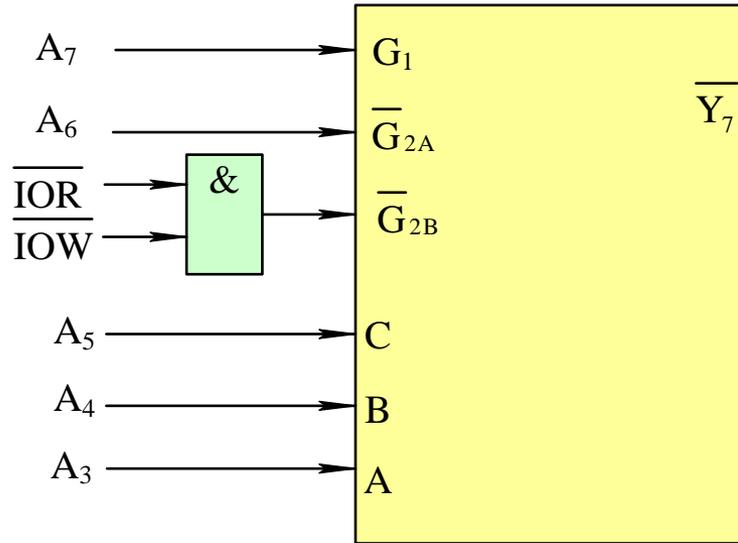
8位ISA总线

8255



A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>
1	0	1	1	1	x	x	x

A口: B8H  
 B口: B9H  
 C口: BAH  
 控制: BBH



74LS138

## 【例8.1】8位ISA总线接口

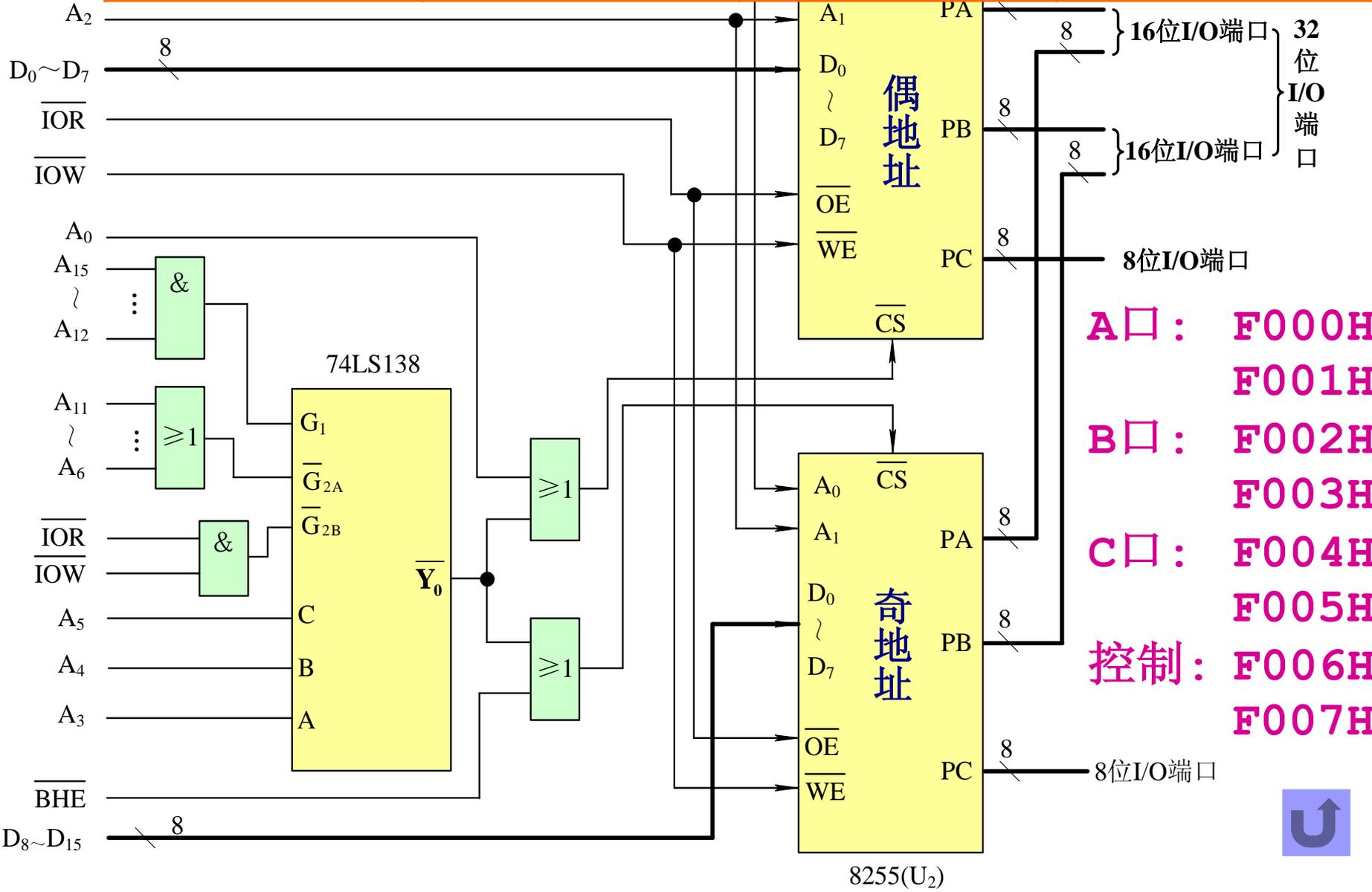
从I/O端口获得**16位数据**:

```
MOV AL, 92H      ;初始化8255,  
OUT 0BBH, AL    ;A、B口方式0, 输入  
.....  
IN AL, 0B9H     ;从8255 B口读入数据高8位  
MOV AH, AL  
IN AL, 0B8H     ;从8255 A口读入数据低8位
```

**8位ISA总线接口：16位数据传输必须通过两次I/O操作才能完成。**

## 【例8.2】 16位ISA总线接口

16位ISA	A <sub>15</sub>	A <sub>14</sub>	A <sub>13</sub>	A <sub>12</sub>	A <sub>11</sub>	A <sub>10</sub>	A <sub>9</sub>	A <sub>8</sub>	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>
A <sub>1</sub>	1	1	1	1	0	0	0	0	0	0	0	0	0	x	x	



## 【例8.2】16位ISA总线接口

从I/O端口（U1的C口）获得8位数据：

```
MOV DX, 0F006H
```

```
MOV AL, 89H
```

```
OUT DX, AL ;初始化8255, C口输入
```

```
.....
```

```
MOV DX, 0F004H
```

```
IN AL, DX ;从8255 C口读入8位数据
```

## 【例8.2】16位ISA总线接口

从I/O端口（U1、U2的A口）输出**16位数据**：

```
MOV DX, 0F006H
```

```
MOV AL, 89H
```

```
OUT DX, AL ;初始化U1, A口方式0, 输出
```

```
MOV DX, 0F007H
```

```
MOV AL, 89H
```

```
OUT DX, AL ;初始化U2, A口方式0, 输出
```

```
.....
```

```
MOV DX, 0F000H
```

```
MOV AX, 5555H
```

```
OUT DX, AX ;16位数据从U1和U2的A口同时送出
```

## 【例8.2】16位ISA总线接口

从I/O端口(U1、U2的A口、B口)输出**32位数据**：  
高16位在IFOH存储单元中，低16位在IFOL存储单元中

```
MOV DX, 0F006H
```

```
MOV AX, 8989H
```

```
OUT DX, AX ;初始化U1和U2, A、B口方式0, 输出  
.....
```

```
MOV DX, 0F000H
```

```
MOV AX, IFOL
```

```
OUT DX, AX ;低16位数据从U1和U2的A口同时送出
```

```
MOV DX, 0F002H
```

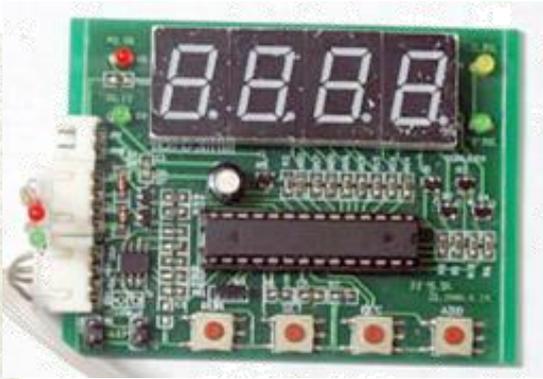
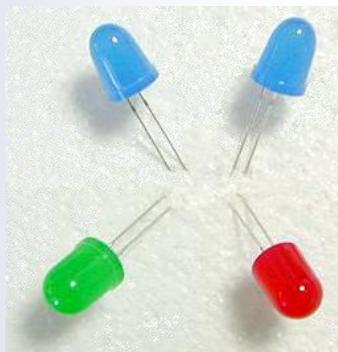
```
MOV AX, IFOH
```

```
OUT DX, AX ;高16位数据从U1和U2的B口同时送出
```

# 微机原理及接口技术

## 第8章 基于总线的I/O接口设计

### 8.1 基于ISA总线的I/O接口设计



#### 8.1.1 LED接口

# 8.1.1 LED接口

## 一、LED数码管

{ 共阴 → 动态  
  共阳 → 静态

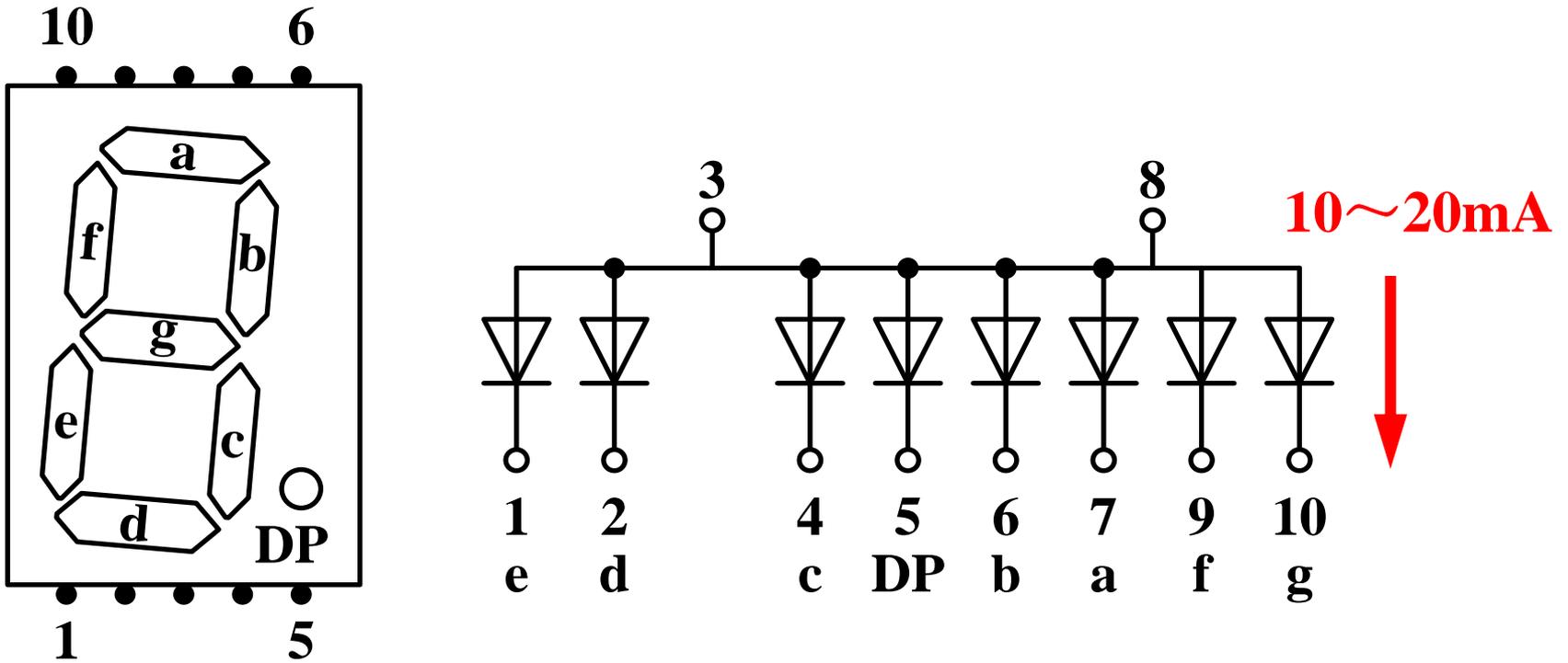


图8.3 共阳LED数码管的示意图

## 8.1.1 LED接口

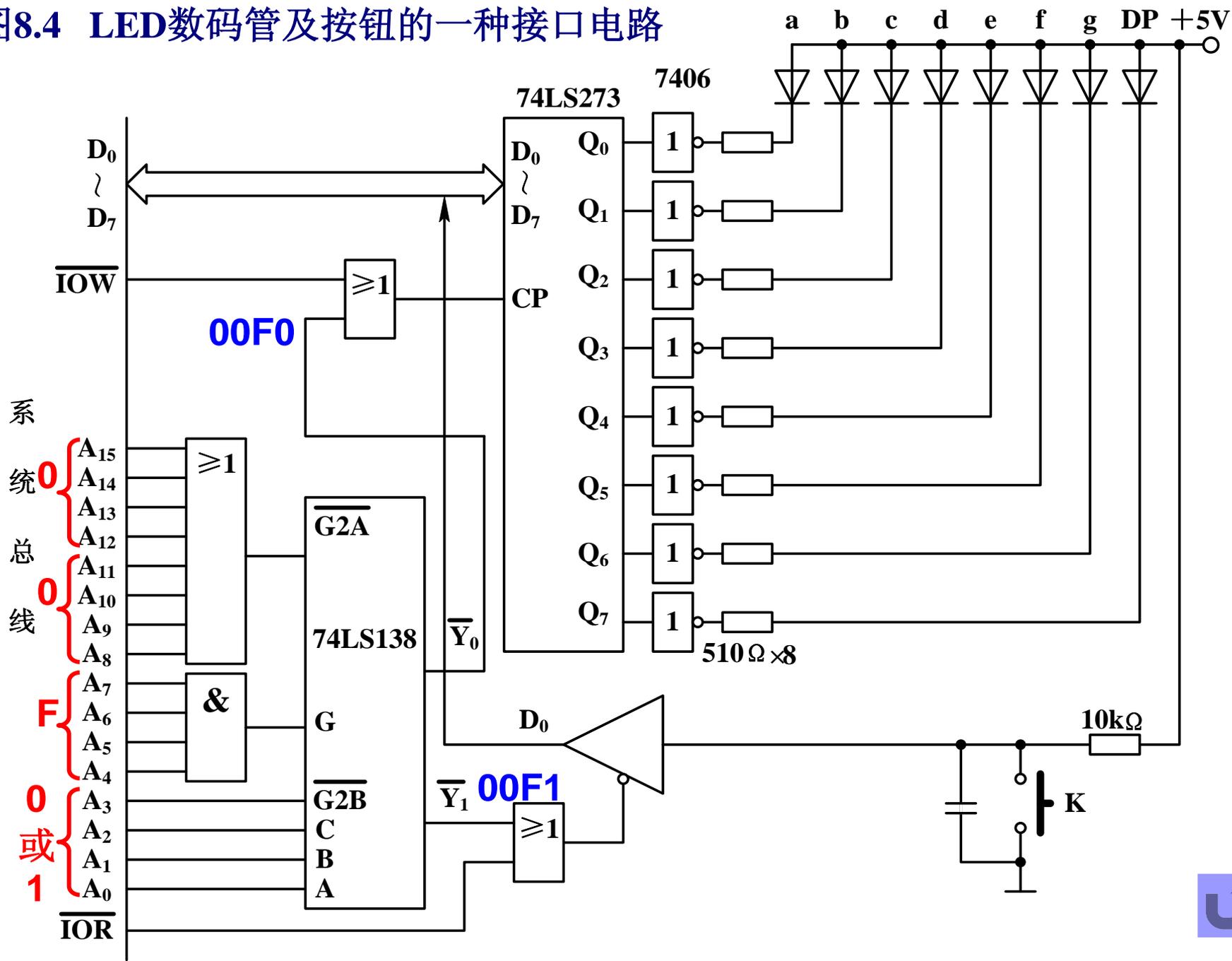
### 二、接口电路

### 1) 用通用并行接口芯片作接口

P345, 图8.4

- 输出接口: 锁存器 + 驱动器 → LED数码管  
74LS273 74LS06 (OC)
- 输入接口: 三态门 ← 按钮K
- $510\Omega \rightarrow 8.4\text{ mA}$   
 $330\Omega \rightarrow 13\text{ mA} (4.3\text{V}/330\Omega)$

图8.4 LED数码管及按钮的一种接口电路



# 8.1.1 LED接口

## 二、接口电路

### 1) 用通用并行接口芯片作接口

```
START: MOV DX, 00F1H
```

读K

```
IN AL, DX
```

; 读键盘

判断

```
TEST AL, 01H
```

```
JNZ KOPEN
```

; 未按下 → KOPEN

按下

```
MOV DX, 00F0H
```

```
MOV AL, 4FH
```

```
OUT DX, AL
```

```
JMP START
```

; 4FH → 【00F0H】 端口 0100 1111

; dp g f e d c b a

```
KOPEN: MOV DX, 00F0H
```

未按下

```
MOV AL, 7DH
```

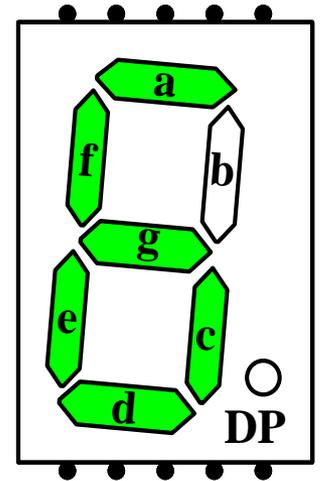
```
OUT DX, AL
```

```
JMP START
```

; 7DH → 【00F0H】 端口 0111 1101

; dp g f e d c b a

图8.4

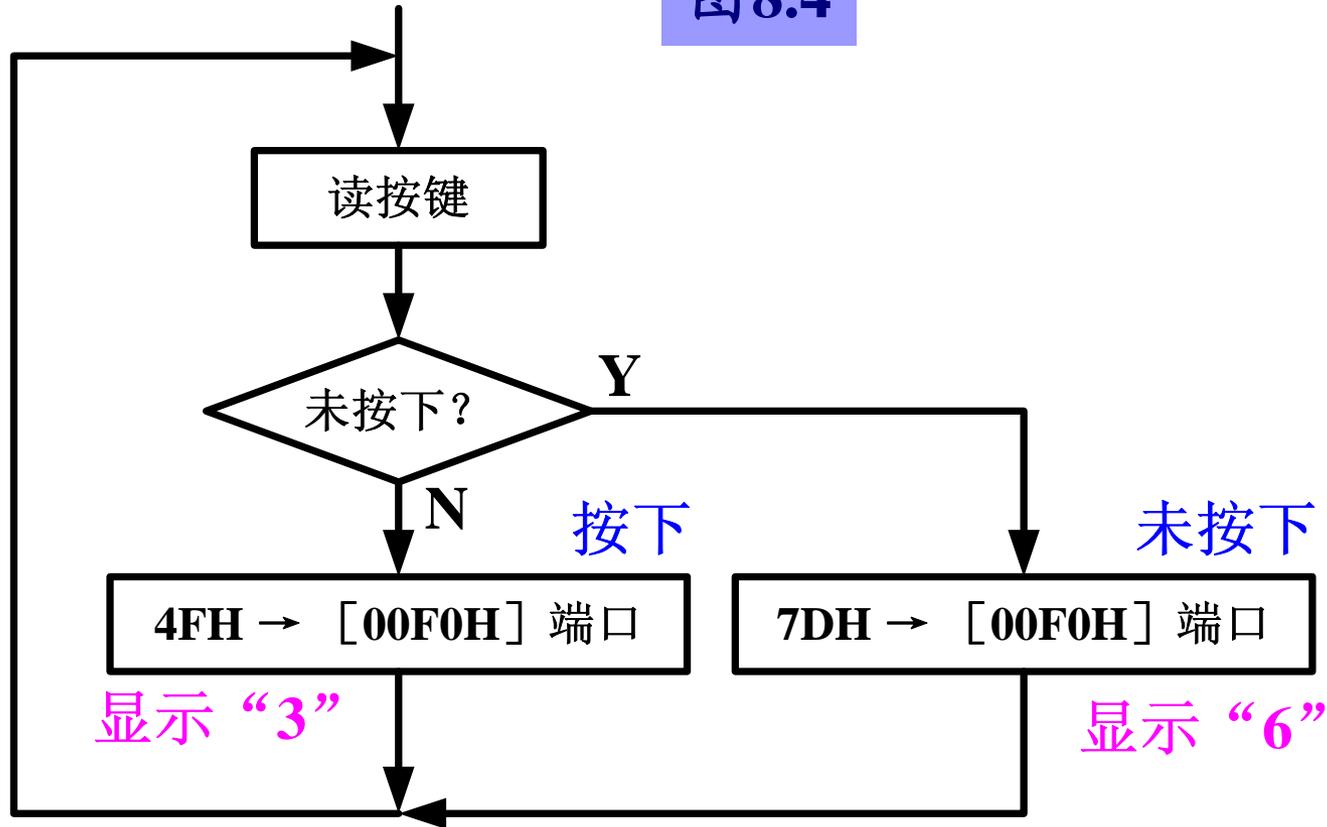


## 8.1.1 LED接口

### 二、接口电路

### 1) 用通用并行接口芯片作接口

图8.4



## 8.1.1 LED接口

### 二、接口电路

### 2) 用LED译码器作接口: DM9368

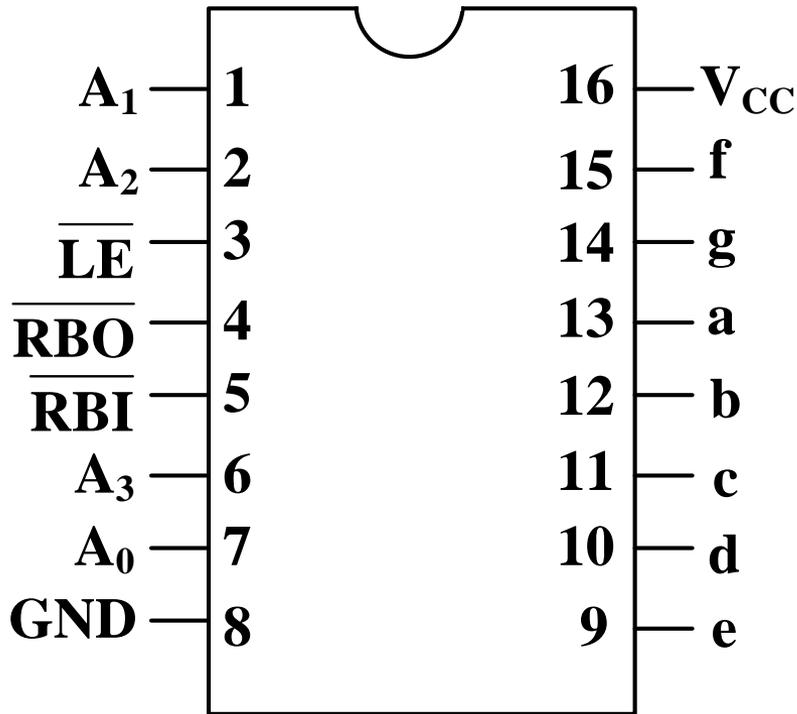
#### DM9368: 7段译码驱动锁存器

- 由Fairchild Semiconductor公司生产;
- 内部加入了输入锁存电路和用于直接驱动共阴极型LED显示的恒流输出电路。
- 功能: 接收一个4位二进制代码, 对其译码生成数字"0"~"9"和字母"A"~"F"的7段显示信号, 并驱动7段数码管相应的段。
- 引脚、真值表

## 8.1.1 LED接口

### 二、接口电路

### 2) 用LED译码器作接口: DM9368



(a)

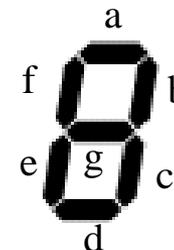
管脚名	描述
$A_0 \sim A_3$	地址(数据)输入
$\overline{RBO}$	波纹消隐输出(低有效)
$\overline{RBI}$	波纹消隐输入(低有效)
$a \sim g$	段驱动, 输出
$\overline{LE}$	锁存允许输入(低有效)

(b)

## DM9368芯片

(a) 引脚图; (b) 引脚功能描述

二进制 状态	输入						输出							显示		
	$\overline{\text{LE}}$	$\overline{\text{RBI}}$	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	a	b	c	d	e	f	g		$\overline{\text{RBO}}$	
—	H	*	×	×	×	×	← 不变 →							H	不变	
0	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	空白
0	L	H	L	L	L	L	H	H	H	H	H	H	L	H	H	0
1	L	×	L	L	L	H	L	H	H	L	L	L	L	H	H	1
2	L	×	L	L	H	L	H	H	L	H	H	L	H	H	H	2
3	L	×	L	L	H	H	H	H	H	L	L	H	H	H	H	3
4	L	×	L	H	L	L	L	H	H	L	L	H	H	H	H	4
5	L	×	L	H	L	H	H	L	H	H	L	H	H	H	H	5
6	L	×	L	H	H	L	H	L	H	H	H	H	H	H	H	6
7	L	×	L	H	H	H	H	H	H	L	L	L	L	H	H	7
8	L	×	H	L	L	L	H	H	H	H	H	H	H	H	H	8
9	L	×	H	L	L	H	H	H	H	L	L	H	H	H	H	9
10	L	×	H	L	H	L	H	H	H	L	H	H	H	H	H	A
11	L	×	H	L	H	H	L	L	H	H	H	H	H	H	H	b
12	L	×	H	H	L	L	H	L	L	H	H	H	L	H	H	C
13	L	×	H	H	L	H	L	H	H	H	H	L	H	H	H	d
14	L	×	H	H	H	L	H	L	L	H	H	H	H	H	H	E
15	L	×	H	H	H	H	H	L	L	L	H	H	H	H	H	F
×	×	×	×	×	×	×	L	L	L	L	L	L	L	L	L <sup>**</sup>	空白



\* 只有当二进制的零被存储在锁存器中时， $\overline{\text{RBI}}$ 才消隐显示。

\*\*  $\overline{\text{RBO}}$  用作输入时超越所有其他输入条件。

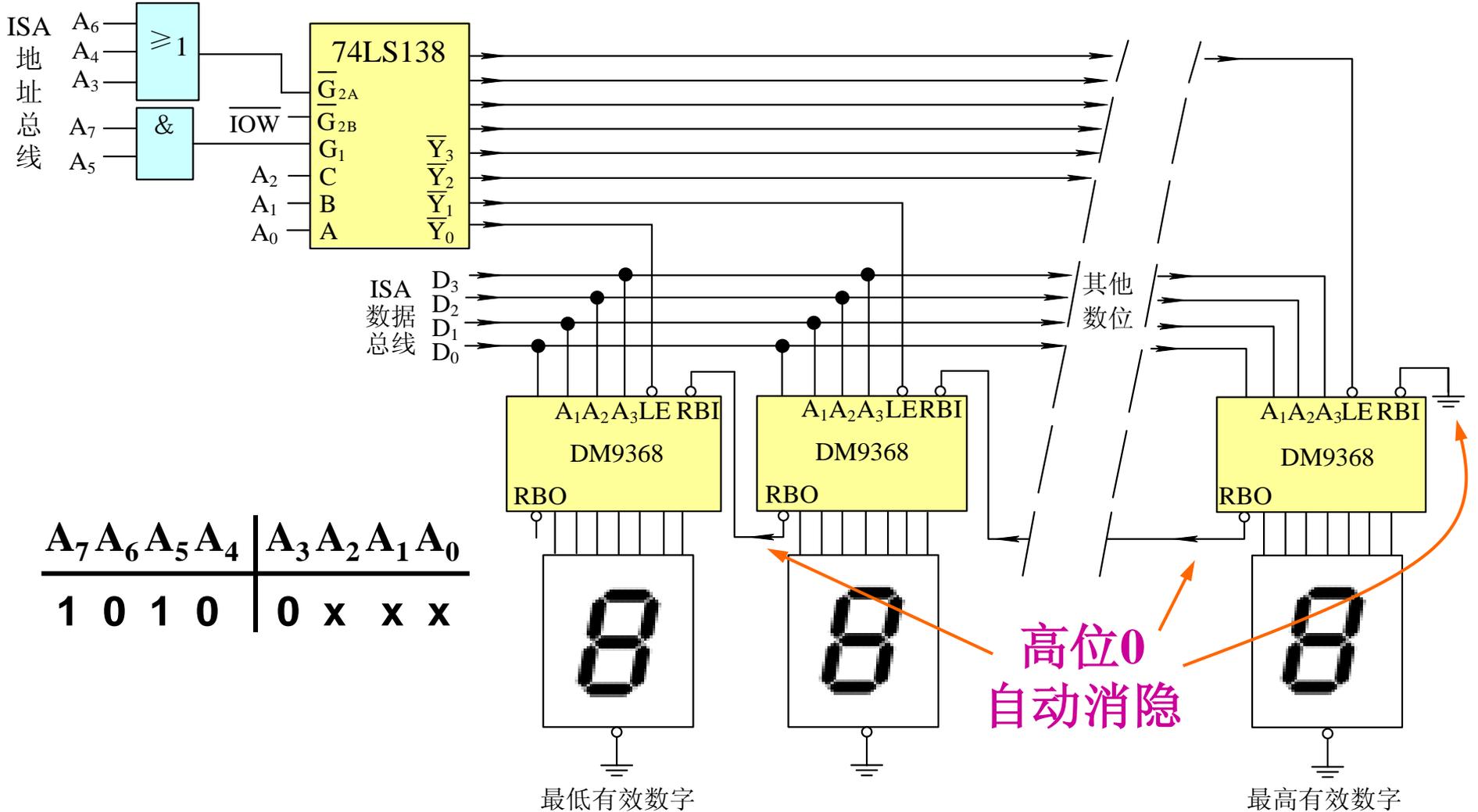
## DM9368芯片的真值表



# 8.1.1 LED接口

## 二、接口电路

## 2) 用LED译码器作接口: DM9368



利用DM9368实现LED数码管接口电路

## 8.1.1 LED接口

### 二、接口电路      2) 用LED译码器作接口: DM9368

利用上图电路中的4个数码管，实现将存储单元 *BUF* 中的4位十六进制数加以显示，并每经过1秒，重新读取 *BUF* 单元数据，更新显示，则控制程序如下：

```

MOV    CL, 4
RP:   MOV    AX, BUF           ;AX:16bit, 4个4bit数等待显示
      MOV    BL, AL
      AND    AL, 0FH
      OUT    0A0H, AL         ;显示bit0~bit3
      MOV    AL, BL
      AND    AL, 0F0H
      SHR    AL, CL           ;左移4位
      OUT    0A1H, AL         ;显示bit4~bit7
      MOV    AL, AH
      AND    AL, 0FH
      OUT    0A2H, AL         ;显示bit8~bit11
      MOV    AL, AH
      AND    AL, 0F0H
      SHR    AL, CL           ;左移4位
      OUT    0A3H, AL         ;显示bit12~bit15
      CALL   DLY1s           ;DLY1s为1s延迟程序
      JMP    RP

```

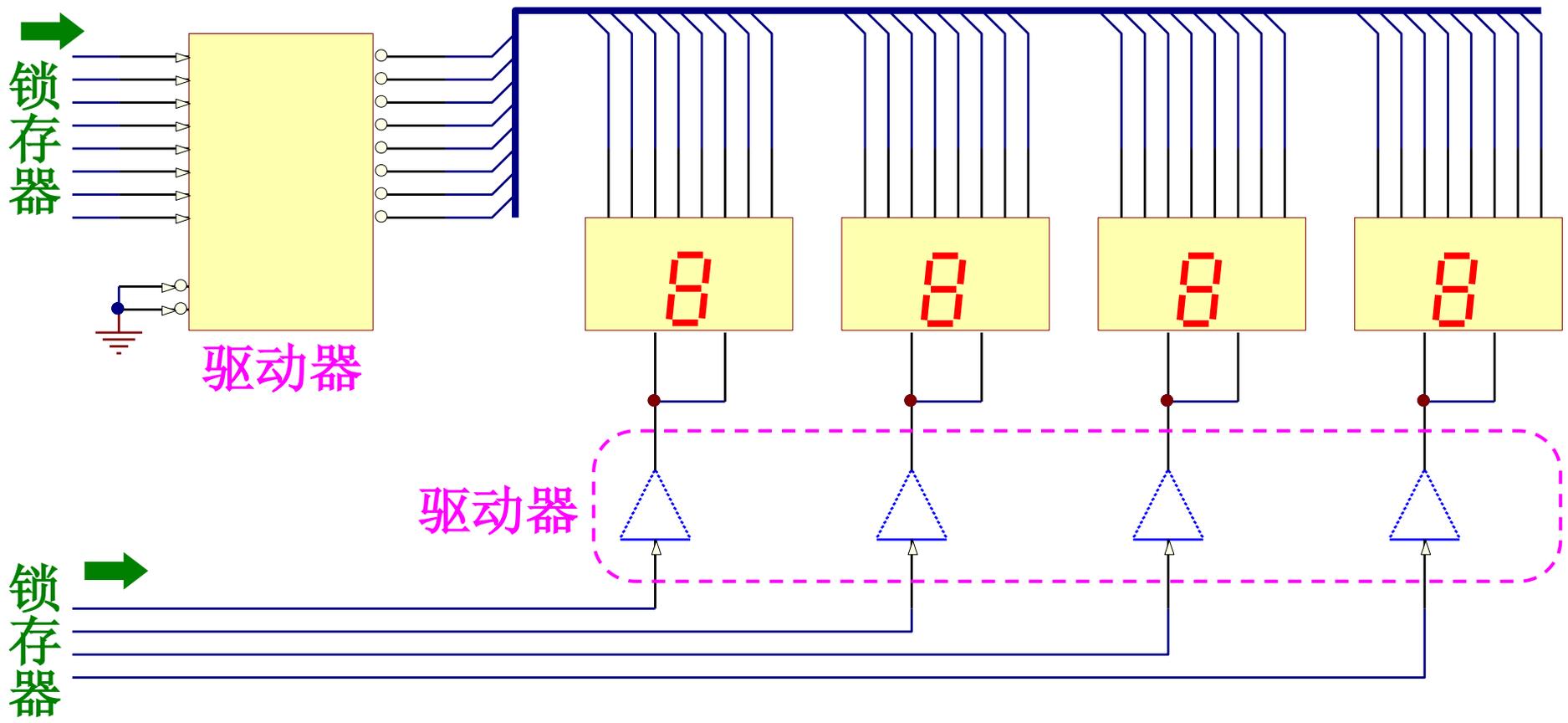
电路图

## 8.1.1 LED接口

### 三、动态显示的接口电路

#### 1) 用通用接口芯片

减轻提供电流驱动器的负载；  
增加吸收电流驱动器的负载。 → 共阴极LED数码管



LED数码管动态显示电路

## 8.1.1 LED接口

### 三、动态显示的接口电路

#### 2) 用专用接口芯片：MM74C912/917

MM74C912/917：6位数字BCD/Hex显示控制驱动器

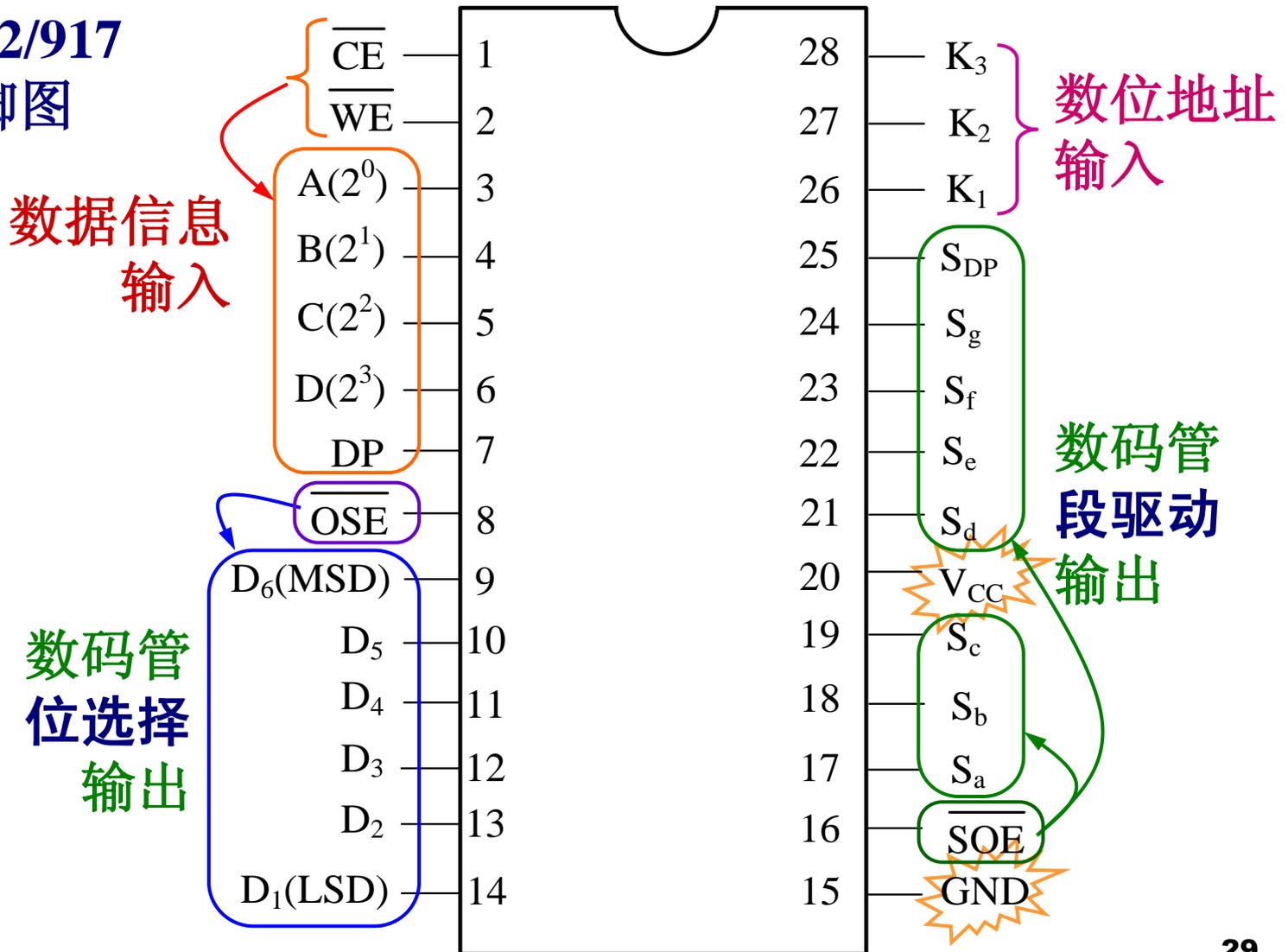
- 由Fairchild Semiconductor公司生产。
- 一个DM9368芯片只能驱动1个7段数码管；  
一个MM74C912/917芯片可控制驱动6个8段数码管。
- 功能：芯片内部带有存储器，可驱动6个8段LED显示。通过5个数据输入A、B、C、D和DP接收数据信息，并通过3个地址输入K1、K2和K3接收数位信息。
- 引脚、真值表、显示字符格式、工作时序

# 8.1.1 LED接口 三、动态显示的接口电路



## 2) 用专用接口芯片: MM74C912/917

MM74C912/917  
芯片的引脚图



## 8.1.1 LED接口 三、动态显示的接口电路

引脚

### 2) 用专用接口芯片: MM74C912/917

真值表

$\overline{CE}$	Digit Address			$\overline{WE}$	Operation
	K3	K2	K1		
0	0	0	0	0	Write Digit 1
0	0	0	0	1	Latch Digit 1
0	0	0	1	0	Write Digit 2
0	0	0	1	1	Latch Digit 2
0	0	1	0	0	Write Digit 3
0	0	1	0	1	Latch Digit 3
0	0	1	1	0	Write Digit 4
0	0	1	1	1	Latch Digit 4
0	1	0	0	0	Write Digit 5
0	1	0	0	1	Latch Digit 5
0	1	0	1	0	Write Digit 6
0	1	0	1	1	Latch Digit 6
0	1	1	0	0	Write Null Digit
0	1	1	0	1	Latch Null Digit
0	1	1	1	0	Write Null Digit
0	1	1	1	1	Latch Null Digit
1	X	X	X	X	Disable Writing

# 8.1.1 LED接口 三、动态显示的接口电路

引脚

## 2) 用专用接口芯片: MM74C912/917

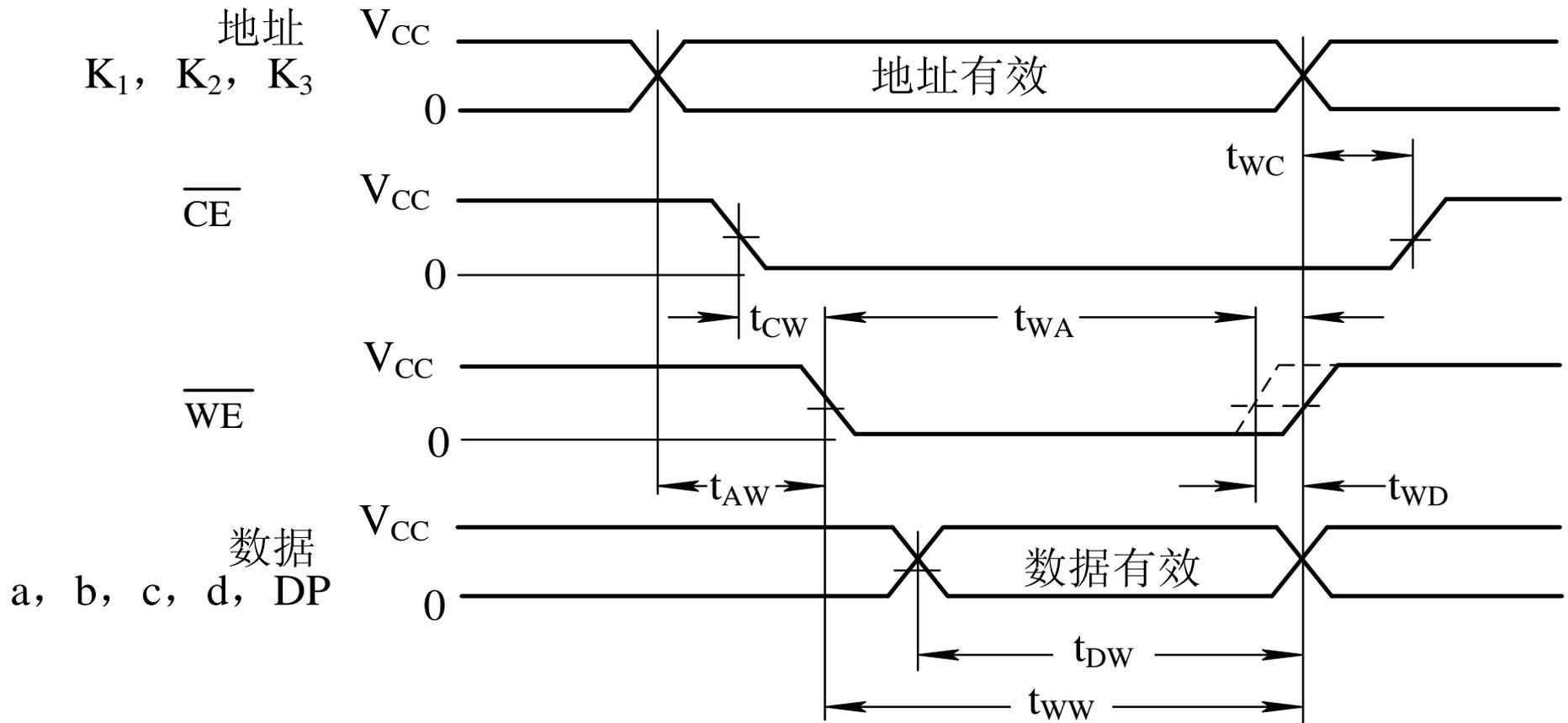
### 显示字符格式

MM74C917	高阻	0	1	2	3	4	5	6	7	8	9	A	b	C	d	E	F	F.
MM74C912	高阻	0	1	2	3	4	5	6	7	8	9	o	o	'	-	-		.
输入数据 A(2 <sup>0</sup> )	×	0	1	0	1	0	0	0	1	0	1	0	1	0	1	0	1	1
B(2 <sup>1</sup> )	×	0	0	1	1	0	1	1	1	0	0	1	1	0	0	1	1	1
C(2 <sup>2</sup> )	×	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	1
D(2 <sup>3</sup> )	×	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
DP	×	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
输出允许 $\overline{\text{SOE}}$	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

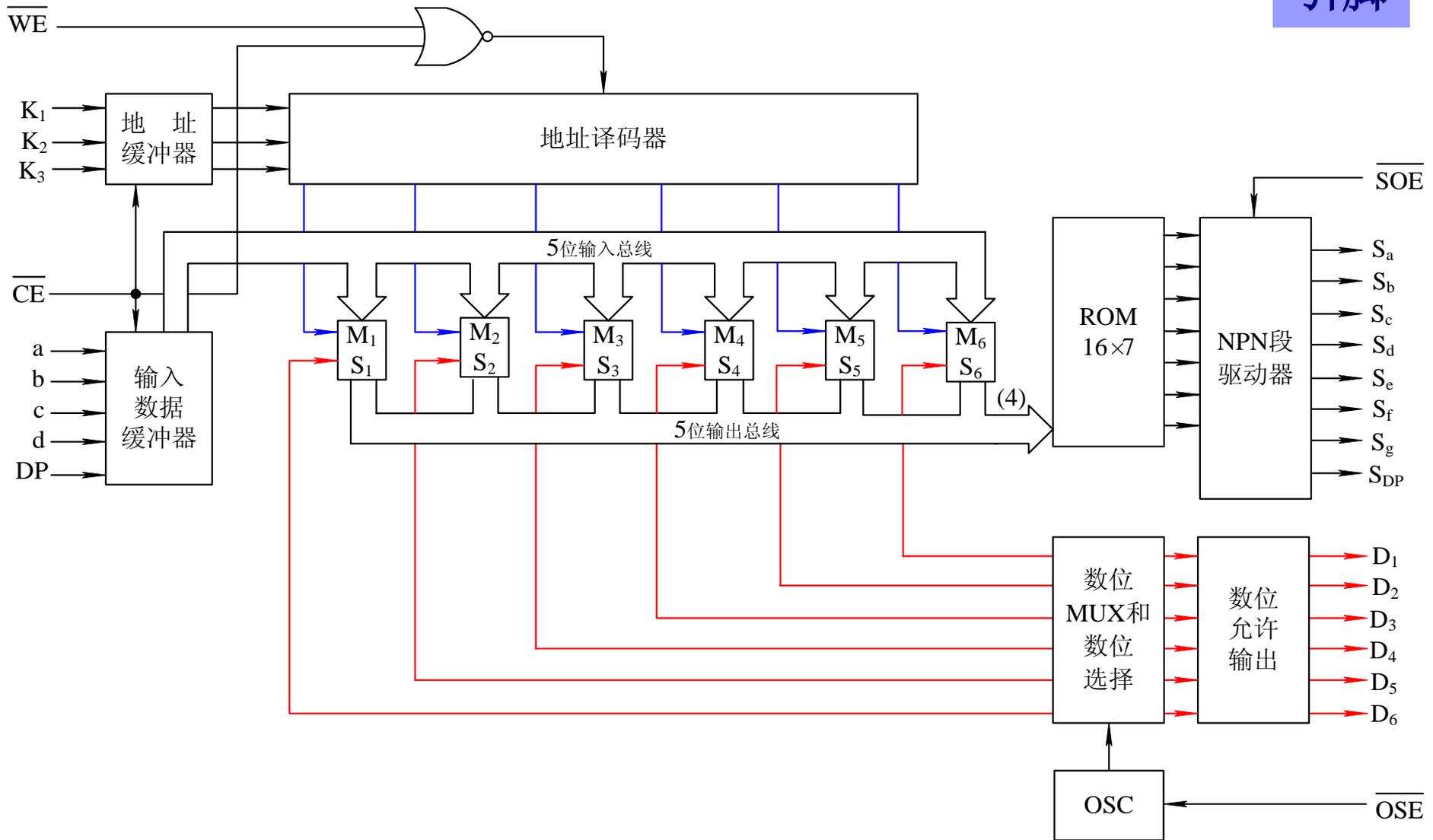
# 8.1.1 LED接口 三、动态显示的接口电路

引脚

## 2) 用专用接口芯片: MM74C912/917



MM74C912/917的工作时序

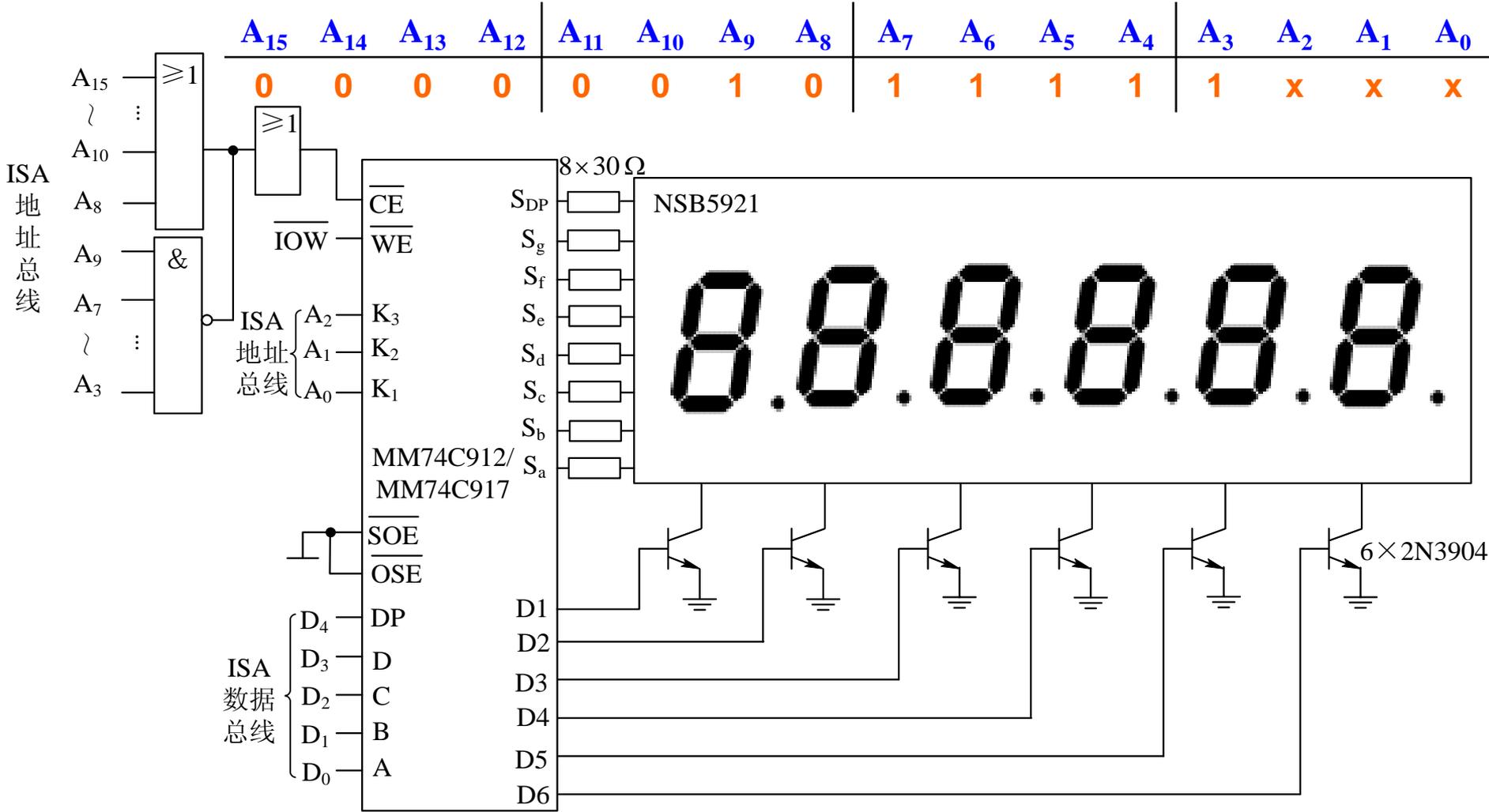


MM74C912/917显示控制器的内部电路框图

# 8.1.1 LED接口 三、动态显示的接口电路



## 2) 用专用接口芯片: MM74C912/917



MM74C912/917实现的LED数码管接口电路

## 8.1.1 LED接口 三、动态显示的接口电路

电路图

### 2) 用专用接口芯片：MM74C912/917

- 上图电路中采用MM74C912芯片作为LED数码管接口，可实现十进制数显示。
- 现在在以DATA为首地址的存储单元中存放有一个6位非压缩型BCD数(十进制数)，高4位为整数部分，低2位为小数部分，数码高位放在高地址中，依次排列。将这个带有小数点的6位数加以显示的控制子程序如下：

## 电路图

```
LEA SI, DATA
MOV DX, 02F8H
MOV AL, [SI+5]
OUT DX, AL      ;千位数输出到数码管1
INC DX
MOV AL, [SI+4]
OUT DX, AL      ;百位数输出到数码管2
INC DX
MOV AL, [SI+3]
OUT DX, AL      ;十位数输出到数码管3
INC DX
MOV AL, [SI+2]
OR AL, 10H
OUT DX, AL      ;个位数+小数点输出到数码管4
INC DX
MOV AL, [SI+1]
OUT DX, AL      ;十分之一位数输出到数码管5
INC DX
MOV AL, [SI]
OUT DX, AL      ;百分之一位数输出到数码管6
```

```
LEDDISPLAY: PUSH SI
              PUSH DX
              PUSH AX
              .
              .
              .
              POP AX
              POP DX
              POP SI
              RET
```

## 8.1.1 LED接口

P345, 图8.4

74LS244 → 8个三态门 → 可接8个按钮

按键多 (101个, 104个, ……)

→ **矩阵结构的键盘接口**

# 微机原理及接口技术

## 第8章 基于总线的I/O接口设计

### 8.1 基于ISA总线的I/O接口设计



#### 8.1.2 键盘接口

## 8.1.2 键盘接口

- 键盘接口需要检测：

- ① 有按键按下？

- ② 哪个键按下？

- 分类：

- 非编码式：**CPU检测** → 成本低，主机效率低
  - 编码式：键盘中有单片机 → 价高，主机效率高

## 8.1.2 键盘接口

### 一、非编码式键盘

1. 结构：矩阵

2. 接口：

行选 → 输出端口 **DIGLH**

		L5	L4	L3	L2	L1	L0
--	--	----	----	----	----	----	----

列选 → 输入端口 **KBSEL**

			R4	R3	R2	R1	R0
--	--	--	----	----	----	----	----

扫描过程：图8.11





图8.11 矩阵式键盘及其接口

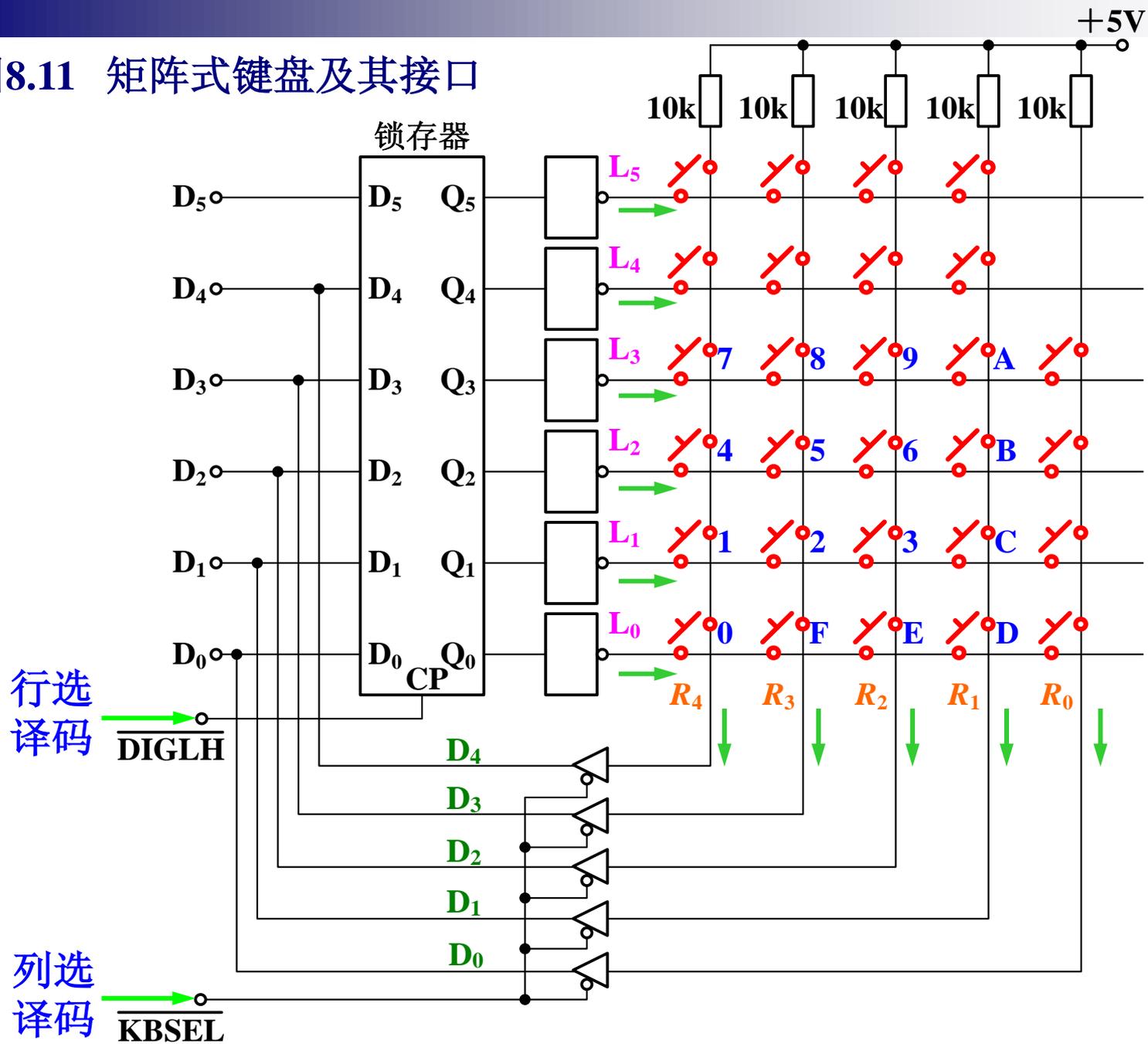


图8.11 矩阵式键盘及其接口

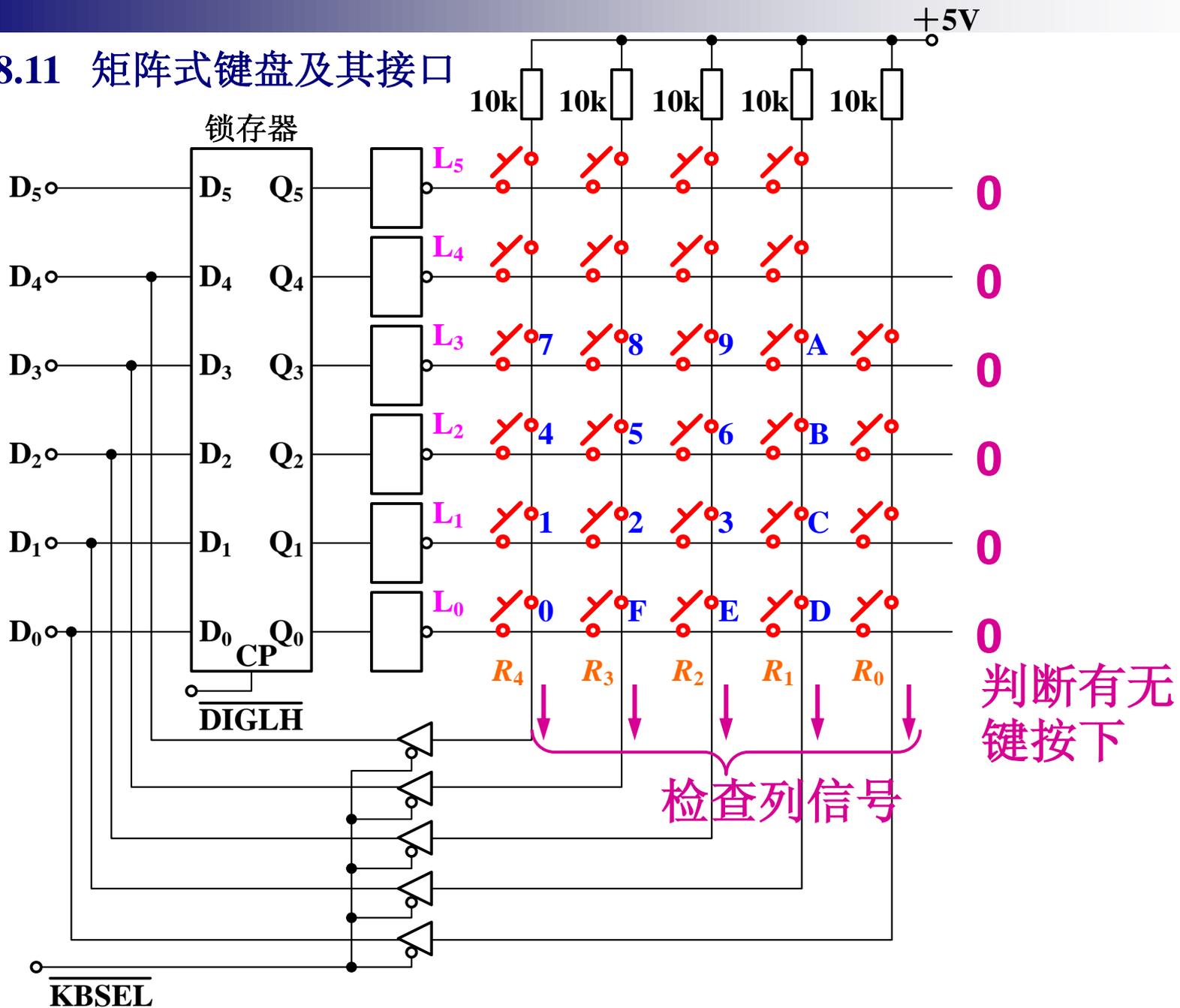


图8.11 矩阵式键盘及其接口

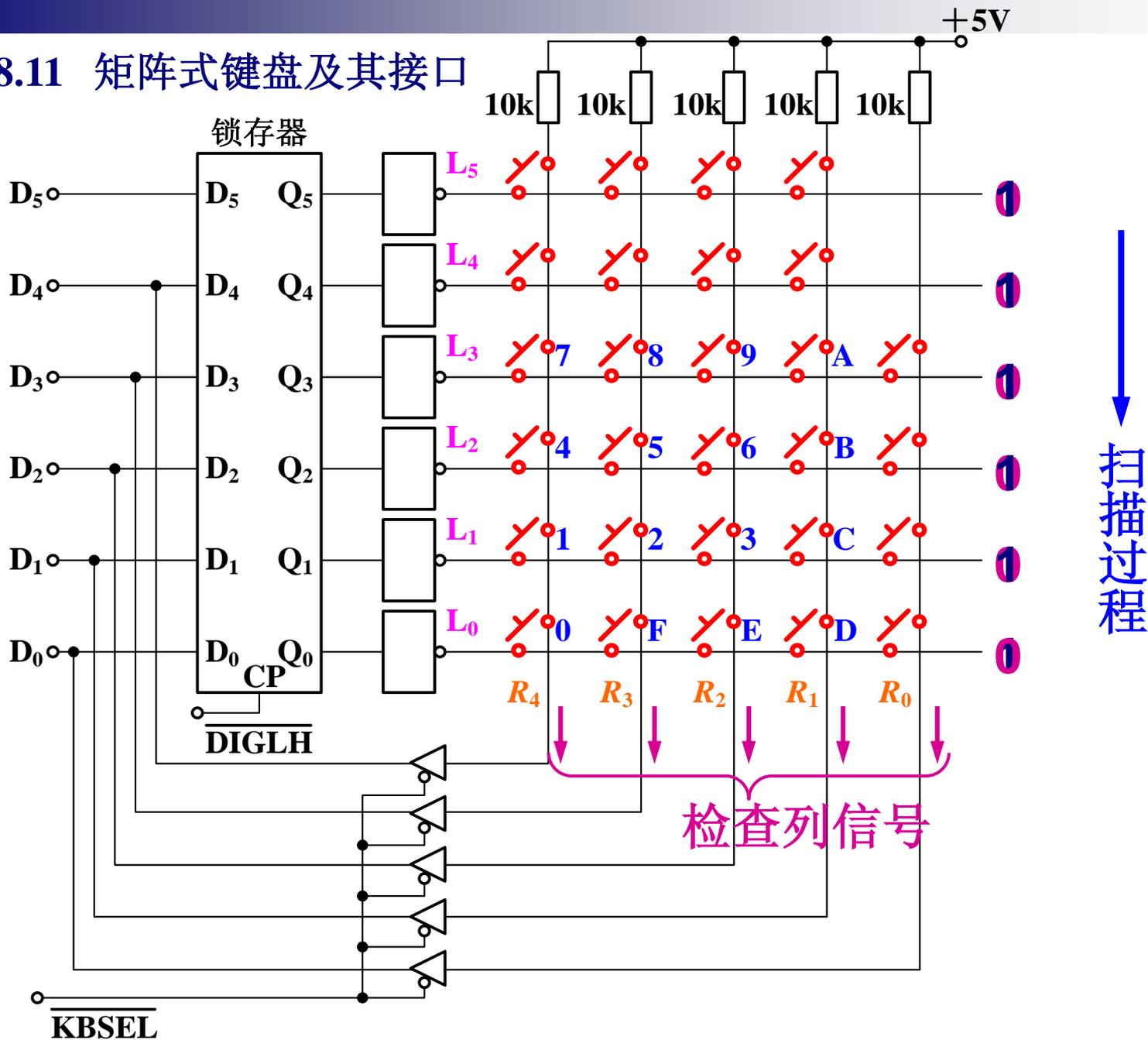


图8.11 矩阵式键盘及其接口

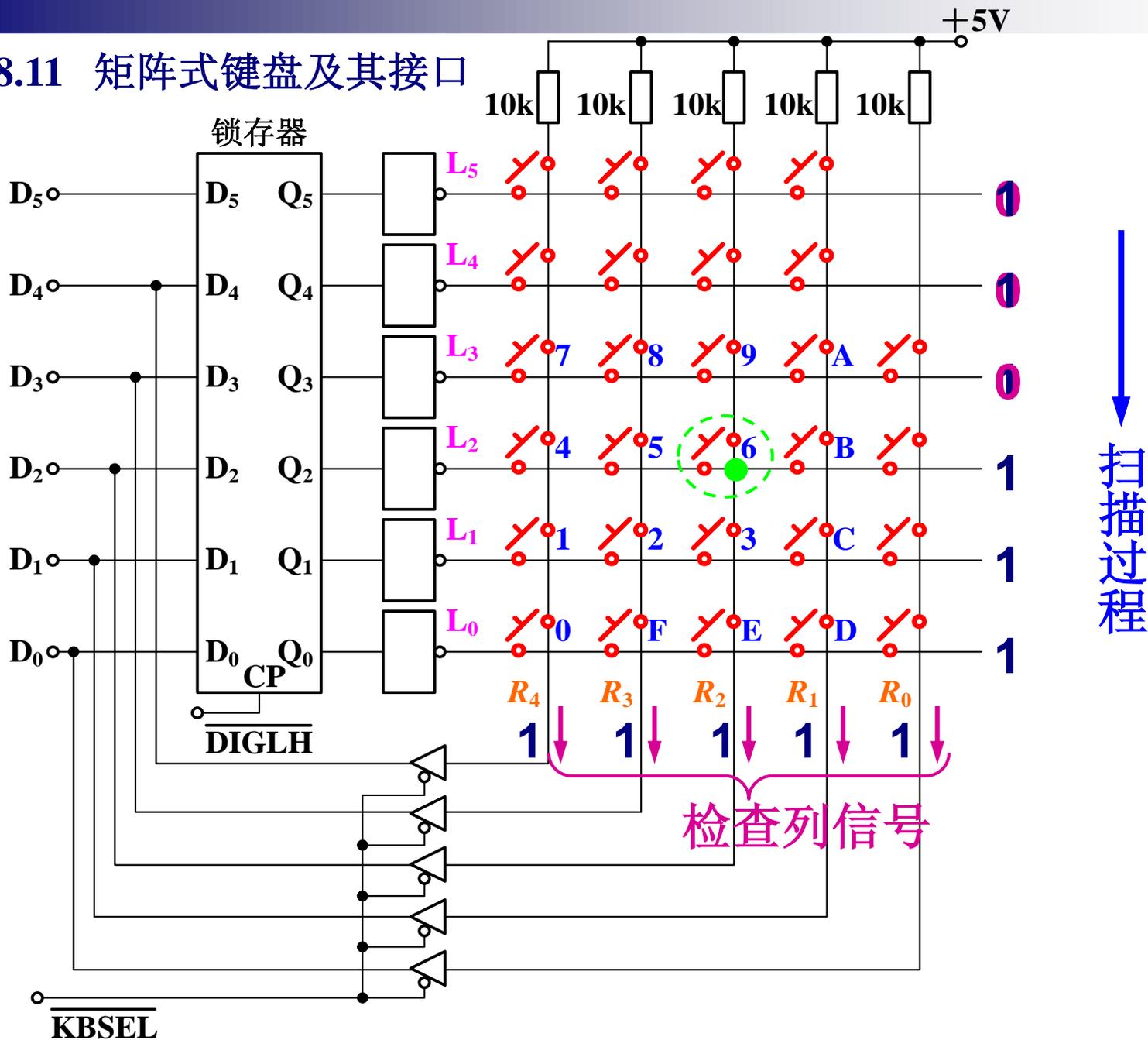
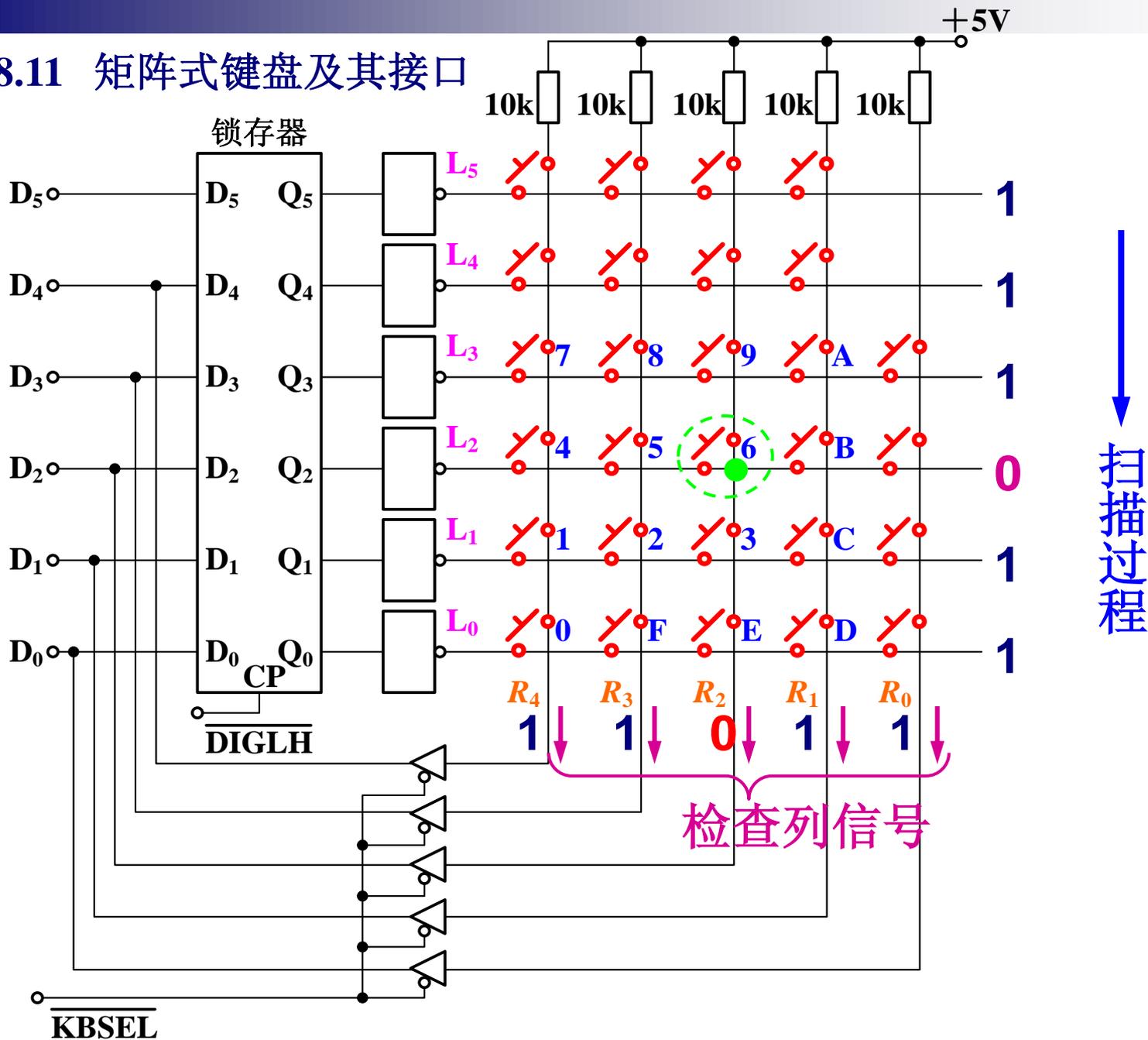


图8.11 矩阵式键盘及其接口



## 8.1.2 键盘接口

### 一、非编码式键盘

应注意的问题：

#### ① 键抖动：几ms~十几ms

- 硬件消抖：电容；RS触发器
- 软件消抖：延时20ms再检测

#### ② 串键：同时按下两个以上的键。

- 无效处理
- 等待释放
- 硬件封锁

#### ③ 防止按一次键而产生多次处理：

检测键抬起（等待），键抬起 → 程序继续。

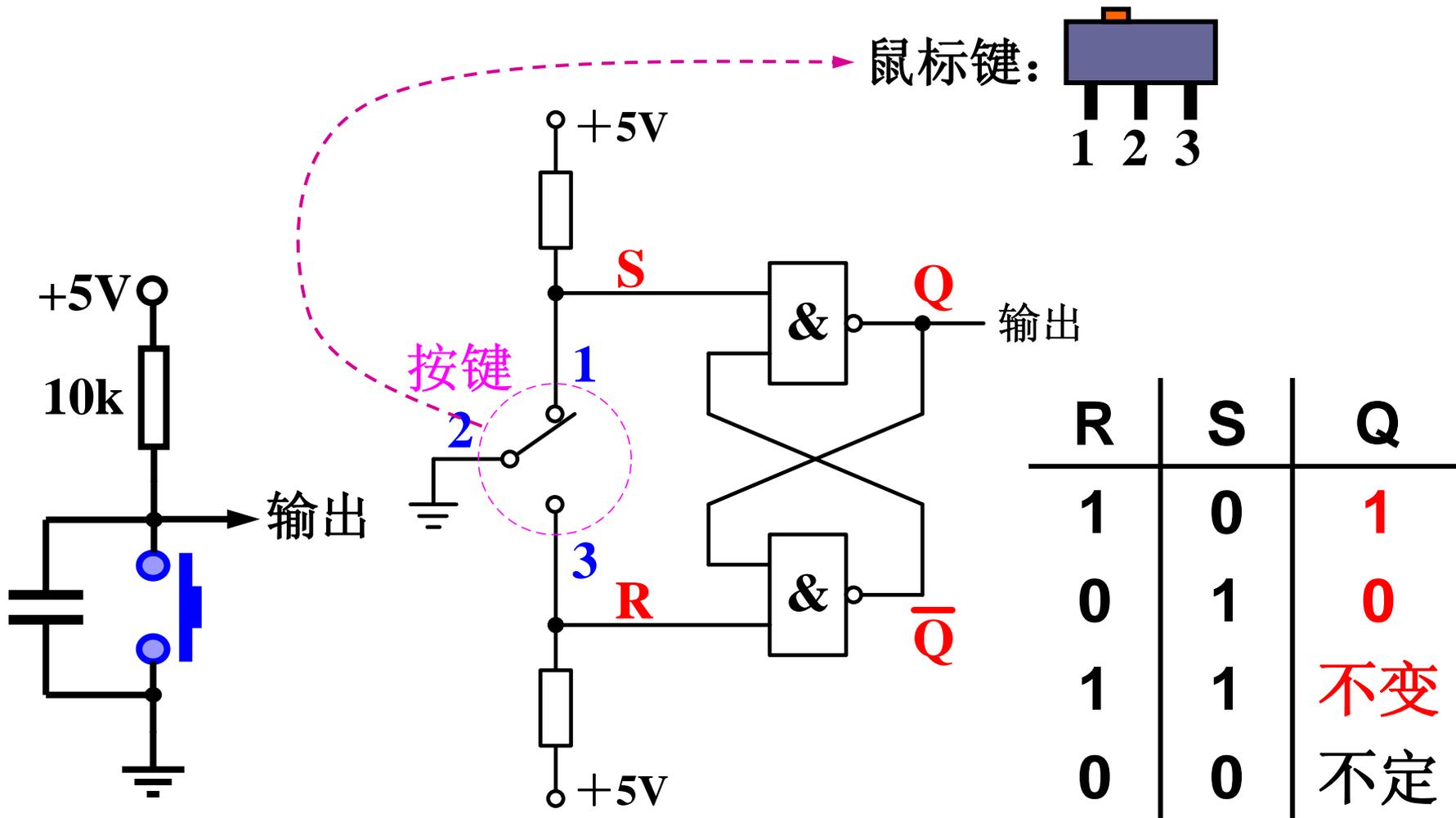


图8.12 按键的硬件消抖电路

## 8.1.2 键盘接口

### 一、非编码式键盘

#### 3. 确定键值

行寄存器

列寄存器

$(FFH - \text{行号}) \times 16 + \text{列寄存器}$

查表值 → 键值

#### 4. 键盘扫描及译码程序：P356~P358

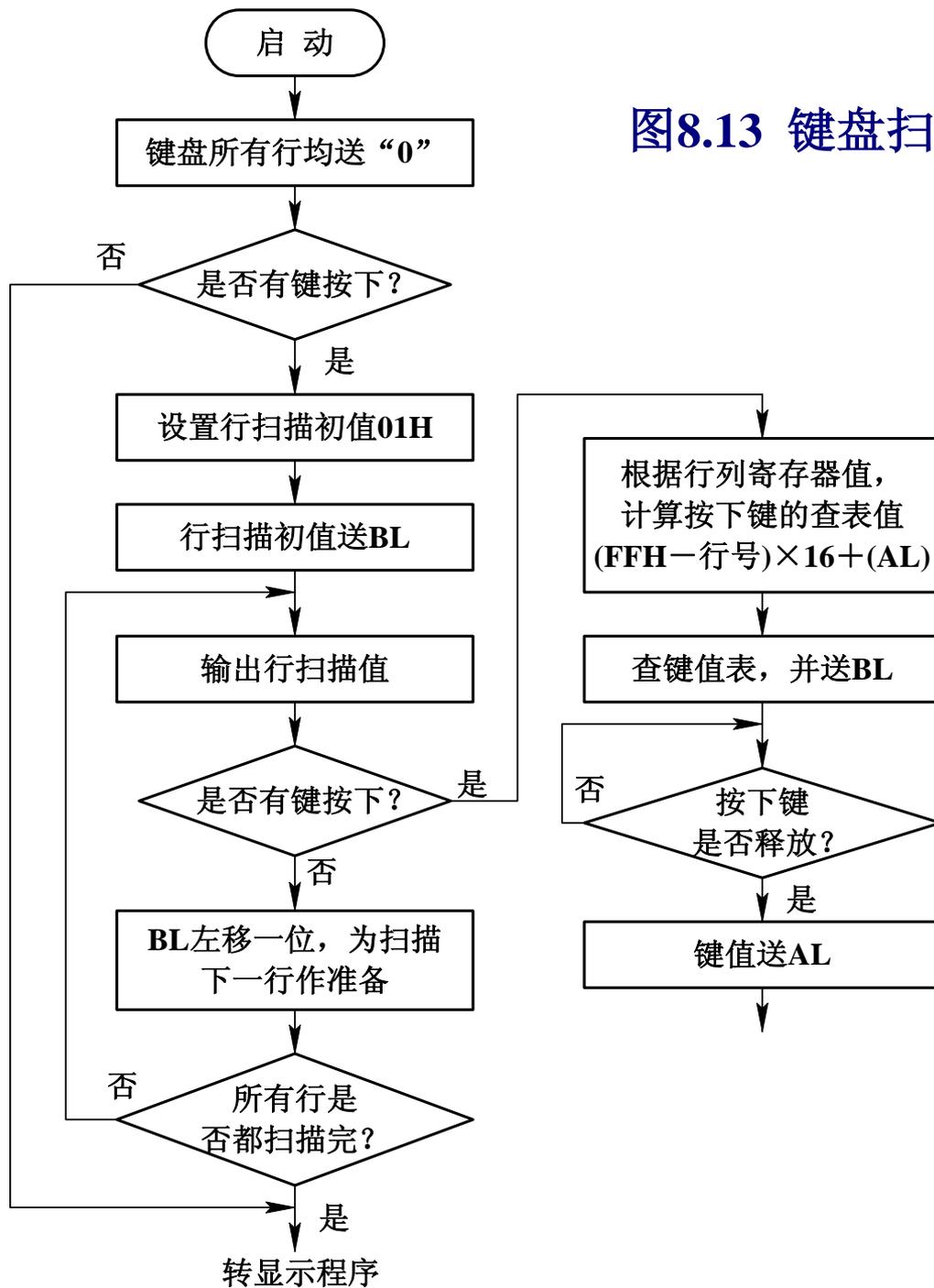


图8.13 键盘扫描及译码程序流程图

图8.11

```

;键盘扫描程序
DECKY:  MOV AL, 3FH      ;0  0  1  1  1  1  1  1
        MOV DX, DIGLH  ;行
        OUT DX, AL     ;行选输出, 行线全部置为低电平
        MOV DX, KBSEL  ;列
        IN AL, DX      ;列选输入
        AND AL, 1FH
        CMP AL, 1FH    ;判有无键闭合
        JZ DISUP       ;无键闭合转显示程序
        CALL D20MS     ;消除键抖动, D20MS为20ms延时子程序
KEYDN1: MOV BL, 01H    ;初始化行扫描值, BL:行寄存器值
        MOV DX, DIGLH
        MOV AL, BL
        OUT DX, AL     ;行扫描
        MOV DX, KBSEL
        IN AL, DX      ;该行是否有键闭合, AL:列寄存器值
        AND AL, 1FH    ;有则转译码程序
        CMP AL, 1FH
        JNZ KEYDN2
        SHL BL, 1     ;左移:01 02 04 08 10 20 → 40H
        MOV AL, 40H
        CMP AL, BL
        JNZ KEYDN1
        JMP DISUP     ;扫描完转显示 (扫描完一遍, 无键按下)

```

按下时  
消抖

```

KEYDN2: MOV    CH, 00H           ;键盘译码程序
KEYDN3: { DEC    CH           ;CH初值=FFH, 出循环后:CH=FFH-行号
        SHR    BL, 1         ;BL右移至其为0, 移位的次数即行号
        JNZ    KEYDN3
        SHL    CH, 1
        SHL    CH, 1
        SHL    CH, 1
        SHL    CH, 1 } ×16
        ADD    AL, CH         ;实现 (FFH-行号) × 16 + 列
        MOV    DI, OFFSET KYTBL ;端口值 键值表入口地址
KEYDN4: { CMP    AL, [DI]     ;寻找键值
        JZ     KEYDN5
        INC    DI
        INC    BL           ;表序号加1
        JMP    KEYDN4
KEYDN5: MOV    DX, KBSEL
KEYDN6: IN     AL, DX
        AND    AL, 1FH
        CMP    AL, 1FH
        JNZ    KEYDN6
        CALL   D20MS
        MOV    AL, BL
        .....

```

加之前为列寄存器值，加之后

实现 (FFH-行号) × 16 + 列

端口值 键值表入口地址

寻找键值

表序号加1

等待键释放

检测键是否释放  
未释放继续检测  
消除键抖动  
键值送AL

BL初值为0

释放时  
消抖

图8.11

## 8.1.2 键盘接口

### 二、编码式键盘

#### 1. 专用键盘接口芯片

- **8279**（并）
- **SSK814**（串）
- **HD7279A**（串）：可同时驱动**8**位共阴式数码管，可连接多达**64**键的键盘矩阵（有消抖电路）。

#### 2. **PC**键盘：采用专用控制器负责键盘扫描。 （键盘中有专用控制器）

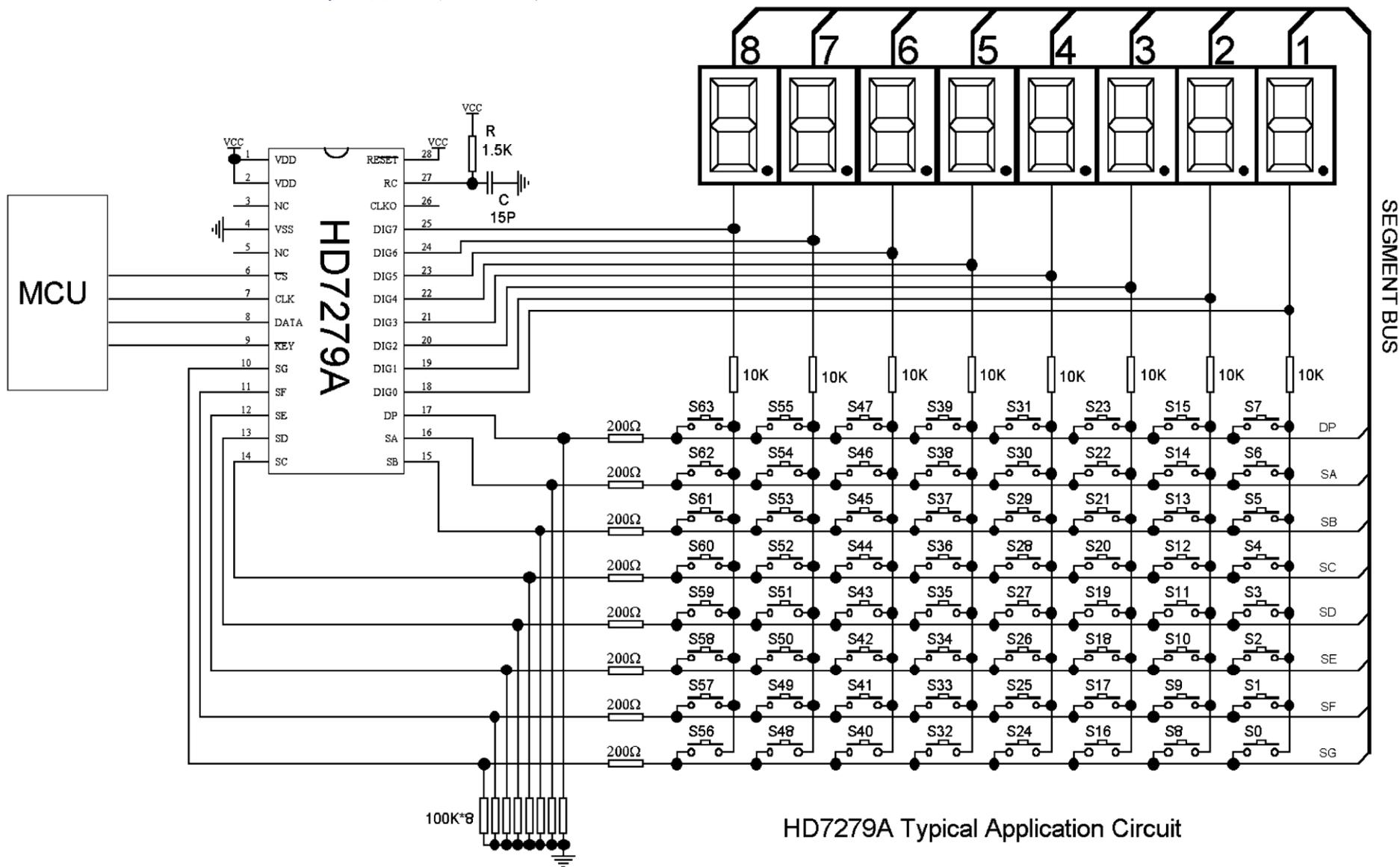
## HD7279A: 单片LED数码管显示和键盘接口芯片。

主要特点:

- 无需外围电路，直接驱动数码管；
- 多种译码方式，各显示位可分别控制译码方式(BCD、16进制)；
- 段寻址功能，可以单独控制显示段，便于使用独立LED；
- 具有左移、右移、闪烁、消隐等多种显示控制指令；
- 键盘部分具有去抖动功能，按键有效指示输出；
- 与CPU的接口采用SPI串行接口方式，并具有片选端可以多片联用。



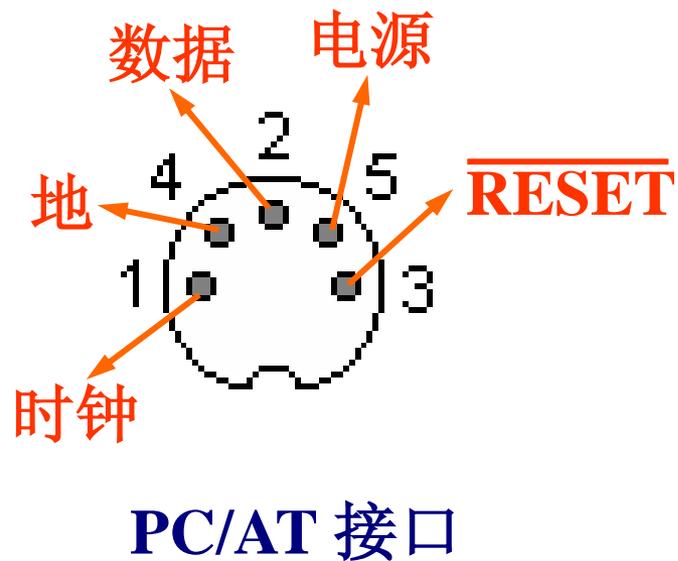
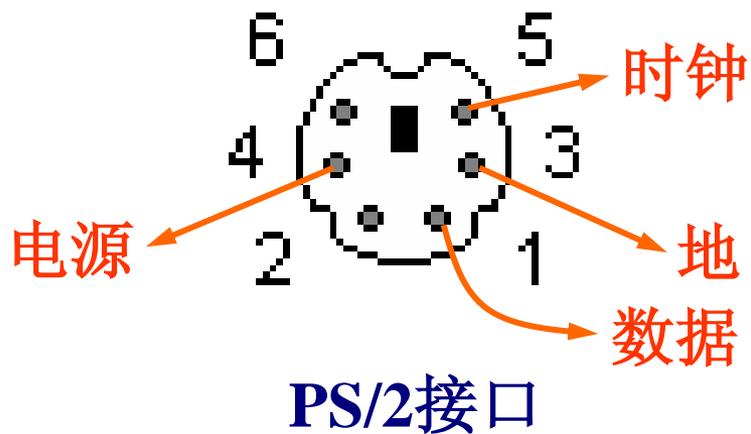
# HD7279的典型应用:



HD7279A Typical Application Circuit

## PC机键盘接口:

- 有PC/AT (大口)
- PS/2 (小口)
- USB
- 红外
- 无线



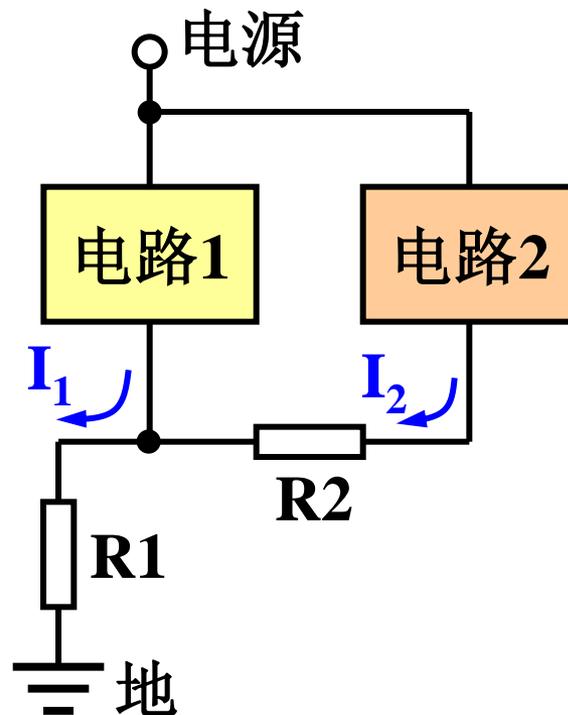
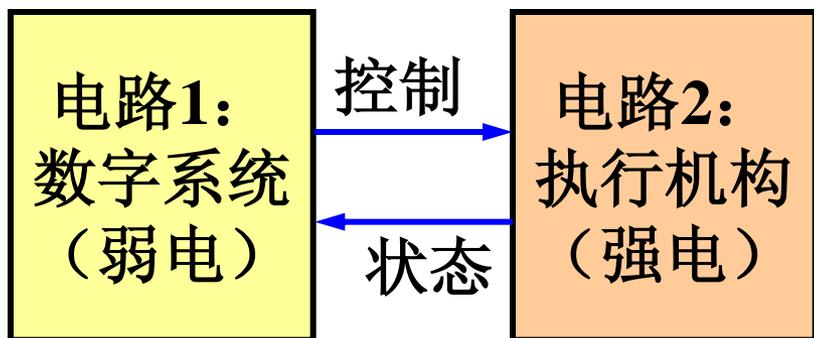
# 微机原理及接口技术

## 第8章 基于总线的I/O接口设计

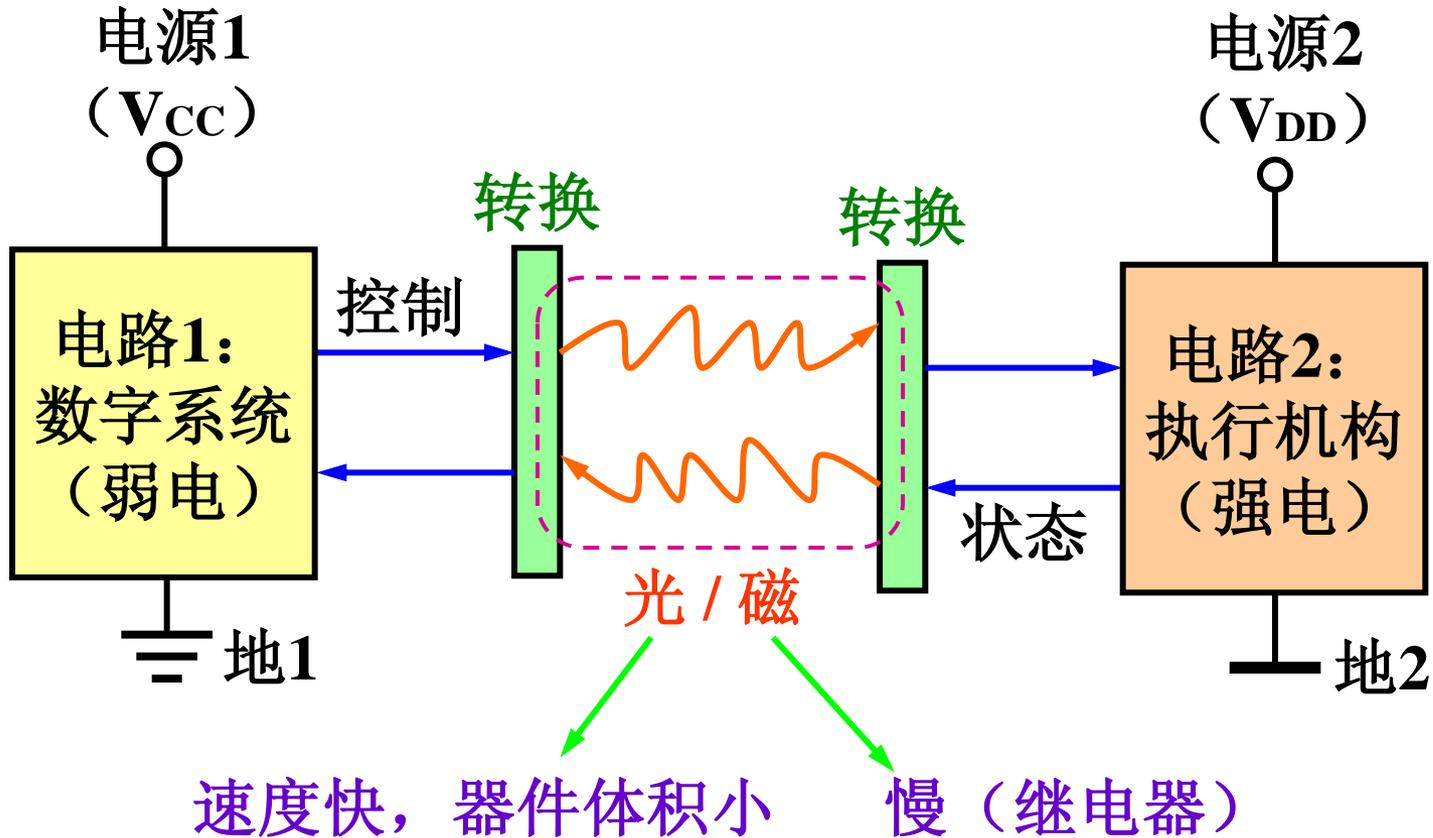
### 8.1 基于ISA总线的I/O接口设计

#### 8.1.3 光电隔离接口

## 8.1.3 光电隔离I/O接口



## 8.1.3 光电隔离I/O接口



## 8.1.3 光电隔离I/O接口

### 一、光电隔离器件

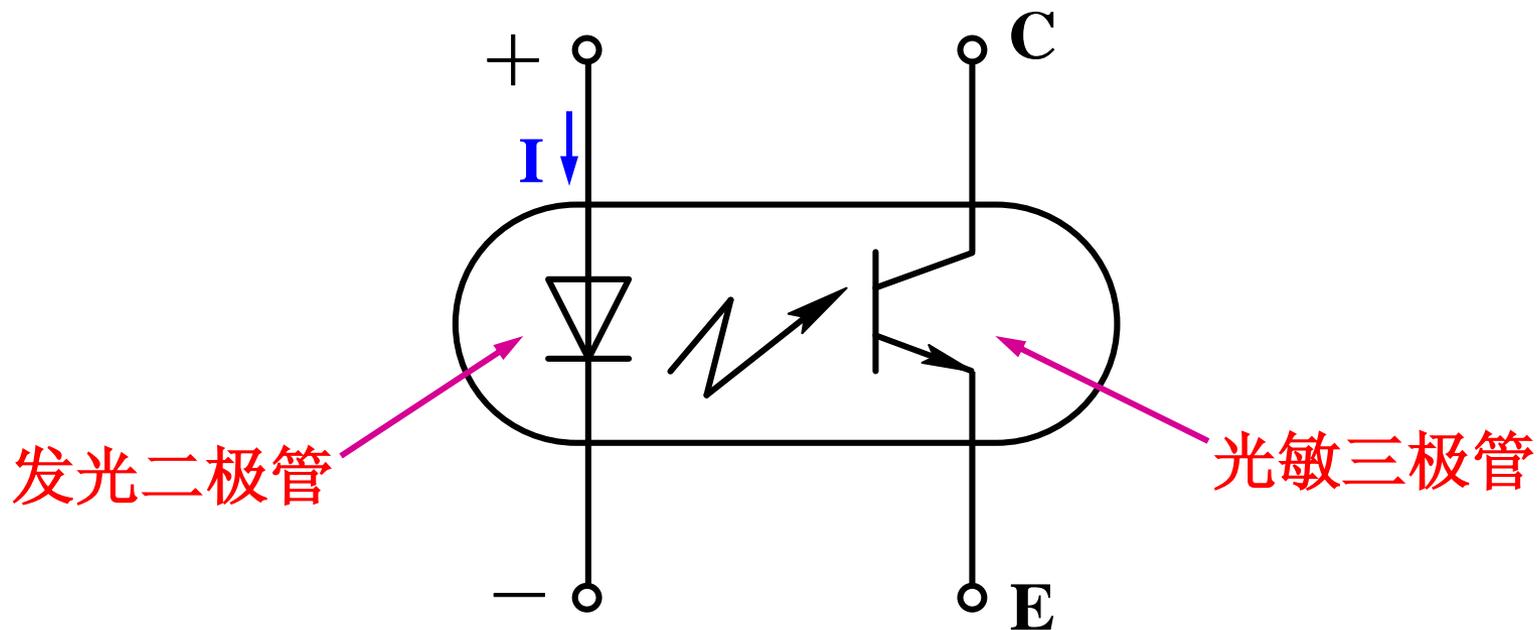
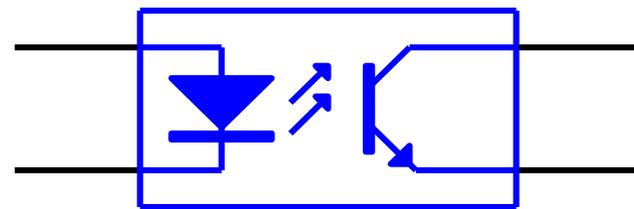
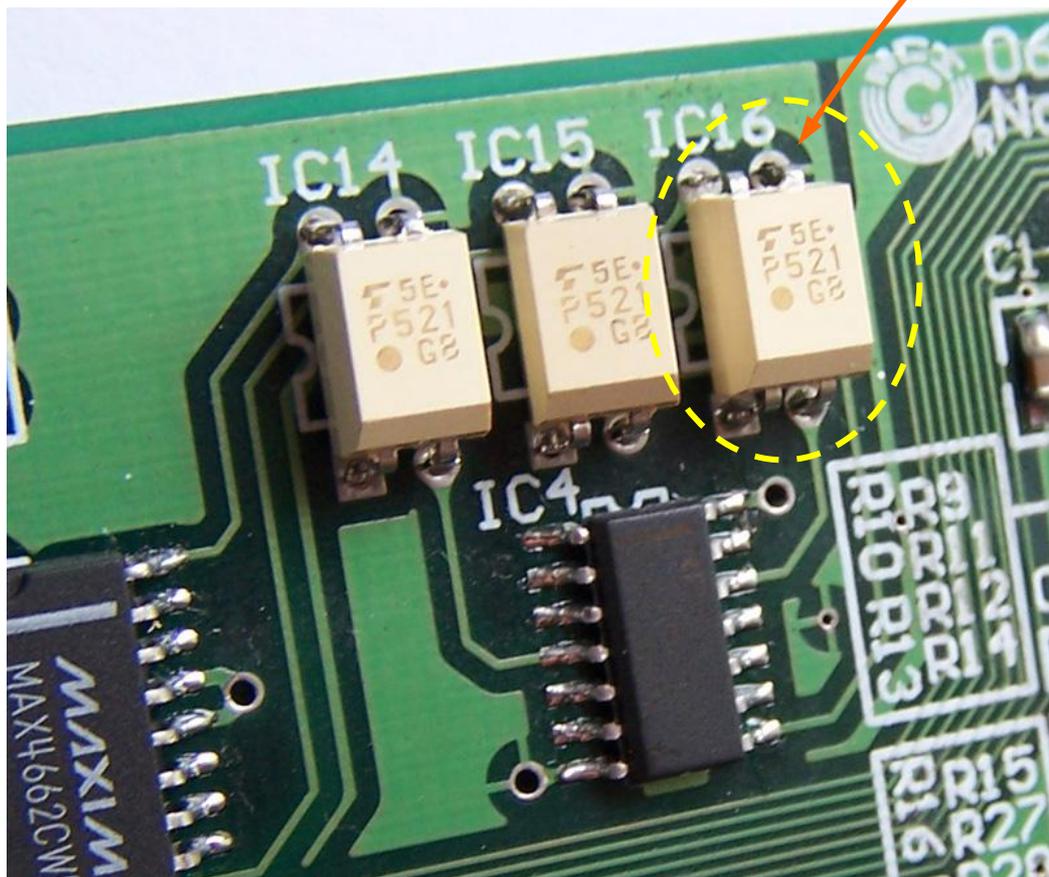


图8.17 光电隔离器件电原理图

## 8.1.3 光电隔离I/O接口

### 一、光电隔离器件



光电隔离器件及原理图符号

## 8.1.3 光电隔离I/O接口

### 二、光电隔离I/O接口电路

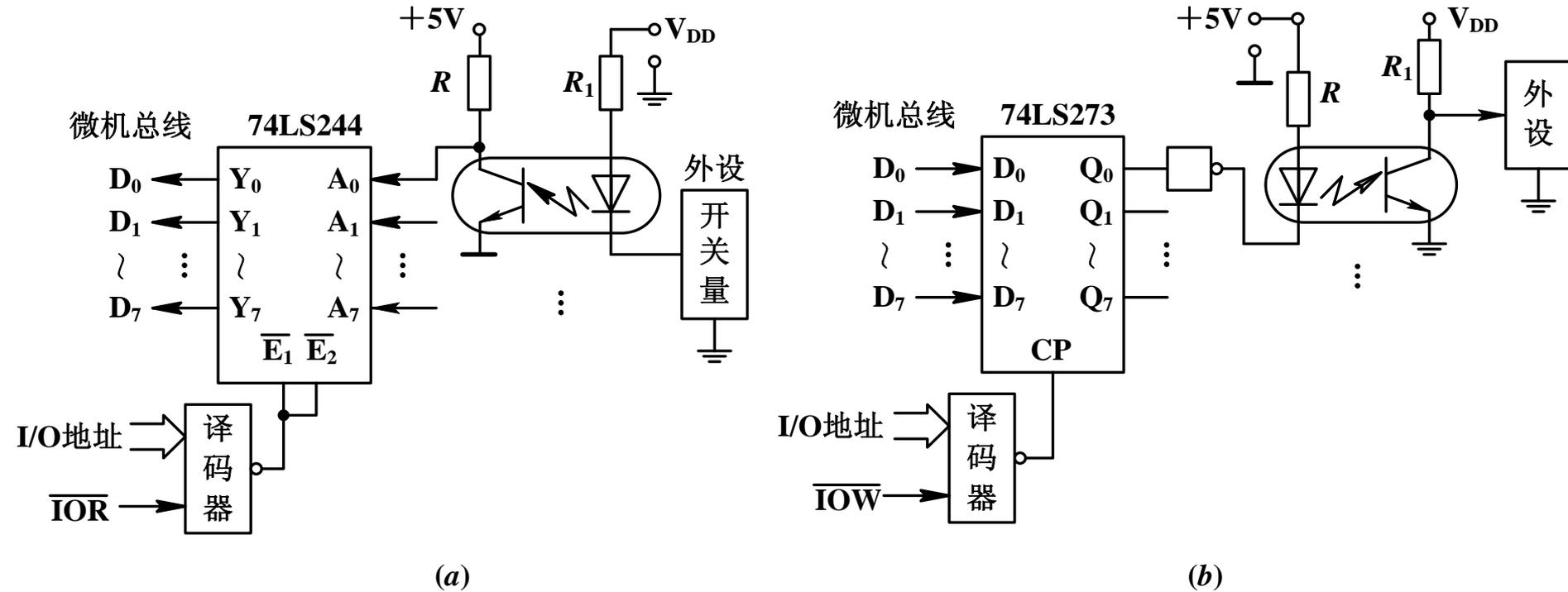


图8.18 光电隔离输入/输出接口电路

(a) 输入接口; (b) 输出接口

## 8.1.3 光电隔离I/O接口

### 三、光电隔离接口应用举例

常闭触点 常开触点

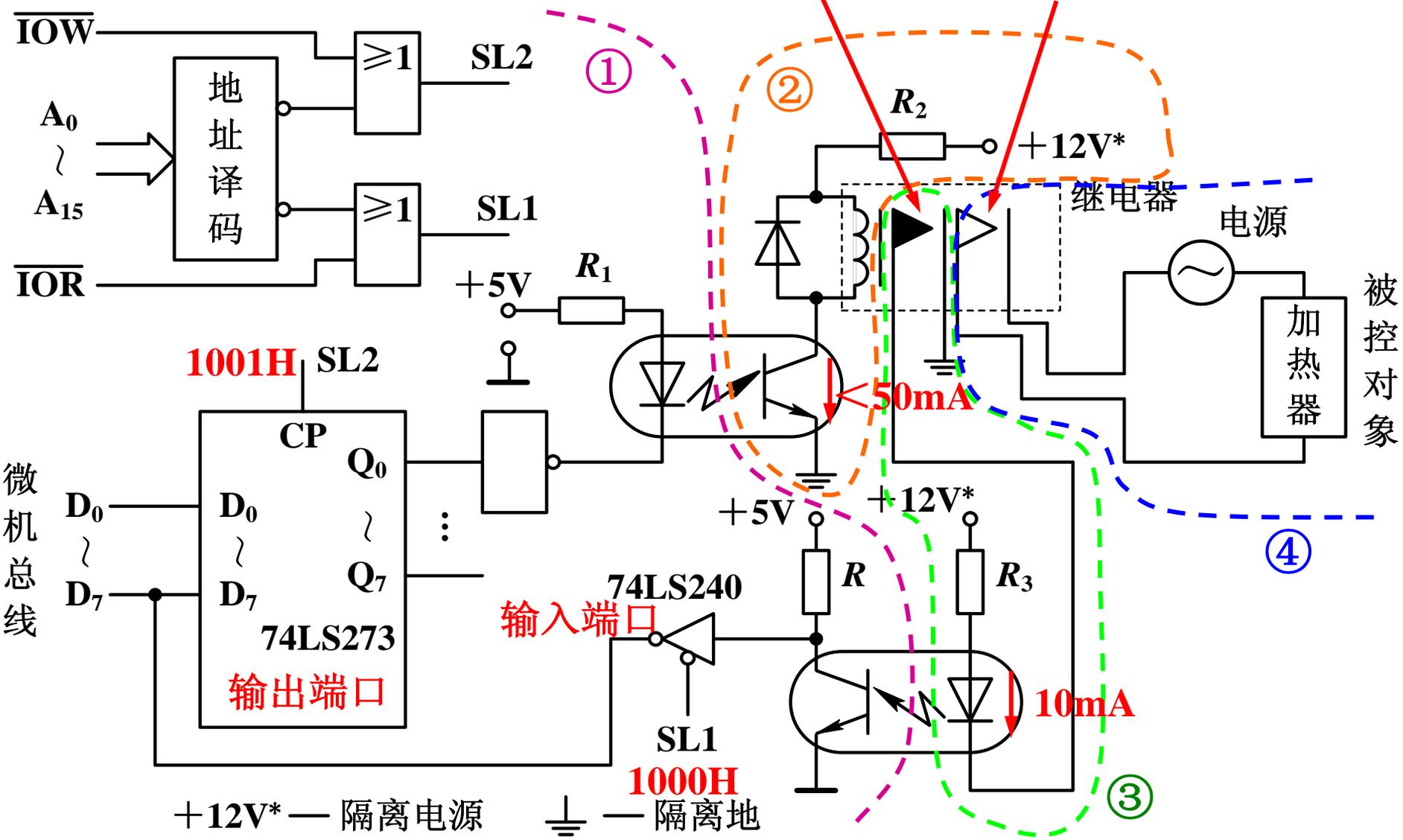


图8.19 光电隔离输入/输出接口电路实例

	<b>FLG</b>	<b>DB ?</b>	
<b>START:</b>	<b>MOV</b>	<b>DX, 1001H</b>	<b>; 锁存器端口地址</b>
	<b>MOV</b>	<b>AL, 01H</b>	
	<b>OUT</b>	<b>DX, AL</b>	<b>; 加热器通电</b>
	<b>CALL</b>	<b>DLY10ms</b>	<b>; DLY10ms为10ms延迟程序, 等待吸合</b>
	<b>MOV</b>	<b>DX, 1000H</b>	<b>; 三态门端口地址</b>
	<b>IN</b>	<b>AL, DX</b>	
	<b>AND</b>	<b>AL, 80H</b>	<b>; 判D7是否为0, 不为0 → Error</b>
	<b>JNZ</b>	<b>ERR</b>	
	<b>CALL</b>	<b>DLY10min</b>	<b>; 延时20分钟</b>
	<b>CALL</b>	<b>DLY10min</b>	
	<b>MOV</b>	<b>DX, 1001H</b>	<b>; 锁存器端口地址</b>
	<b>MOV</b>	<b>AL, 00H</b>	
	<b>OUT</b>	<b>DX, AL</b>	<b>; 加热器断电</b>
	<b>CALL</b>	<b>DLY10ms</b>	<b>; 等待继电器断开</b>
	<b>MOV</b>	<b>DX, 1000H</b>	<b>; 三态门端口地址</b>
	<b>IN</b>	<b>AL, DX</b>	
	<b>AND</b>	<b>AL, 80H</b>	
	<b>JZ</b>	<b>ERR</b>	
	<b>CALL</b>	<b>DLY10min</b>	<b>; 延时10分钟</b>
	<b>JMP</b>	<b>START</b>	
<b>ERR:</b>	<b>MOV</b>	<b>FLG, 0AAH</b>	
	<b>NOP</b>		
	<b>HLT</b>		

## 应注意的问题:

- ① 速度: 几十~几百KHz
- ② 绝缘电压: 0.5~10 KV
- ③ 两边不共地、电源独立
- ④ 动作控制  $\longleftrightarrow$  反馈检测
- ⑤ R1 R2 R3: 限流电阻, 几百 $\Omega$   
R: 保护电阻, 几K $\Omega$

