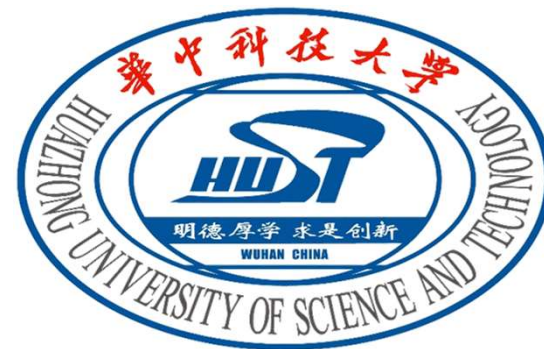


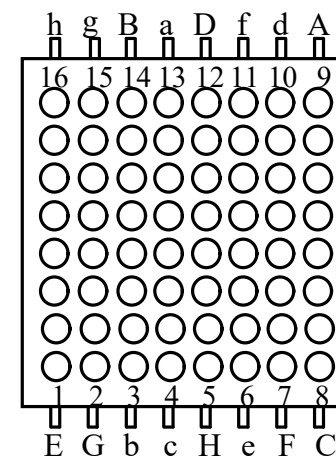
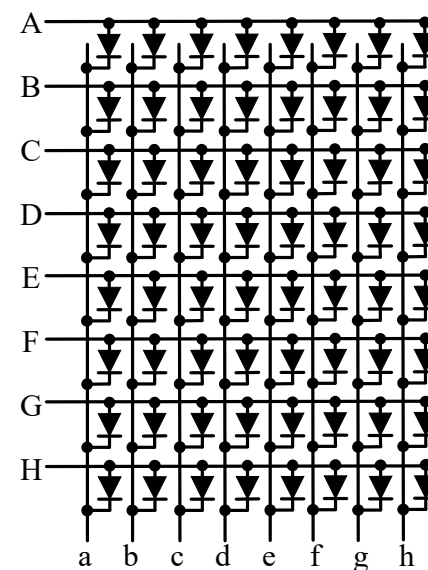
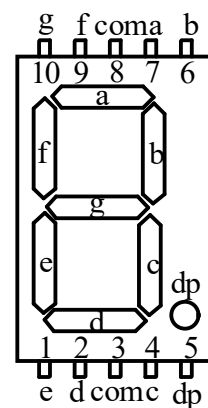
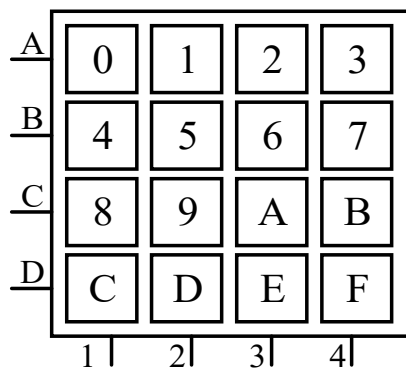
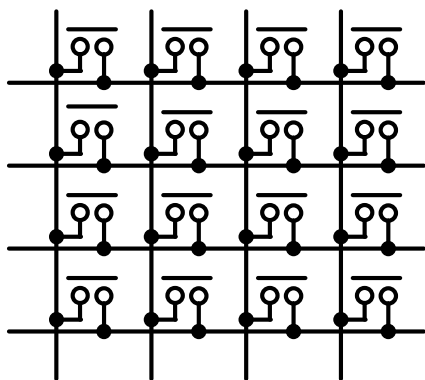
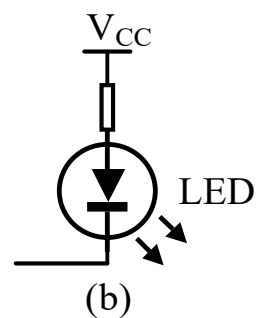
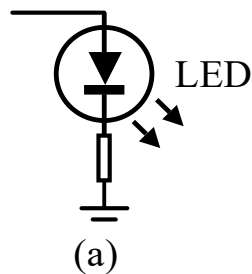
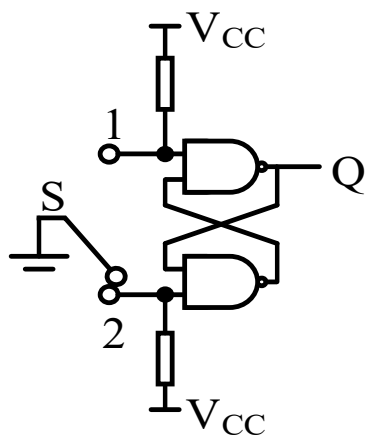
微机原理与接口技术

常用并行数字IO设备接口

华中科技大学 左冬红



常用并行数字IO设备



常用并行数字IO设备特点

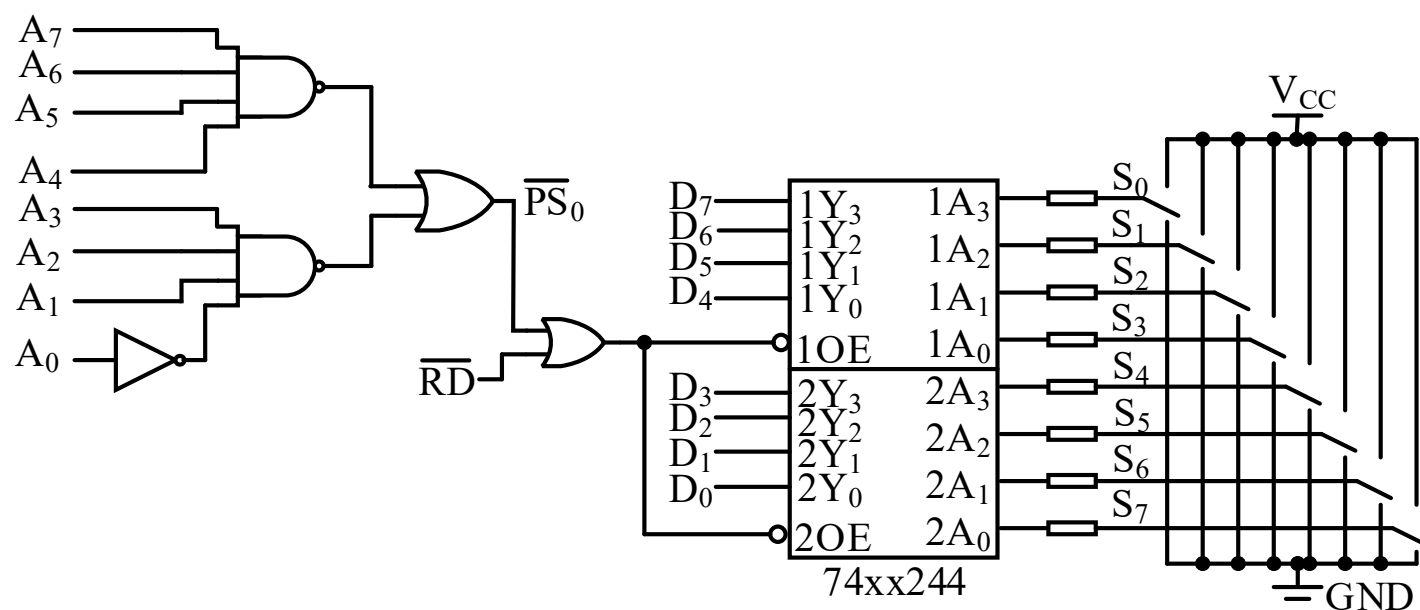
输出设备不具有数据保持功能

输入设备不具有数据缓冲功能

数据缓冲、锁存都必须由接口电路实现

独立开关接口电路示例

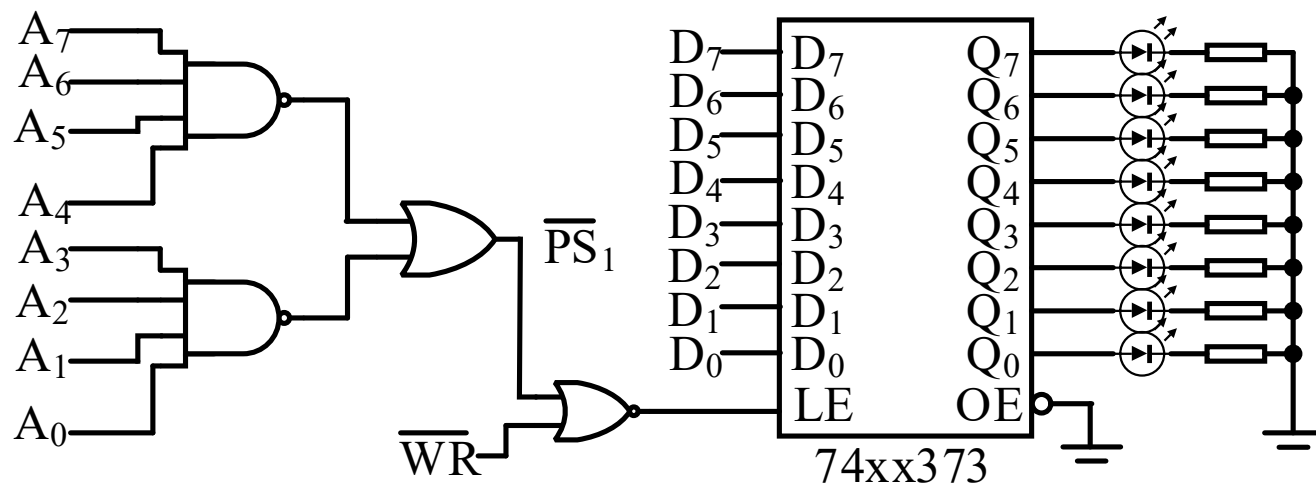
已知某计算机系统具有8位地址总线A7~0、8位数据总线D7~0，采用存储器映像IO寻址方式，要求为该计算机系统设计一个8位独立开关输入接口电路，且端口地址为0xfe，并编写控制程序段读入8位开关的状态。



```
Switch=Xil_In8(0xfe);
```

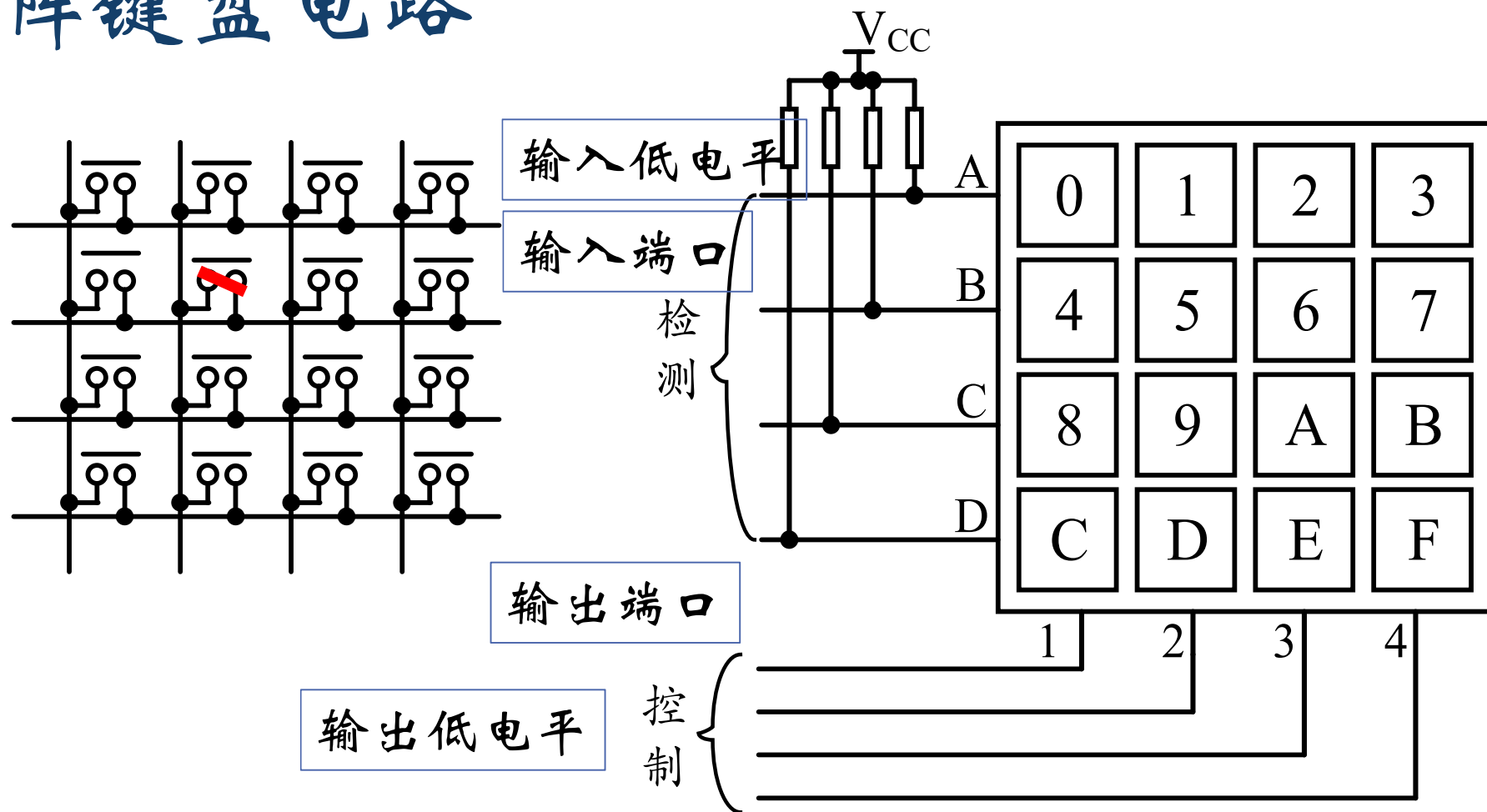
独立LED灯接口电路示例

已知某计算机系统具有8位地址总线A7~0、8位数据总线D7~0，采用存储器映像IO寻址方式，要求为该计算机系统设计一个8位发光二极管输出接口电路，且端口地址为0xff，并编写控制程序段将8个发光二极管点亮。



```
Xil_Out8( 0xff,0xff);
```

矩阵键盘电路



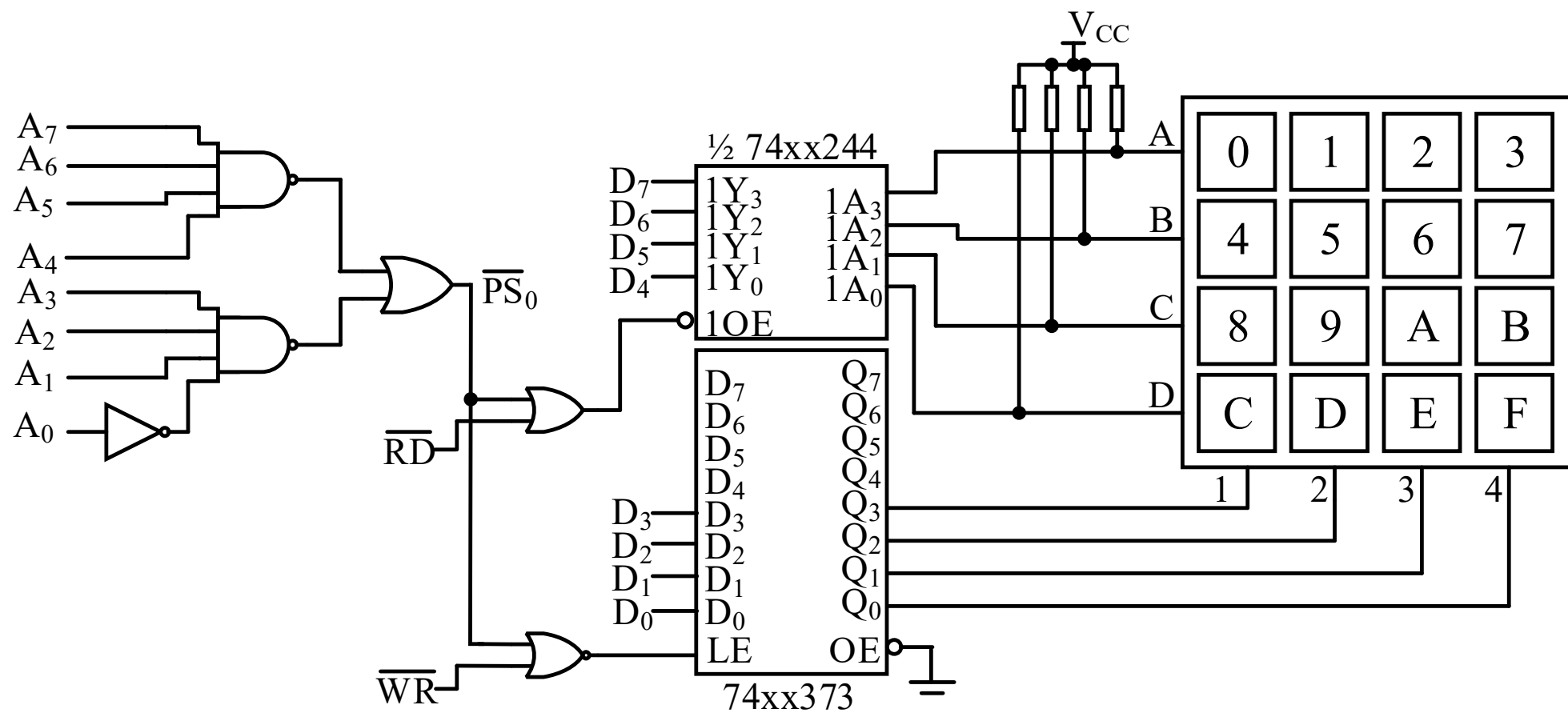
矩阵键盘接口电路示例

已知某计算机系统具有8位地址总线A7~0、8位数据总线D7~0，采用存储器映像IO寻址方式，要求为该计算机系统设计一个4×4矩阵键盘输入设备接口电路，且仅具有一个端口地址0xfe。编写控制程序识别按键并输出各个按键所表示的字符。

8位端口，4位输入，4位输出，输入输出由控制信号控制

端口地址0xfe分别与RD\WR信号译码控制输入、输出

矩阵键盘接口电路示例



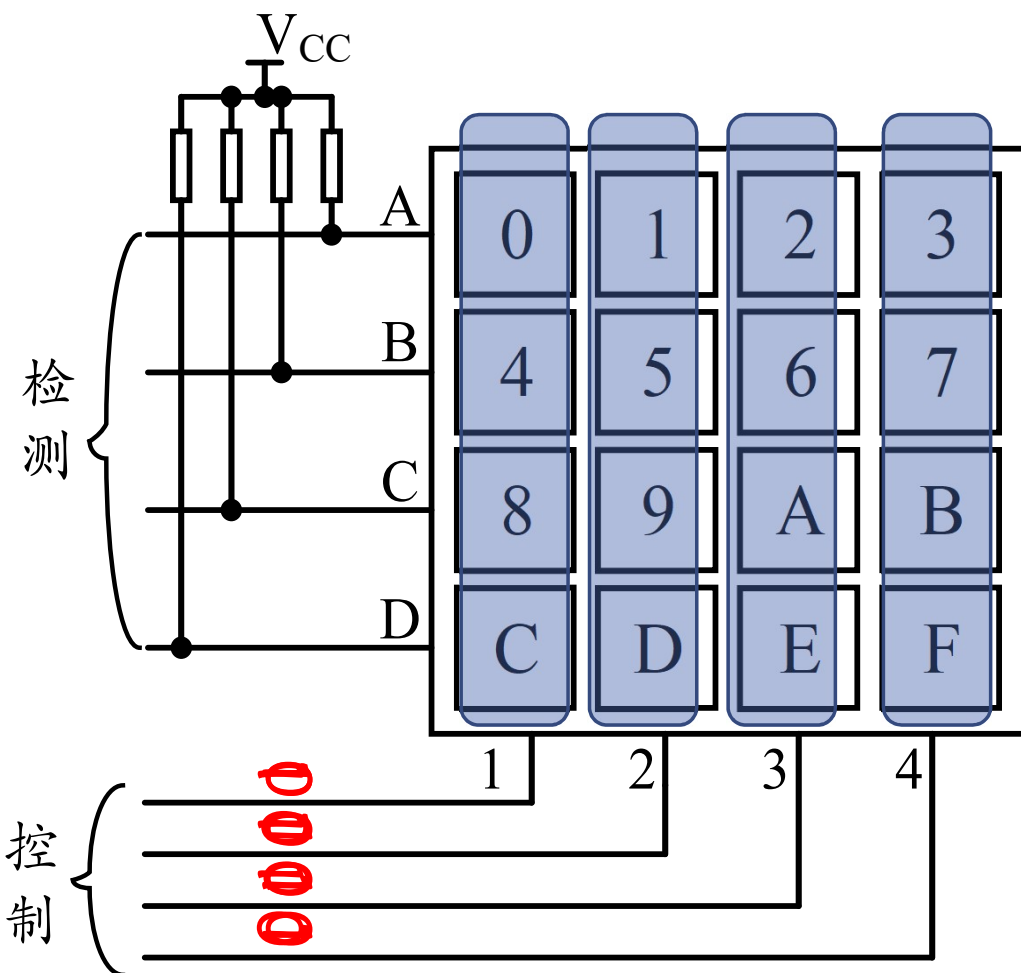
矩阵键盘按键识别

是否有键按下?

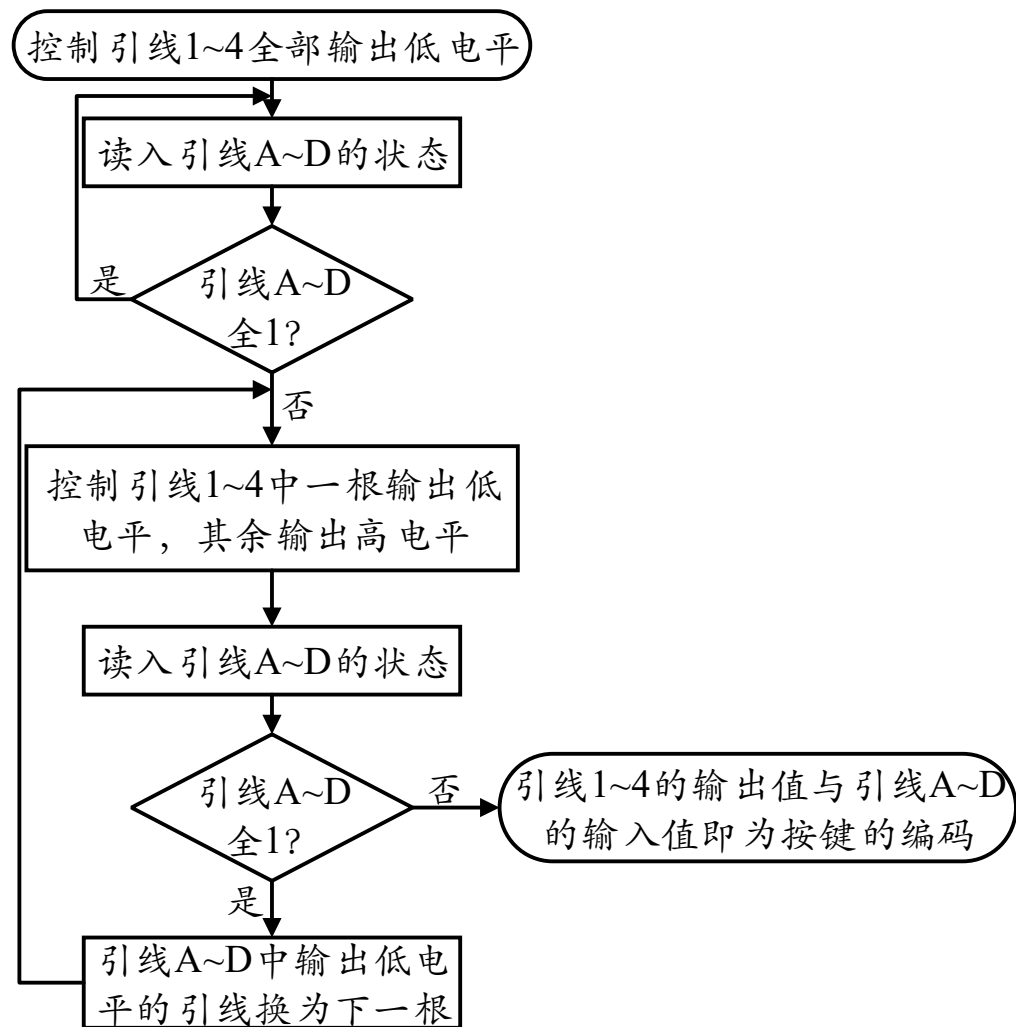
程序控制简单

哪个按键按下?

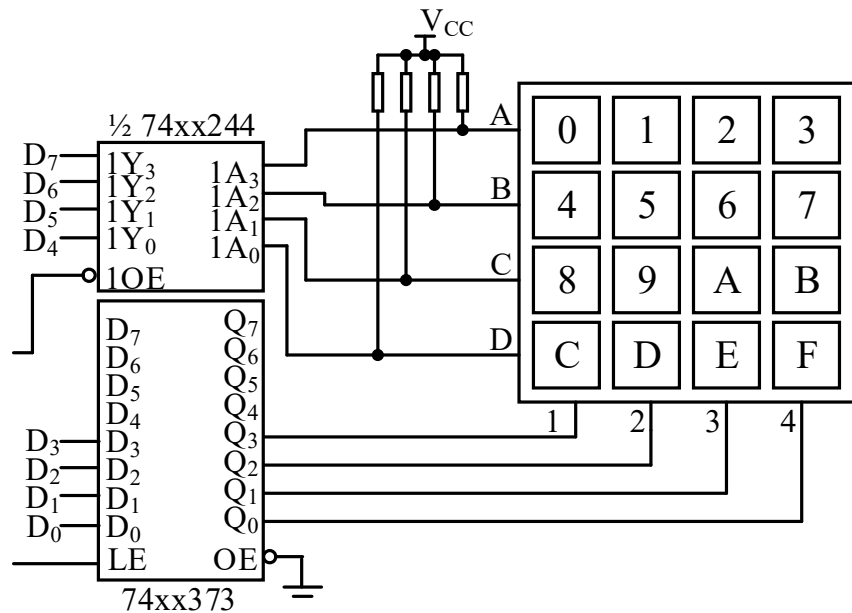
准确识别



矩阵键盘控制流程



控制程序



```

unsigned char KeyScancode;
char Row, Col=0xf7;
Xil_Out8(0xfe, 0x00); //
While ((Row=Xil_In8(0xfe)&0xf0) == 0xf0); //
Xil_Out8 (0xfe, Col); //
while ((Row=Xil_In8 (0xfe)&0xf0) == 0xf0) //
{
    Col=Col>>1; //
    Xil_Out8 (0xfe, Col);
}
KeyScancode=Row | (Col&0xf);
    
```

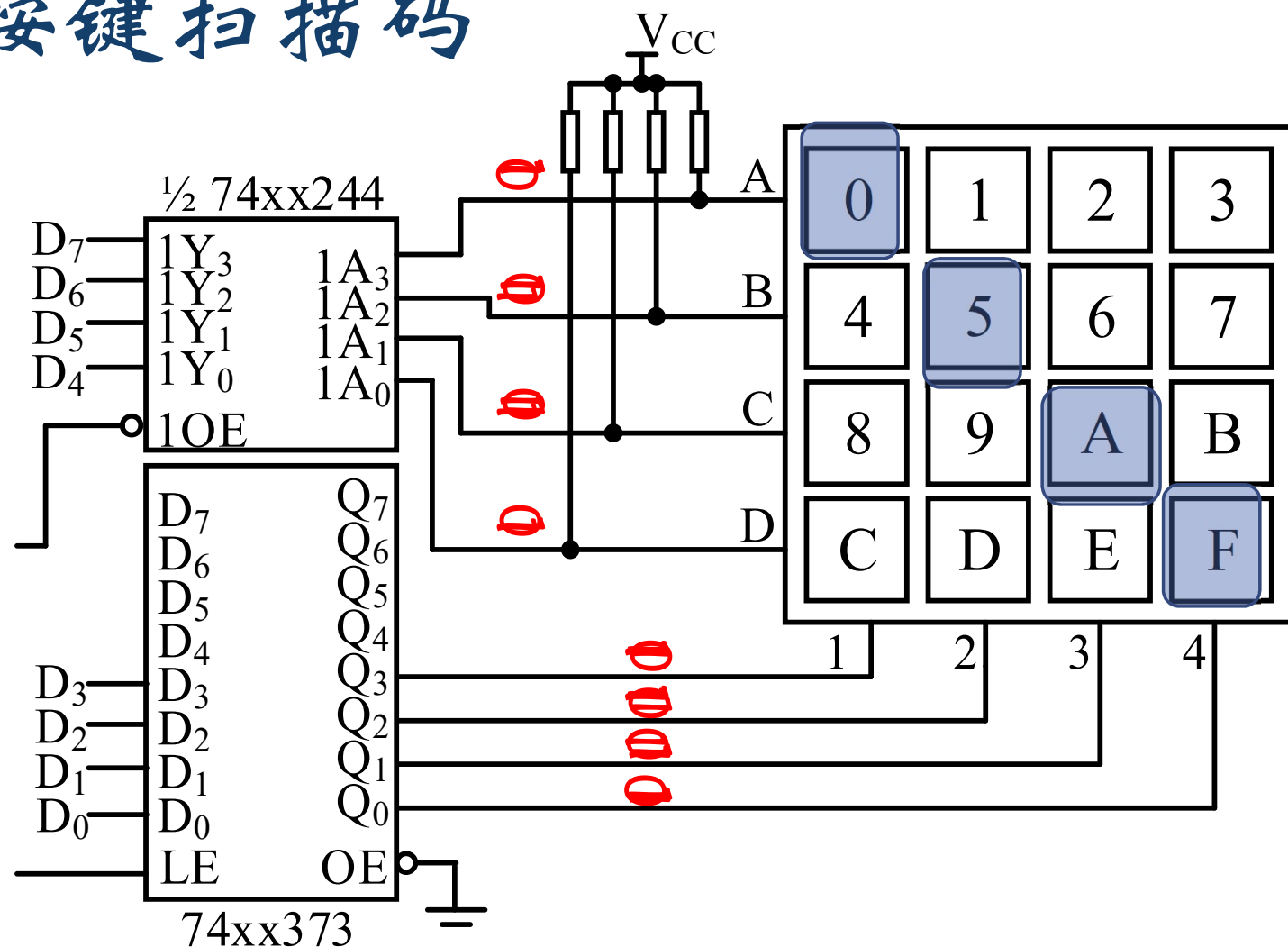
矩阵键盘按键扫描码

0的扫描码

5的扫描码

A的扫描码

F的扫描码



矩阵键盘按键扫描码

按键	引线A (D ₇)	引线B (D ₆)	引线C (D ₅)	引线D (D ₄)	引线1 (D ₃)	引线2 (D ₂)	引线3 (D ₁)	引线4 (D ₀)	编码
0	0	1	1	1	0	1	1	1	0x77
1	0	1	1	1	1	0	1	1	0x7b
2	0	1	1	1	1	1	0	1	0x7d
3	0	1	1	1	1	1	1	0	0x7e
4	1	0	1	1	0	1	1	1	0xb7
5	1	0	1	1	1	0	1	1	0xbb
6	1	0	1	1	1	1	0	1	0xbd
7	1	0	1	1	1	1	1	0	0xbe
8	1	1	0	1	0	1	1	1	0xd7
9	1	1	0	1	1	0	1	1	0xdb
A	1	1	0	1	1	1	0	1	0xdd
B	1	1	0	1	1	1	1	0	0xde
C	1	1	1	0	0	1	1	1	0xe7
D	1	1	1	0	1	0	1	1	0xeb
E	1	1	1	0	1	1	0	1	0xed
F	1	1	1	0	1	1	1	0	0xee

矩阵键盘按键键值

扫描码到数字

unsigned char

```
hex_table[16]={0x77,0x7b,0x7d,0x7e,0xb7,0xbb,0xbd,0xbe,  
0xd7,0xdb,0xdd,0xde,0xe7,0xeb,0xed,0xee}
```

扫描码到ASCII字符

unsigned char

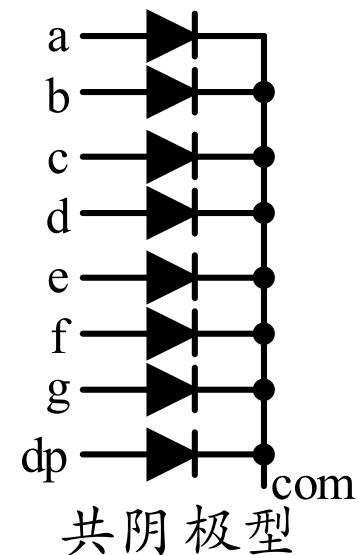
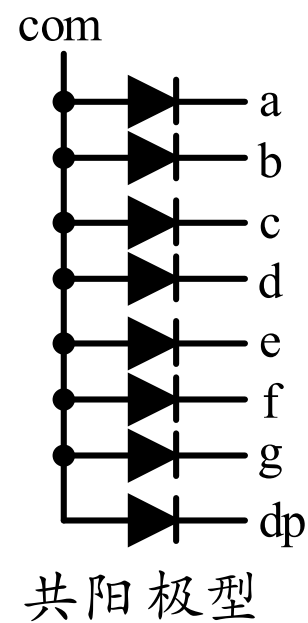
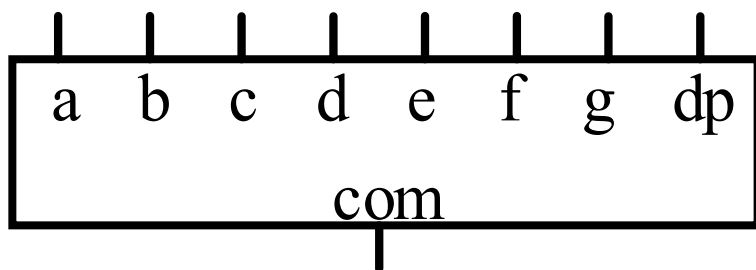
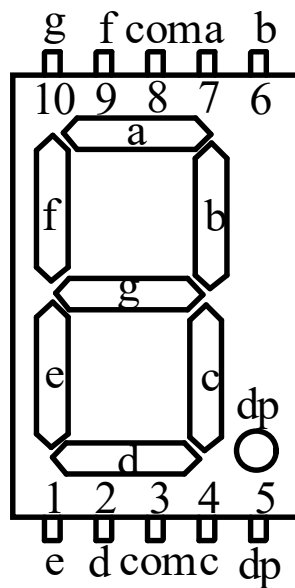
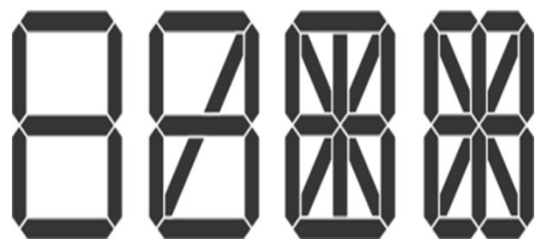
```
ascii_table[16][2]={0x77,0x30,0x7b,0x31,0x7d,0x32,0x7e,0x33,  
0xb7,0x34,0xbb,0x35,0xbd,0x36,0xbe,0x37,  
0xd7,0x38,0xdb,0x39,0xdd,0x41,0xde,0x42,  
0xe7,0x43,0xeb,0x44,0xed,0x45,0xee,0x46}
```

矩阵键盘按键键值获取程序段

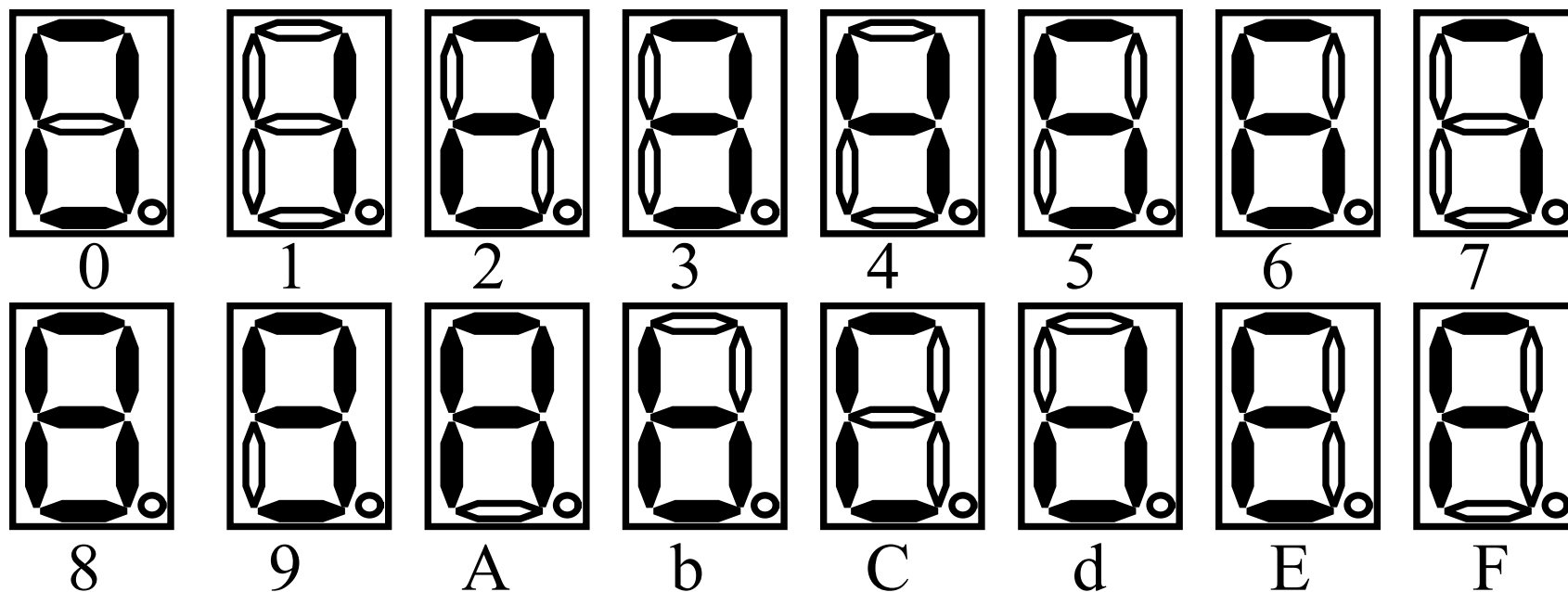
```
unsigned char Key_Hex, Key_Ascii;  
for(int i=0;i<16;i++) // 获取按键ASCII字符  
{  
    if (ascii_table[i][0]==KeyScancode)  
    {  
        Key_Ascii = ascii_table[i][1];  
        break;  
    }  
}
```

```
for(int i=0;i<16;i++) // 按键十六进制键值  
{  
    if (hex_table[i]==KeyScancode)  
    {  
        Key_Hex = i;  
        break;  
    }  
}
```

七段数码管



七段数码管十六进制数字字符字型

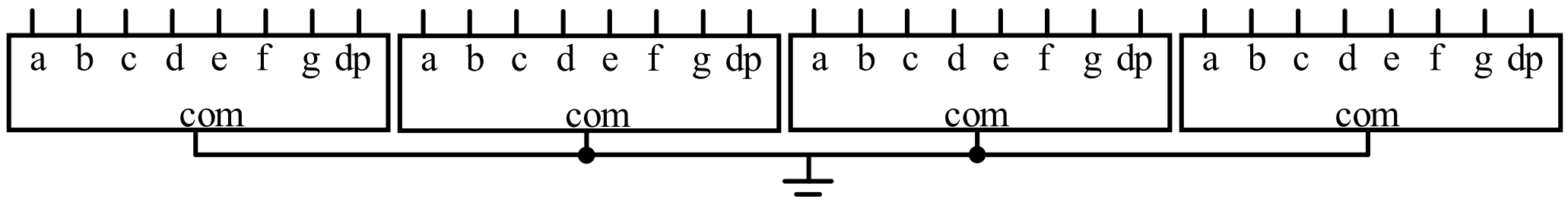


段码

字型对应的数字编码,编码值与电路连接相关

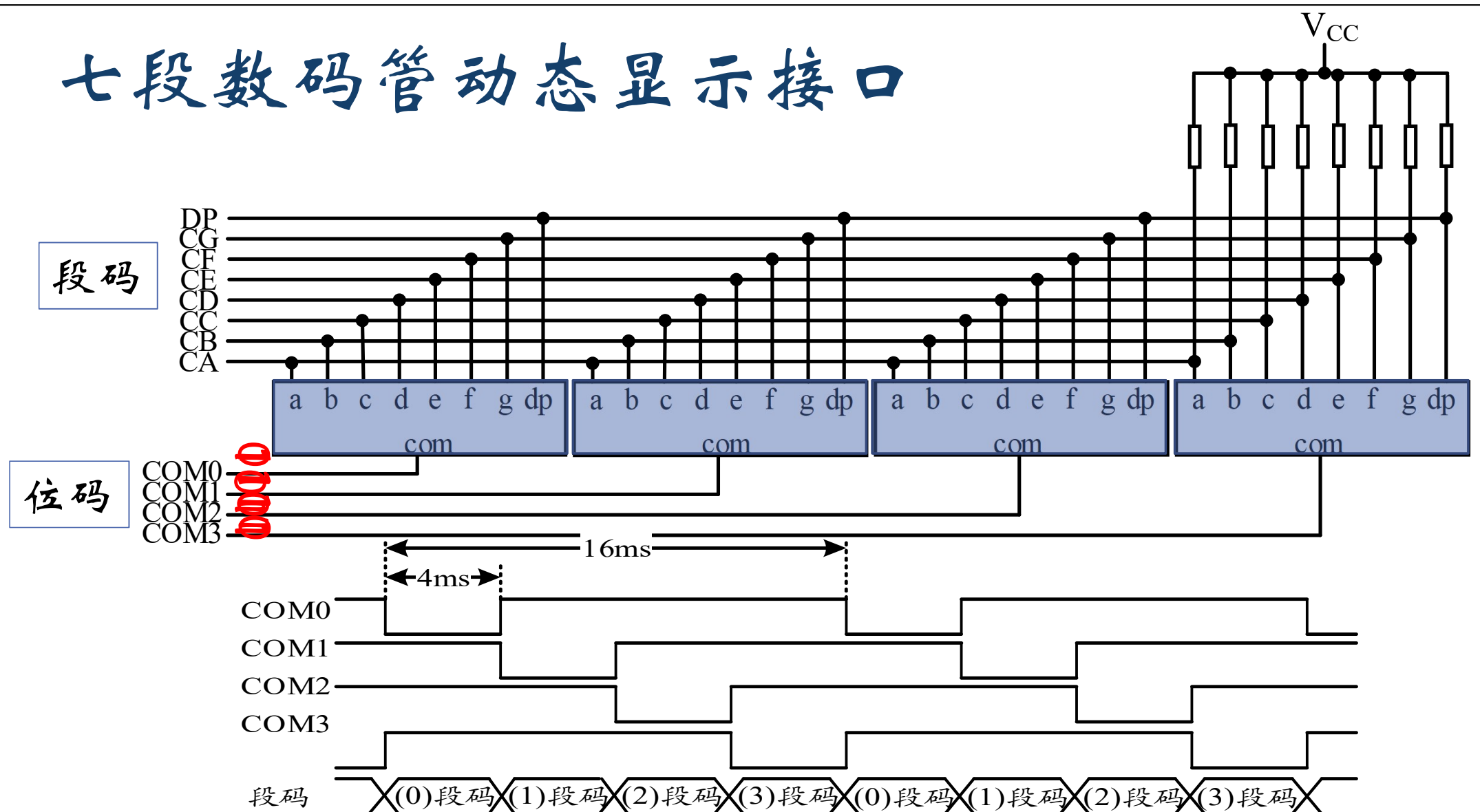
七段数码管静态显示接口

输出引脚多



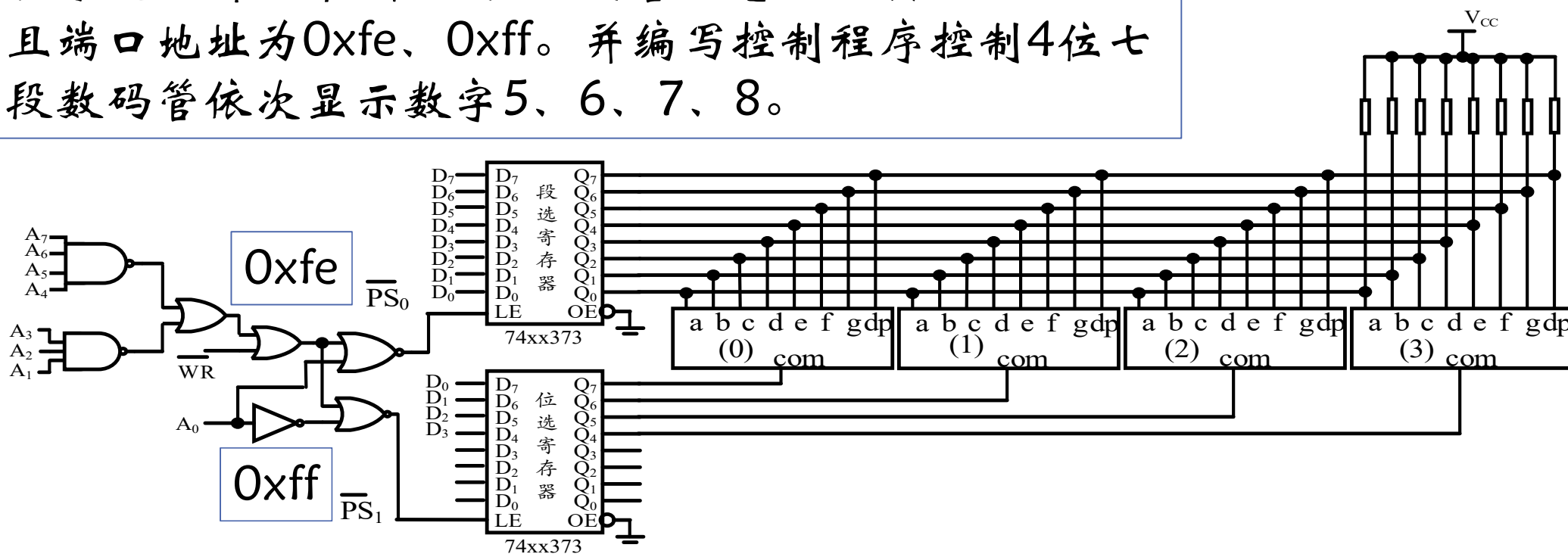
人眼具有“视觉暂留”效应，光的作用结束后，视觉形象并不立即消失

七段数码管动态显示接口



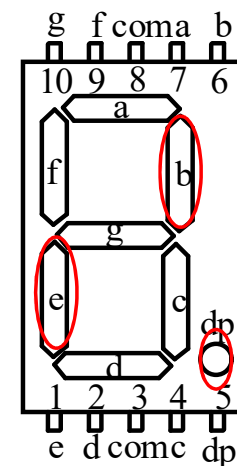
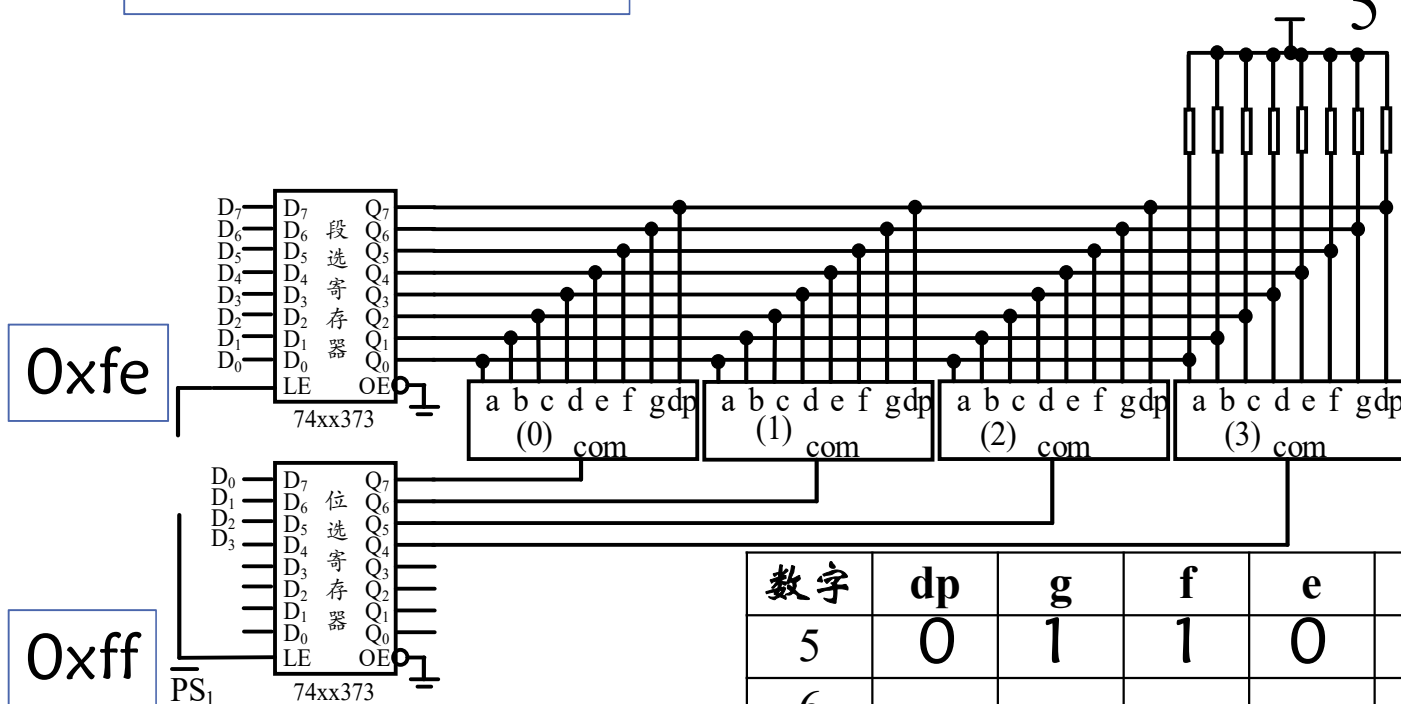
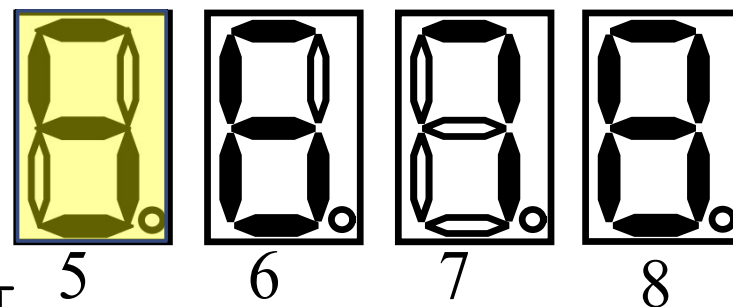
七段数码管应用示例

已知某计算机系统具有8位地址总线 $A_7 \sim A_0$ 、8位数据总线 $D_7 \sim D_0$ ，采用存储器映像IO寻址方式，要求为该计算机系统设计一个4位七段数码管动态显示接口电路，且端口地址为0xfe、0xff。并编写控制程序控制4位七段数码管依次显示数字5、6、7、8。



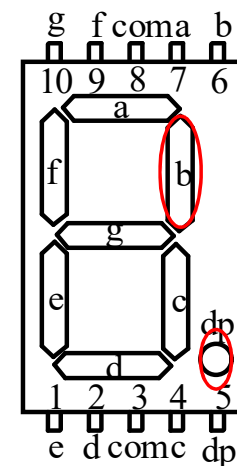
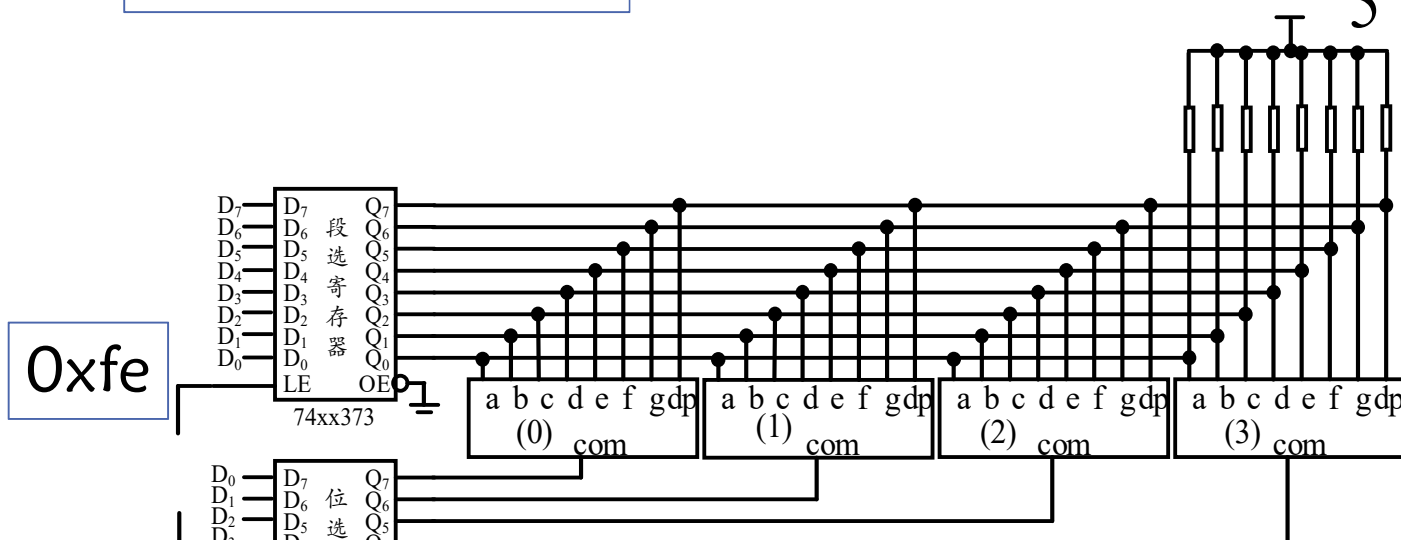
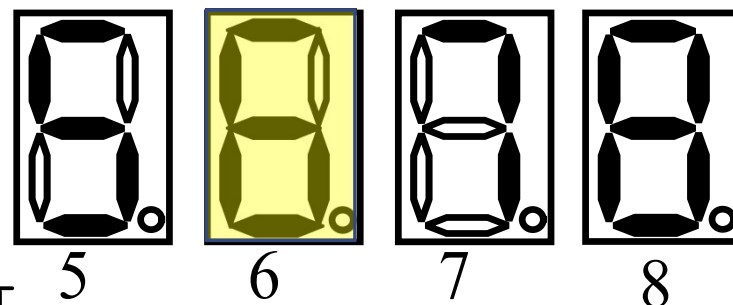
七段数码管应用示例

显示数字 5,6,7,8

[illegible]

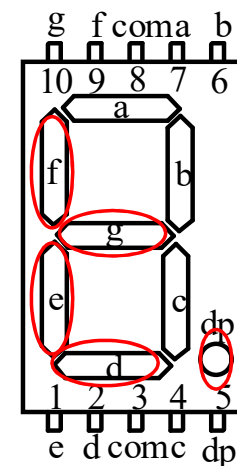
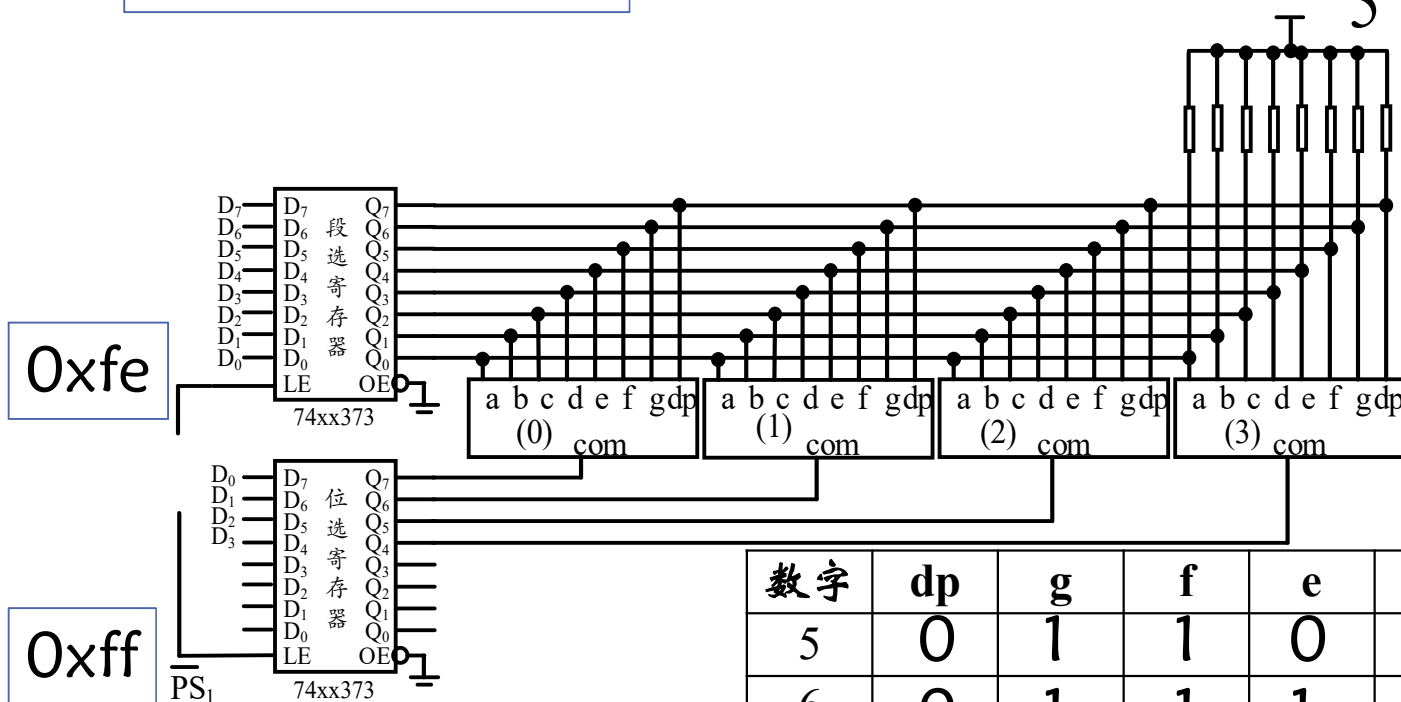
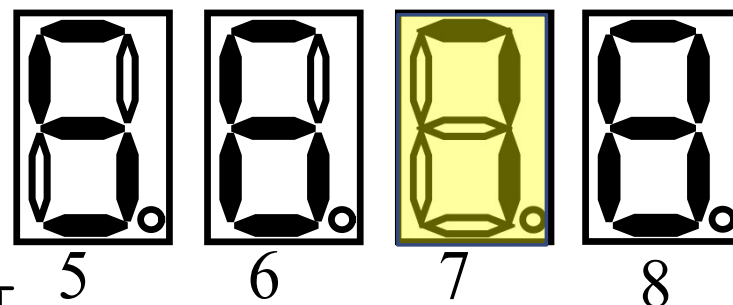
七段数码管应用示例

显示数字5,6,7,8

[illegible]

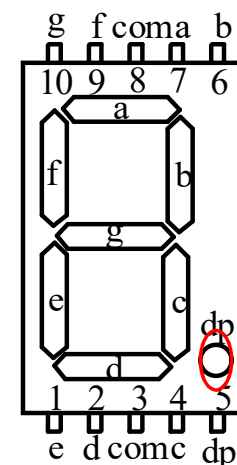
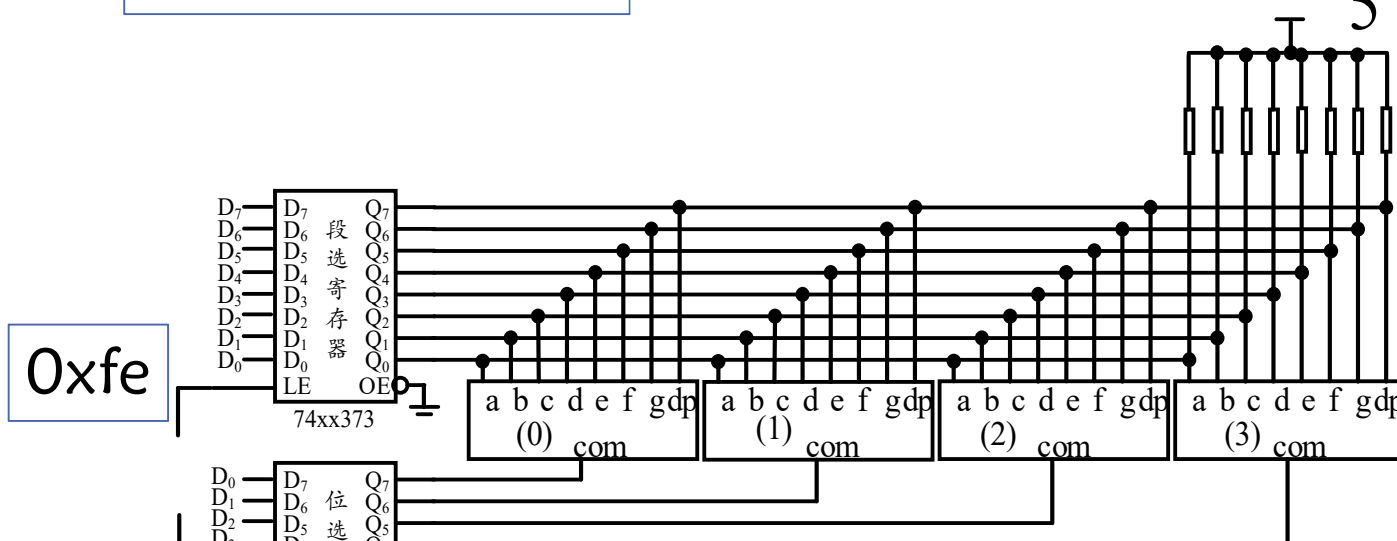
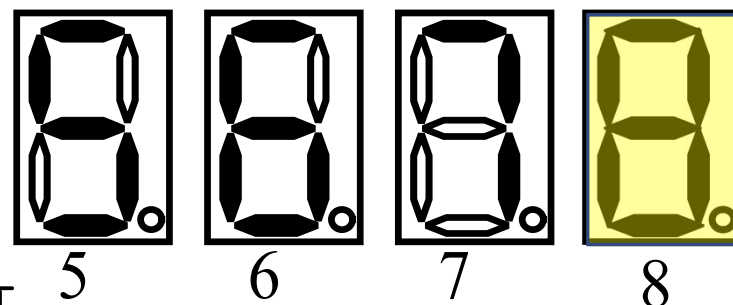
七段数码管应用示例

显示数字 5,6,7,8

[illegible]

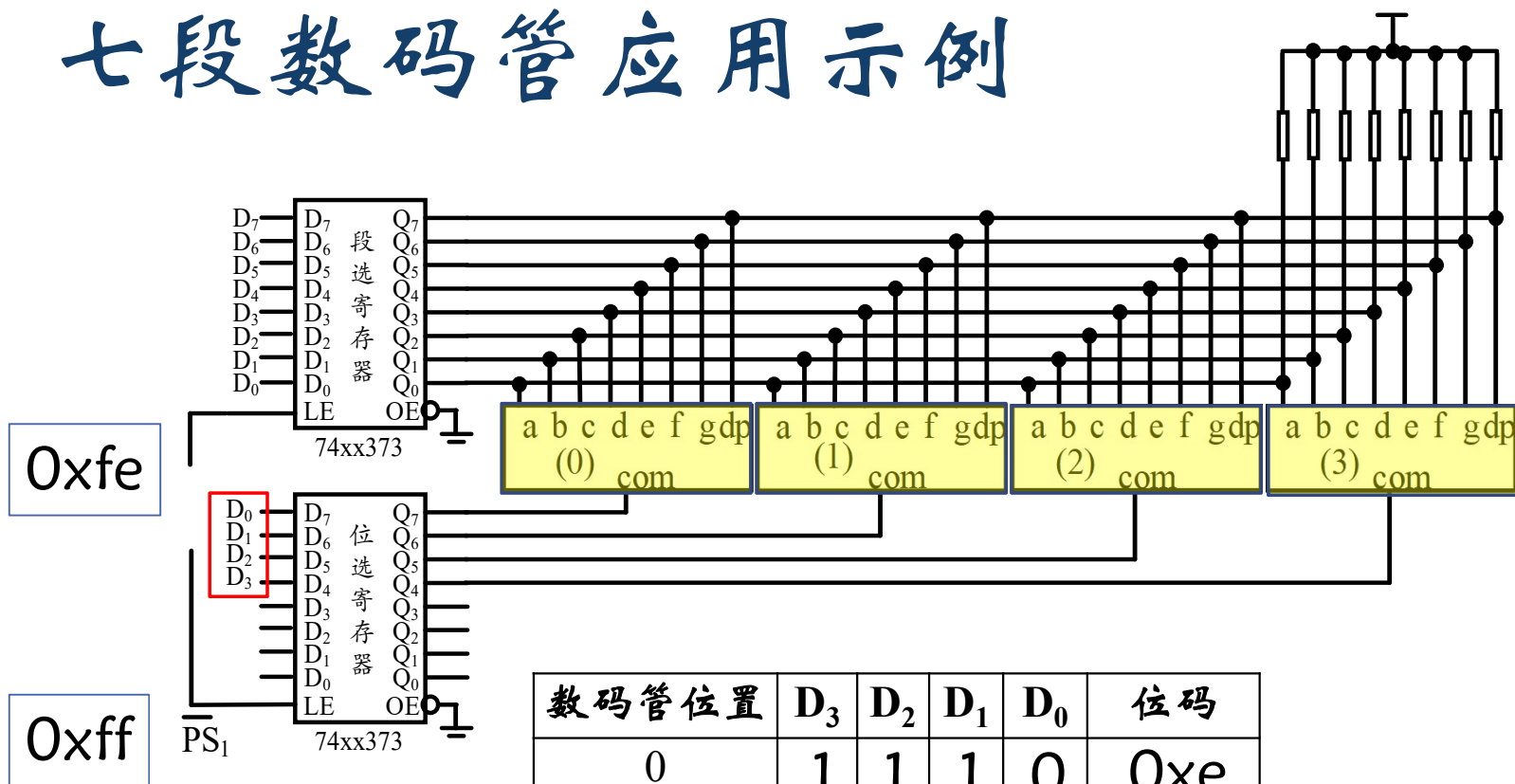
七段数码管应用示例

显示数字5,6,7,8



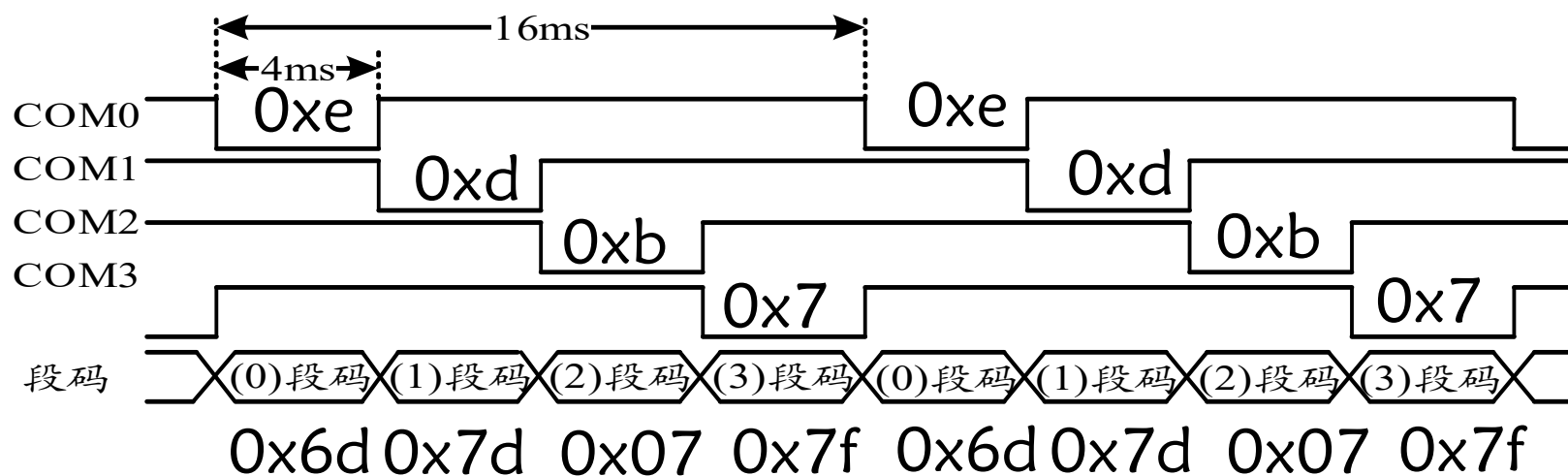
数字	dp	g	f	e	d	c	b	a	段码
5	0	1	1	0	1	1	0	1	0x6d
6	0	1	1	1	1	1	0	1	0x7d
7	0	0	0	0	0	1	1	1	0x07
8	0	1	1	1	1	1	1	1	0x7f

七段数码管应用示例



数码管位置	D ₃	D ₂	D ₁	D ₀	位码
0	1	1	1	0	0xe
1	1	1	0	1	0xd
2	1	0	1	1	0xb
3	0	1	1	1	0x7

七段数码管应用示例



七段数码管应用示例

显示缓冲区

```
unsigned char seg_code[4]={ 0x6d,0x7d,0x07,0x7f};  
unsigned char position=0x01;  
Xil_Out8(0xff,0xf);//  
while(1) //  
{   position=0x01;  
    for(int i=0;i<4;i++)  
    {      Xil_Out8(0xfe, seg_code[i]); //输出第i位的段码  
          Xil_Out8(0xff,~position); //输出第i位的位码  
          delay_ms(); //延时  
          position=position << 1;//位码指向下一个七段数码管  
    }  
}
```

软件延时

指令周期

指令执行需经历一定的时间

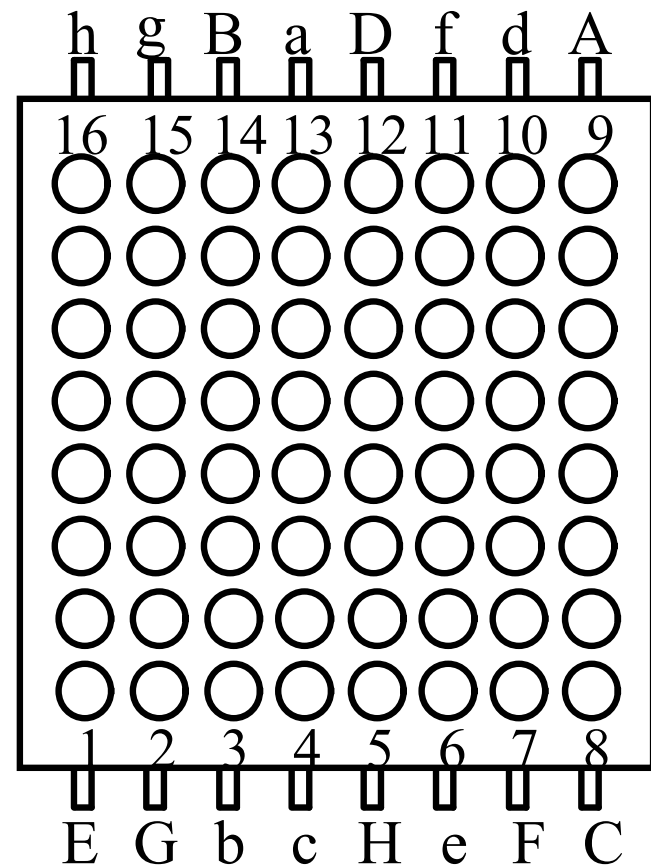
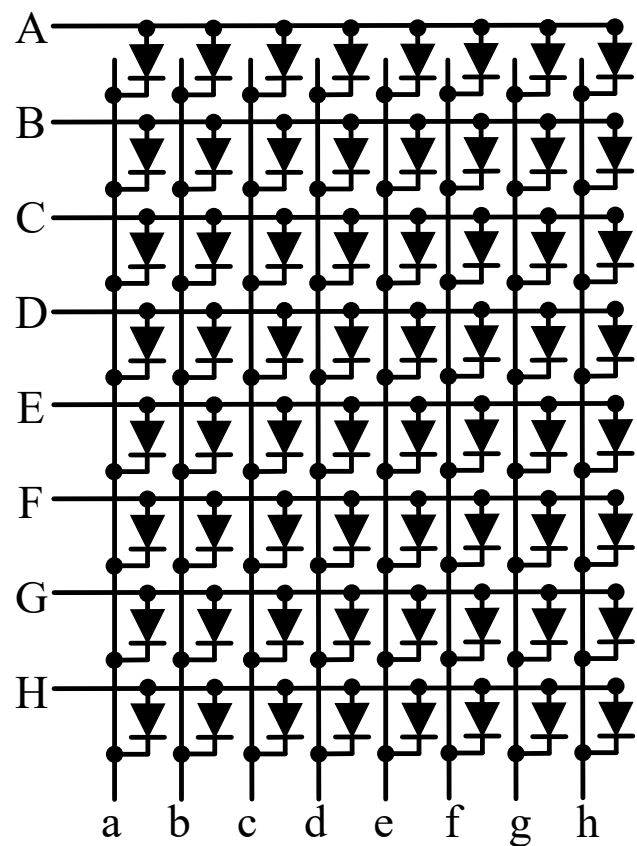
从取指令到存储结果的完整过程需经历的时间，

以微处理器时钟周期为单位

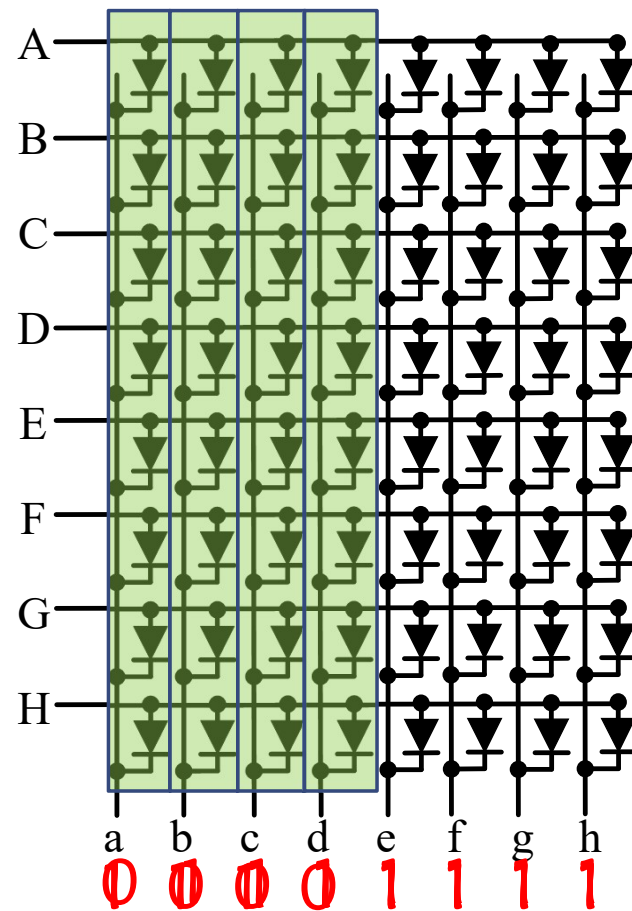
微处理器执行不对寄存器、外设以及存储器产生影响的指令

```
void delay_ms( int mil_sec)
{
for(int i=0;i<mil_sec*0x10000;i++);
}
```

LED点阵接口

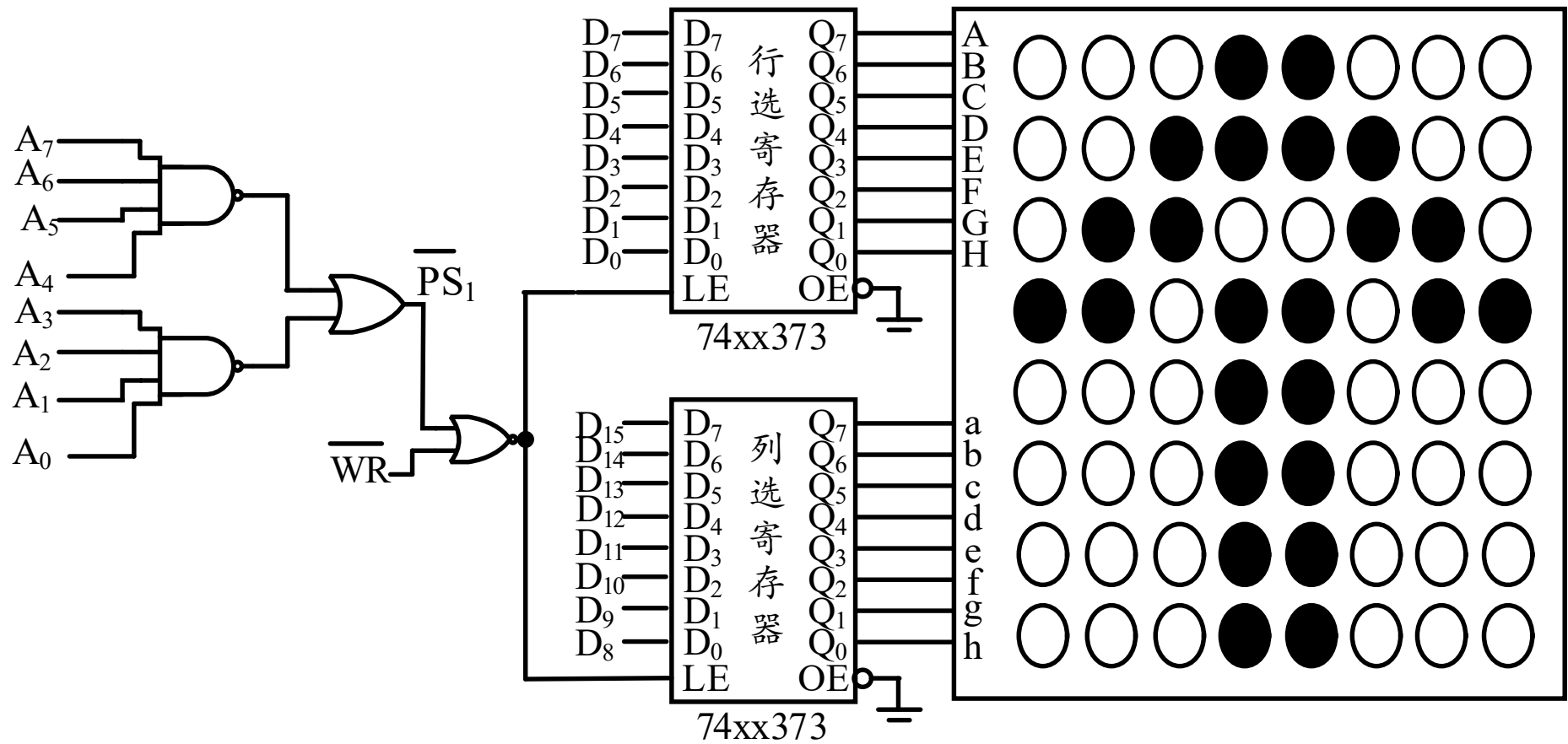


LED点阵工作原理



人眼视觉暂留效应

LED点阵接口



LED点阵接口

```
unsigned char Col_Code[8]={0xe7,0xc3,0x99,0x24,0xe7,0xe7,0xe7,0xe7};
unsigned char Rol_Code=0x80;
unsigned short code;
while(1) //无限循环控制8×8 LED点阵显示图案
{
    Rol_Code=0x80;
    for(int i=0;i<8;i++)
    {
        code=((unsigned short) Col_Code [i]<<8)| Rol_Code;
        Xil_Out16(0xff,code); //输出第i行的行选和列编码
        delay_ms( ); //延时
        Rol_Code = Rol_Code >> 1;
    }
}
```


小结

- 并行数字IO设备接口
 - 输入缓冲
 - 输出锁存
 - 动态扫描
 - 人眼视觉暂留效应-循环扫描
 - 软件延时

下一讲：模拟设备并行IO接口