

本章分为五节，主要介绍：

7.1 存储器的扩展

7.2 输入/输出及其控制方式

7.3 并行接口的扩展

7.4 8279接口芯片

7.5 显示器及键盘接口

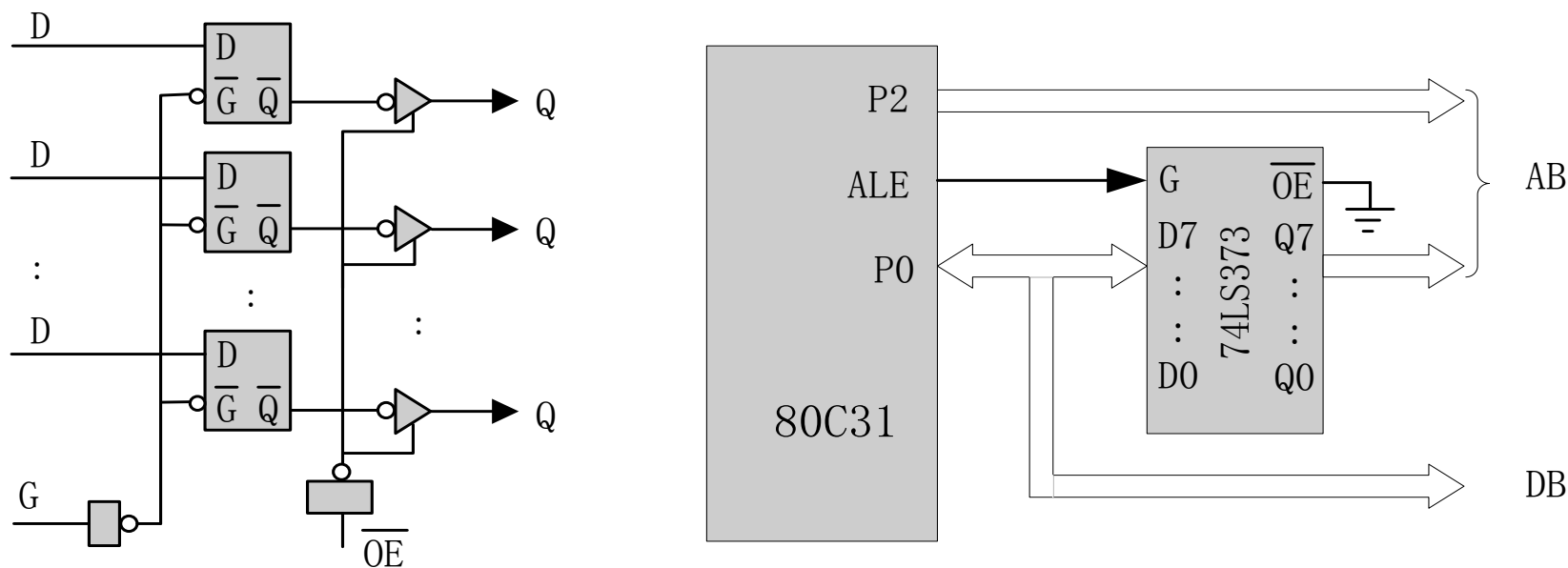
## 7.1 存储器的扩展

### 7.1.1 程序存储器的扩展

#### 一、扩展总线

- 数据总线宽度为 8 位，由 P0 口提供；↵
- 地址总线宽度为 16 位，可寻址范围达  $2^{16}$ ，即 64K。低 8 位 A7~A0 由 P0 口经地址锁存器提供，高 8 位 A15~A8 由 P2 口提供。由于 P0 口是数据、地址分时复用，所以 P0 口输出的低 8 位地址必须用地址锁存器进行锁存；↵
- 控制总线由  $\overline{RD}$ 、 $\overline{WR}$ 、 $\overline{PSEN}$ 、ALE 和  $\overline{EA}$  等信号组成，用于读/写控制、片外 ROM 选通、地址锁存控制和片内、片外 ROM 选择。↵

地址锁存器一般选用带三态缓冲输出的 8D 锁存器 74LS373。



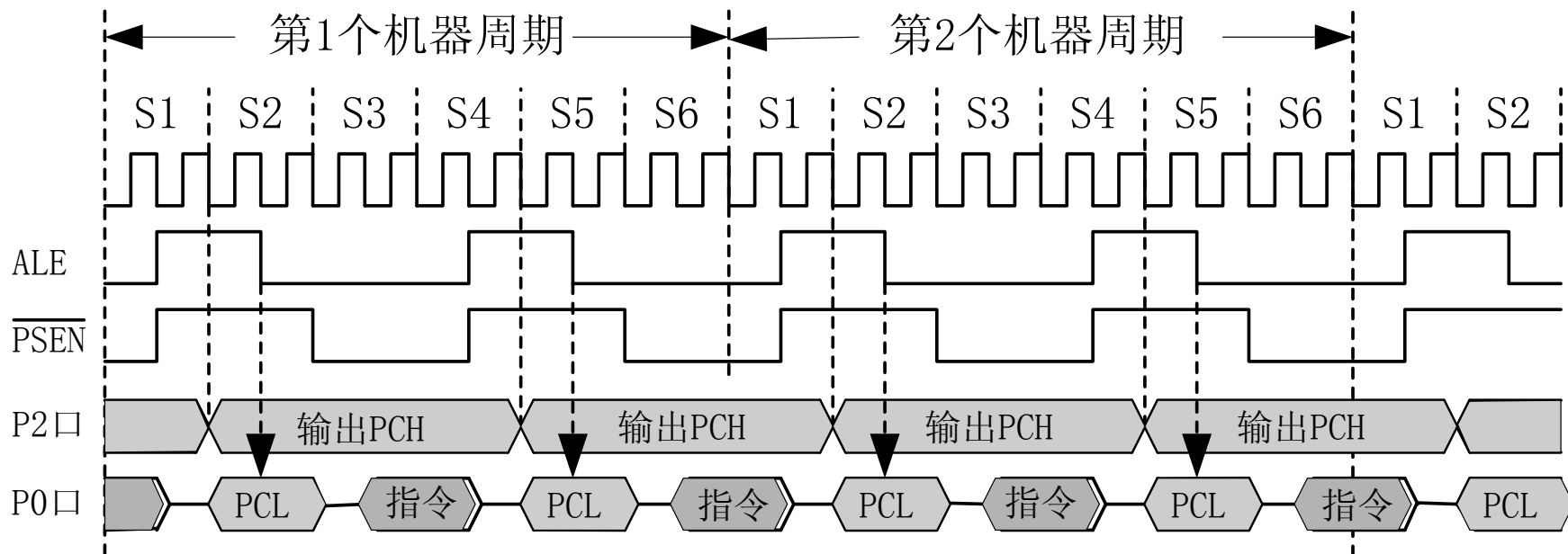
**74LS373**是有输出三态门的电平允许**8D**锁存器。当**G**（使能端）为高电平时，锁存器的数据输出端**Q**的状态与数据输入端**D**相同（透明的）。当**G**端从高电平返回到低电平时（下降沿后），输入端的数据就被锁存在锁存器中，数据输入端**D**的变化不再影响**Q**端输出。

## 二、片外ROM操作时序

进行ROM的扩展，其扩展方法较为简单容易，这是由单片机的优良扩展性能决定的。单片机的地址总线为16位，扩展的片外ROM的最大容量为64KB，地址为0000H~FFFFH。扩展的片外RAM的最大容量也为64KB，地址为0000H~FFFFH。

- 由于80C51采用不同的控制信号和指令，尽管ROM与RAM的地址是重叠的，也不会发生混乱。
- 80C51对片内和片外ROM的访问使用相同的指令，两者的选择是由硬件实现的。
- 芯片选择现在多采用线选法，地址译码法用的渐少。ROM与RAM共享数据总线和地址总线。

## 访问片外ROM的时序：



**80C51**系列单片机的**CPU**在访问片外**ROM**的一个机器周期内，信号**ALE**出现两次（正脉冲），**ROM**选通信号也两次有效，这说明在一个机器周期内，**CPU**两次访问片外**ROM**，也即在一个机器周期内可以处理两个字节的指令代码，所以在**80C51**系列单片机指令系统中有很多单周期双字节指令。

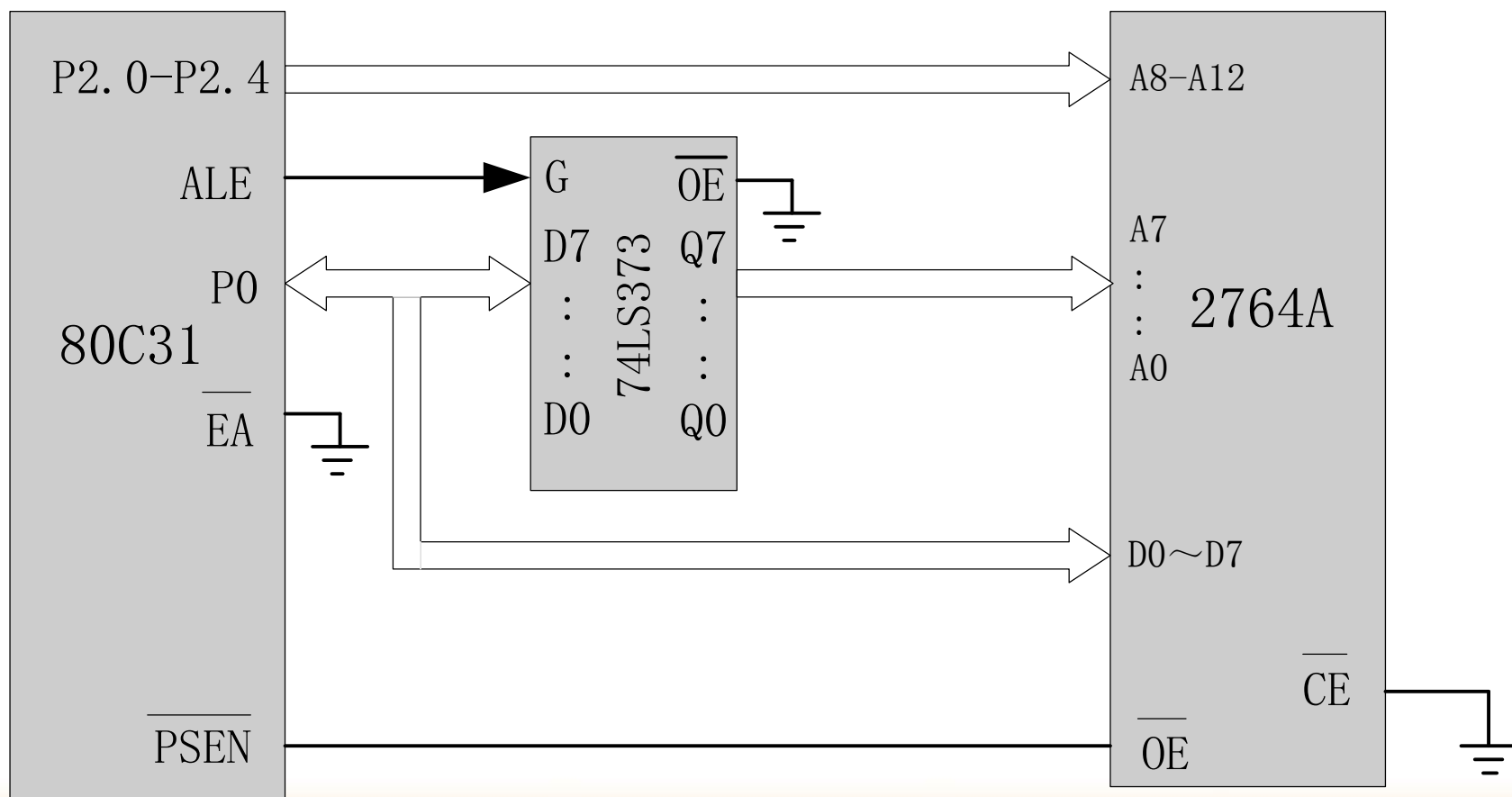
## 三、ROM芯片及扩展方法

### 1、EPROM存储器及扩展

常用的芯片有2732、2764、27128、27256、27512等。

27512	27256	27128	2764		2764	27128	27256	27512	
A15	V <sub>pp</sub>	V <sub>pp</sub>	V <sub>pp</sub>	1	28	V <sub>cc</sub>	V <sub>cc</sub>	V <sub>cc</sub>	V <sub>cc</sub>
A12	A12	A12	A12	2	27	PGM	PGM	A14	A14
A7	A7	A7	A7	3	26	NC	A13	A13	A13
A6	A6	A6	A6	4	25	A8	A8	A8	A8
A4	A4	A4	A4	5	24	A9	A9	A9	A9
A4	A4	A4	A4	6	23	A11	A11	A11	A11
A3	A3	A3	A3	7	22	OE	OE	OE	OE/V <sub>pp</sub>
A2	A2	A2	A2	8	21	A10	A10	A10	A10
A1	A1	A1	A1	9	20	CE	CE	CE	CE
A0	A0	A0	A0	10	19	Q7	Q7	Q7	Q7
Q0	Q0	Q0	Q0	11	18	Q6	Q6	Q6	Q6
Q1	Q1	Q1	Q1	12	17	Q5	Q5	Q5	Q5
Q2	Q2	Q2	Q2	13	16	Q4	Q4	Q4	Q4
GND	GND	GND	GND	14	15	Q3	Q3	Q3	Q3

## EPR0M存储器扩展电路:



## 2、EEPROM存储器及扩展

常用的EEPROM芯片有2864、2817等。

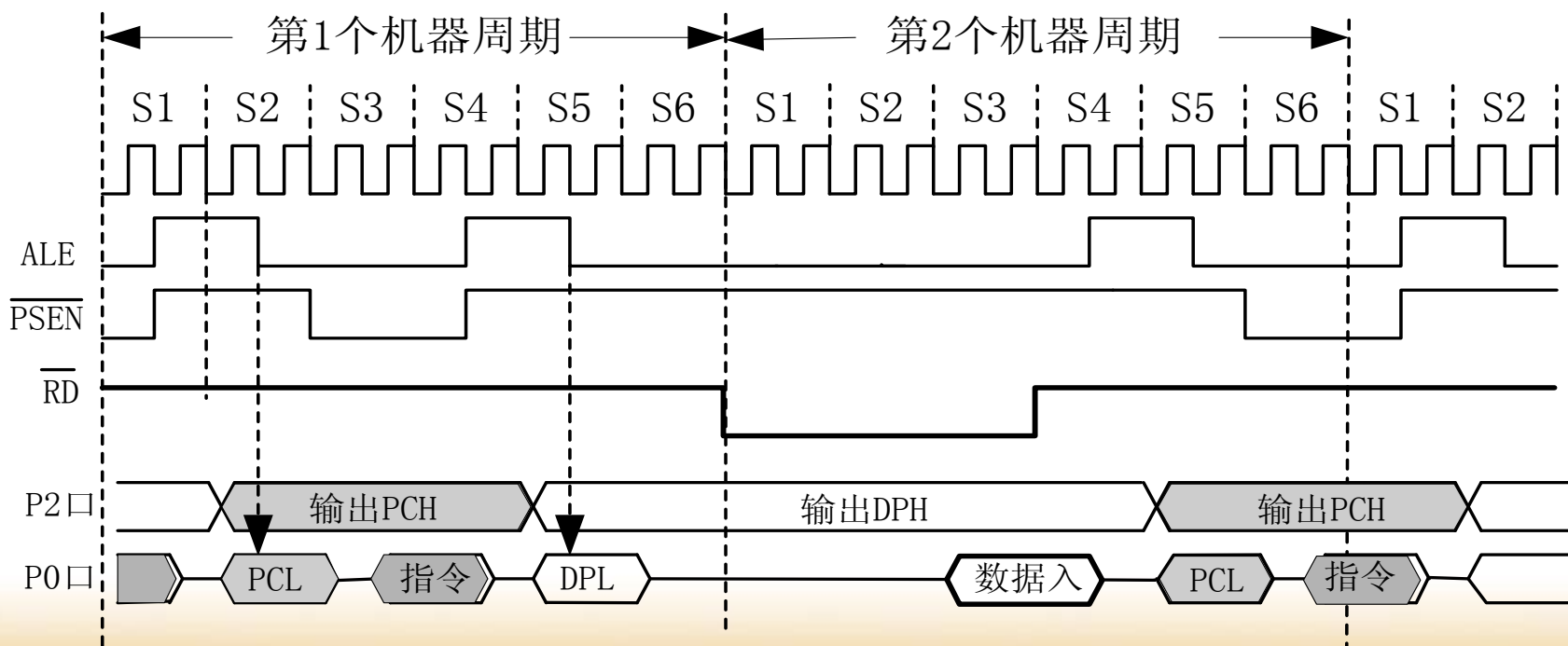
NC	1	28	$V_{CC}$		1	28	$V_{CC}$
A12	2	27	WE	RDY/ $\overline{BUSY}$	2	27	WE
A7	3	26	NC	A12	3	26	NC
A6	4	25	A8	A7	4	25	A8
A5	5	24	A9	A6	5	24	A9
A4	6	23	A11	A5	6	23	A11
A3	7	22	$\overline{OE}$	A4	7	22	$\overline{OE}$
A2	8	21	A10	A3	8	21	A10
A1	9	20	CE	A2	9	20	CE
A0	10	19	I/07	A1	10	19	I/07
I/00	11	18	I/06	A0	11	18	I/06
I/01	12	17	I/05	I/00	12	17	I/05
I/02	13	16	I/04	I/01	13	16	I/04
GND	14	15	I/03	I/02	14	16	I/04
				GND	15	15	I/03



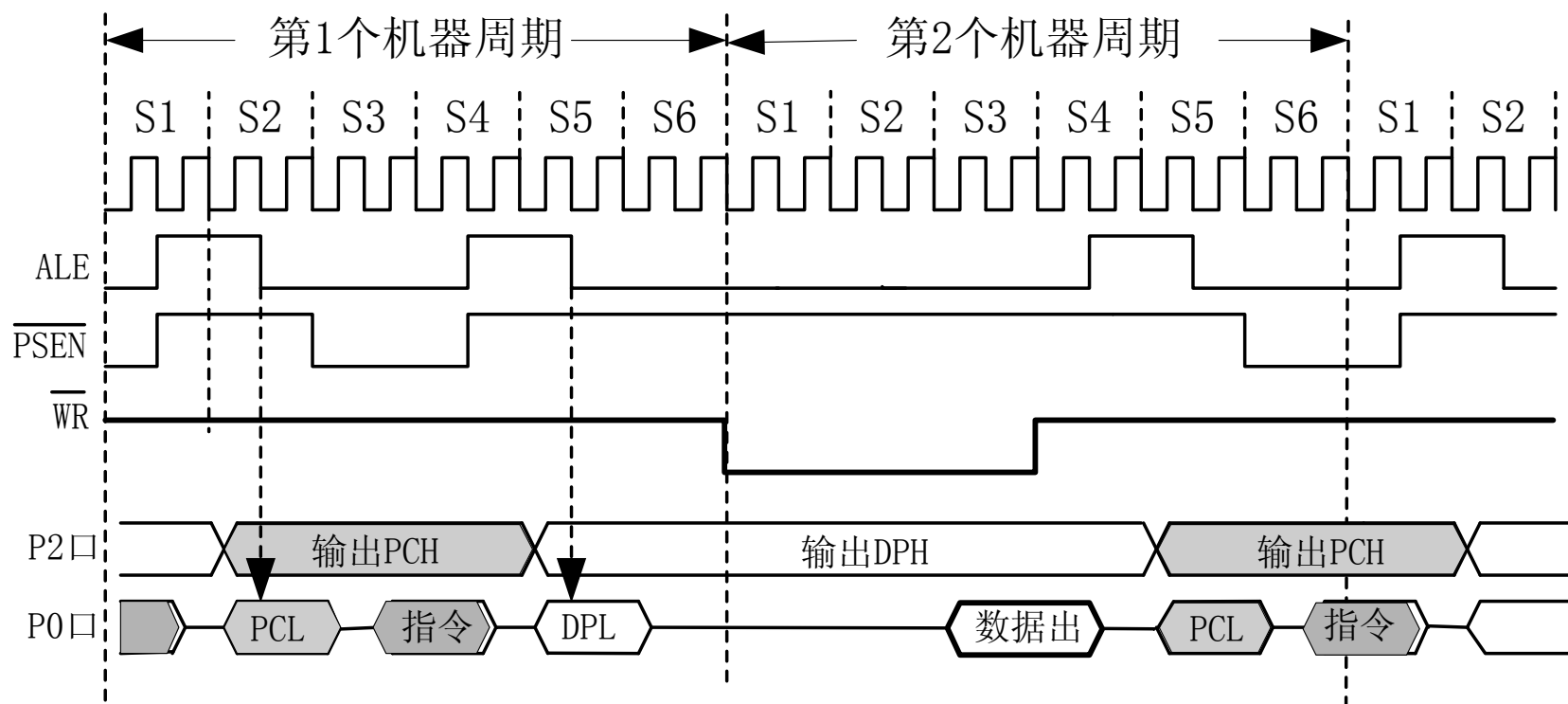
## 7.1.2 数据存储器的扩展

### 一、RAM扩展原理

扩展RAM和扩展ROM类似，由P2口提供高8位地址，P0口分时地作为低8位地址线和8位双向数据总线。**外部RAM读时序为：**



## 外部RAM写时序为：



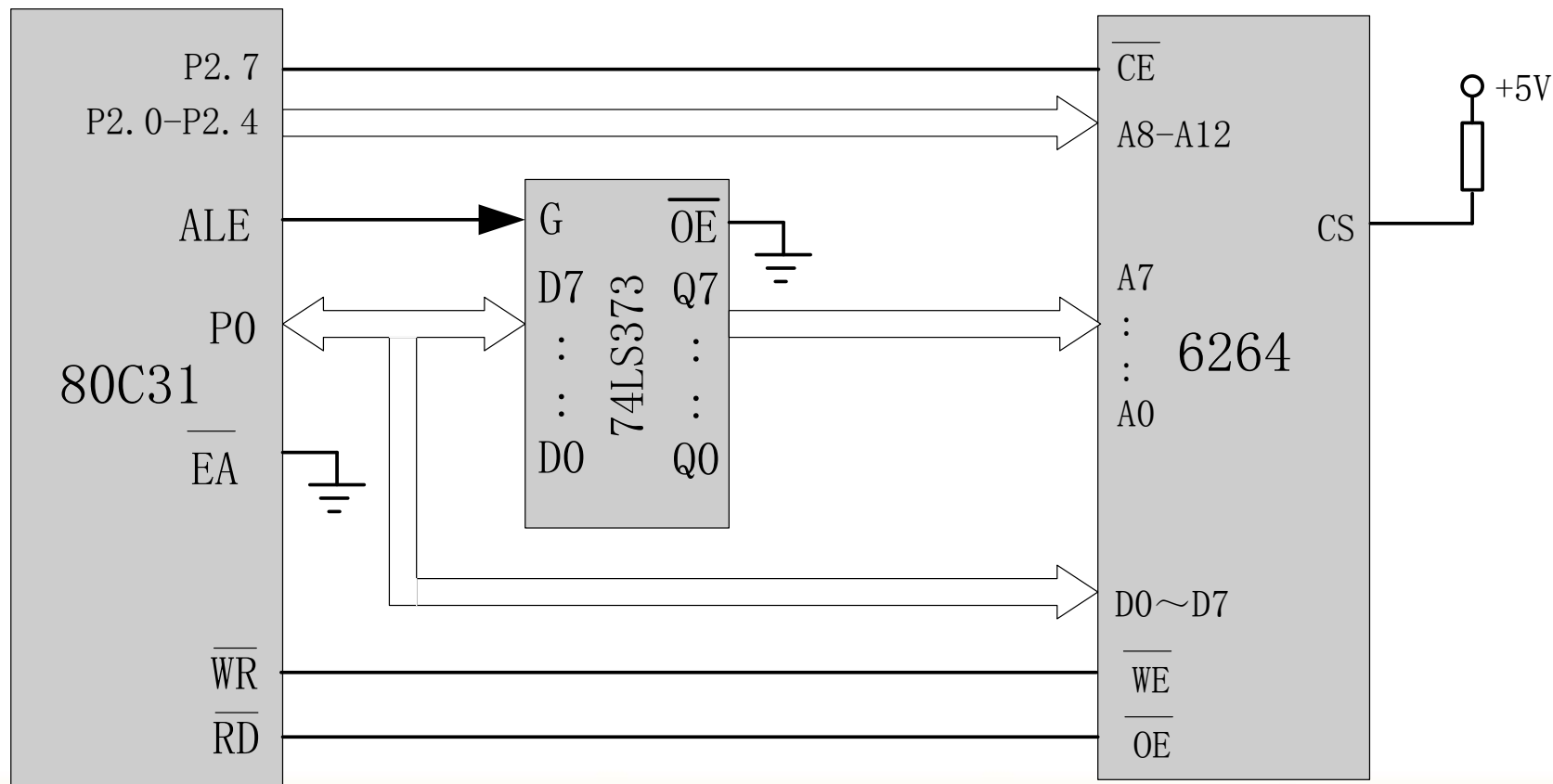
## 二、RAM扩展方法

### 1、数据存储器

常用的数据存储器SRAM芯片有**6116**、**6264**、**62256**等。

6116				62256 62128 6264			6266 62128 62256				
A7	1	24	Vcc	A14	NC	NC	1	28	Vcc	Vcc	Vcc
A6	2	23	A8	A12	A12	A12	2	27	$\overline{WE}$	$\overline{WE}$	$\overline{WE}$
A4	3	22	A9	A7	A7	A7	3	26	CS	A13	A13
A4	4	21	$\overline{WE}$	A6	A6	A6	4	25	A8	A8	A8
A3	5	20	$\overline{OE}$	A4	A4	A4	5	24	A9	A9	A9
A2	6	19	A10	A4	A4	A4	6	23	A11	A11	A11
A1	7	18	$\overline{CE}$	A3	A3	A3	7	22	$\overline{OE}$	$\overline{OE}$	$\overline{OE}/RFSH$
A0	8	17	D7	A2	A2	A2	8	21	A10	A10	A10
D0	9	16	D6	A1	A1	A1	9	20	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$
D1	10	15	D5	A0	A0	A0	10	19	D7	D7	D7
D2	11	14	D4	D0	D0	D0	11	18	D6	D6	D6
GND	12	13	D3	D1	D1	D1	12	17	D5	D5	D5
				D2	D2	D2	13	16	D4	D4	D4
				GND	GND	GND	14	15	D3	D3	D3

## 2、数据存储扩展电路

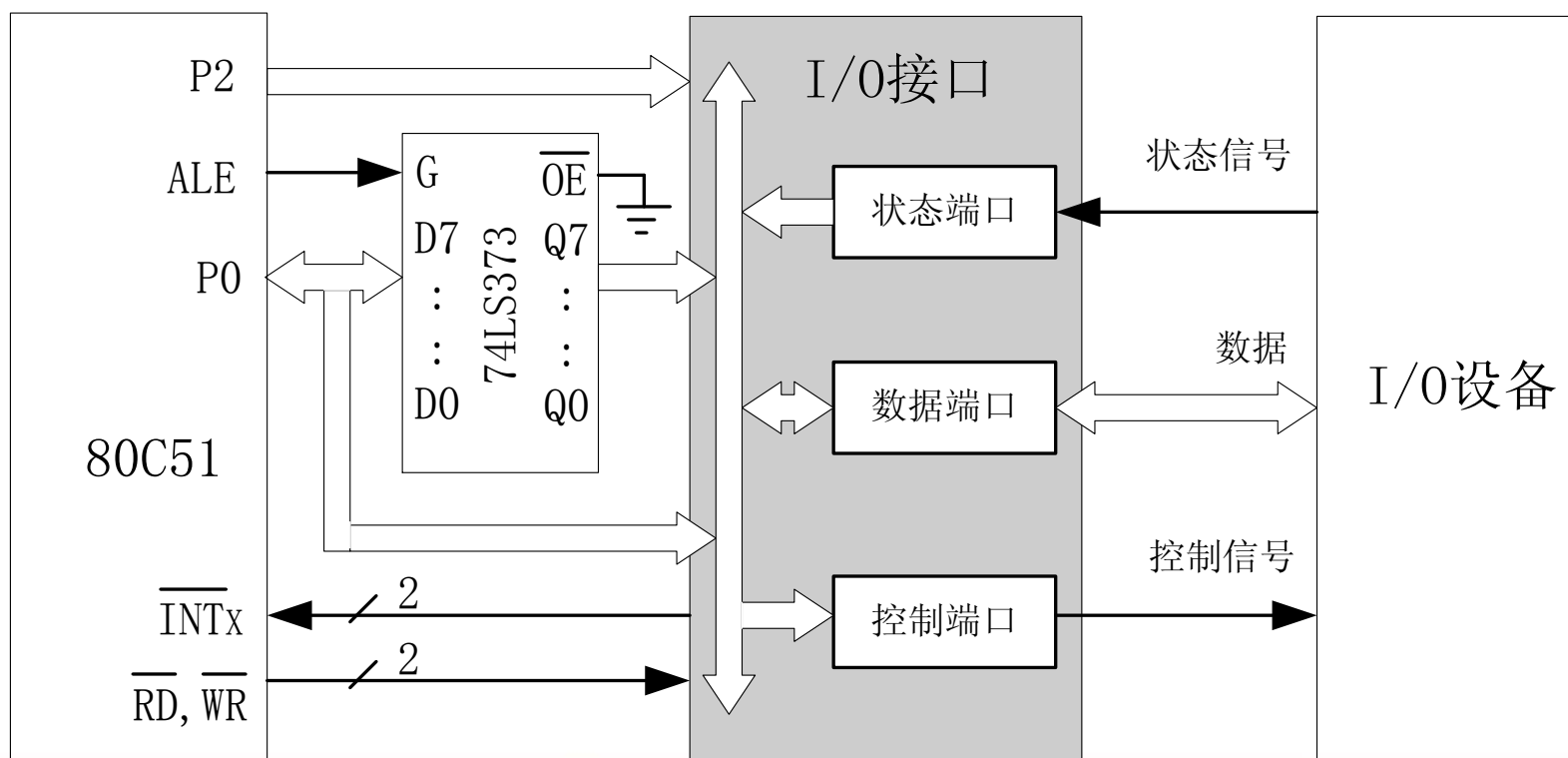


## 7.2 输入/输出及其控制方式

- 原始数据或现场信息要利用输入设备输入到单片机中，单片机对输入的数据进行处理加工后，还要输出给输出设备。常用的输入设备有键盘、开关及各种传感器等，常用的输出设备有LED（或LCD）显示器、微型打印机及各种执行机构等。
- 80C51单片机内部有四个并行口和一个串行口，对于简单的I/O设备可以直接连接。当系统较为复杂时，往往要借助I/O接口电路（简称I/O接口）完成单片机与I/O设备的连接。现在，许多I/O接口已经系列化、标准化，并具有可编程功能。

## 7.2.1 输入/输出接口的功能

单片机与I/O设备的关系:



## I/O接口的功能是：

### 1、对单片机输出的数据锁存

锁存数据线上瞬间出现的数据，以解决单片机与I/O设备的速度协调问题。

### 2、对输入设备的三态缓冲

外设传送数据时要占用总线，不传送数据时必须对总线呈高阻状态。利用I/O接口的三态缓冲功能，可以实现I/O设备与数据总线的隔离，便于其它设备的总线挂接。

### 3、信号转换

信号类型（数字与模拟、电流与电压）、信号电平（高与低、正与负）、信号格式（并行与串行）等的转换。

### 4、时序协调

**不同的I/O设备定时与控制逻辑是不同的**，并与**CPU**的时序往往是不一致的，这就需要**I/O**接口进行时序的协调。



## 7.2.2 单片机与I/O设备的数据传送方式

### 一、无条件传送

适用于对**简单的I/O设备**（如开关、**LED显示器**、继电器等）的操作，或者**I/O设备的定时固定或已知**的场合。

### 二、查询状态传送

**效率低。**

### 三、中断传送方式

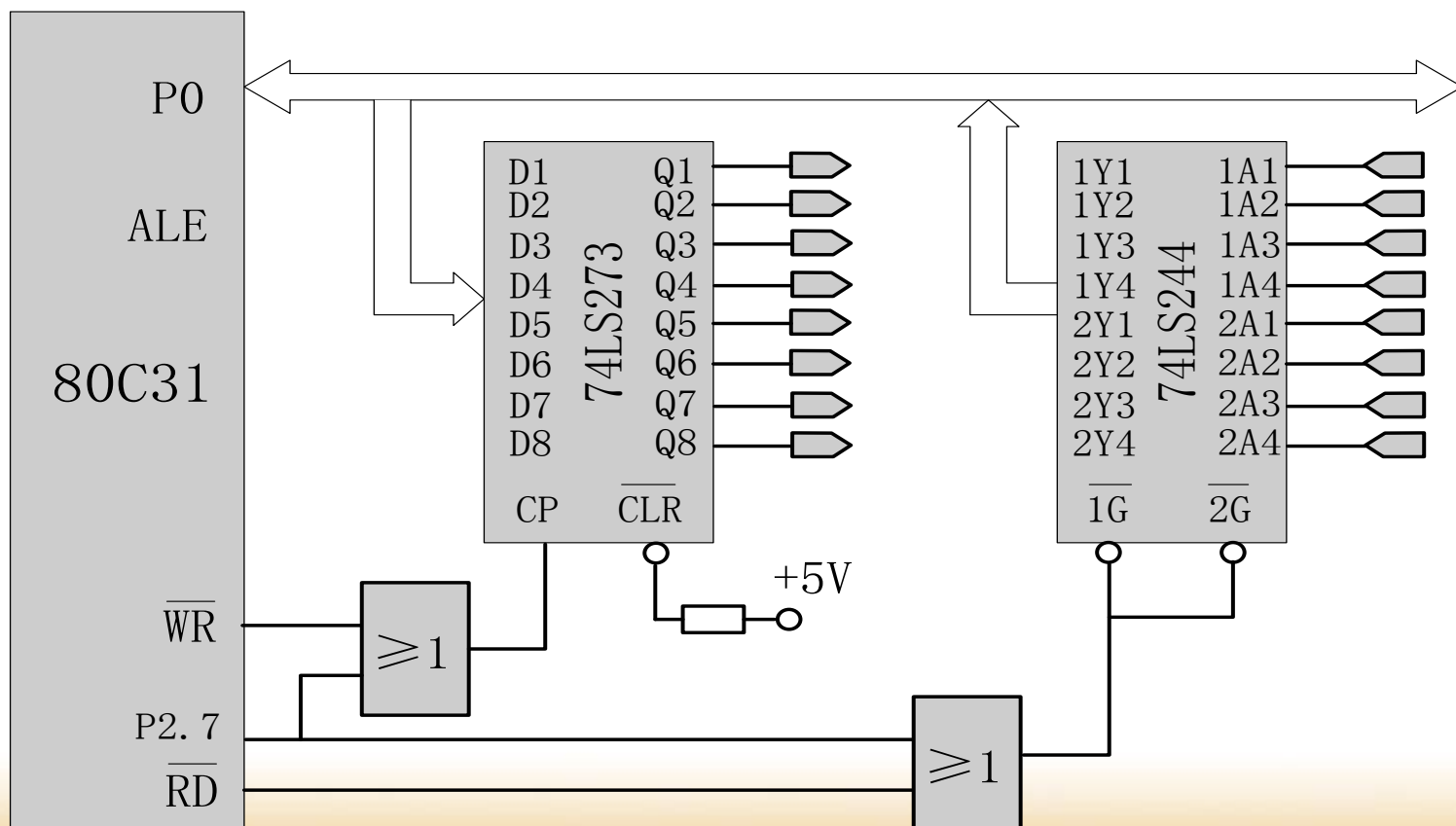
**效率大大提高。**

### 四、直接存储器存取（DMA）方式

适于**高速外设以及成组交换数据**的场合。

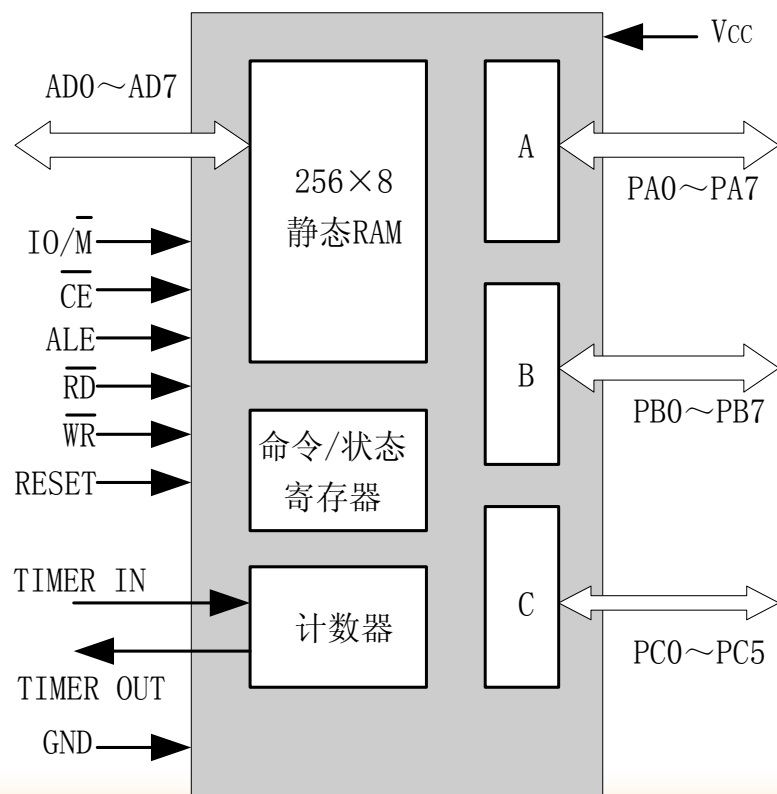
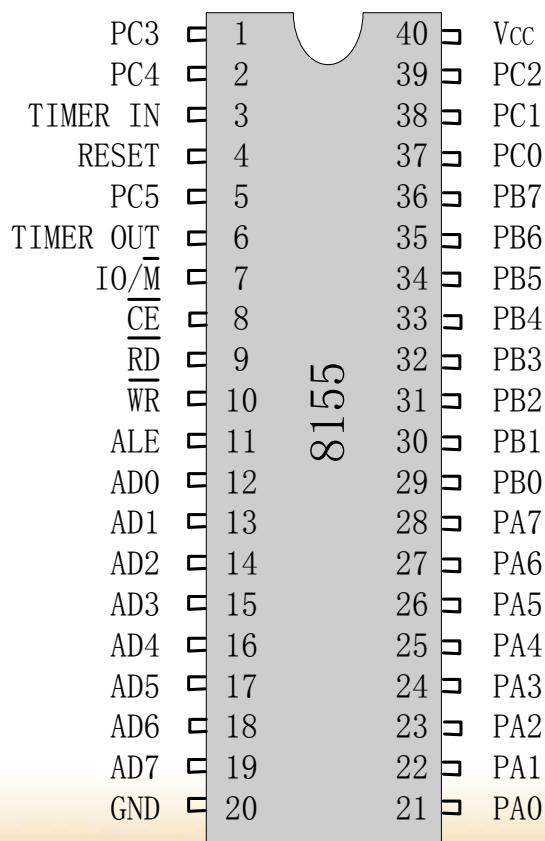
## 7.3 并行接口的扩展

### 7.3.1 并行输入/输出口的简单扩展



## 7.3.2 可编程接口8155的扩展

### 一、8155的引脚及结构



## 二、8155的内部编址

内部RAM地址为：00H~FFH。

内部端口地址为：000-----命令/状态寄存器

001-----A口

010-----B口

011-----C口

100-----计数器低8位

101-----计数器高6位及计数器方式设置位

## 三、工作方式设置及状态字格式

### 1、方式设置

位	7	6	5	4	3	2	1	0	
地址: 000	TM2	TM1	IEB	IEA	PC2	PC1	PB	PA	

**PA:** A口数据传送方向设置位。0--输入；1--输出。

**PB:** B口数据传送方向设置位。0--输入；1--输出。

**PC1、PC2:** C口工作方式设置位。

C口工作方式

PC2PC1	工作方式	说明
00	ALT1	A、B口为基本 I/O, C口方向为输入
11	ALT2	A、B口为基本 I/O, C口方向为输出
01	ALT3	A口为选通 I/O, PC0~PC2 作为 A口的选通应答 B口为基本 I/O, PC3~PC5 方向为输出
10	ALT4	A口为选通 I/O, PC0~PC2 作为 A口的选通应答 B口为选通 I/O, PC3~PC5 作为 B口的选通应答

**IEA:** A口的中断允许设置位。0--禁止；1--允许。

**IEB:** B口的中断允许设置位。0--禁止；1--允许。

**TM2、TM1:** 计数器工作方式设置位。

定时/计数器命令字

TM2TM1	工作方式	说明
00	方式 0	空操作，对计数器无影响
01	方式 1	使计数器停止计数
10	方式 2	减 1 计数器回 0 后停止工作
11	方式 3	未计数时，送完初值及方式后立即启动计数； 正在计数时，重置初值后，减 1 计数器回 0 则按新计数初值计数

## 2、状态字格式

位 <sup>↕</sup>	7 <sup>↕</sup>	6 <sup>↕</sup>	5 <sup>↕</sup>	4 <sup>↕</sup>	3 <sup>↕</sup>	2 <sup>↕</sup>	1 <sup>↕</sup>	0 <sup>↕</sup>
地址: 000 <sup>↕</sup>	<sup>↕</sup>	TIMER	INTEB	BFB <sup>↕</sup>	INTRB	INTEA	BFA <sup>↕</sup>	INTRA

- **INTRX**: 中断请求标志。 **INTRX=1**, 表示A或B口有中断请求; **INTRX=0**, 表示A或B口无中断请求。
- **BFX**: 口缓冲器空/满标志。 **BFX=1**, 表示口缓冲器已装满数据, 可由外设或单片机取走; **BFX=0**, 表示口缓冲器为空, 可以接受外设或单片机发送数据。
- **INTEX**: 口中断允许/禁止标志。 **INTEX=1**, 表示允许口中断; **INTEX=0**, 表示禁止口中断。
- **TIMER**: 计数器计满标志。 **TIMER=1**, 表示计数器的原计数初值已计满回零; **TIMER=0**, 表示计数器尚未计满。

## 四、计数器输出模式

计数器是一个**14**位的减法计数器，它能对输入的脉冲进行计数，在到达最后一个计数值时，输出一个矩形波或脉冲。

装入计数长度寄存器的值为**2H~3FFFH**。**15、14**两位用于规定计数器的输出方式。

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	M2	M1	T13	T12	T11	T10	T9	T8	T7	T6	T5	T4	T3	T2	T1	T0

计数器输出方式

M2M1	输出方式	说 明
00	方式 0	<b>单方波输出</b> 。计数期间为低电平，计数器回 0 后输出高电平。
01	方式 1	<b>连续方波输出</b> 。计数长度前半部分输出高电平，后半部分输出低电平。
10	方式 2	<b>单脉冲输出</b> 。计数器回 0 后输出一个单脉冲。
11	方式 3	<b>连续脉冲输出（计数值自动重装）</b> 。计数器回 0 后输出单脉冲，又自动向计数器重装原计数值，回 0 后又输出单脉冲，如此循环。



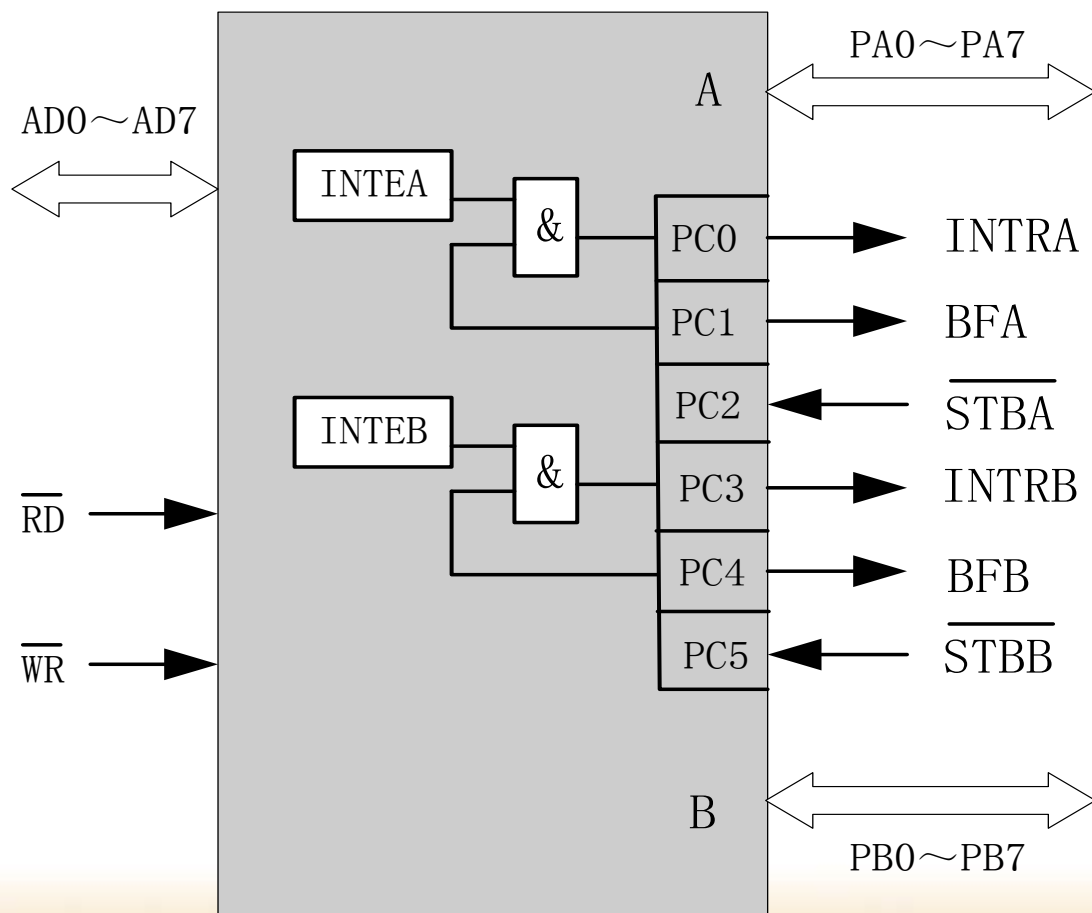
## 五、选通I/O的组态

对8155命令字的PC2、PC1位编程，使A或B口工作在选通方式时，C口的PC0~PC5就被定义为A或B口选通I/O方式的应答和控制线。

C口的控制分配表

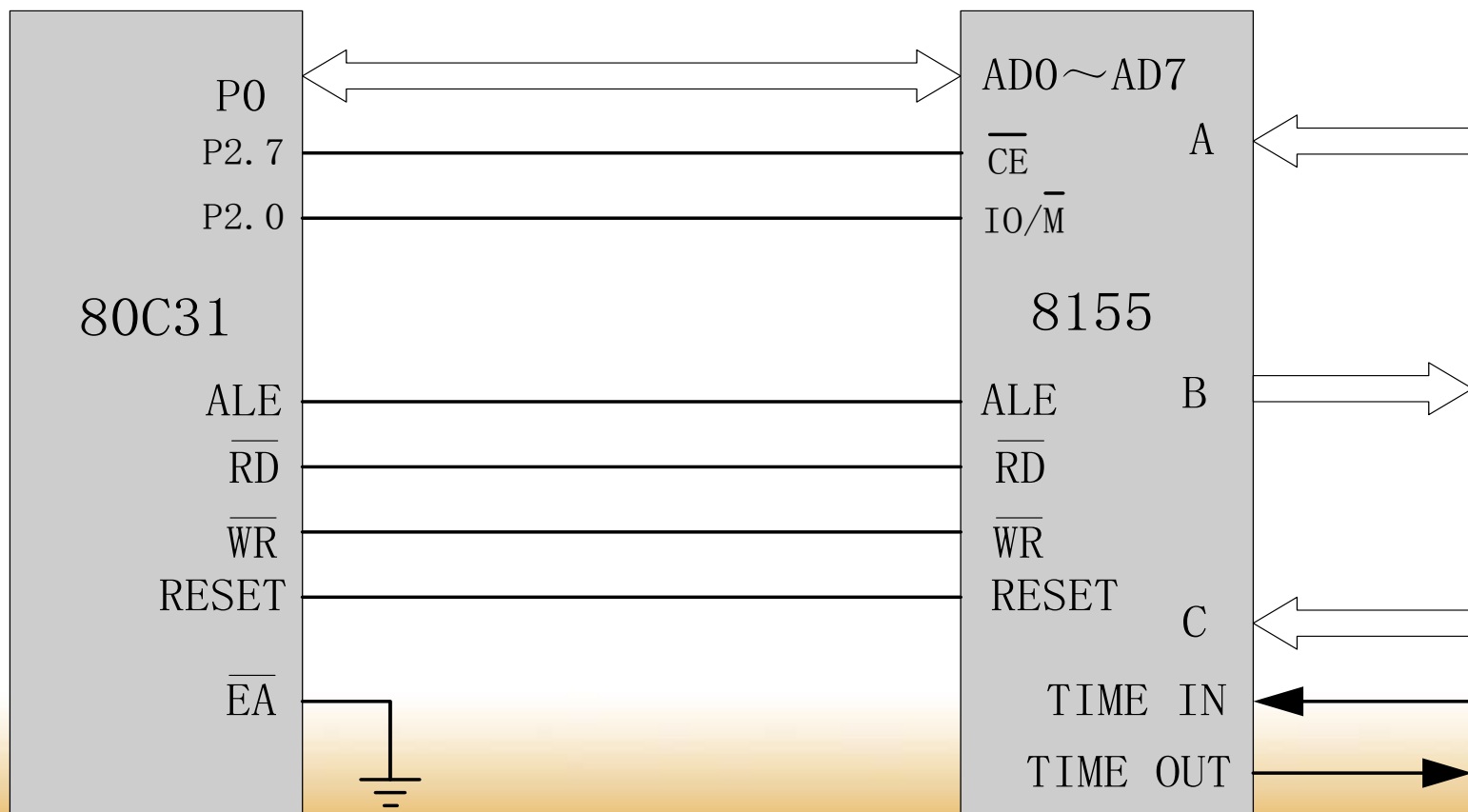
工作方式	PC5	PC4	PC3	PC2	PC1	PC0
ALT1	输入					
ALT2	输出					
ALT3	输出			$\overline{\text{STBA}}$	BFA	INTRA
ALT4	$\overline{\text{STBB}}$	BFB	INTRB	$\overline{\text{STBA}}$	BFA	INTRA

## 选通方式的组态逻辑：



## 六、8155芯片与单片机的接口

80C51单片机可以与8155直接连接而不需要附加任何电路。使系统增加256字节的RAM，22位I/O线及一个计数器。



8155中**RAM地址**因P2.7 (A15) =0 及P2.0 (A8) =0, 故可选为**01111110 00000000B (7E00H) ~ 01111110 11111111B (7EFFH)** ;

**I/O端口的地址为: 7F00H~7F05H。**

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	I/O口
0	X	X	X	X	X	X	1	X	X	X	X	X	0	0	0	命令/状态口
0	X	X	X	X	X	X	1	X	X	X	X	X	0	0	1	A口
0	X	X	X	X	X	X	1	X	X	X	X	X	0	1	0	B口
0	X	X	X	X	X	X	1	X	X	X	X	X	0	1	1	C口
0	X	X	X	X	X	X	1	X	X	X	X	X	1	0	0	计数器低8位
0	X	X	X	X	X	X	1	X	X	X	X	X	1	0	1	计数器高6位及方式

若A口、B口定义为基本输入方式，计数器作为方波发生器，对80C31输入脉冲进行24分频（但需要注意8155的计数最高频率约为4MHZ），则8155 I/O口初始化程序如下：

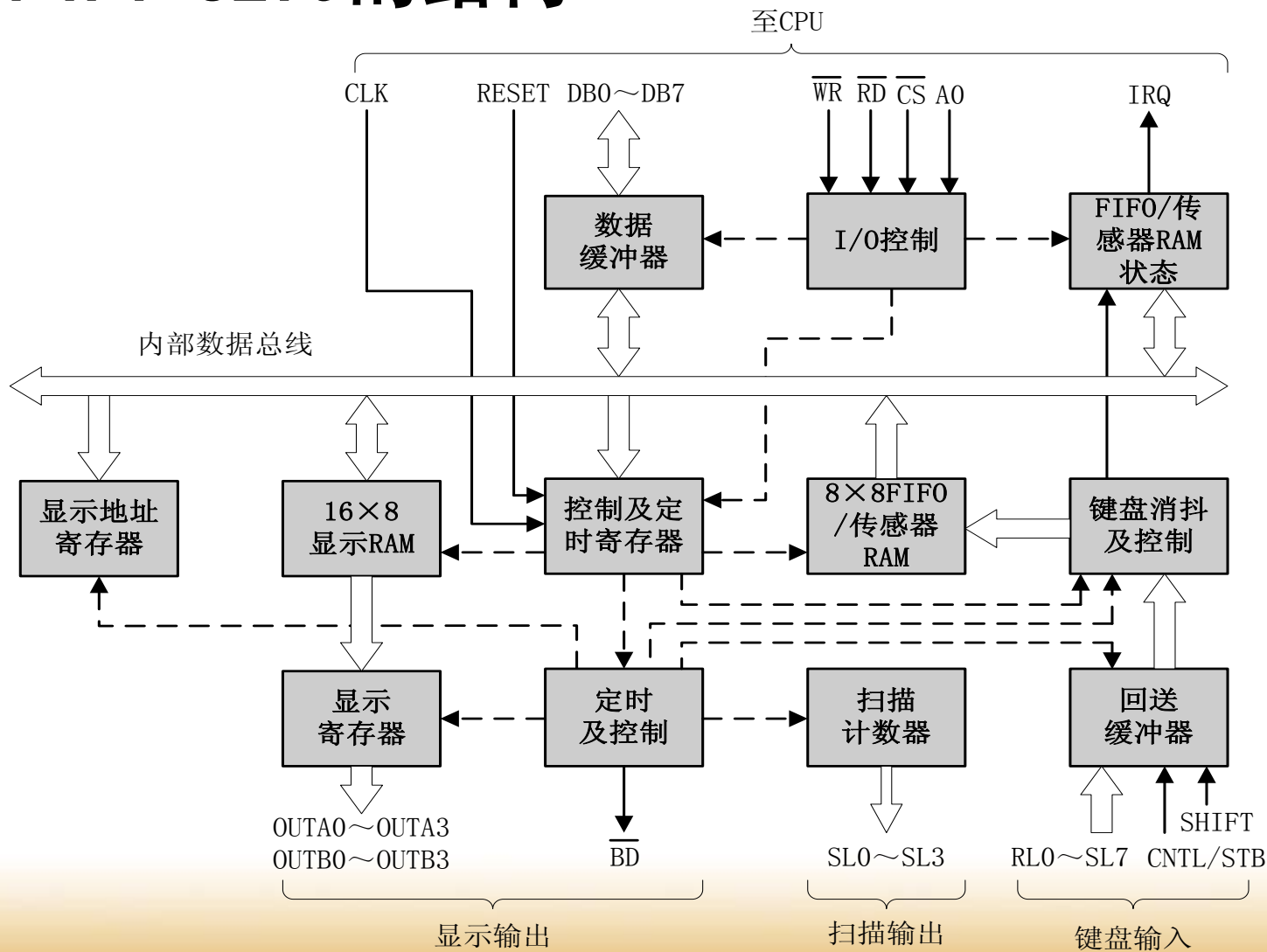
```
START: MOV DPTR, #7F04H ; 计数寄存器低8位
      MOV A, #18H ; 计数器初值#18H (24D)
      MOVX @DPTR, A ; 计数器寄存器低8位赋值
      INC DPTR; 指向计数器寄存器高6位及方式位
      MOV A, #40H ; 计数器为连续方波方式
      MOVX @DPTR, A ; 计数寄存器高6位赋值
      MOV DPTR, #7F00H ; 命令寄存器
      MOV A, #0C2H ; 设命令字
      MOVX @DPTR, A ; 送命令字
```

## 7.4 8279接口芯片

- 由**80C51**系列单片机构成的小型测控系统或智能仪表中，常常需要扩展显示器和键盘以实现人机对话功能。**8279**芯片在扩展显示器和键盘时功能强、使用方便。

- 8279**是Intel公司为8位微处理器设计的通用键盘/显示器接口芯片，其功能是：**接收来自键盘的输入数据并作预处理；完成数据显示的管理和数据显示器的控制。**单片机应用系统采用**8279**管理键盘和显示器，软件编程极为简单，显示稳定，且减少了主机的负担。

# 7.4.1 8279的结构



- **数据缓冲器**将双向三态8位内部数据总线D0~D7与系统总线相连，用于传送CPU与8279之间的命令和状态。
- **控制和定时寄存器**用于寄存键盘和显示器的工作方式，锁存操作命令，通过译码器产生相应的控制信号，使8279的各个部件完成相应的控制功能。
- **定时器**包含一些计数器，其中有一个可编程的5位计数器（计数值在2~31间），对CLK输入的时钟信号进行分频，产生100 KHz的内部定时信号（此时扫描时间为5.1ms，消抖时间为10.3ms）。外部输入时钟信号周期不小于500ns。



- **扫描计数器**有两种输出方式：**一是编码方式**，计数器以二进制方式计数，4位计数状态从扫描线**SL3~SL0**输出，经外部译码器可以产生**16位**的键盘和显示器扫描信号；**另一种是译码方式**，扫描计数器的低两位经内部译码后从**SL3~SL0**输出，直接作为键盘和显示器的扫描信号。
- **回送缓冲器、键盘消抖及控制**完成对键盘的自动扫描以搜索闭合键，锁存**RL7~RL0**的键输入信息，消除键的抖动，将键输入数据写入内部先进先出存储器（**FIFO RAM**）。**RL7~RL0**为回送信号线作为键盘的检测输入线，由回送缓冲器缓冲并锁存，当某一键闭合时，附加的移位状态**SHIFT**、控制状态**CNTL**及扫描码和回送信号拼装成一个字节的“键盘数据”送入**8279**内部的**FIFO**（先进先出）**RAM**。

## 键盘的数据格式为：

位	7	6	5	4	3	2	1	0	
	CNTL	SHIFT	扫描（闭合键行号）			回送（闭合键列号）			

在传感器矩阵方式和选通方式时，回送线**RL7~RL0**的内容被直接送往相应的**FIFO RAM**。输入数据即为

**RL7~RL0**。数据格式为：

位	7	6	5	4	3	2	1	0
	RL7	RL6	RL5	RL4	RL3	RL2	RL1	RL0

## FIFO/传感器RAM是具有双功能的 $8 \times 8$ RAM

- 在键盘或选通方式时，它作为FIFO RAM，依先进先出的规则输入或读出，其状态存放在FIFO/传感器RAM状态寄存器中。只要FIFO RAM不空，状态逻辑将置中断请求 $IRQ=1$ ；
- 在传感器矩阵方式，作为传感器RAM，当检测出传感器矩阵的开关状态发生变化时，中断请求信号 $IRQ=1$ 。在外部译码扫描方式时，可对 $8 \times 8$ 矩阵开关的状态进行扫描，在内部译码扫描方式时，可对 $4 \times 8$ 矩阵开关的状态进行扫描。

**显示RAM用来存储显示数据，容量是 $16 \times 8$ 位**

- 在显示过程中，存储的**显示数据**轮流从显示寄存器输出。显示寄存器输出分成两组，即**OUTA0~OUTA3**和**OUTB0~OUTB3**，两组可以单独送数，也可以组成一个**8位**的字节输出，该输出与位选扫描线**SL0~SL3**配合就可以实现动态扫描显示。
- **显示地址寄存器**用来寄存**CPU**读/写显示**RAM**的地址，可以设置为每次读出或写入后自动递增。

## 7.4.2 8279的引脚定义

RL2	□	1	8279	40	□	Vcc
RL3	□	2		39	□	RL1
CLK	□	3		38	□	RL0
IRQ	□	4		37	□	CNTL/STB
RL4	□	5		36	□	SHIFT
RL5	□	6		35	□	SL3
RL6	□	7		34	□	SL2
RL7	□	8		33	□	SL1
RESET	□	9		32	□	SL0
$\overline{RD}$	□	10		31	□	OUTB0
$\overline{WR}$	□	11		30	□	OUTB1
DB0	□	12		29	□	OUTB2
DB1	□	13		28	□	OUTB3
DB2	□	14		27	□	OUTA0
DB3	□	15		26	□	OUTA1
DB4	□	16		25	□	OUTA2
DB5	□	17		24	□	OUTA3
DB6	□	18		23	□	$\overline{BD}$
DB7	□	19		22	□	$\overline{CS}$
GND	□	20		21	□	A0

**DB7~DB0**为双向外部数据总线；  
 $\overline{CS}$  为片选信号线，低电平有效；  
 $\overline{RD}$  和 $\overline{WR}$ 为读和写选通信号线；

**IRQ**为中断请求输出线。

**RL7~RL0**为键盘回送线。

**SL3~SL0**为扫描输出线。

**OUTB3~OUTB0、OUTA3~OUTA0**为显示寄存器数据输出线。

**RESET**为复位输入线。

**SHIFT**为换档键输入线。

**CNTL/STB**为控制/选通输入线。

**CLK**为外部时钟输入线。

$\overline{BD}$  为显示器消隐控制线。

# 7.4.3 8279的操作命令

命令特征位			功能特征位				
D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0, 左输入 1, 右输入	0, 8 字符 1, 16 字符	00, 双键互锁		0, 编码 1, 译码
					01, N 键轮回		
					10, 传感器矩阵		
					11, 选通输入		
0	0	1	2~31				
0	1	0	0, 仅读 1 个单元 1, 每次读后地址加 1	X	3 位传感器 RAM 起始地址		4 位显示 RAM 起始地址
0	1	1					
1	0	0					
1	0	1					
1	0	1	X	1. A 组不变 0. A 组可变	1. B 组不变 0. B 组可变	1. A 组消隐 0. 恢复	1. A 组消隐 0. 恢复
1	1	0	0. 不消除 (CA=0) 1. 允许消除	00, 全清为 0 01, 全清为 0 10, 清为 20H 11, 清为全 1		CF 清 FIFO 使之为空, 且 IBQ=0 读出地址 0	CA: 总清 清显示 清 FIFO
1	1	1	E	X	X	X	X

## 一、显示器和键盘方式设置命令

**D7 D6 D5=000** 是键盘/显示方式命令特征字。

**D4 D3=DD** 为显示器方式设置位。

**D2 D1 D0 =KKK** 为键盘工作方式设置位。

**8279** 可外接**8位或16位 LED** 显示器，显示器的每一位对应一个**8位**的显示器缓冲单元。左端输入方式较为简单，显示缓冲器**RAM地址0~15**分别对应于显示器的**0位(左)~15位(右)**。**CPU**依次从**0地址**或某一地址开始将段数据写入显示缓冲器。右端输入方式是移位，输入数据总是写入右端的显示缓冲器，数据写入显示缓冲器后，原来缓冲器的内容左移一个字节。

- **内部译码**的扫描方式时，扫描信号由 **SL3~SL0** 输出，仅能提供 **4选1** 扫描线。
- **外部译码**工作方式时，内部计数器作二进制计数，**4位二进制计数器的计数状态**从扫描线 **SL3~SL0** 输出，并在外部进行译码。可为键盘/显示器提供 **16选1** 扫描线。
- **双键互锁**工作方式时，键盘中同时有两个以上的键被按下，任何一个键的编码信息均不能进入 **FIFO RAM**，直至仅剩下一个键闭合时，该键的编码信息方能进入 **FIFO RAM**。



- **N键轮回**工作方式时, 如有多个键按下, 键盘扫描能够根据发现它们的顺序, 依次将它们的状态送入**FIFO RAM**。
- **传感器矩阵**工作方式, 是指片内的去抖动逻辑被禁止掉, 传感器的开关状态直接输入到**FIFO RAM**中。因此, 传感器开关的闭合或断开均可使**IRQ**马上为**1**, 向**CPU**快速申请中断。

## 二、时钟编程命令

**D7D6D5=001** 为时钟编程命令特征位。

8279的内部定时信号是由外部输入时钟经分频后产生的，分频系数由时钟编程命令确定。**D4~D0** 用来设定对**CLK**端输入时钟的分频次数**N**，**N=2~31**。利用这条命令，可以将来自**CLK**引脚的外部输入时钟分频，以取得**100KHz**的内部时钟信号。例如**CLK**输入时钟频率为**2MHz**，获得**100KHz**的内部时钟信号，则需要**20**分频。

### 三、读FIFO /传感器RAM命令

**D7D6D5=010** 为该命令的特征位。

**D2~D0 (AAA)** 为起始地址。**D4 (AI)** 为多次读出时的地址自动增量标志，**D3**无用。在键扫描方式中，**AI AAA**均被忽略，**CPU**总是按先进先出的规律读键输入数据，直至输入键全部读出为止。在传感器矩阵方式中，若**AI=1**，则**CPU**从起始地址开始依次读出，每读出一个数据地址自动加1；**AI=0**，**CPU**仅读出一个单元的内容。

#### 四、读显示RAM命令

**D7D6D5=011**为该命令的特征位。

**D3~D0 (AAAA)** 用来寻址显示RAM的16个存储单元，**AI**为自动增量标志，若**AI=1**，则每次读出后地址自动加1。

#### 五、写显示RAM命令

**D7 D6 D5=100**为该命令的特征位。

**D4 (AI)** 为自动增量标志，**D3~D0 (AAAA)** 为起始地址，数据写入按左端输入或右端输入方式操作。若**AI=1**，则每次写入后地址自动加1，直至所有显示RAM全部写完。

## 六、显示器写禁止/消隐命令

**D7 D6 D5=101**为该命令的特征位。该命令用以禁止写**A组**和**B组**显示**RAM**。

在双**4位**显示器使用时，即**OUTA3~OUTA0**和**OUTB3~OUTB0**独立地作为两个半字节输出时，可改写显示**RAM**中的低半字节而不影响高半字节的状态，反之亦可改写高半字节而不影响低半字节。**D1、D0**位是消隐显示器特征位，要消隐两组显示器，必须使之同时为**1**，为**0**时则恢复显示。

## 七、清除命令

**D7 D6 D5=110**为该命令的特征位。**CPU**将清除命令写入**8279**，使显示缓冲器呈初态（暗码），该命令同时也能清除输入标志和中断请求标志。

**D4 D3 D2 (CDCDCD)**用来设定清除显示**RAM**的方式。

**D1 (CF) =1**为清除**FIFO RAM**的状态标志，**FIFO RAM**被置成空状态（无数据），并复位中断请求线**IRQ**时，传感器 **RAM**的读出地址也被置成**0**。

**D0 (CA)**是总清的特征位，它兼有**CD**和**CF**的联合效用。当**CA =1**时，对显示**RAM**的清除方式仍由**D3、D2**编码确定。

## 八、结束中断/错误方式设置命令

**D7 D6 D5=101**为该命令的特征位。此命令用来结束传感器RAM的中断请求。

**D4 (E) =0** 为结束中断命令。在传感器工作方式中使用。每当传感器状态出现变化时，扫描检测电路就将其状态写入传感器RAM，并启动中断逻辑使IRQ变高，向CPU请求中断，并且禁止写入传感器RAM。此时，若传感器RAM读出地址的自动增量特征位未设置 (**AI=0**)，则中断请求IRQ在CPU第一次从传感器RAM读出数据时就被清除。若**AI=1**，则CPU对传感器RAM读出并不能清除IRQ，而必须通过给**8279**写入结束中断/设置出错方式命令才能使IRQ变低。

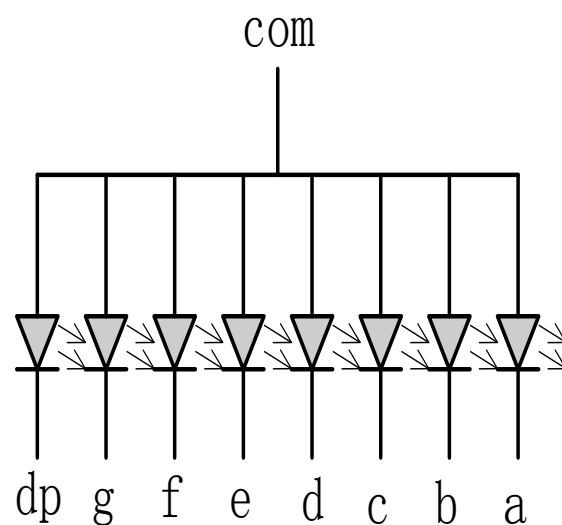
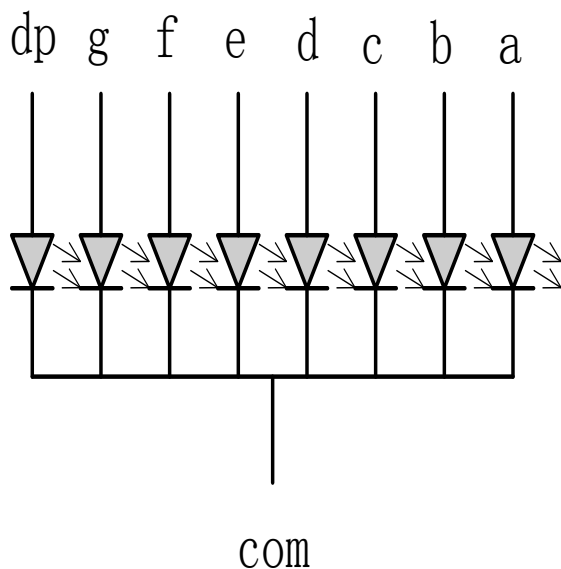
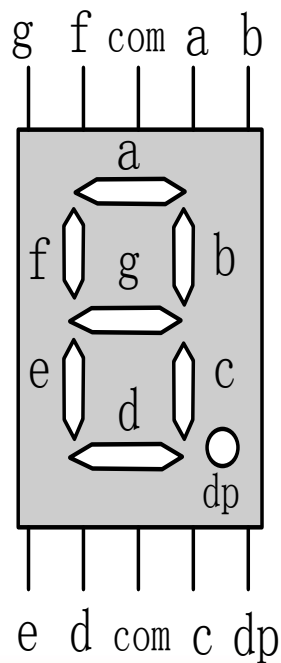
**D4 (E) =1** 为特定错误方式命令。在**8279**已被设定为键盘扫描**N**键轮回方式后，如果**CPU**给**8279**又写入结束中断/错误方式命令 (**E=1**)，则**8279**将以一种特定的错误方式工作。这种方式的特点是：在**8279**消抖周期内，如果发现多个按键同时按下，则**FIFO**状态字中的错误特征位**S/E**将置**1**，并产生中断请求信号和阻止写入**FIFO RAM**。



## 7.5 显示器及键盘接口

### 7.5.1 七段显示器的原理

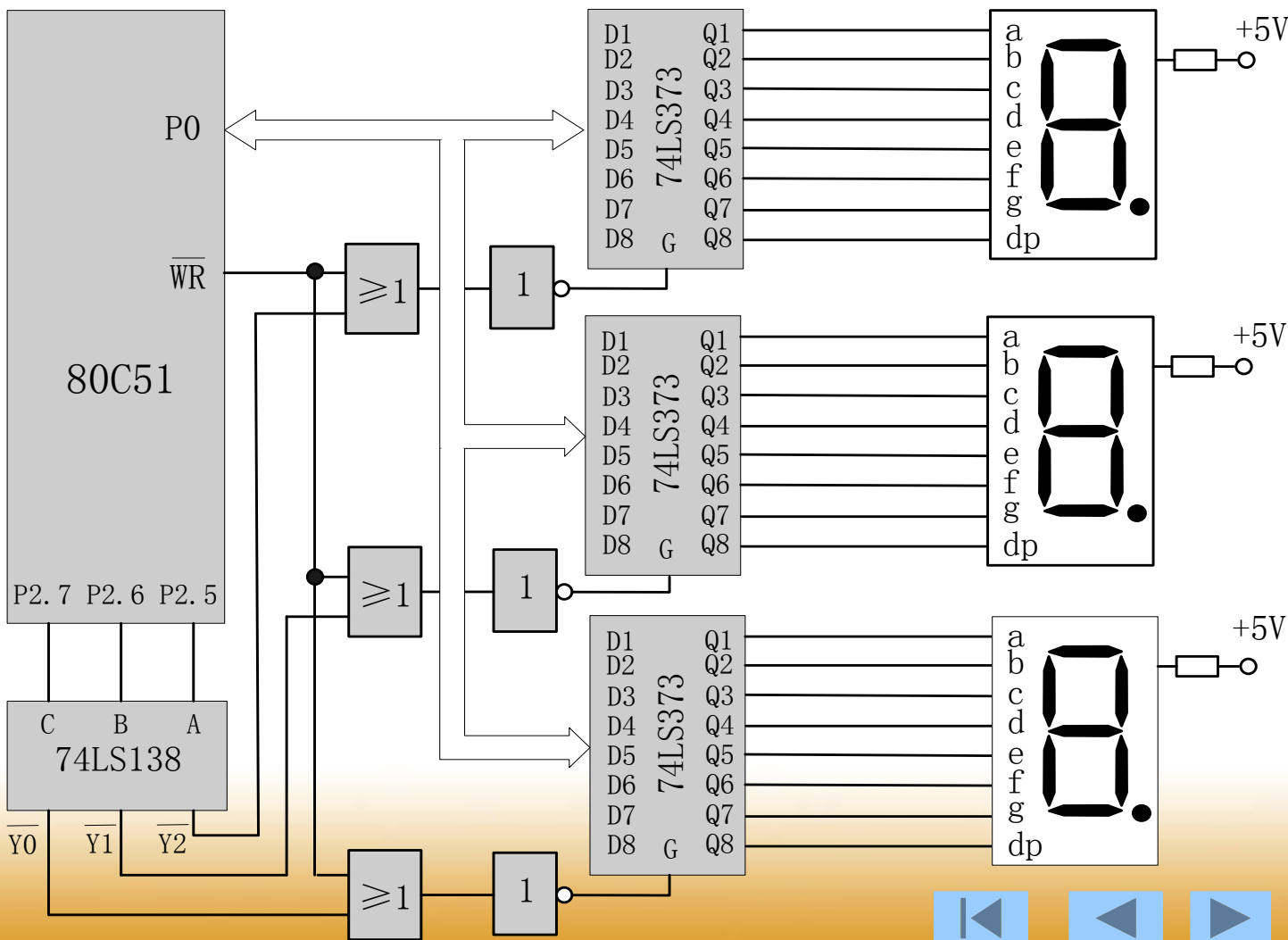
#### 一、七段显示器的原理



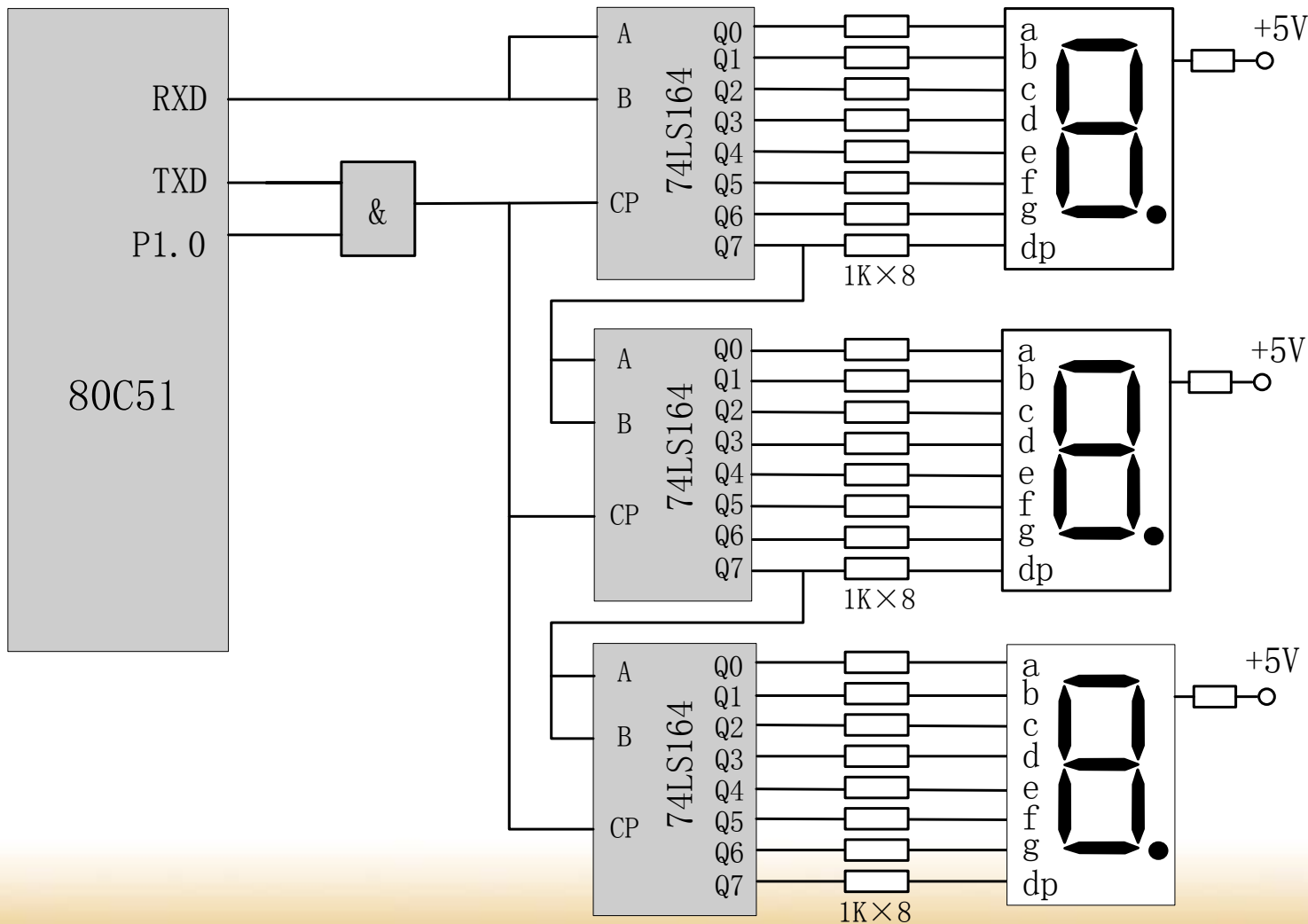
## 二、显示方式及接口

### 1、静态显示

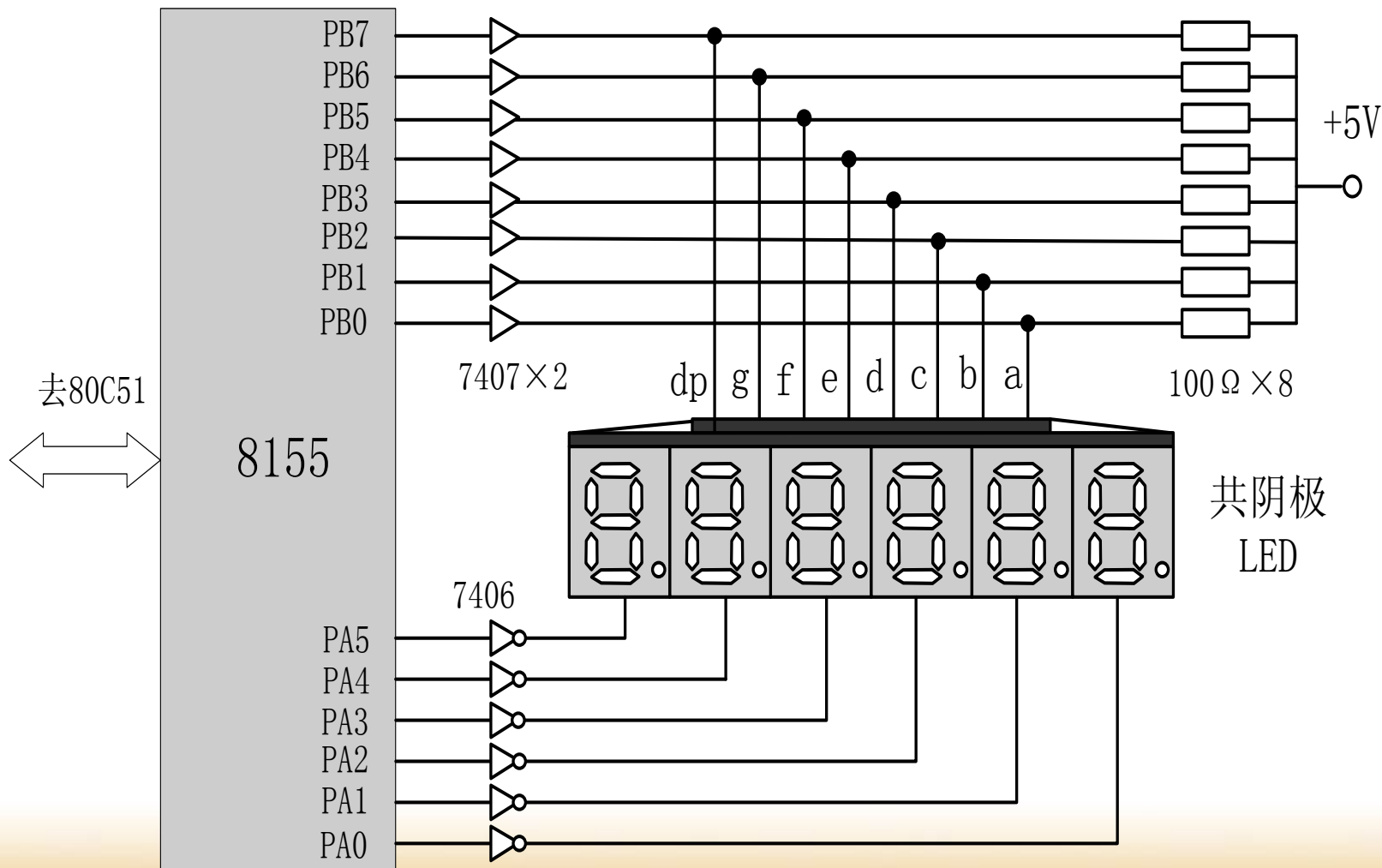
并行输出:



串行输出:



## 2、动态显示

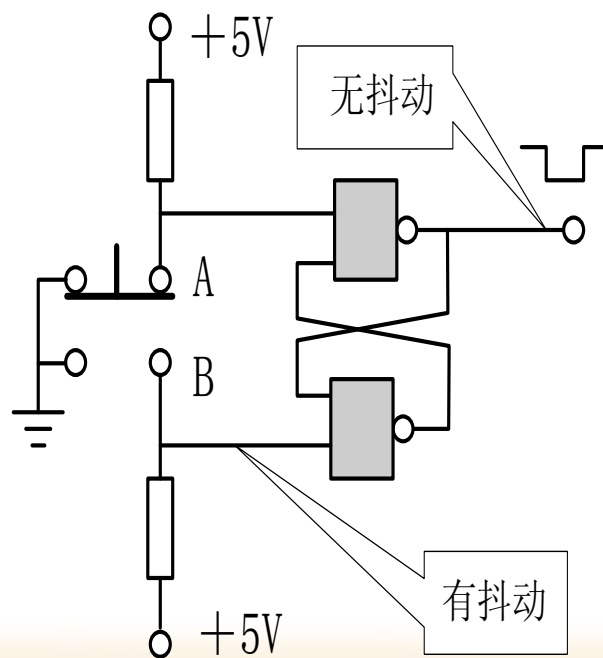
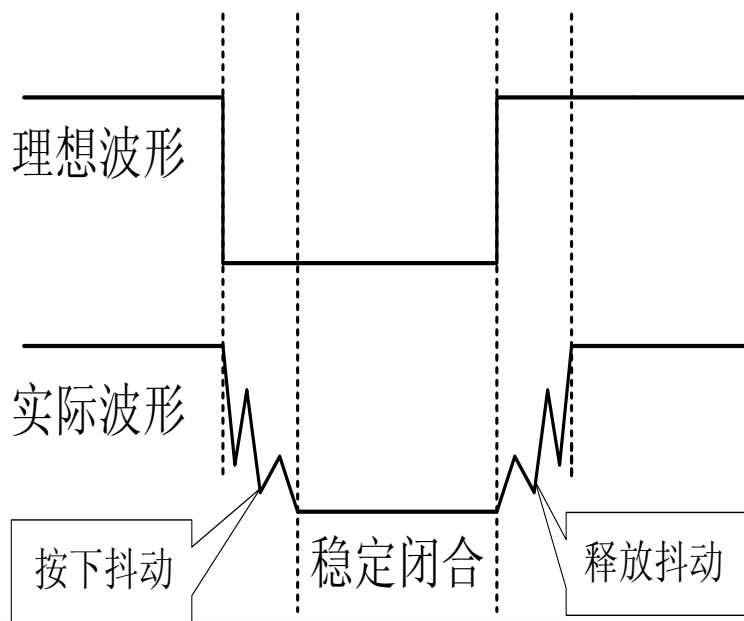


```
DIS: MOV R0, #79H      ; 显示数据缓冲区首地址送R0
      MOV R3, #01H     ; 使显示器最右边位亮
      MOV A, R3        ;
LD0:  MOV DPTR, #7F01H ; 数据指针指向A口
      MOVX @DPTR, A    ; 送扫描值
      INC DPTR         ; 数据指针指向B口
      MOV A, @R0       ; 取欲显示的数据
      ADD A, #0DH      ; 加上偏移量
      MOVC A, @A+PC    ; 取出字型码
      MOVX @DPTR, A    ; 送显示
      ACALL DL1        ; 调用延时子程序
      INC R0           ; 指向下一个显示段数据地址
      MOV A, R3        ;
      JB ACC.5, ELD1   ; 扫描到第六个显示器否?
      RL A             ; 未到, 扫描码左移1位
      MOV R3, A
      AJMP LD0
ELD1: RET
```

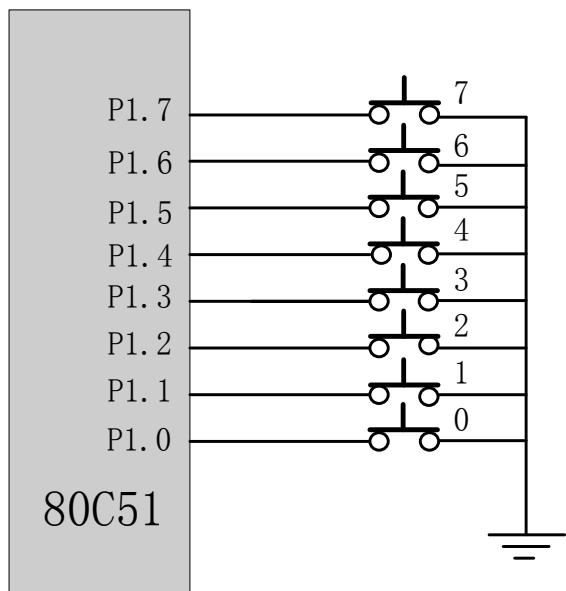
```
DSEG: DB 3FH, 06H, 5BH, 4FH, 66H, 6DH
       DB 7DH, 07H, 7FH, 6FH, 77H, 7CH
       DB 39H, 5EH, 79H, 71H, 40H, 00H
DL1:  MOV R7, #02H    ; 延时1ms子程序
      DL: MOV R6, #0FFH
DL6:  DJNZ R6, DL6
      DJNZ R7, DL
      RET
```

## 7.5.2 键盘及其接口

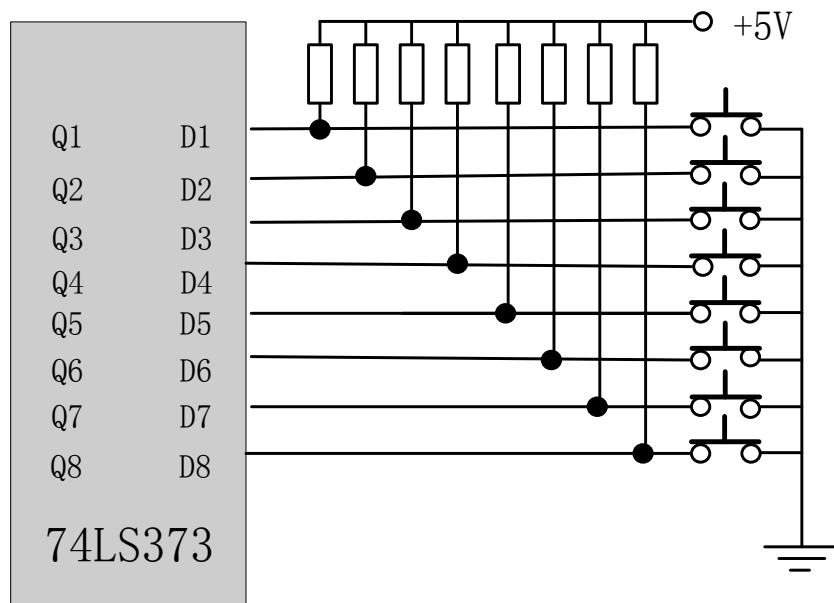
按键在闭合和断开时，触点会存在抖动现象：



# 一、独立式按键及其接口



芯片内有上拉电阻



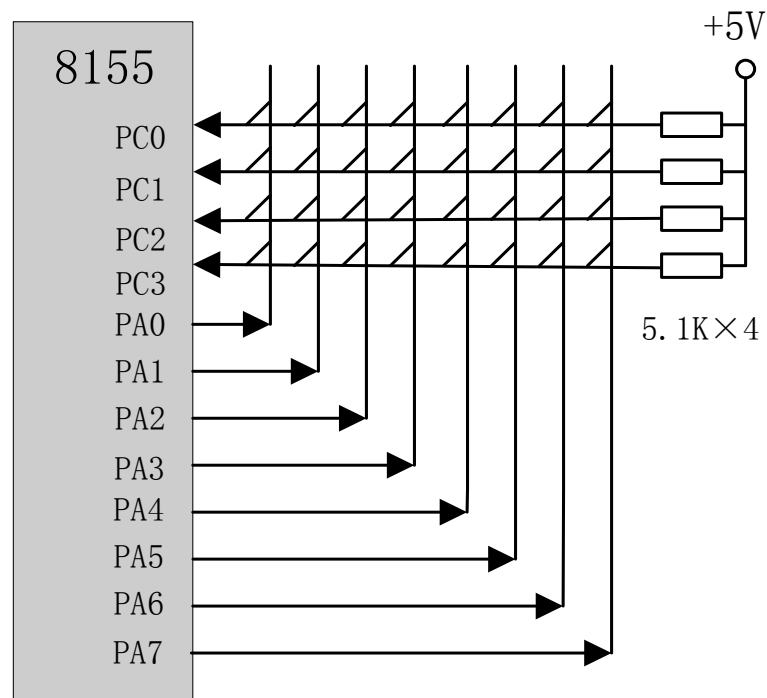
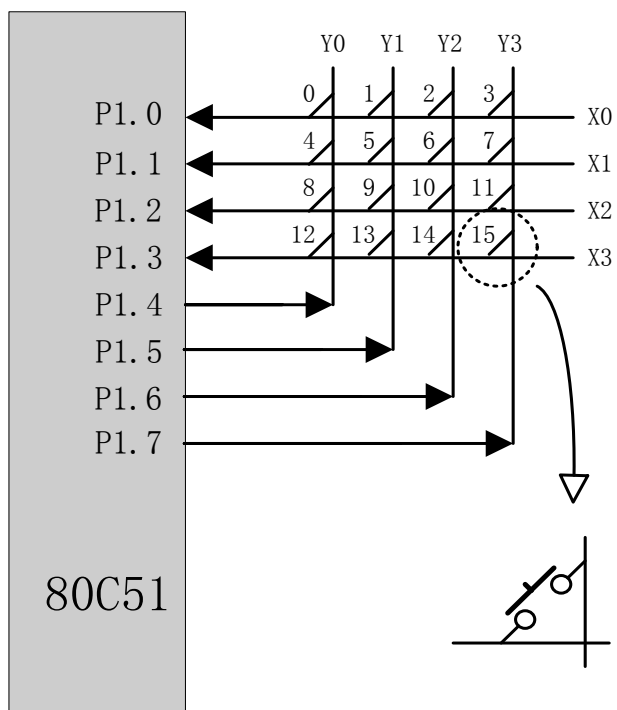
芯片内无上拉电阻



随机扫描程序（也可以用定时扫描或中断扫描）如下：

```
SMKEY: ORL  P1, #0FFH ; 置P1口为输入方式
        MOV  A, P1    ; 读P1口信息
        JNB  ACC.0, P0F ; 0号键按下, 转0号键处理
        JNB  ACC.1, P1F ; 1号键按下, 转1号键处理
        ... ..
        JNB  ACC.7, P7F ; 7号键按下, 转7号键处理
        LJMP SMKEY
P0F:    LJMP  PROG0
P1F:    LJMP  PROG1
        ... ..
P7F:    LJMP  PROG7
PROG0:  ... ..
        LJMP  SMKEY
PROG1:  ... ..
        LJMP  SMKEY
        ... ..
PROG7:  ... ..
        LJMP  SMKEY
```

## 二、矩阵式键盘及其接口



判有无键按下；  
判按下的是哪一键；  
键处理。

**SMKEY: MOV P1, #0FH ; 置P1口高4位为“0”、低4位为输入状态**

**MOV A, P1 ; 读P1口**

**ANL A, #0FH ; 屏蔽高4位**

**CJNE A, #0FH, HKEY ; 有键按下, 转HKEY**

**SJMP SMKEY ; 无键按下转回**

**HKEY: LCALL DELAY10 ; 延时10ms, 去抖**

**MOV A, P1 ;**

**ANL A, #0FH ;**

**CJNE A, #0FH, WKEY ; 确认有键按下, 转判哪一键按下**

**SJMP SMKEY ; 是抖动转回**

**WKEY: MOV P1, #1110 1111B ; 置扫描码, 检测P1.4列**

**MOV A, P1 ;**

**ANL A, #0FH ;**

**CJNE A, #0FH, PKEY ; P1.4列 (Y0) 有键按下, 转键处理**

```
MOV P1, #1101 1111B ; 置扫描码, 检测P1.5列
MOV A, P1           ;
ANL A, #0FH        ;
CJNE A, #0FH, PKEY ; P1.5列 (Y1) 有键按下, 转键处理
MOV P1, #1011 1111B ; 置扫描码, 检测P1.6列
MOV A, P1           ;
ANL A, #0FH        ;
CJNE A, #0FH, PKEY ; P1.6列 (Y2) 有键按下, 转键处理
MOV P1, #0111 1111B ; 置扫描, 检测P1.7列
MOV A, P1           ;
ANL A, #0FH        ;
CJNE A, #0FH, PKEY ; P1.7列 (Y3) 有键按下, 转键处理
LJMP SMKEY          ;
PKEY: ... ..       ; 键处理
```

线反转法示例程序：

```
SMKEY: MOV P1, #0FH      ; 置P1口高4位为“0”、低4位为输入状态
        MOV A, P1        ; 读P1口
        ANL A, #0FH      ; 屏蔽高4位
        CJNE A, #0FH, HKEY ; 有键按下, 转HKEY
        SJMP SMKEY       ; 无键按下转回
HKEY:   LCALL DELAY10    ; 延时10ms, 去抖
        MOV A, P1        ;
        ANL A, #0FH      ;
        MOV B, A         ; 行线状态在B的低4位
        CJNE A, #0FH, WKEY ; 确认有键按下, 转判哪一键按下
        SJMP SMKEY       ; 是抖动转回
WKEY:   MOV P1, #0F0H    ; 置P1口高4位为输入、低4位为“0”
        MOV A, P1        ;
        ANL A, #0F0H     ; 屏蔽低4位
        ORL A, B         ; 列线状态在高4位, 与行线状态合成于B中
        ... ..         ; 键处理
```

• **键处理**。键处理是根据所按键散转进入相应的功能程序。为了散转的方便，通常应先得到按下键的键号。键号是键盘的每个键的编号，可以是**10进制**或**16进制**。键号一般通过键盘扫描程序取得的键值求出。键值是各键所在行号和列号的组合码。如图所  
示接口电路中的键“9”所在行号为**2**，所在列号为**1**，键值可以表示为“**21H**”（也可以表示为‘**12H**’，表示方法并不是唯一的，要根据具体按键的数量及接口电路而定）。根据键值中行号和列号信息就可以计算出键号，如：

• 键号 = 所在行号 × 键盘列数 + 所在列号，

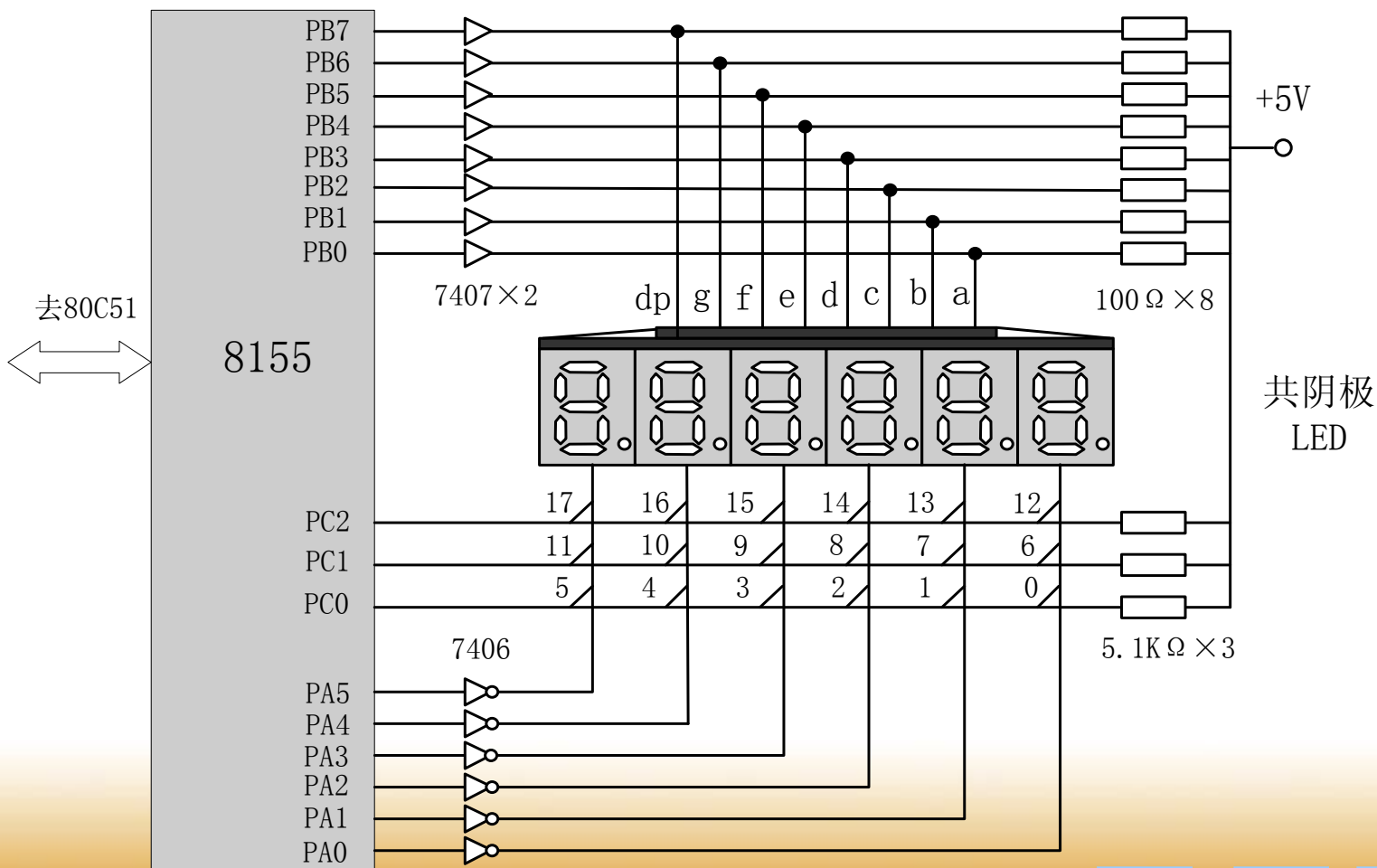
$$\text{即 } 2 \times 4 + 1 = 9$$

根据键号就可以方便地通过散转进入相应键的功能程序。



# 7.5.3 键盘和显示器接口示例

## 一、8155的键盘及显示接口



```
KD1:MOV A, #00000011B; 8155初始化: PA、PB为基本输出, PC为输入  
MOV DPTR, #7F00H ;  
MOVX @DPTR, A ;  
KEY1:ACALL KS1 ; 查有无键按下  
JNZ LK1 ; 有, 转键扫描  
ACALL DIS ; 调显示子程序  
AJMP KEY1 ;  
LK1: ACALL DIS ; 键扫描  
ACALL DIS ; 两次调显示子程序, 延时12ms  
ACALL KS1 ;  
JNZ LK2 ;  
ACALL DIS ; 调显示子程序  
AJMP KEY1
```



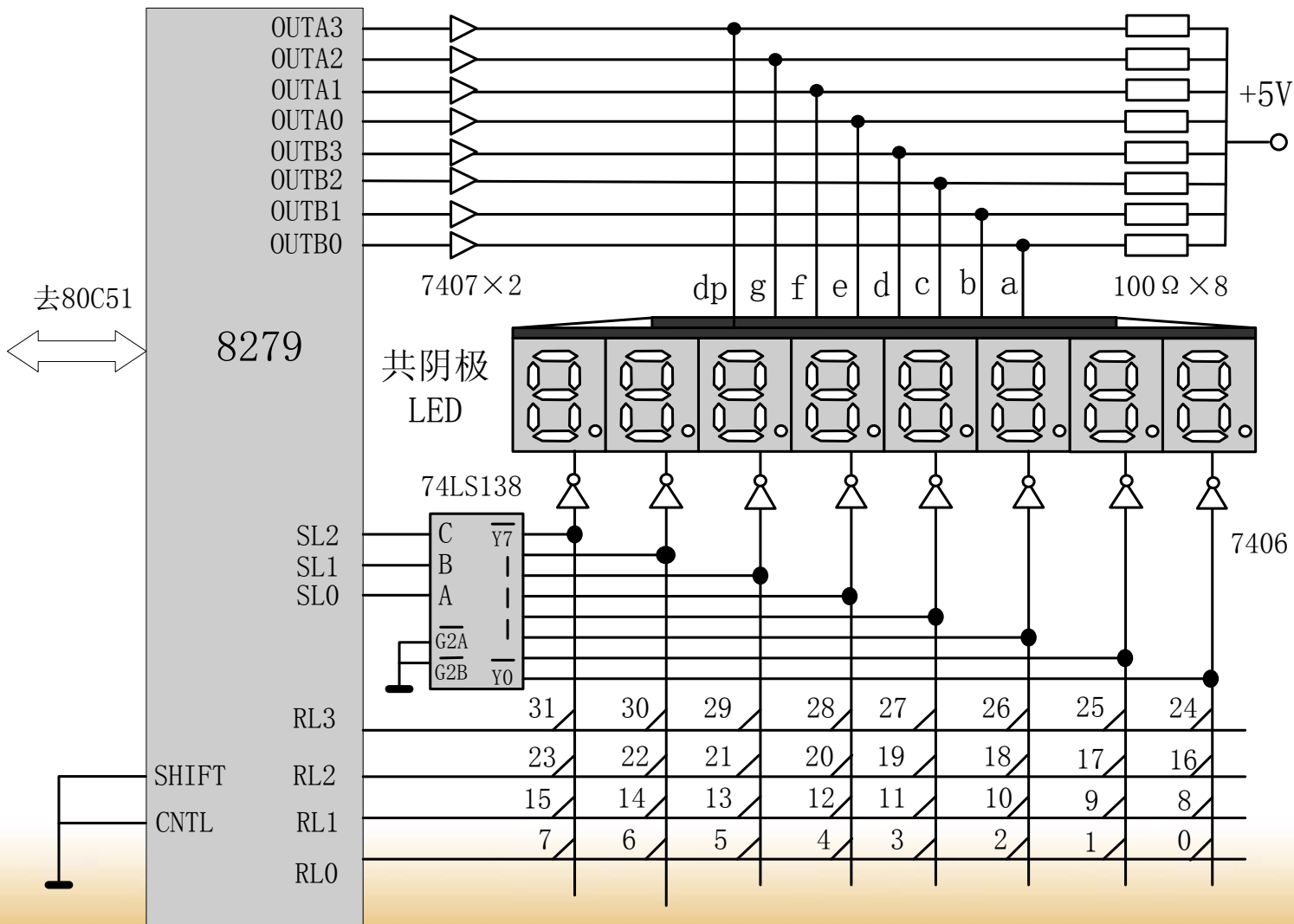
```
LK2: MOV R2,#0FEH ; 从首列开始
      MOV R4, #00H ; 首列号送R4
LK4: MOV DPTR,#7F01H ;
      MOV A, R2 ;
      MOVX A, @DPTR ;
      INC DPTR ;
      INC DPTR ; 指向C口
      MOVX @DPTR, A ;
      JB ACC.0, LONE ; 第0行无键按下, 转查第1行
      MOV A, #00H ; 第0行有键按下, 该行首键号送A
      AJMP LKP ; 转求键号
LONE: JB ACC.1, LTWO ; 第1行无键按下, 转查第2行
      MOV A, #08H ; 第1行有键按下, 该行首键号送A
      AJMP LKP ; 转求键号
LTWO: JB ACC.2, NEXT ; 第2行无键按下, 转查下一列
      MOV A, #10H ; 第2行有键按下, 该行首键号送A
LKP: ADD A, R4 ; 求键号。键号=行首键号+列号
      PUSH ACC ; 保护键号
LK3: ACALL DIS ; 等待键释放
      ACALL KS1 ;
      JNZ LK3 ;
      POP ACC ;
      RET ; 键扫描结束。此时A的内容为按下键的键号
```



```
NEXT:INC R4      ; 指向下一列
      MOV A, R2   ;
      JNB ACC.5, KND ; 判6列扫描完没有。
      RL A        ; 未完, 扫描字对应下一列
      MOV R2, A   ;
      AJMP LK4    ; 转下一列扫描
KND: AJMP KEY1   ; 扫完, 转入新一轮扫描
KS1: MOV DPTR, #7F01H ; 查有无键按下子程序。先指向A口
      MOV A, #00H ;
      MOVX @DPTR, A ; 送扫描字“00H”
      INC DPTR    ;
      INC DPTR    ; 指向C口
      MOVX A, @DPTR ;
      CPL A       ; 变正逻辑
      ANL A,#0FH ; 屏蔽高位
      RET        ; 子程序出口, A的内容非0则有键按下
```



## 二、8279的键盘及显示接口



初始化程序如下：

```
INIT:MOV DPTR, #7FFFH ; 置8279命令/状态口地址
      MOV A, #0D1H ; 置清显示命令字
      MOVX @DPTR, A ; 送清显示命令
WEIT:MOVX A, @DPTR ; 读状态
      JB ACC.7, WEIT ; 等待清显示RAM结束
      MOV A, #34H ; 置分频系数, 晶振12MHz
      MOVX @DPTR, A ; 送分频系数
      MOV A, #00H ; 置键盘/显示命令
      MOVX @DPTR, A ; 送键盘/显示命令
      MOV IE, #84H ; 允许8279中断
      RET
```

显示子程序如下：

```
DIS:MOV DPTR, #7FFFH ; 置8279命令/状态口地址
MOV R0, #30H ; 字段码首地址
MOV R7, #08H ; 8位显示
MOV A, #90H ; 置显示命令字
MOVX @DPTR, A ; 送显示命令
MOV DPTR, #7FFE0H ; 置数据口地址
LP:MOV A, @R0 ; 取显示数据
ADD A, #6 ; 加偏移量
MOVC A, @A+PC ; 查表, 取得数据的段码
MOVX @DPTR, A ; 送段码显示
INC R0 ; 调整数据指针
DJNZ R7, LP ;
RET
```

**SEG:DB 3FH, 06H, 5BH, 4FH, 66H, 6DH**

； 字符0、1、2、3、4、5段码

**DB 7DH, 07H, 7EH, 6FH, 77H, 7CH**

； 字符6、7、8、9、A、b段码

**DB 39H, 5EH, 79H, 71H, 73H, 3EH**

； 字符C、d、E、F、P、U段码

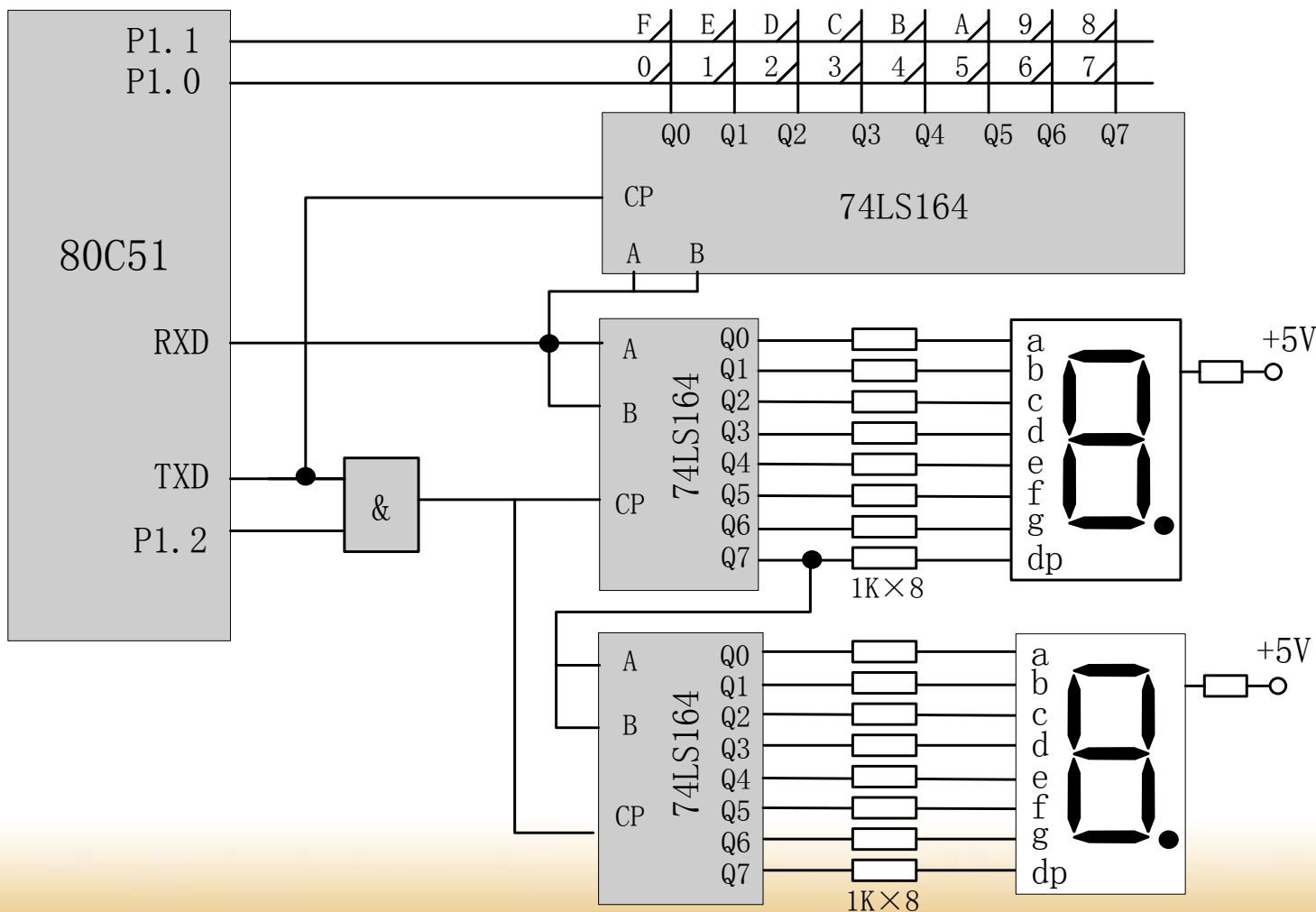
**DB 76H, 38H, 40H, 6EH, FFH, 00H**

； 字符H、L、-、Y、日、“空”段码

键盘中断子程序如下:

```
KEY:PUSH PSW
      PUSH DPL
      PUSH DPH
      PUSH ACC
      PUSH B
      SETB PSW.3
      MOV DPTR, #7FFFH ; 置状态口地址
      MOVX A, @DPTR ; 读FIFO状态
      ANL A, #0FH ;
      JZ PKYR ;
      MOV A, #40H ; 置读FIFO命令
      MOVX @DPTR, A ; 送读FIFO命令
      MOV DPTR, #7FFEh ; 置数据口地址
      MOVX A, @DPTR ; 读数据
      LJMP KEY1 ; 转键值处理程序
PKYR: POP B
      POP ACC
      POP DPH
      POP DPL
      POP PSW
      RETI ;
KEY1: ... .. ; 键值处理程序
```

### 三、串行口键盘及显示接口电路





## 思考题与习题

- 1、试以80C31为主机，用2片2764 EPROM扩展16K ROM，画出硬件接线图。
- 2、设计扩展2KB RAM和4KB EPROM的电路图。
- 3、当单片机应用系统中数据存储器RAM地址和程序存储器EPROM地址重叠时，是否会发生数据冲突，为什么？
- 4、80C51单片机在应用中P0和P2是否可以直接作为输入/输出连接开关、指示灯等外围设备？
- 5、七段LED显示器有动态和静态两种显示方式，这两种显示方式要求80C51系列单片机如何安排接口电路？
- 6、设计80C51键盘显示接口，采用中断扫描方式扩展3×6共18个键分别为0~9、A~F、RUN和RESET键，具体要求如下：
  - a、按下RESET键后，单片机复位。
  - b、按下RUN键后，系统进入地址为2000H的用户程序。
  - c、按下0~9、A~F键后，键值存入内部RAM，首地址为40H。试画出接口电路的硬件连接图并编写相应程序。
- 7、试编写图7-27所示接口电路的实现程序