

# 第8章 并行通信及其接口电路





# 第8章 并行通信及其接口电路

## 8.1 并行通信概述

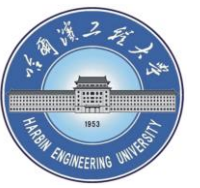
## 8.2 简单并行接口电路

## 8.3 可编程并行接口电路

## 8.4 可编程并行接口芯片8255A

## 8.5 8255A芯片的应用

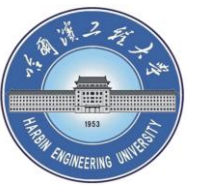
# 学习目的



通过对本章的学习，应该能够达到下列要求：

- 并行通信的概念
- 8255A芯片的编程结构
- 8255A控制器的应用

# 学习目的



## 重点

- 并行通信的概念
- 8255A芯片的结构和命令
- 8255A芯片的工作方式
- LED七段码显示
- 8255A芯片应用编程



# 8.1 并行接口电路概述

- 在计算机和数据通信系统中，有两种基本的数据传送方式，即串行数据传送方式和并行数据传送方式，也称**串行通信和并行通信**。
- 数据在单条一位宽的传输线上按时间先后一位一位地进行传送，称为**串行传送**；
- 数据在多位宽的传输线上各位同时进行传送，称为**并行传送**。
- 和串行传送相比，在同样的时钟速率下，并行传送的数据传输率较高。



## 8.2 简单的并行接口电路

- 在输入输出接口电路中，经常要对所传送的信号进行缓冲、驱动和锁存。
- 能实现这种功能的接口芯片通常是简单的**数据锁存器、缓冲器以及双向总线收发器等。**
- 三种常用的简单并行输入输出接口芯片：
  - **锁存器74LS373；**
  - **缓冲器74LS244；**
  - **数据总线收发器74LS245。**



## 8.3 可编程并行接口电路

### 8.3.1 可编程并行接口的组成及其与CPU和外设的连接

- 通常，一个可编程并行接口电路应包括下列组成部分：
  - (1) 两个或两个以上具有缓冲能力的数据寄存器。
  - (2) 可供CPU访问的控制及状态寄存器。
  - (3) 片选和内部控制逻辑电路。



## 8.3 可编程并行接口电路

(4)与外设进行数据交换的控制与联络信号线。

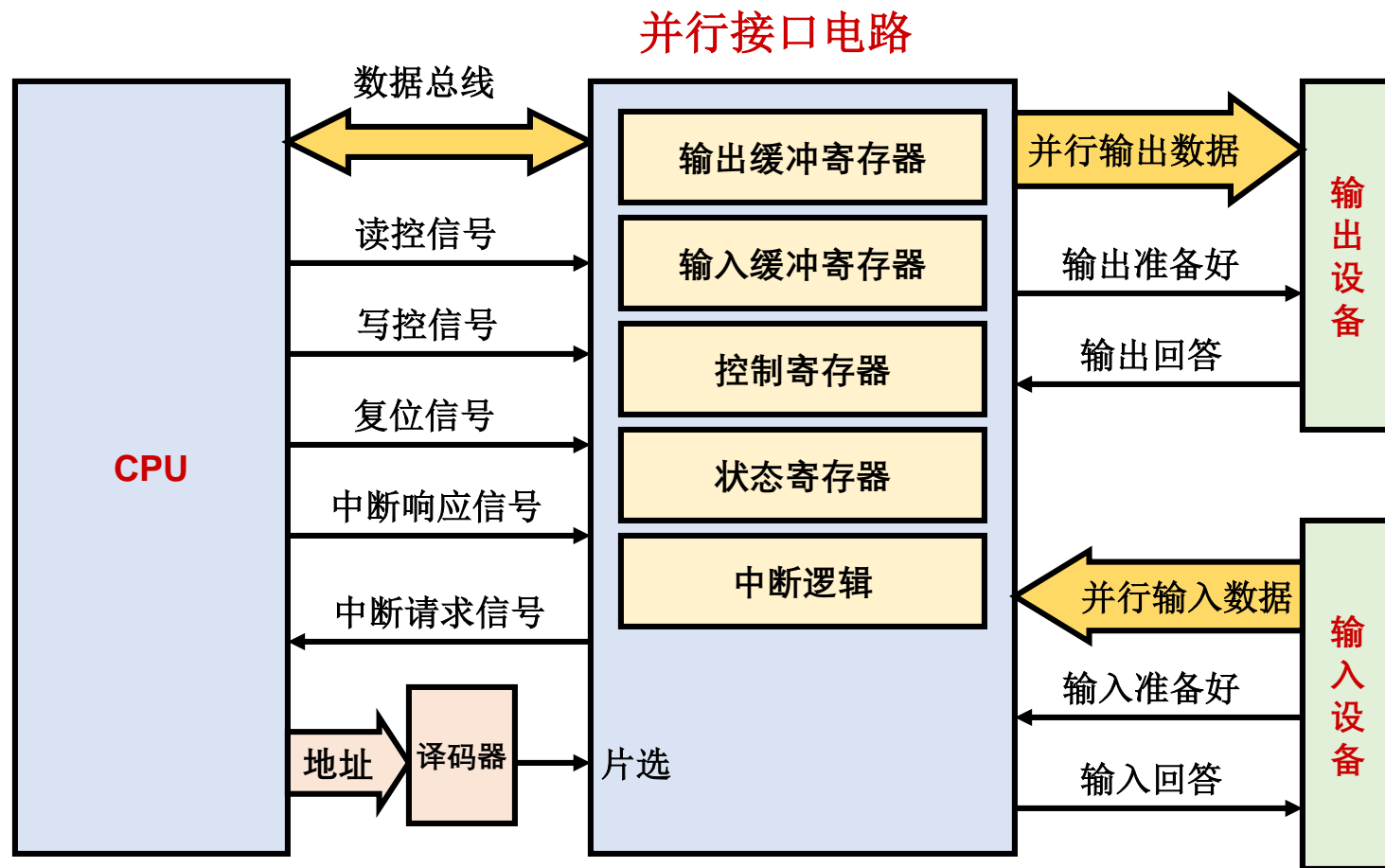
(5)与CPU用中断方式传送数据的相关中断控制电路。

- 典型的可编程并行接口及其与CPU和外设的连接示意图如图8.1所示。



# 8.3 可编程并行接口电路

图8.1 可编程并行接口电路及其与CPU和外设的连接





## 8.3 可编程并行接口电路

---

- 8.3.2 可编程并行接口的数据输入输出过程

将以8255A为例进行讨论。



# 8.4 可编程并行接口芯片8255A

## 8.4.1 8255A芯片概要

- **Intel 8255A**是一个为Intel 8080和8086微机系统设计的通用可编程并行接口芯片，也可应用于其他微机系统之中。
- 8255A采用**40脚**双列直插封装，单一+5V电源，全部输入输出与TTL电平兼容。
- 用8255A连接外部设备时，通常不需要再附加其他电路，给使用带来很大方便。



## 8.4 可编程并行接口芯片8255A

- 8255A 有三个输入输出端口：**端口A、端口B、端口C。**
- 每个端口都可通过编程设定为**输入端口或输出端口**，但有各自不同的方式和特点。
- **端口C**可作为一个独立的端口使用，但通常是**配合端口A和端口B的工作**，为这两个端口的输入输出提供**控制联络信号**。



# 8.4 可编程并行接口芯片8255A

## • 8.4.2 8255A芯片引脚分配及引脚信号说明

8255A芯片引脚分配如图8.2所示。

■ 8255A芯片的40条引脚，大致可分为三类：

(1) 电源与地线共2条：Vcc、GND。

(2) 与外设相连的共24条：

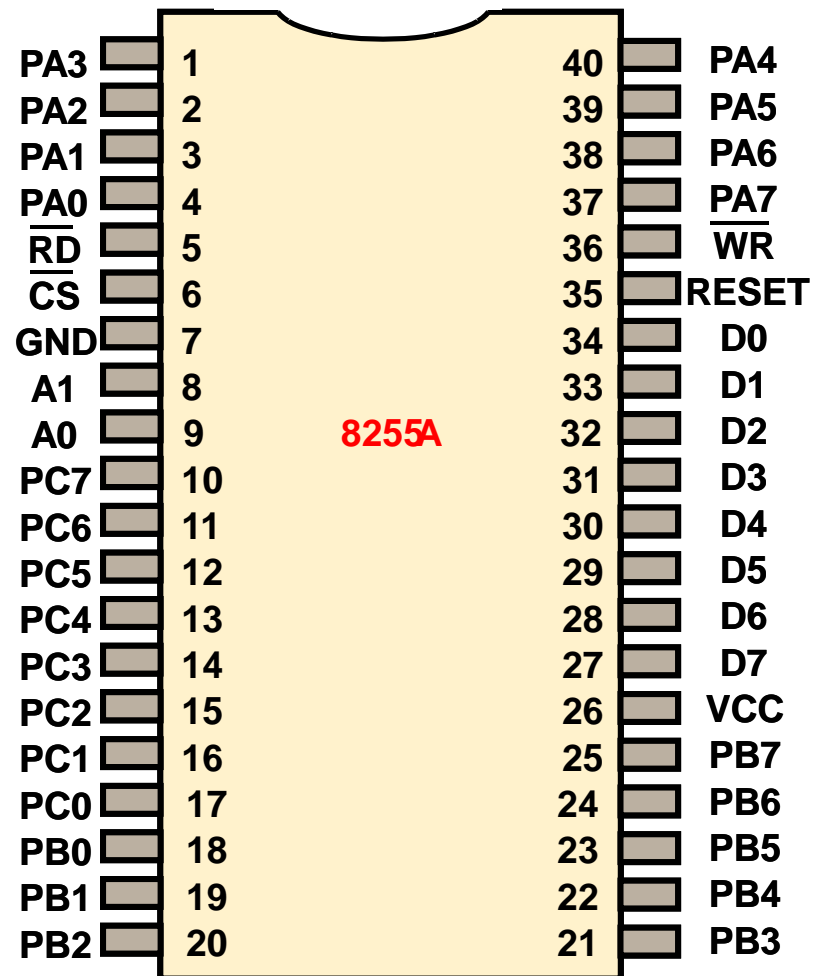
□PA7~PA0： 端口A数据信号。

□PB7~PB0： 端口B数据信号。

□PC7~PC0： 端口C数据信号。

# 8.4 可编程并行接口芯片8255A

图8.2 8255A芯片引脚分配





## 8.4 可编程并行接口芯片8255A

### (3) 与CPU相连的共14条：

- **RESET**: 复位信号，高电平有效。当RESET信号有效时，所有内部寄存器都被清除。同时，3个数据端口被自动设置为**输入端口**。
- **D7~D0**: 双向数据线，在8080系统中，8255A的D7~D0与系统的**8位**数据总线相连；
- 在8086系统中，采用16位数据总线，8255A的D7~D0通常是接在16位数据总线的**低8位**上。



## 8.4 可编程并行接口芯片8255A

- **$\overline{CS}$** : 片选信号，低电平有效。该信号来自译码器的输出，只有当 $\overline{CS}$ 有效时，读信号 $\overline{RD}$ 和写信号 $\overline{WR}$ 才对8255A有效。
- **$\overline{RD}$** : 读信号，低电平有效。它控制从8255A读出数据或状态信息。
- **$\overline{WR}$** : 写信号，低电平有效。它控制把数据或控制命令字写入8255A。





## 8.4 可编程并行接口芯片8255A

- **A1、A0**： 端口选择信号。8255A内部共有4个端口(即寄存器)： 3个数据端口(端口A、端口B、端口C) 和1个控制端口,当片选信号 $\overline{CS}$ 有效时,规定A1、A0为**00、01、10、11**时,分别选中**端口A、端口B、端口C和控制端口**。
- $\overline{CS}$ 、 $\overline{RD}$ 、 $\overline{WR}$ 、A1、A0 这五个信号的组合决定了对三个数据端口和一个控制端口的读写操作,如表8-1所示。



# 8.4 可编程并行接口芯片8255A

表8-1 8255A端口选择和基本操作

A1	A0	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	输入操作 (读)
0	0	0	1	0	端口A→数据总线
0	1	0	1	0	端口B→数据总线
1	0	0	1	0	端口C→数据总线
					输出操作 (写)
0	0	1	0	0	数据总线→端口A
0	1	1	0	0	数据总线→端口B
1	0	1	0	0	数据总线→端口C
1	1	1	0	0	数据总线→控制字寄存器
					无操作情况
X	X	X	X	1	数据总线为三态 (高阻)
1	1	0	1	0	非法操作
X	X	1	1	0	数据总线为三态 (高阻)



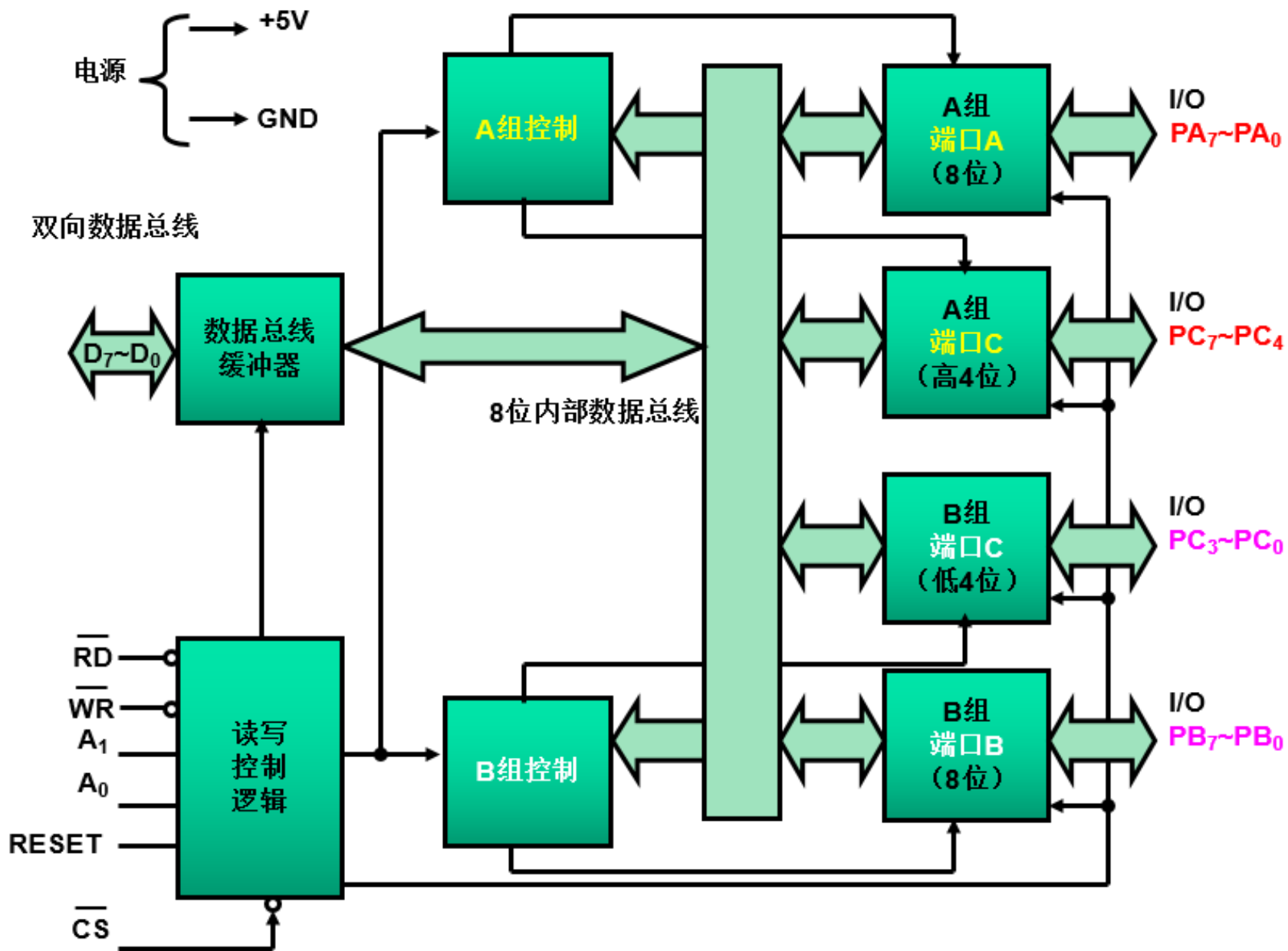
## 8.4 可编程并行接口芯片8255A

---

- 8.4.3 8255A内部结构方块图

- 8255A内部结构方块图如图8.3所示。

图8.3 8255A内部结构方块图





## 8.4 可编程并行接口芯片8255A

- 由图8.3可以看出，8255A由以下几部分组成：

### (1) 数据总线缓冲器

- 这是一个双向三态8位数据缓冲器，它是8255A与CPU数据总线的接口。
- 输入数据、输出数据以及CPU发给8255A的控制字和从8255A读出的状态信息都是通过该缓冲器传送的。



## 8.4 可编程并行接口芯片8255A

### (2) 端口A、端口B、端口C

- 8255A有三个8位端口(端口A、端口B、端口C)，各端口可由程序设定为输入端口或输出端口。
- 在使用中，端口A和端口B常常作为独立的输入端口或输出端口。
- 端口C也可以作为输入端口或输出端口，但往往是用来配合端口A和端口B的工作。
- 在方式字的控制下，端口C可以分成两个4位的端口，分别用来为端口A和端口B提供控制和状态信息。



## 8.4 可编程并行接口芯片8255A

### (3) A组控制和B组控制

- A组控制——控制端口A及端口C的**高4位**。
- B组控制——控制端口B及端口C的**低4位**。

### (4) 读写控制逻辑

- 读写控制逻辑负责管理8255A的数据传输过程。



# 8.4 可编程并行接口芯片8255A

---

- 8.4.4 8255A的控制字

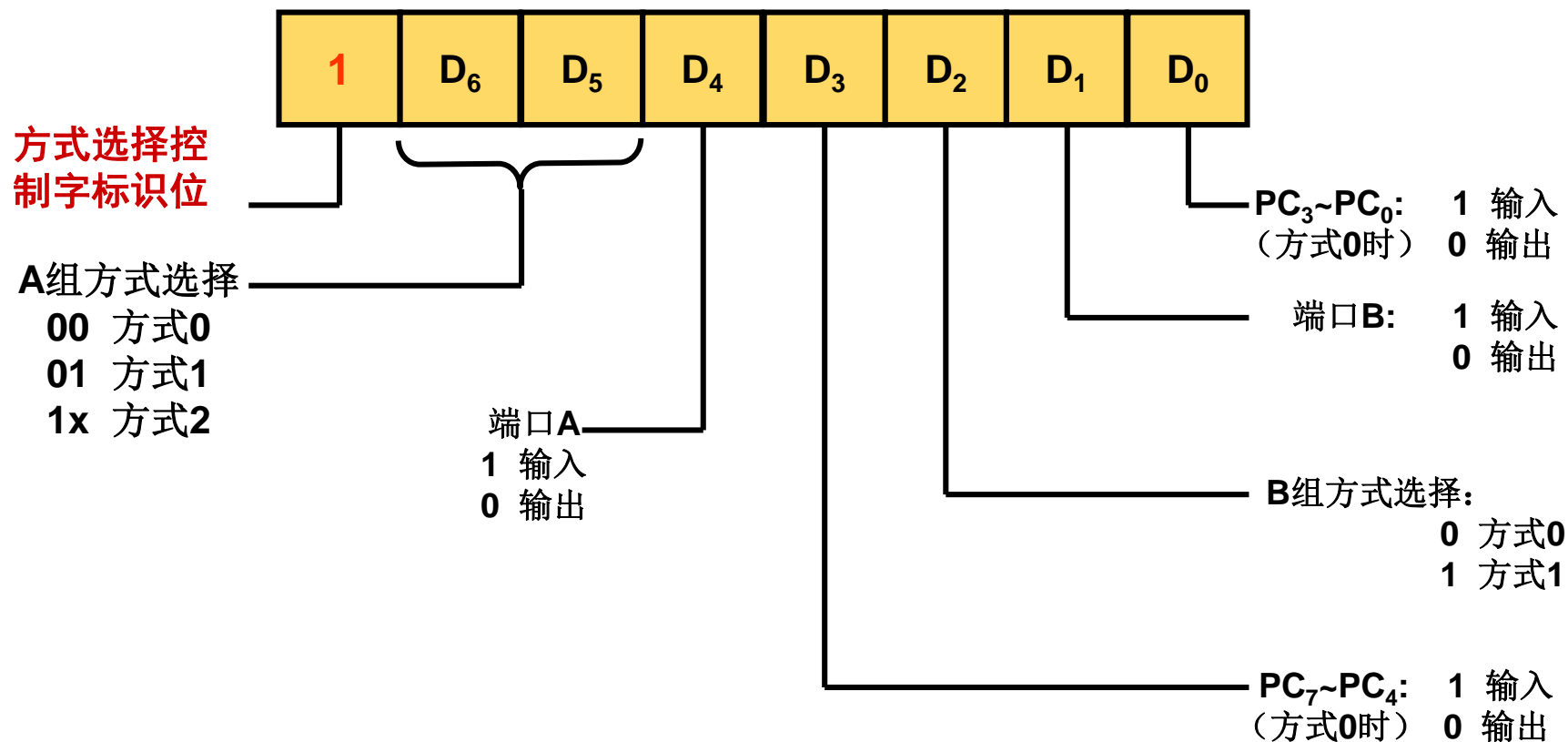
## 1. 方式选择控制字

- 方式选择控制字的格式如图8.4所示。



# 8.4 可编程并行接口芯片8255A

图8.4 8255A方式选择控制字





## 8.4 可编程并行接口芯片8255A

- 假定要求8255A的各个端口工作于如下方式：
- 端口A——方式0，输出；
- 端口B——方式0，输入；
- 端口C的高4位——方式0，输出；
- 端口C的低4位——方式0，输入。
- 那么，相应的方式选择控制字应为**10000011B(83H)**。



## 8.4 可编程并行接口芯片8255A

- 在实验系统中8255A控制口的地址为**28BH**，则执行如下三条指令即可实现上述工作方式的设定。

```
MOV DX, 28BH
```

```
MOV AL, 83H
```

```
OUT DX, AL; 将方式选择控制字写入控制口
```



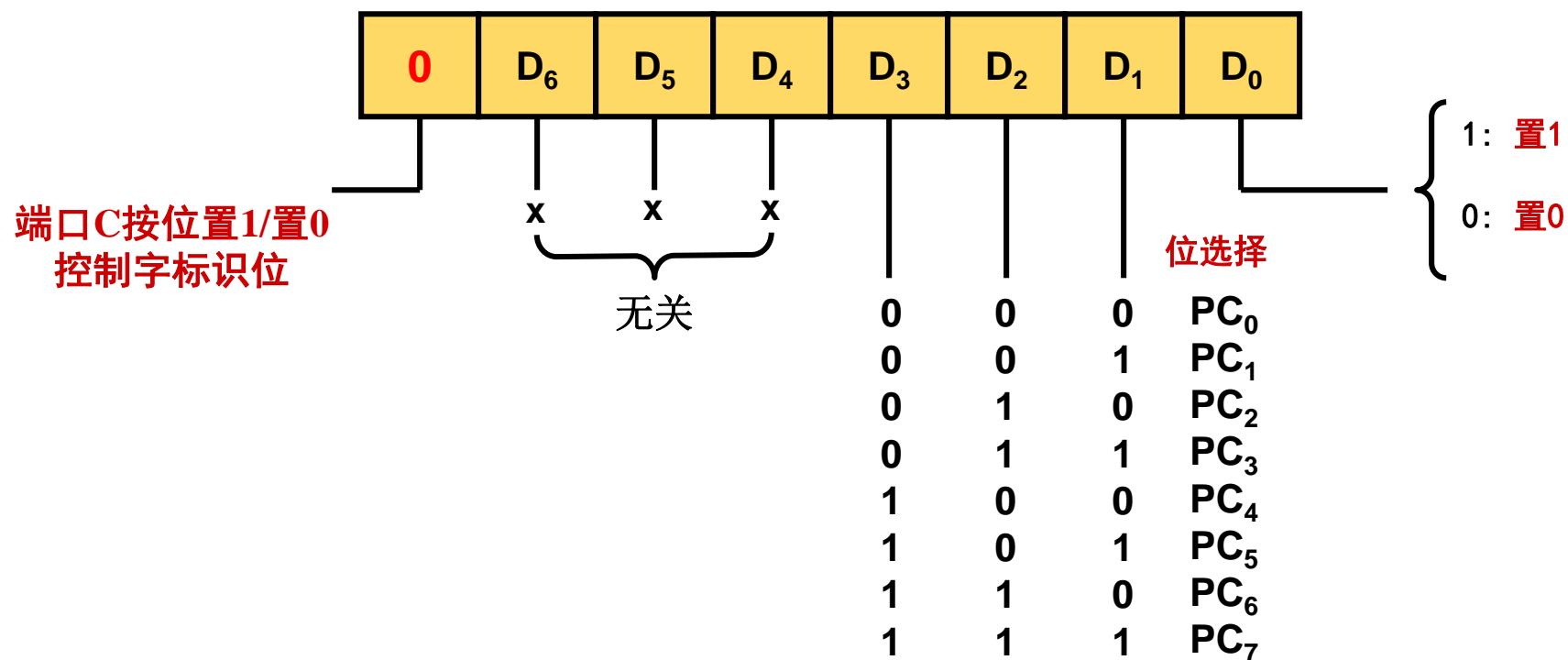
# 8.4 可编程并行接口芯片8255A

## 2. 端口C按位置1 / 置0控制字

- 可以用专门的**控制字**实现对**端口C按位置1 / 置0**操作，用以产生所需的控制功能，这种控制字就是“**端口C按位置1 / 置0控制字**”。
- 该控制字的具体格式如图**8.6**所示。

# 8.4 可编程并行接口芯片8255A

图8.6 端口C按位置1/置0控制字





## 8.4 可编程并行接口芯片8255A

- 需要指出的是，端口C按位置1 / 置0控制字是对端口C的操作控制信息，因此**该控制字必须写入控制口，而不应写入端口C。**
- 控制字的**D0位**决定是置“1”操作还是置“0”操作，但究竟是对端口C的哪一位进行操作，则决定于控制字中的**D3、D2、D1位**。



## 8.4 可编程并行接口芯片8255A

例如，要实现 对端口 C 的 PC6 位置 “0”，则控制字应为 **00001100B(0CH)**。

设8255A的控制口地址同上，则执行下列指令即可实现指定的功能：

**MOV DX, 28BH**

**MOV AL, 0CH**

**OUT DX, AL** ； 将 “端口C按位置1 / 置0控制字” 写入控制口，实现对 PC6位置 “0”



# 8.4 可编程并行接口芯片8255A

## 8.4.5 8255A的工作方式

### 1. 方式0

- 方式0也叫**基本输入 / 输出方式**。
- 在这种方式下，端口A和端口B可以通过**方式选择控制字**规定为输入口或者输出口；
- 端口C分为高4位(PC7~PC4) 和低4位(PC3~PC0) 两个4位端口，这两个4位端口也可由方式选择控制字分别规定为输入口或输出口。





## 8.4 可编程并行接口芯片8255A

- 这四个并行口共可构成 $2^4=16$ 种不同的使用组态。
- 利用8255A的**方式0**进行数据传输时，由于没有规定专门的应答信号，所以这种方式常用于与简单外设之间的数据传送，如向LED显示器的输出，从二进制开关装置的输入等。



# 8.4 可编程并行接口芯片8255A

## 2. 方式1

- 方式1也叫**选通的输入 / 输出方式**。
- 和方式0相比，最主要的差别就是当端口A和端口B工作于方式1时，要利用**端口C**来接收选通信号或提供有关的状态信号，而这些信号是由端口C的固定数位来接收或提供的，即信号与数位之间存在着对应关系。
- **这种关系不可以用程序的方法予以改变。**



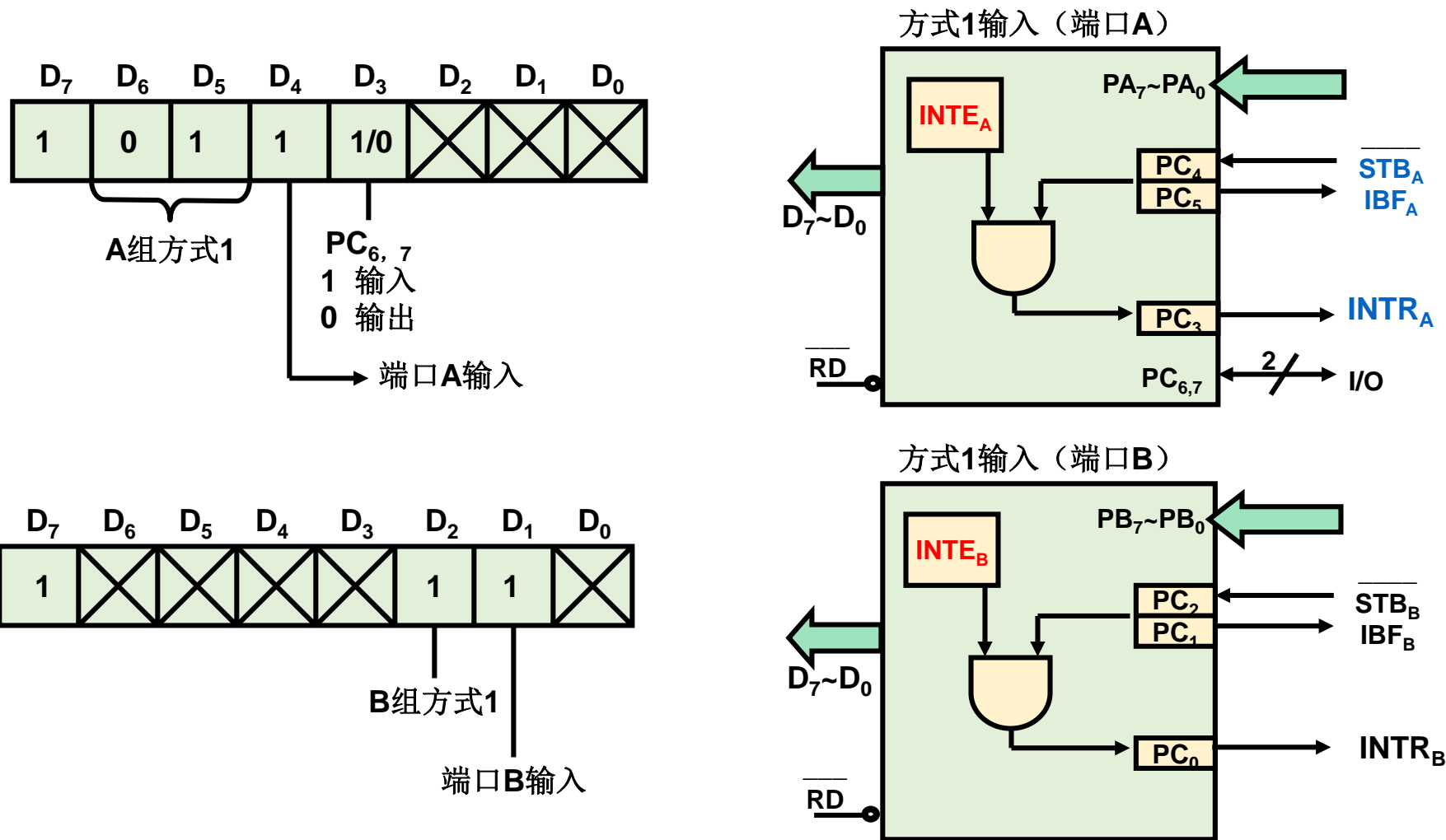
## 8.4 可编程并行接口芯片8255A

### (1) 方式1输入

- 当端口A和端口B工作于“方式1输入”时，端口C控制信号定义如图8.7所示。
- 该图中还给出了相应的方式选择控制字。

# 8.4 可编程并行接口芯片8255A

## 图8.7 8255A方式1输入





## 8.4 可编程并行接口芯片8255A

对于图8.7中所示的控制信号说明如下：

- **STB**(Strobe): **选通信号**，低电平有效。它是由外设送给8255A的输入信号，当其有效时，8255A接收外设送来的一个8位数据。
- **IBF**(Input Buffer Full): “**输入缓冲器满**”信号，高电平有效，它是一个8255A送给外设的联络信号。



## 8.4 可编程并行接口芯片8255A

- 当IBF为高电平时，表示外设的数据已送进输入缓冲器中，但尚未被CPU取走，通知外设不能送新数据；
- 只有当IBF变为低电平时，即CPU已读取数据，输入缓冲器变空时，才允许外设送新数据。
- IBF信号是由 $\overline{\text{STB}}$ 使其置位(变为高电平)，而由读信号 $\overline{\text{RD}}$ 的上升沿使其复位(变为低电平)。



## 8.4 可编程并行接口芯片8255A

- **INTR**(Interrupt Request): 中断请求信号, 高电平有效。
- 它是当 $\overline{STB}$ 为1、**IBF**为1且**INTE**也为1(中断允许)时被置为高电平。
- 也就是说, 当选通信号结束( $\overline{STB}=1$ ), 已将一个数据送进输入缓冲器(**IBF**=1), 并且端口处于中断允许状态(**INTE**=1)时, 8255A的**INTR**端被置为高电平, 向CPU发出中断请求信号。
- 当CPU响应中断读取输入缓冲器中的数据时, 由读信号 $\overline{RD}$ 的下降沿将**INTR**置为低电平。



## 8.4 可编程并行接口芯片8255A

- **INTE<sub>A</sub>**(Interrupt Enable): 端口A中断允许信号。INTE<sub>A</sub>没有外部引出端，它实际上就是端口A内部的中断允许触发器的状态信号。
- **INTE<sub>A</sub>**由PC4 (  $\overline{\text{STB}}$  ) 的置位 / 复位来控制，PC4=1时，使端口A处于中断允许状态。





## 8.4 可编程并行接口芯片8255A

- **INTE<sub>B</sub>** : 端口B中断允许信号。与INTE<sub>A</sub>类似, INTE<sub>B</sub>也没有外部引出端, 它是端口B内部的中断允许触发器的状态信号。
- 由PC<sub>2</sub>的置位 / 复位来控制, PC<sub>2</sub>=1时, 使端口B处于中断允许状态。

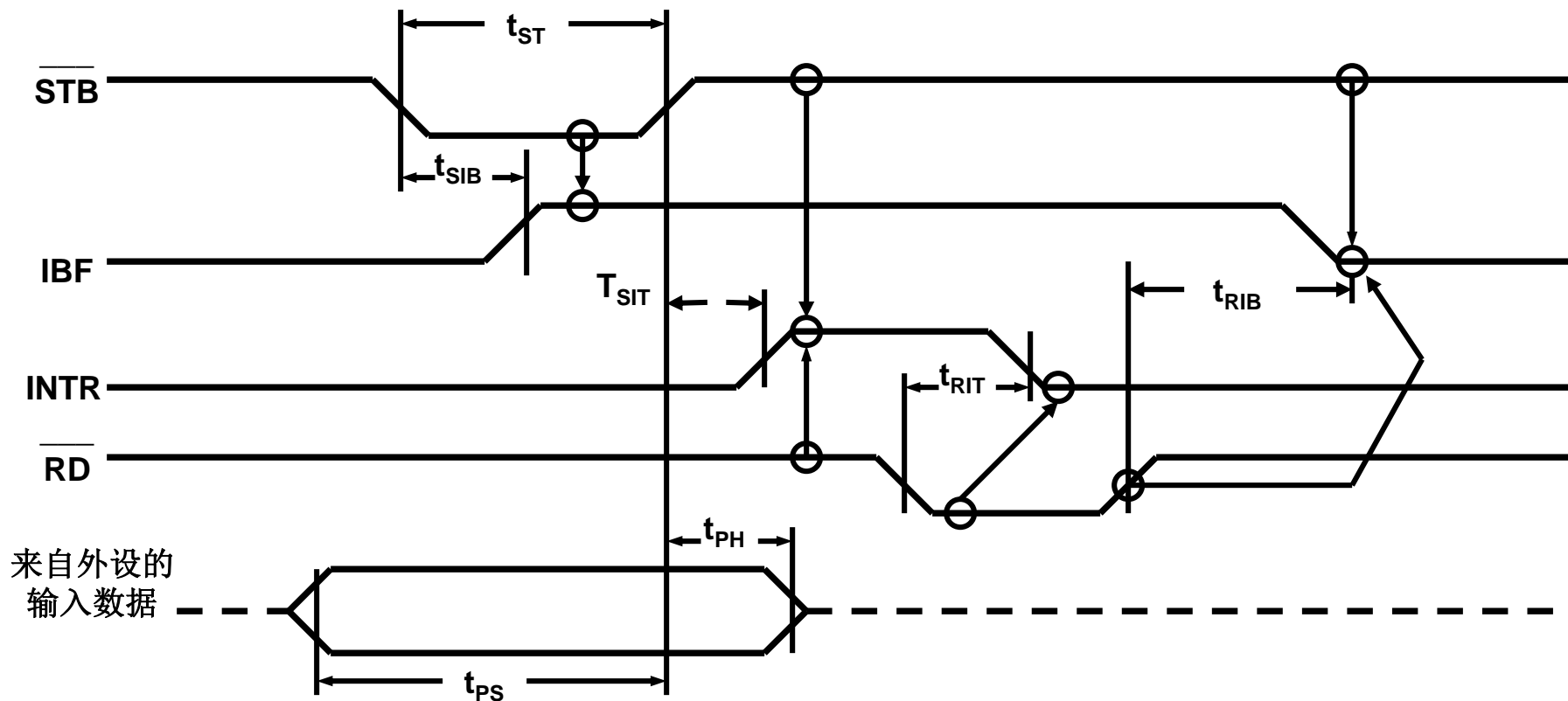


## 8.4 可编程并行接口芯片8255A

- 另外，在方式1输入时，**PC6和PC7**两位还闲着未用。如果要利用它们，可用方式选择控制字中的D3位来设定。
- 方式1输入工作时序图如图8.8所示。

# 8.4 可编程并行接口芯片8255A

图8.8 方式1输入工作时序图





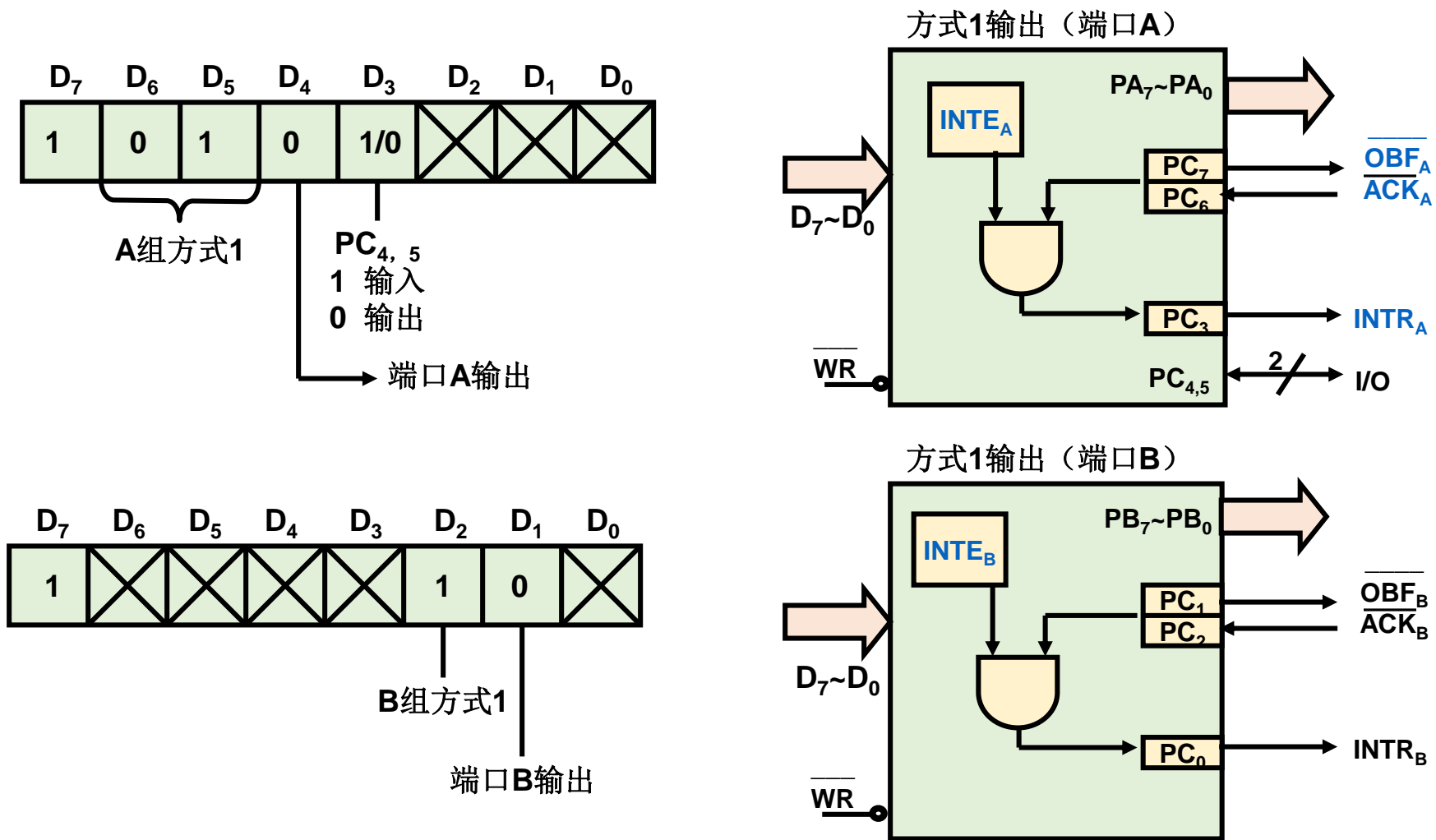
## 8.4 可编程并行接口芯片8255A

### (2) 方式1输出

当端口A和端口B工作于方式1输出时，方式选择控制字及相应的端口C控制信号定义如图8.9所示。

# 8.4 可编程并行接口芯片8255A

## 图8.9 8255A方式1输出





## 8.4 可编程并行接口芯片8255A

- 对图8.9中所示的控制信号说明如下：
  - **OBF**(Output Buffer Full): “输出缓冲器满”信号，低电平有效，它是8255A输出给外设的一个控制信号。
  - 当其有效时，表示CPU已经把数据输出给指定端口，通知外设把数据取走。
  - 它是由写信号 $\overline{WR}$ 的上升沿置成有效(低电平)，而由 $\overline{ACK}$ 信号的有效电平使其恢复为高电平。



## 8.4 可编程并行接口芯片8255A

**ACK**(Acknowledge): 外设响应信号，低电平有效。当其有效时，表明CPU通过8255A输出的数据已经由外设接收。它是对**OBF**的回答信号。

**INTR**(Interrupt Request): 中断请求信号，高电平有效。它是8255A的一个输出信号，用于向CPU发出中断请求。INTR是当**ACK**、**OBF**和**INTE**都为“1”时才被置成高电平(向CPU发出中断请求信号)；写信号**WR**的下降沿使其变为低电平(清除中断请求信号)。



## 8.4 可编程并行接口芯片8255A

**INTEA:** 端口A中断允许信号, 由**PC6** (  $\overline{ACK}$  ) 的置位/复位来控制, PC6=1时, 端口A处于中断允许状态。

**INTEB:** 端口B中断允许信号, 由**PC2**的置位/复位来控制, PC2=1时, 端口B处于中断允许状态。

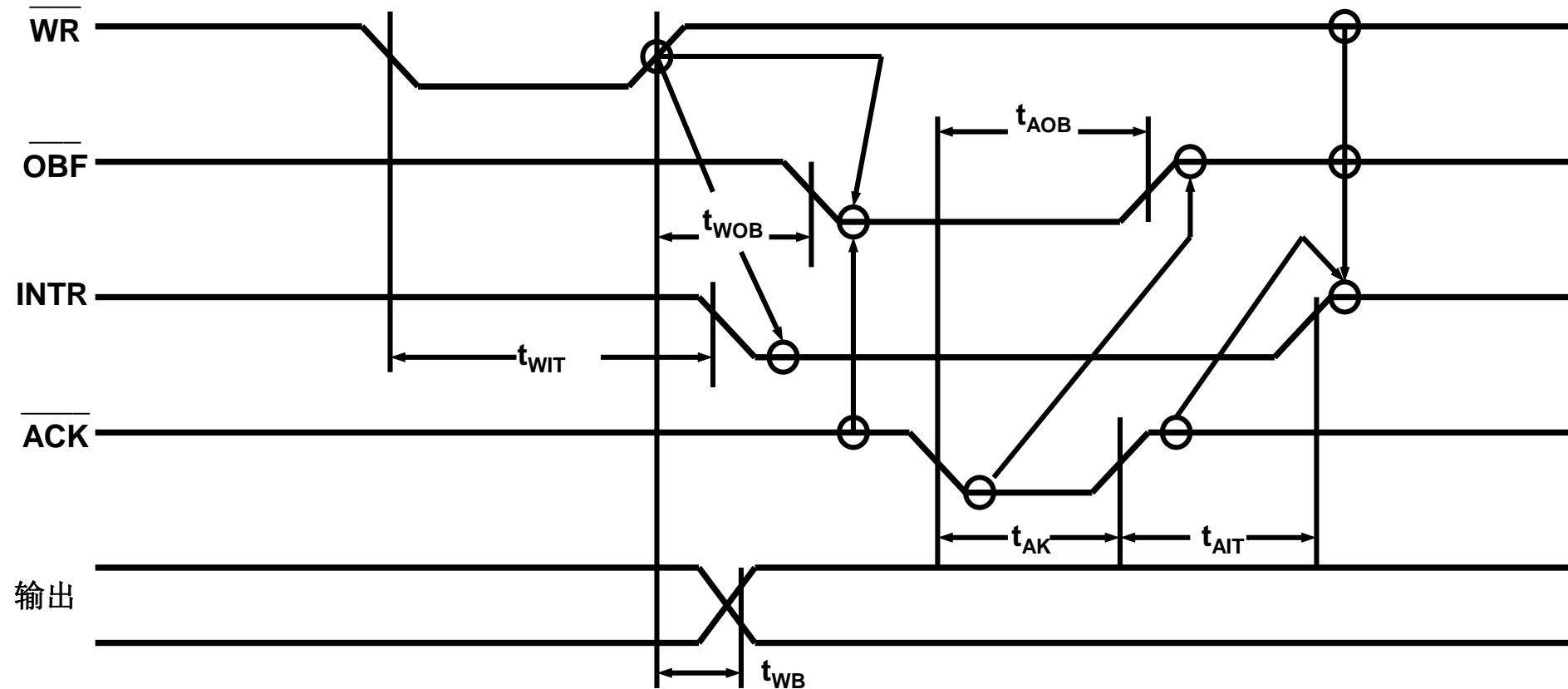
另外, 在方式1输出时, PC4、PC5两位还闲着未用, 如果要利用它们可用方式选择控制字的D3位来设定。



# 8.4 可编程并行接口芯片8255A



图8.10 方式1输出工作时序图



# 8.4 可编程并行接口芯片8255A

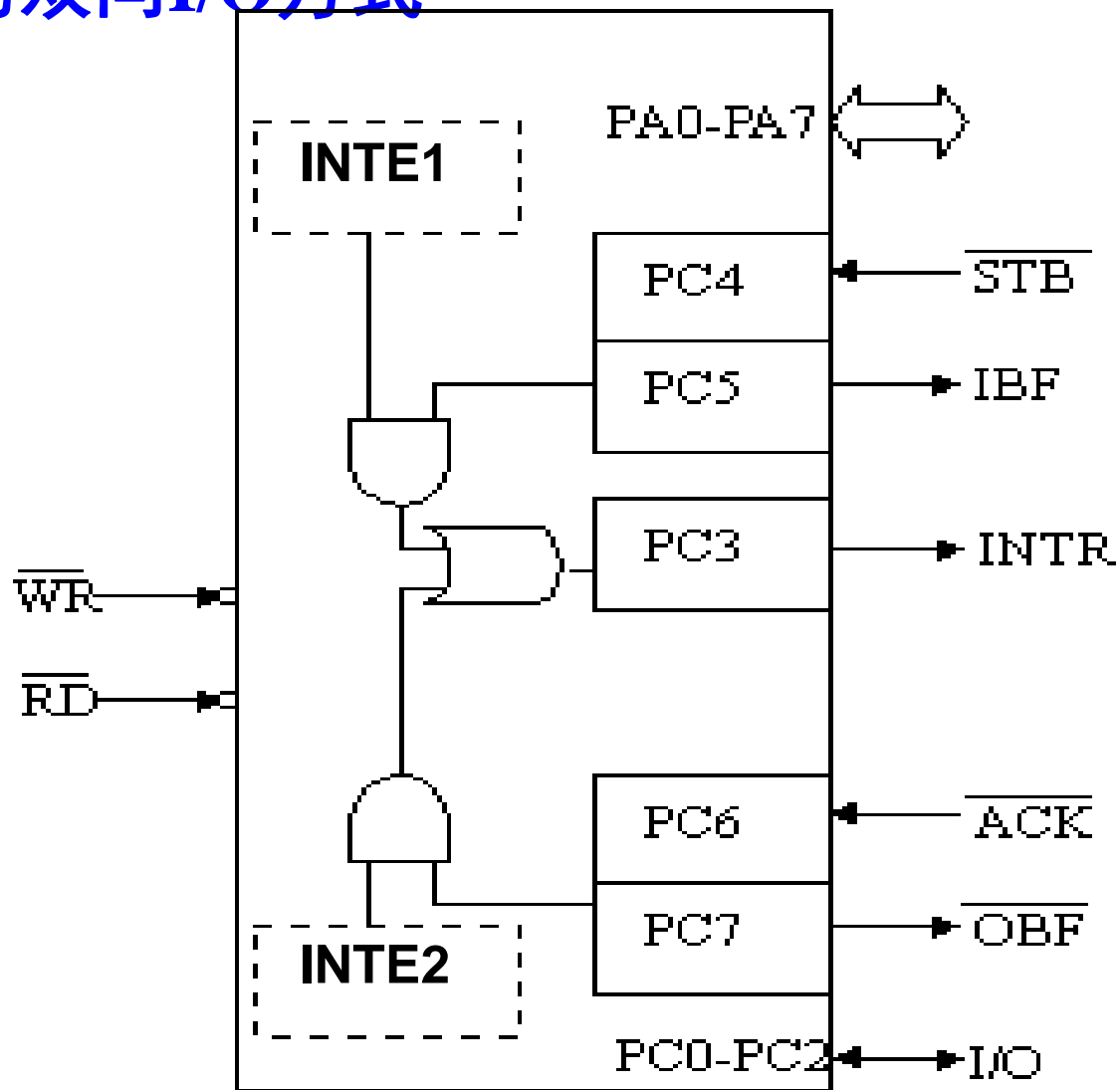
## 3. 方式2——带有选通信号的双向I/O方式

仅有A口可以工作在方式2

A口内部结构

PA7~PA0作为双向数据总线，  
PC3~PC7用作A口的联络控制信号。

PC2~PC0仍用作B口的应答信号线，或作为I/O线。



握手信号线的功能和意义与A口工作在方式1输入和输出大致相同（有两点区别），因此**方式2可以看成是A口工作在方式1输入和输出的叠加。**

**区别1：**方式2中的 $\text{INTR}=1$ 时，可能有三种情况：

- 输入缓冲器满；
- 输出缓冲器空；
- 以上两种情况兼而有之。

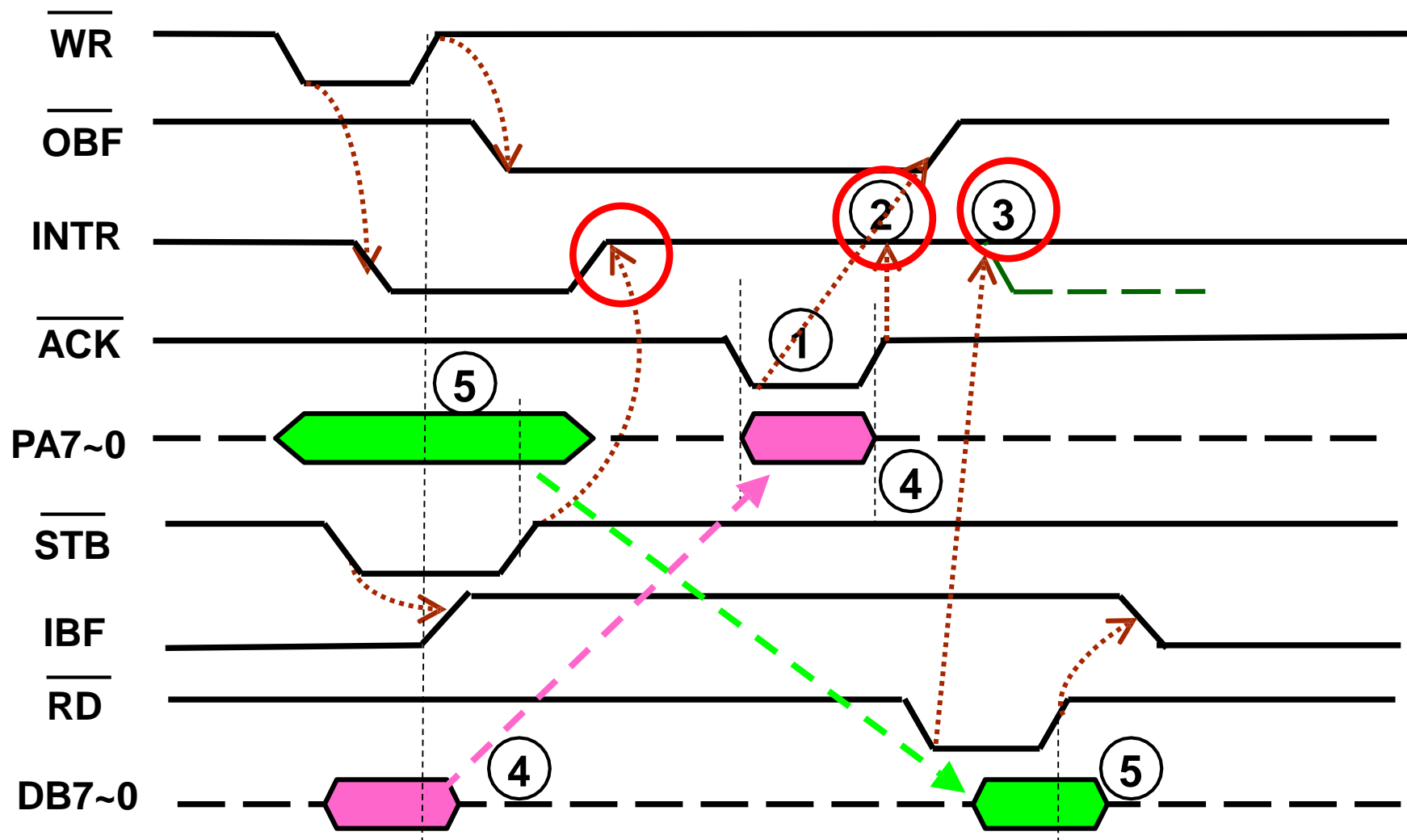
**区别2：**ACK的作用不同。

此外，对应输入输出方式2有两个中断允许触发器

- 输出—— $\text{INTE1}$ ，用PC6设置； 输入—— $\text{INTE2}$ ，用PC4设置

## 方式2时序

假设场景：CPU响应8255A中断请求，向A口写入数据，差不多与此同时，外设向A口输入数据（IBFA=0），……





## 8.4 可编程并行接口芯片8255A

### 几点说明:

- ① 注意 $\overline{\text{ACK}}$ 的作用与方式1的区别。
- ②  $\overline{\text{ACK}}$ 后沿表明外设已将数据读走，应使 $\text{INTR}=1$ ，但此时 $\text{INTR}$ 已经为“1”，此后 $\text{INTR}$ 包含两个请求。
- ③  $\overline{\text{RD}}$ 信号到达，表示已进行中断服务，应将 $\text{INTR}$ 复位，但仍有一个没有服务，因此 $\text{INTR}$ 仍然为“1”。
- ④ 这是CPU输出到外设的数据，注意该数据出现在PA口线上的时间。
- ⑤ 这是外设经A口输入的数据，注意该数据出现在数据总线D7~D0的时间。



## 8.4 可编程并行接口芯片8255A

### ■ 问题？

8255A工作在方式2，当CPU收到INTR信号后，如何判断是需要输入服务还是输出服务？

#### • 几种方法：

- 1、利用不同的硬件信号，分别产生两个对应于输入和输出的中断请求信号。
- 2、读C口得到状态字节，根据状态字节进行判断。



## 8.4 可编程并行接口芯片8255A

### 8.4.6 8255A的状态字

- 8255A工作于方式1和方式2时的状态字是通过读端口C的内容来获得的。

#### 1. 方式1状态字格式

- 方式1状态字格式如图8.11所示。







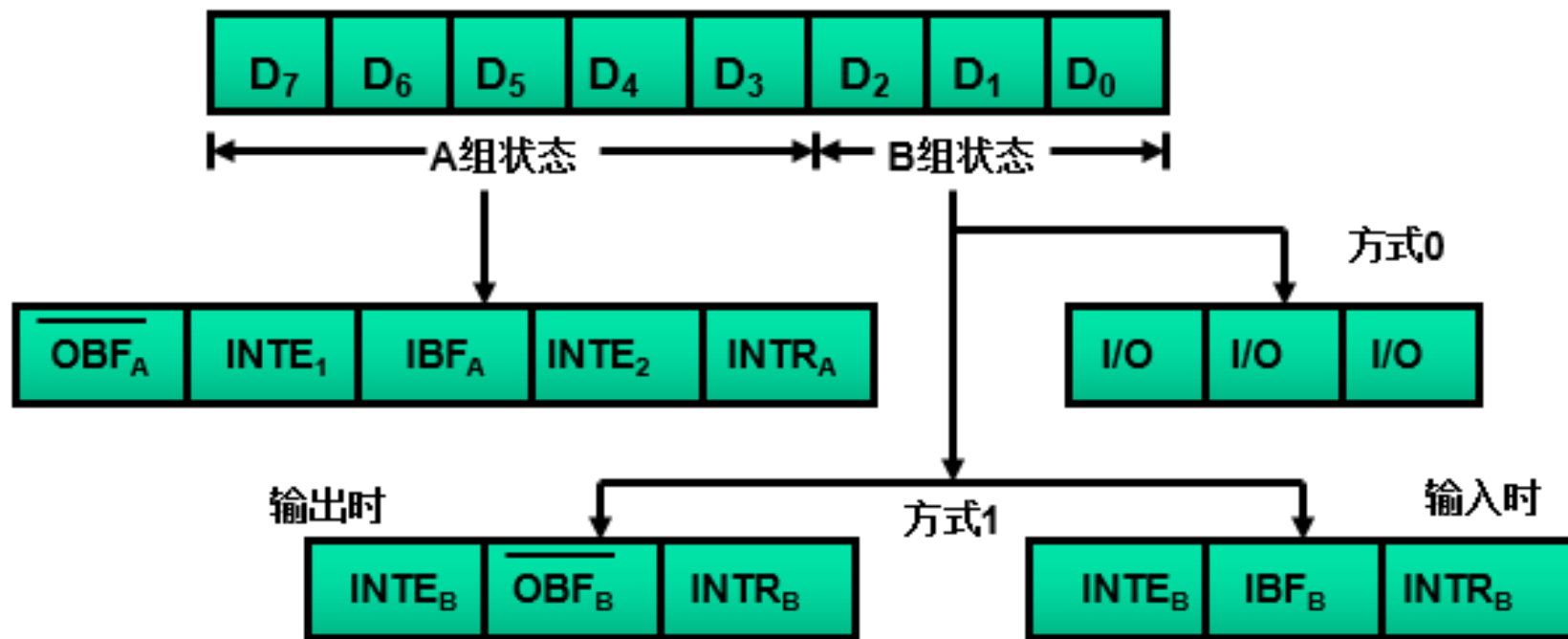
# 8.4 可编程并行接口芯片8255A

## 2. 方式2状态字格式

- 方式2的状态字也是从端口C读取。
- 方式2状态字的格式如图8.12所示。

# 8.4 可编程并行接口芯片8255A

## 图8.12 方式2状态字格式





## 8.4 可编程并行接口芯片8255A

### 注意:

- (1) C口具有位操作功能，把一个置/复位控制字送入8255的控制寄存器，就能将C口的某一位置1或清0而不影响其它位的状态。
- (2) C口控制字虽然是对端口C操作，但应写入到控制口地址，而不是写入到PC数据口。
- (3) 如果A口B口都以方式0工作，无须C口提供控制线，则这个控制字无意义。
- (4) 当A口B口选择了方式1，方式2工作时，可利用C口的按位置位/复位功能控制8255是否申请中断。

### 例题：开A口中断

```
MOV AL, 00001001B
```

```
OUT 83H, AL ;使PC4=1, 即使INTEA=1, 开A口中断
```



## 8.4 可编程并行接口芯片8255A

**例题：** 设8255控制卡地址为0063H，编写下列情况下的初始化程序。

- (1) A组B组工作于方式0，A口B口输入，C口输出。
- (2) A口B口设置成方式1，均为输入，C口输出。
- (3) A口方式1，输入，B口方式0，输出，C口高四位配合A口工作，第四位输入。

**(1) MOV AL, 10010010B**

**MOV DX, 0063H**

**OUT DX, AL**

**(2) MOV AL, 10110110B**

**MOV DX, 0063H**

**OUT DX, AL**

**(3) MOV AL, 10110001B**

**MOV DX, 0063H**

**OUT DX, AL**



## 8.4 可编程并行接口芯片8255A

例题：若C口的PC7位要求置1，PC3位要求置0，且控制口地址为0EEH，请写出该片8255初始化程序。

**MOV AL, 00001111B ;对C口的PC7置位**

**OUT 0EEH, AL**

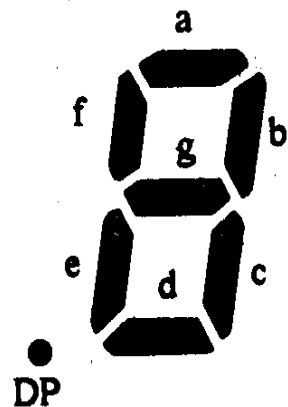
**MOV AL, 00000110B ;对C口的PC3复位**

**OUT 0EEH, AL**

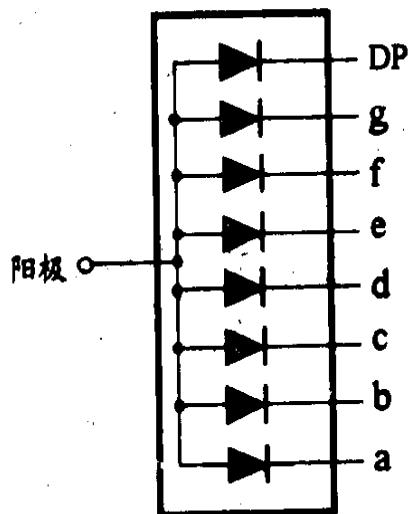
# 8.4 可编程并行接口芯片8255A

## 一、LED七段码显示

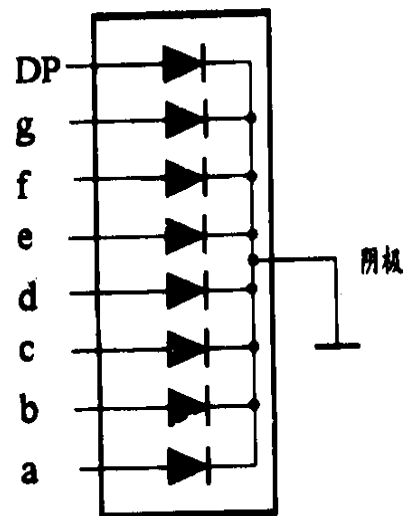
LED (Light-Emitting-Diode中文意思为发光二极管) 是一种能够将电能转化为可见光的半导体，它改变了白炽灯钨丝发光与节能灯三基色粉发光的原理，而采用电场发光。LED的特点是寿命长、光效高、无辐射与低功耗。



(a) 典型的七段 LED



(b) 共阳极 LED



(c) 共阴极 LED

## 7段LED显示器字符段码表

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	.	80H	7FH
5	6DH	92H	P	73H	82H
6	7DH	82H	U	3EH	C1H
7	07H	F8H	T	31H	CEH
8	7FH	80H	Y	6EH	91H
9	6FH	90H	8.	FFH	00H
A	77H	88H	“灭”	00H	FFH
b	7CH	83H	自定义		



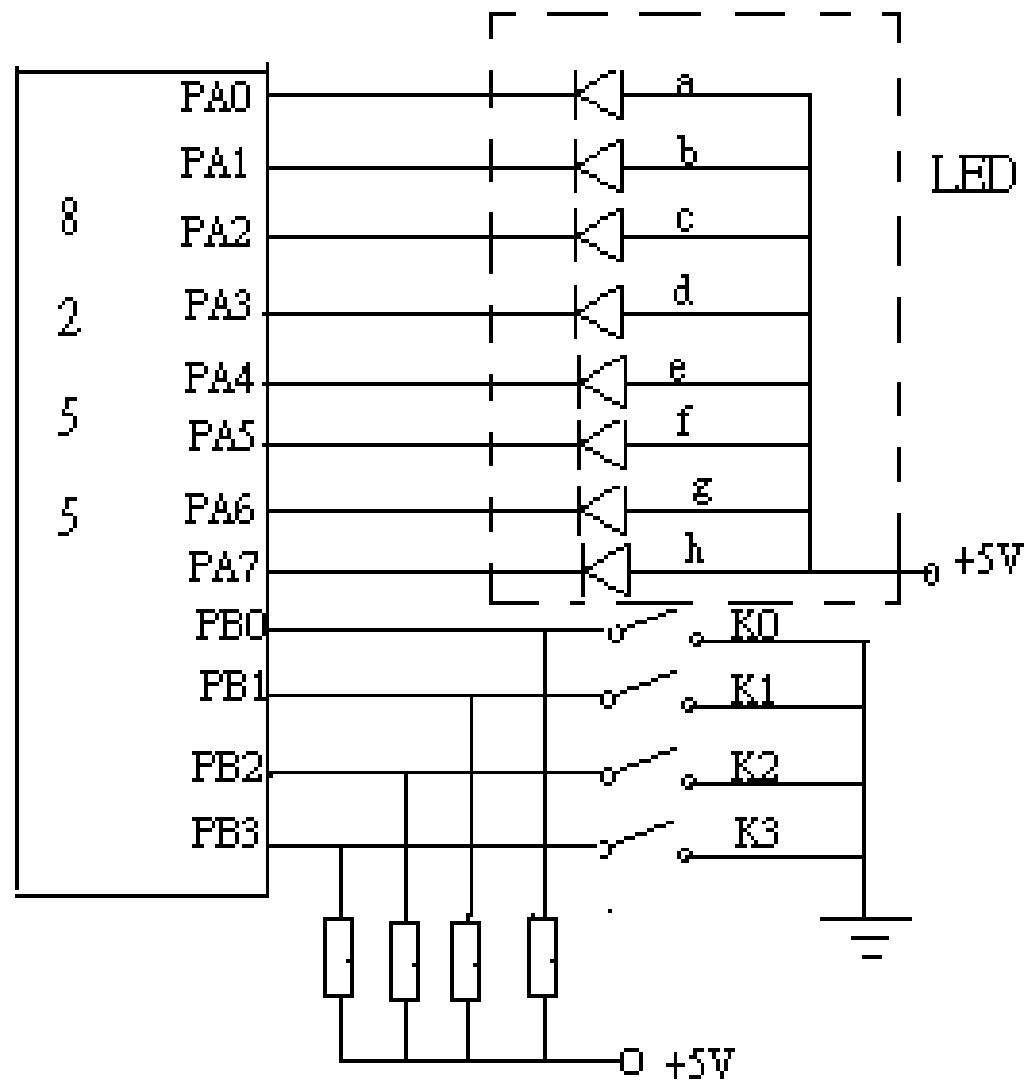
## 8.4 可编程并行接口芯片8255A

8255A连接开关和LED显示器的接口电路设计。要求：8255A的B口连接四个开关K3~K0，设置为方式0输入，A口连接一个共阳极LED显示器，设置为方式0输出，将B口四个开关输入的16种状态0H~0FH送A口输出显示。画出接口电路连接图，并编制汇编语言程序段实现上述功能。（已知8255四个端口地址为80H-83H）。

分析：本题是8255A方式0应用的一个实例。B口的PB3~PB0连接四个开关K3~K0，其输入有16种组合状态，即0000B~1111B（0H~0FH），A口与LED显示器连接，可输出一位十六进制数0~F。



# 8.4 可编程并行接口芯片8255A





## 8.4 可编程并行接口芯片8255A

```
SEGCODE DB 0C0H, 0F9H, A4H, B0H, ..., 8EH
```

```
.....
```

```
MOV AL, 1000010B ; 8255初始化
```

```
OUT 83H, AL
```

```
LOP: IN AL, 81H ; 读B口
```

```
AND AL, 0FH ; 使B口高位清零
```

```
MOV BX, OFFSET SEGCODE
```

```
XLAT ; 换码, 将B口读
```

```
OUT 80H, AL ; 入的数转换成对
```

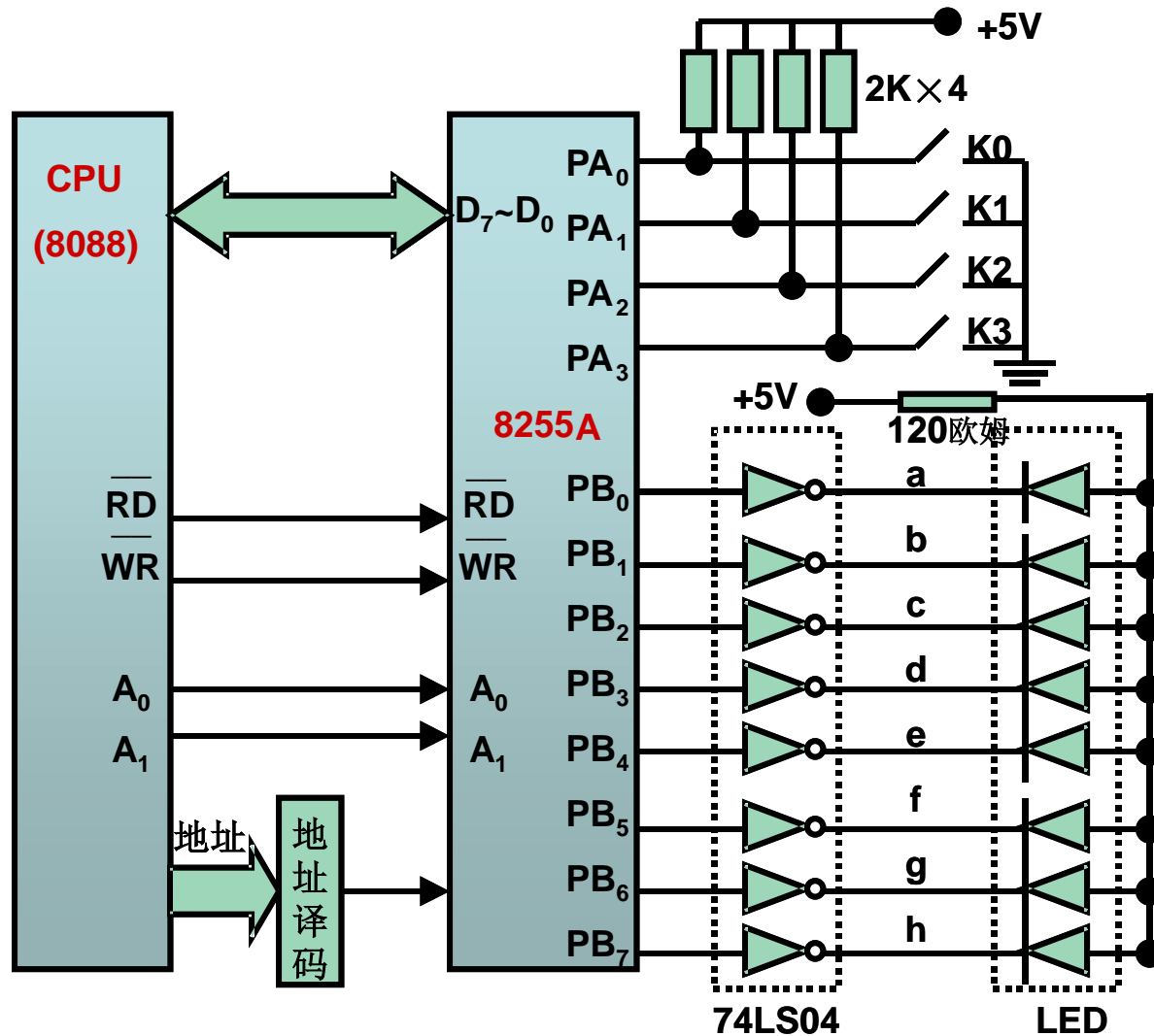
```
JMP LOP ; 应的七段码值
```



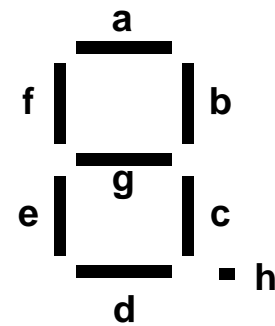
## 8.5 8255A芯片应用

- **例8.1** 8255A工作于方式0，利用8255A将外设开关的二进制状态从端口A输入，经程序转换为对应的LED段选码(字形码)后，再从端口B输出到LED显示器。
  - 具体连线图如图7.18(a)所示。
  - LED显示器如图 7.18(b)所示。

# 8.5 8255A芯片应用



(a) 连线图



(b) LED显示器

设8255A的端口地址为： 端口A——D0H， 端口B——D1H， 端口C——D2H， 控制口——D3H。 则本例的初始化及输入、输出控制程序如下所示。

## DATA SEGMENT

```
SSEGCODE DB 3FH, 06H, 5BH, 4FH, 66H, 6DH, 7DH, 07H
          DB 7FH, 67H, 77H, 7CH, 39H, 5EH, 79H, 71H
```

## DATA ENDS

## CODE SEGMENT

```
ASSUME CS: CODE, DS: DATA
```

```
START: MOV AX, DATA
```

```
        MOV DS, AX
```

```
        MOV AL, 90H           ; 设置方式选择控制字， A口工作于方式0输入， B口工作于方式0输出
```

```
        OUT 0D3H, AL
```

## RDPORTA :

```
        IN AL, 0D0H          ; 读A口
```

```
        AND AL, 0FH          ; 取A口低4位
```

```
        MOV BX, OFFSET SSEGCODE ; 取LED段选码表首地址
```

```
        XLAT                 ; 查表， AL←(BX+AL)
```

**OUT 0D1H, AL ;从B口输出LED段选码, 显示相应字形符号**

**MOV AX, XXXXH ; 延时**

**DELAY: DEC AX ;**

**JNZ DELAY ;**

**MOV AH, 1 ; 判断是否有键按下**

**INT 16H ;**

**JZ RDPORTA ; 若无, 则继续读端口A**

**MOV AH, 4CH ; 否则返回DOS**

**INT 21H**

**CODE ENDS**

**END START**



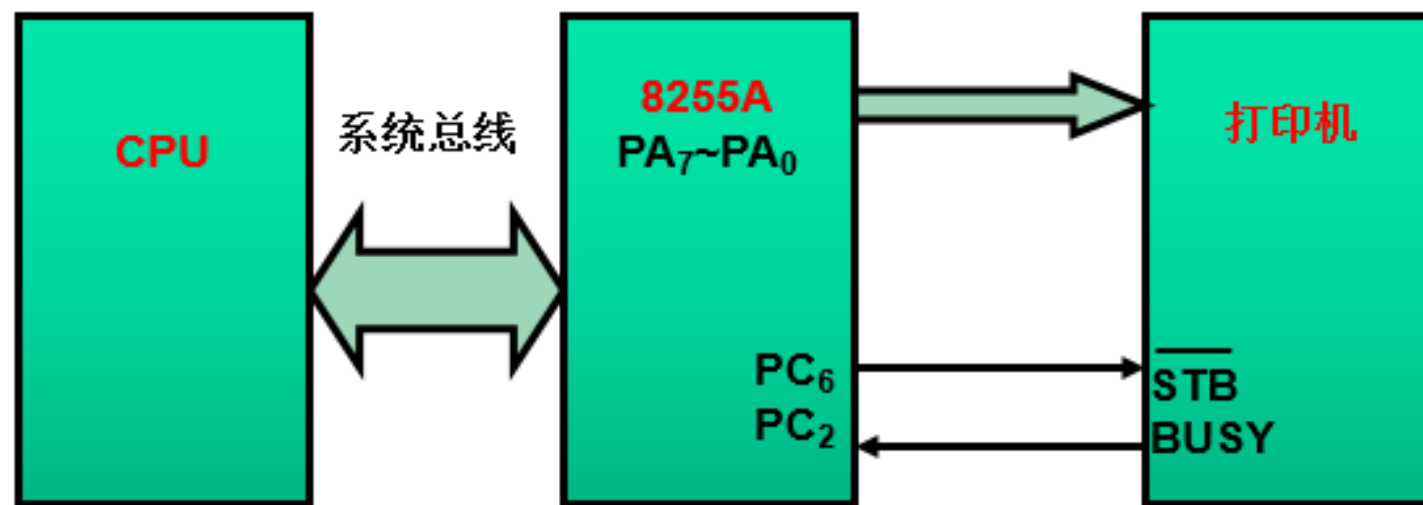
## 8.5 8255A芯片应用

---

**例8.2 8255A工作于方式0，用8255A作为以查询方式工作的打印机接口，如图所示。**

# 8.5 8255A芯片应用

图 8255A作为打印机接口

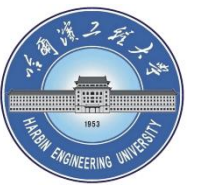






## 8.5 8255A芯片应用

- **工作原理为:** 当主机开始打印输出时, 先测试打印机忙(BUSY) 信号,
- 如果打印机处于忙状态, 则BUSY信号为1;
- 反之, 则BUSY信号为0。
- 当检测到BUSY信号为0时, 则主机可通过8255A向打印机输出一个字符。
- 此时, 还需输出一个选通信号(负脉冲) 给打印机的STB端, 用此负脉冲作为字符送到打印机输入缓冲器的打入脉冲。



## 8.5 8255A芯片应用

- 用8255A的**端口A**作为打印输出数据口，**工作于方式0**；端口B不用；**端口C也工作于方式0**，端口C的PC2作为BUSY信号输入端，所以PC3~PC0应设定为输入方式；
- PC6作为选通信号输出端，所以PC7~PC4应设定为输出方式。
- 设打印字符存放在内存2000H单元。
- **8255A的端口地址为：**
- **端口A—D0H，端口B—D1H，端口C—D2H，控制口—D3H**
- 则初始化及打印控制子程序如下所示：



## 8.5 8255A芯片应用

### BEGIN:

MOV AL, 81H ; 方式选择控制字, 使A、B、C三个端口工作于方式0,  
OUT 0D3H, AL ; 端口A为输出, PC7~PC4为输出, PC3~PC0为输入  
MOV AL, 0DH ; 用“端口C按位置1/置0控制字”置PC6=1, 使STB为高电平  
OUT 0D3H, AL

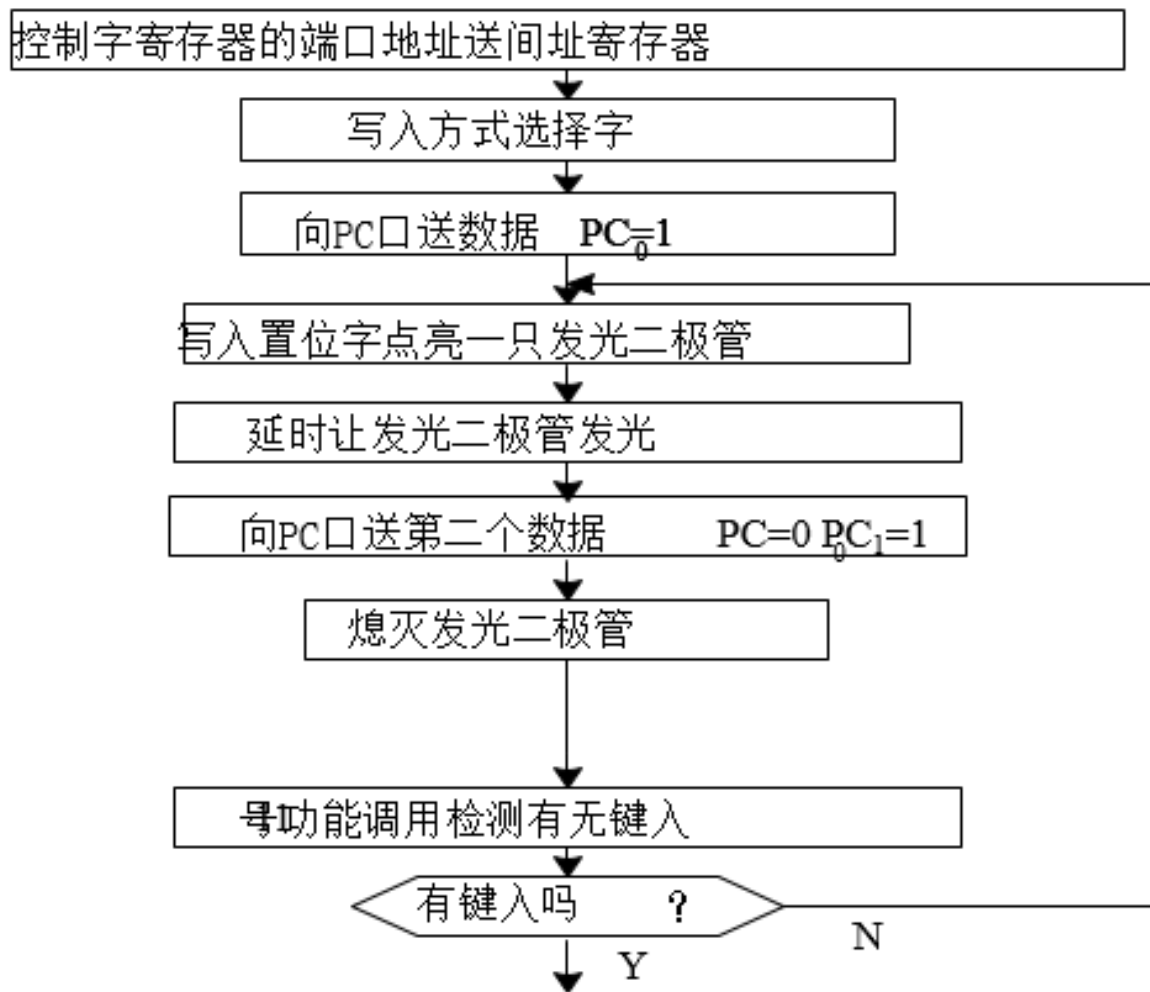
### TESBY:

IN AL, 0D2H ; 读端口C  
TEST AL, 04H ; 测试PC2(BUSY)  
JNZ TESBY ; 如PC2=1(BUSY信号有效) 打印机处于忙状态, 继续查询  
MOV AL, DS: [2000H] ; PC2=0(BUSY信号无效), 取打印字符  
OUT 0D0H, AL ; 由端口A输出打印字符 —  
MOV AL, 0CH ; 用“端口C按位置1/置0控制字”置PC6=0, 使STB为低电平  
OUT 0D3H, AL  
INC AL  
OUT 0D3H, AL ; 置PC6=1, 再使STB为高电平, 从而产生一个负选通脉冲信号  
RET



# 8.5 8255A芯片应用

## 程序框图



## 参考程序

```
stack    segment stack 'stack'  
         dw 32 dup (0)  
stack    ends  
code     segment  
begin    proc far  
         assume ss: stack, cs: code  
         PUSH DS  
         SUB AX, AX  
         PUSH AX  
         MOV DX,383H; 383H为控制字寄存器的端口地址  
         MOV AL,80H  
         OUT DX,AL; 写入方式选择字
```



## 8.5 8255A芯片应用

```
MOV DX,382H ; C端口的端口地址送DX
MOV AL,1 ; C端口的输出值
AGAIN: OUT DX,AL ; 给C口送数据, 点亮一只二极管
LOOP $ ; 延时
LOOP $ ; 延时
PUSH AX
MOV AH,11 ; 11号功能调用: 检查键盘有无输入
INT 21H ; 无: 0送AL; 有: -1送AL
INC AL ; 有键入, AL=-1, AL增1, AL=0
JZ BACK
POP AX
ROL AL,1 ; 通过循环移位指令改变C端口的输出值, 灭第一盏灯点第二盏灯
JMP AGAIN ; 给端口送数据
BACK: ret
```



## 8.5 8255A芯片应用

```
begin endp  
code ends  
end begin
```

方式0也可作为查询式输入或输出的接口电路，此时端口A和B分别可作为一个数据端口，而取端口C的某些位作为这两个数据端口的控制和状态信息。



## 例8.4: LED/开关接口

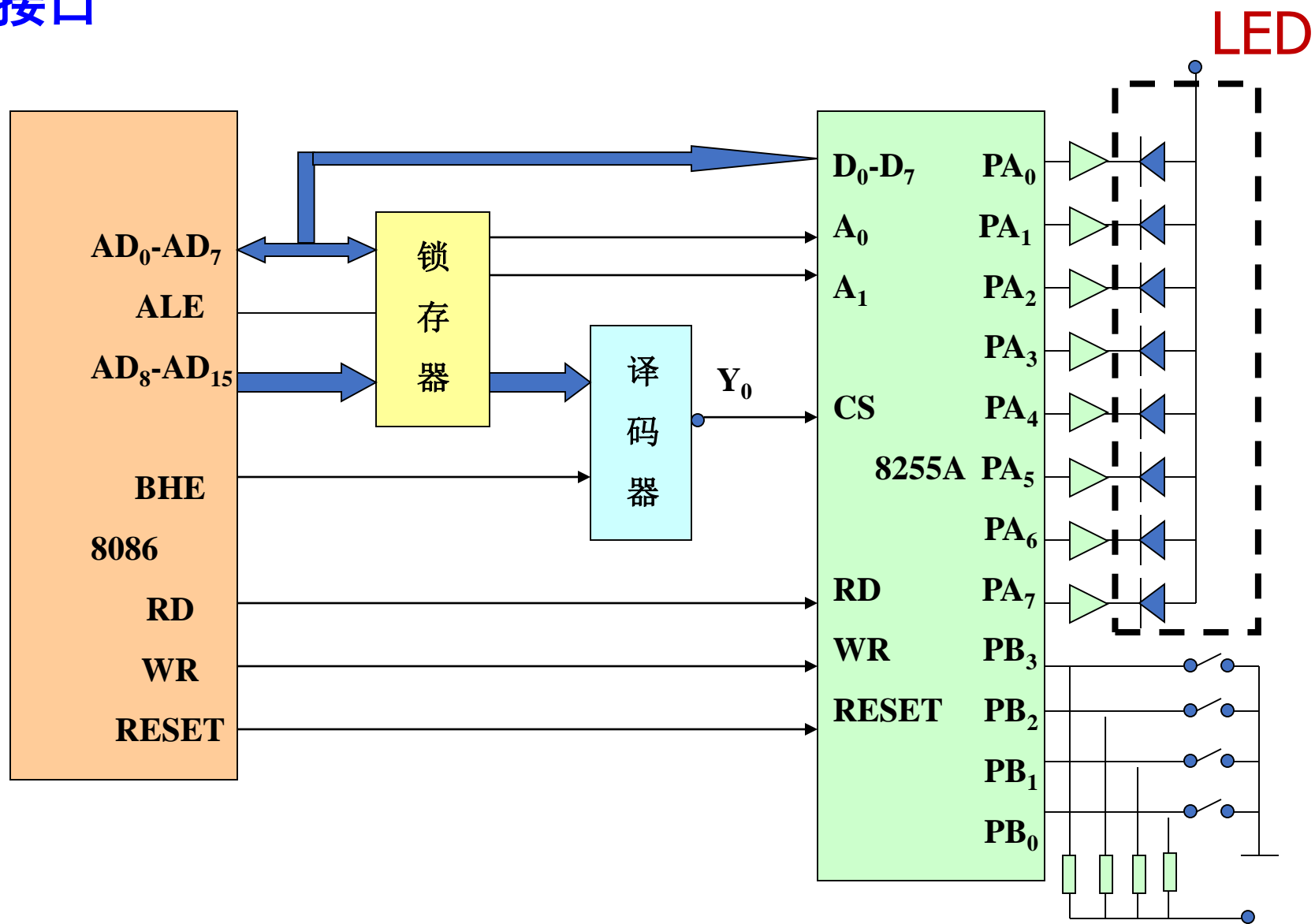


图8-14 8086CPU、8255A同开关7段LED的接口

**如果要求7段LED循环显示0-F十六个数，每个显示5秒，显示20遍，  
则程序为：**

```
ORG 2000H
MOV AL, 80H (1000 0000B) ; 8255A方式字
MOV DX, 0FFFEH ; 8255A控制字端口地址
OUT DX, AL ; 送方式字
MOV BX, 20 ; 循环次数20次
DISFLOP: LEA DI, SSEGCODE ; 求段码首地址
MOV CX, 16 ; 显示字符个数
LOP: MOV AL, [DI] ; 取显示字符送A口
MOV DL, 0F8H ; 设A口地址
OUT DX, AL ; 段码输出到A口显示
INC DI ; 修改显示指针
CALL DELAY5S ; 延时5秒子程序
LOOP LOP ; 循环16次
DEC BX ; 修改循环20次的计数值
```

**JNZ DISPLOP ; 返回再次读B口内容**

**HLT ; 暂停**

**ORG 2100H**

**DELAY5S: :**

**:**

**ORG 2500H**

**SSEGCODE DB 0C0H, 0F9H, 0A4H, 0B0H, 99H,**

**DB 92H, 82H, 0F8H, 80H, 98H, 88H**

**DB 83H, 0C6H, 0A1H, 86H, 8EH**

**; 段码定义（共阳）**

**例如：0C0H=1100 0000B；显示“0”**

**PA<sub>0</sub>=a段=“0”点亮 PA<sub>1</sub>=b段=“0”点亮**

**PA<sub>2</sub>=c段=“0”点亮 PA<sub>3</sub>=d段=“0”点亮**

**PA<sub>4</sub>=e段=“0”点亮 PA<sub>5</sub>=f段=“0”点亮**

**PA<sub>6</sub>=g段=“1”灭 PA<sub>7</sub>=DP段=“1”灭**