

## 第9章 单片机系统的并行扩展

### 学习要求

- ◆熟悉系统三总线的概念；
- ◆掌握程序存储器的扩展；
- ◆掌握外部数据存储器的扩展；
- ◆掌握8255芯片扩展I/O接口的方法。

### 9.1 系统并行扩展技术

#### 9.2 外部数据存储器的扩展方法

#### 9.3 片内Flash存储器的编程

#### 9.4 E2PROM的并行扩展

#### 9.5 并行I/O芯片82C55的设计

### 9.1 系统并行扩展技术

#### 9.1.1 系统并行扩展结构

单片机系统并行扩展结构见图9-1。

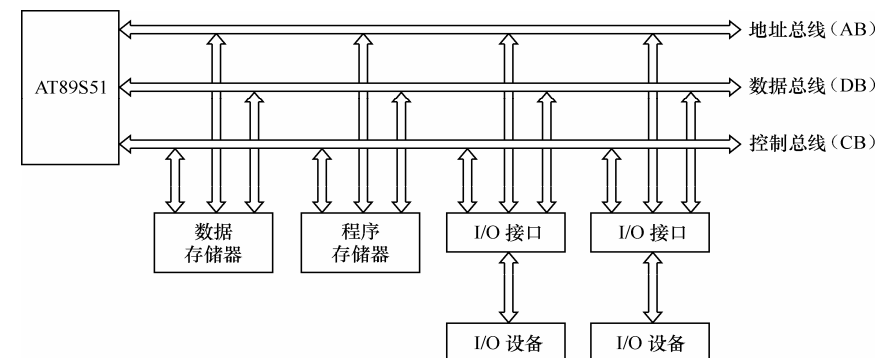


图9-1 单片机系统扩展结构



表9-1 AT89S5x 系列单片机片内的存储器资源

| 型 号     | 片内闪烁存储器容量 | 片内 RAM 存储器容量 |
|---------|-----------|--------------|
| AT89S52 | 8 k B     | 256 B        |
| AT89S53 | 12k B     | 256 B        |
| AT89S54 | 16k B     | 256B         |
| AT89S55 | 20k B     | 256 B        |



## 扩展 方法

- 1、根据单片机的型号和应用系统的要求  
确定扩展存储器的类型、容量或I/O口的类型  
包括芯片的类型、个数、容量、特点、要求
- 2、确定作程序存储器 /PSEN  
数据存储器 /RD、/WR  
混合存储器 /RD./PSEN
- 3、进行地址统筹，为各器件分配地址
- 4、根据位扩展、字扩展，确定片选方式
- 5、如何与三总线连接？



## 系统总线及总线构造

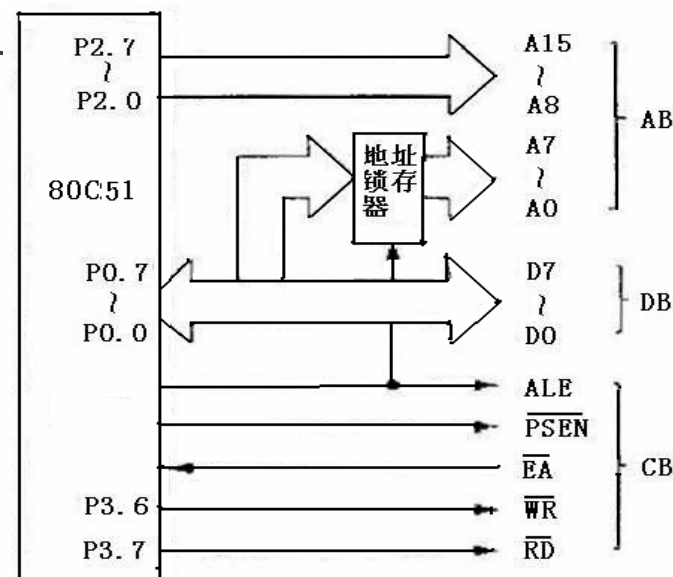
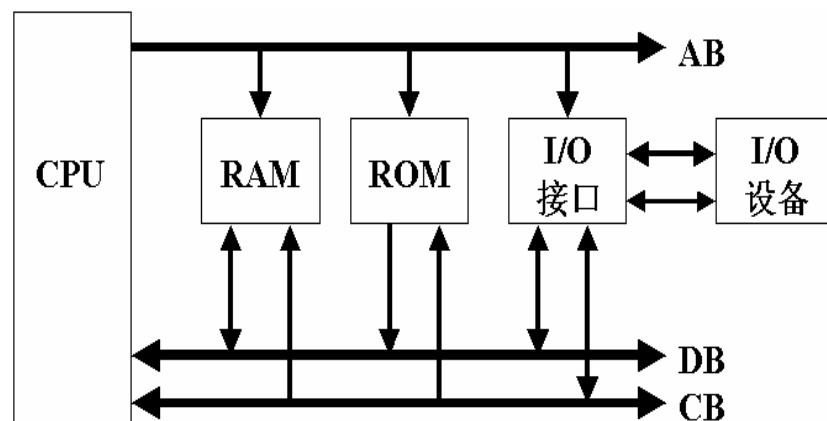


图8-2 8051系列单片微机的三总线结构



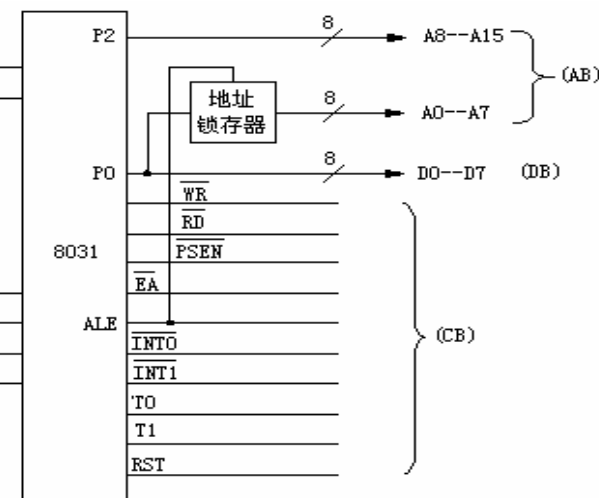
表4-4 P3各口线与专用功能

| 口线   | 替代的专用功能                             |
|------|-------------------------------------|
| P3.0 | RXD(串行输入口)                          |
| P3.1 | TXD(串行输出口)                          |
| P3.2 | $\overline{\text{INT0}}$ (外部中断0)    |
| P3.3 | $\overline{\text{INT1}}$ (外部中断1)    |
| P3.4 | T0(定时器0的外部输入)                       |
| P3.5 | T1(定时器1的外部输入)                       |
| P3.6 | $\overline{\text{WR}}$ (外部数据存储器写选通) |
| P3.7 | $\overline{\text{RD}}$ (外部数据存储器读选通) |

### P3各口线与专用功能

## 89S51单片机对外扩展三总线

三总线方式:  
1. P0口为低8位地址/数据复用总线,  
2. P2口的口线作高位地址线。  
3. 控制信号线:  
ALE, /PSEN, /EA  
/RD/WR信号。(见P3口多功能)



如何来构造系统的三总线。

### 1. P0口作为低8位地址/数据总线

AT89S51受引脚数目限制, P0口既用作低8位地址总线, 又用作数据总线(分时复用), 因此需增加一个8位地址锁存器。AT89S51访问外部扩展的存储器单元或I/O接口寄存器时, 先发出低8位地址送地址锁存器锁存, 锁存器输出作为系统的低8位地址(A7~A0)。随后, P0口又作为数据总线口(D7~D0), 如图8-2所示。

### 2. P2口的口线作为高位地址线

P2口用作系统的高8位地址线, 再加上地址锁存器提供的低8位地址, 便形成了系统完整的16位地址总线。

使单片机系统的寻址范围达到64KB。

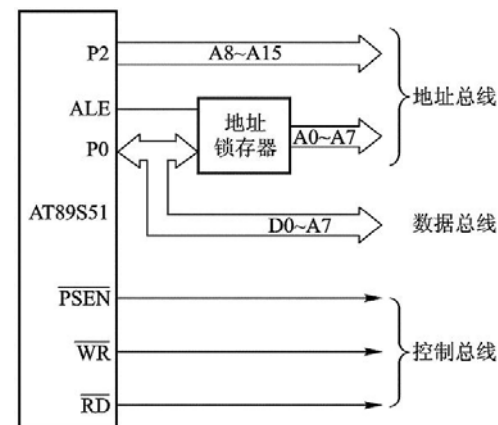


图8-2 AT89S51单片机扩展的片外三总线

### 3. 控制信号线

除地址线和数据线外，还要有系统的控制总线。这些信号有的就是单片机引脚的第一功能信号，有的则是P3口第二功能信号。包括：

- (1)  $\overline{\text{PSEN}}$ 作为外扩程序存储器的读选通控制信号。
- (2)  $\overline{\text{RD}}$ 和 $\overline{\text{WR}}$ 为外扩数据存储器 and I/O的读、写选通控制信号。
- (3)  $\text{ALE}$ 作为P0口发出的低8位地址锁存控制信号。
- (4)  $\overline{\text{EA}}$ 为片内、片外程序存储器的选择控制信号。

可见，AT89S51的4个并行I/O口，由于系统扩展的需要，真正作为数字I/O用，就剩下P1和P3的部分口线了。



13

### 9.1.2 地址空间分配和外部地址锁存器

本节讨论如何进行存储器空间的地址分配，并介绍用于输出低8位地址的常用的地址锁存器。

#### 存储器地址空间分配

实际系统设计中，既需要扩展程序存储器，又需要扩展数据存储器，如何把片外的两个64KB地址空间分配给各个程序存储器、数据存储器芯片，使一个存储单元只对应一个地址，避免单片机发出一个地址时，同时访问两个单元，发生数据冲突。这就是存储器地址空间分配问题。



14

AT89S51单片机发出的地址码用于选择某个存储器单元，外扩多片存储器芯片中，单片机必须进行两种选择：一是选中该存储器芯片，这称为“片选”，未被选中的芯片不能被访问。二是在“片选”的基础上再根据单片机发出的地址码来对“选中”芯片的某一单元进行访问，即“单元选择”。

为实现片选，存储器芯片都有片选引脚。同时也都有多条地址线引脚，以便进行单元选择。注意，“片选”和“单元选择”都是单片机通过地址线一次发出的地址信号来完成选择。



15

通常把单片机系统的地址线笼统地分为低位地址线和高位地址线，“片选”都是使用高位地址线。实际上，16条地址线中的高、低位地址线的数目并不是固定的，只是习惯上把用于“单元选择”的地址线，都称为低位地址线，其余的为高位地址线。

常用的存储器地址空间分配方法有两种：线性选择法（简称线选法）和地址译码法（简称译码法），下面介绍。



## 1. 线选法

是直接利用系统的某一高位地址线作为存储器芯片（或I/O接口芯片）的“片选”控制信号。为此，只需要把用到的高位地址线与存储器芯片的“片选”端直接连接即可。



### 存储器容量：

是指一片存储器最多能够存储多少个单位信息，二进制信息单位多用字节来表示。

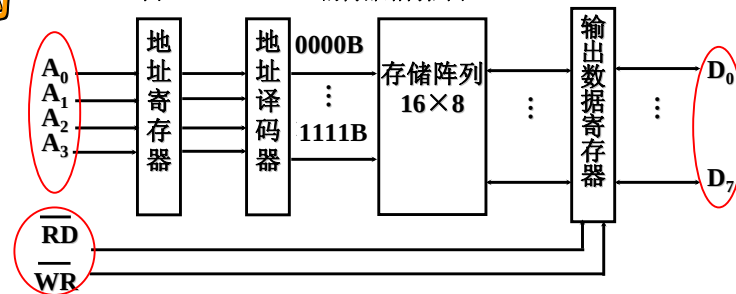
在标注存储器容量时，经常同时标出存储单元的数目和位数。

地址线条数决定

存储器芯片容量=单元数×数据线条数

## 结构

图1-2 16×8bitRAM的内部结构框图



ROM和RAM芯片均有分四组引脚线：（ROM另有特殊的引脚线）

- 1、地址线 传送存储器的地址码，其根数决定存储单元个数即字数
- 2、数据线 传送对某一单元进行读/写的数，双向  
决定一个单元内存储二进制数的位数，即字长
- 3、控制线 传送读/写控制信号，以控制读/写操作
- 4、电源线 +5V 和GND线

例：

intel 2114芯片容量=1kB× 4位/片

62256芯片容量=32KB× 8位/片

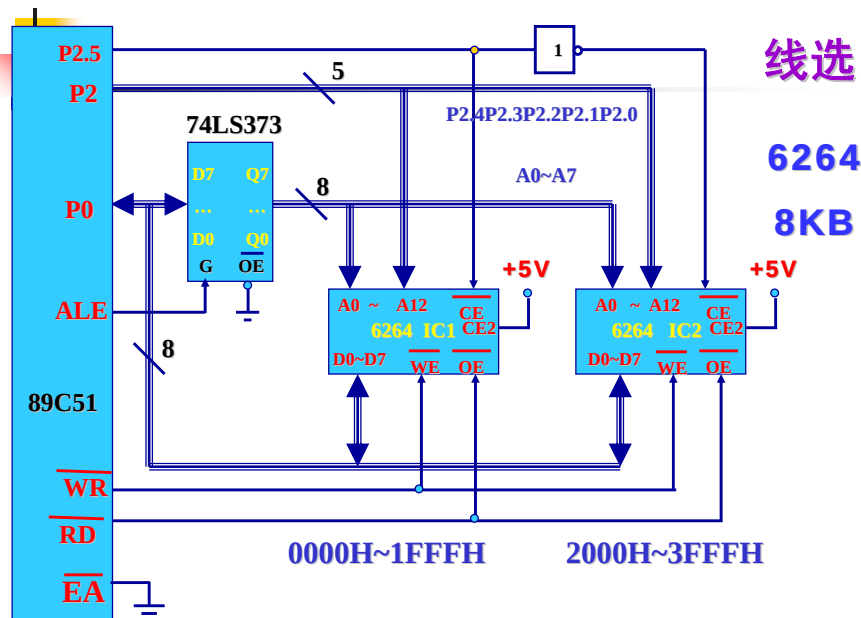
$2^{15}=32KB$ ,即15条地址线 数据线条数:8条

2864芯片容量= 8KB× 8位/片 ?

## 地址空间——地址段不唯一

| A15  | A14  | A13  | A12                     | A11  | A10  | A9   | A8   | A7~A0 | 存储器 |
|------|------|------|-------------------------|------|------|------|------|-------|-----|
| P2.7 | P2.6 | P2.5 | P2.4                    | P2.3 | P2.2 | P2.1 | P2.0 | P0    | IC1 |
| X    | X    | 0    | 0                       | 0    | 0    | 0    | 0    | 0~0   |     |
| 000B |      |      | 0000H、4000H、8000H、C000H |      |      |      |      |       |     |
| 010B |      |      | 1                       | 1    | 1    | 1    | 1    | 1~1   |     |
| 100B |      |      | 1FFFH、5FFFH、9FFFH、DFFFH |      |      |      |      |       |     |
| 110B |      |      |                         |      |      |      |      |       |     |
| X    | X    | 1    | 0                       | 0    | 0    | 0    | 0    | 0~0   | IC2 |
| 001B |      |      | 2000H、6000H、A000H、E000H |      |      |      |      |       |     |
| 011B |      |      | 1                       | 1    | 1    | 1    | 1    | 1~1   |     |
| 101B |      |      | 3FFFH、7FFFH、BFFFH、FFFFH |      |      |      |      |       |     |
| 111B |      |      |                         |      |      |      |      |       |     |

片外剩2根线，每片有4段地址段



**线选法优点**是电路简单，不需要另外增加地址译码器硬件电路，体积小，成本低。**缺点**是可寻址的芯片数目受到限制。另外，地址空间不连续，每个存储单元的地址不唯一，这会给程序设计带来不便，只适用于外扩芯片数目不多的单片机系统的存储器扩展。

### 2. 译码法

使用译码器对AT89S51单片机的高位地址进行译码，译码输出作为存储器芯片的片选信号。这种方法能够有效地利用存储器空间，适用于多芯片的存储器扩展。常用的译码器芯片有74LS138（3线-8线译码器）、74LS139（双2线-4线译码器）和74LS154（4线-16线译码器）。

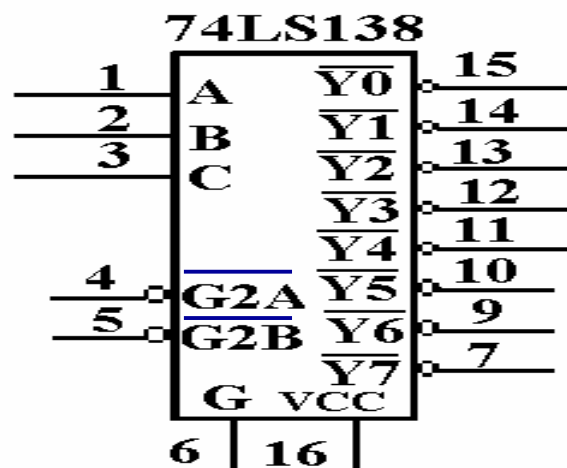
若全部高位地址线都参加译码，称为**全译码**；若仅部分高位地址线参加译码，称为**部分译码**。部分译码存在着部分存储器地址空间相重叠的情况。

下面介绍常用的译码器芯片。

#### (1) 74LS138

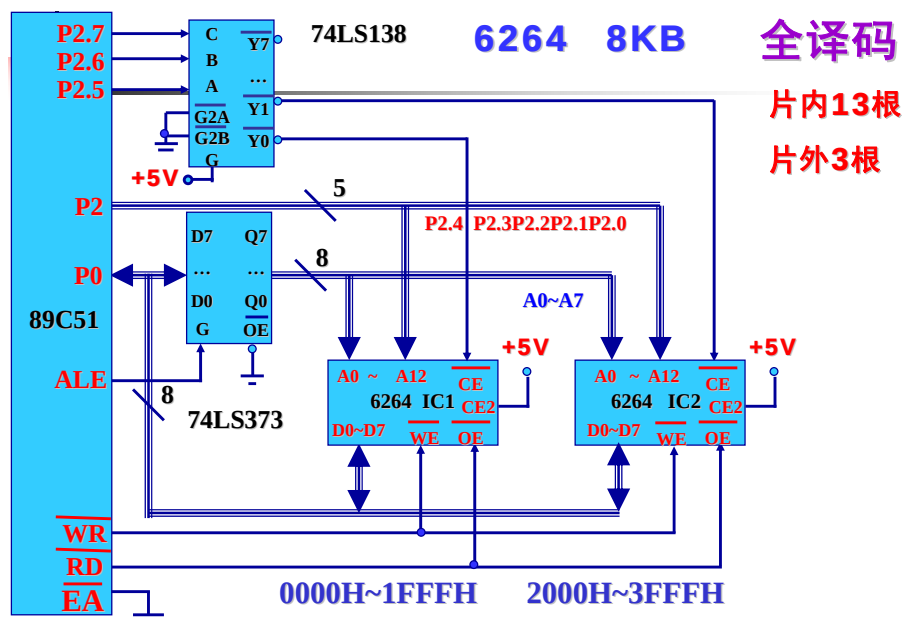
3线-8线译码器，有3个数据输入端，经译码产生8种状态。引脚如图8-3所示，真值表见表8-1。由表8-1可见，当译码器的输入为某一固定编码时，其输出仅有一个固定的引脚输出为低电平，其余的为高电平。输出为低电平的引脚就作为某一存储器芯片的片选信号。

## 3—8译码器 74LS138



## 74LS138真值表

| $\overline{G_1}$ | $\overline{G_{2A}}$ | $\overline{G_{2B}}$ | C | B | A | $\overline{Y_7}$ | $\overline{Y_6}$ | $\overline{Y_5}$ | $\overline{Y_4}$ | $\overline{Y_3}$ | $\overline{Y_2}$ | $\overline{Y_1}$ | $\overline{Y_0}$ |
|------------------|---------------------|---------------------|---|---|---|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|
| 1                | 0                   | 0                   | 0 | 0 | 0 | 1                | 1                | 1                | 1                | 1                | 1                | 1                | 0                |
| 1                | 0                   | 0                   | 0 | 0 | 1 | 1                | 1                | 1                | 1                | 1                | 1                | 0                | 1                |
| 1                | 0                   | 0                   | 0 | 1 | 0 | 1                | 1                | 1                | 1                | 1                | 0                | 1                | 1                |
| 1                | 0                   | 0                   | 0 | 1 | 1 | 1                | 1                | 1                | 0                | 1                | 1                | 1                | 1                |
| 1                | 0                   | 0                   | 1 | 0 | 0 | 1                | 1                | 1                | 0                | 1                | 1                | 1                | 1                |
| 1                | 0                   | 0                   | 1 | 0 | 1 | 1                | 1                | 0                | 1                | 1                | 1                | 1                | 1                |
| 1                | 0                   | 0                   | 1 | 1 | 0 | 1                | 0                | 1                | 1                | 1                | 1                | 1                | 1                |
| 1                | 0                   | 0                   | 1 | 1 | 1 | 0                | 1                | 1                | 1                | 1                | 1                | 1                | 1                |
| 其它状态             | ×                   | ×                   | × | × | × | 1                | 1                | 1                | 1                | 1                | 1                | 1                | 1                |



## 全译码——各芯片地址惟一

| A15                                   | A14  | A13  | A12   | A11  | A10  | A9   | A8   | A7~A0 | 存储器 |     |
|---------------------------------------|------|------|-------|------|------|------|------|-------|-----|-----|
| P2.7                                  | P2.6 | P2.5 | P2.4  | P2.3 | P2.2 | P2.1 | P2.0 | P0    | IC1 |     |
| C                                     | B    | A    |       |      |      |      |      |       |     |     |
| 0                                     | 0    | 0    | 0     | 0    | 0    | 0    | 0    | 0~0   |     |     |
| 000B<br>$\overline{Y0}—\overline{CE}$ |      |      | 0000H |      |      |      |      |       |     |     |
|                                       |      |      | 1     | 1    | 1    | 1    | 1    | 1~1   |     |     |
|                                       |      |      | 1FFFH |      |      |      |      |       |     |     |
| 0                                     | 0    | 1    | 0     | 0    | 0    | 0    | 0    | 0~0   |     | IC2 |
| 001B<br>$\overline{Y1}—\overline{CE}$ |      |      | 2000H |      |      |      |      |       |     |     |
|                                       |      |      | 1     | 1    | 1    | 1    | 1    | 1~1   |     |     |
|                                       |      |      | 3FFFH |      |      |      |      |       |     |     |

3—8译码器的其它输出端代表的地址是什么？



## 线选法与译码法比较

### ③ 线选法选址

- 电路连接简单
- 地址空间利用率低
- 地址空间重叠严重

### ③ 译码法选址

- 采用译码器电路
- 部分译码仍有重叠的地址空间
- 全译码地址空间利用率高，地址唯一

### 9.1.3 外部地址锁存器

AT89S51单片机受引脚数的限制，P0口兼用数据线和低8位地址，为了将它们分离出来，需要在单片机外部增加地址锁存器。目前，常用的地址锁存器芯片有74LS373、74LS573等。

### 1. 锁存器74LS373

带有三态门的8D锁存器，其引脚见图9-6，内部结构如图9-7所示。

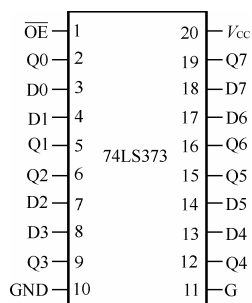


图9-6 锁存器74LS373的引脚

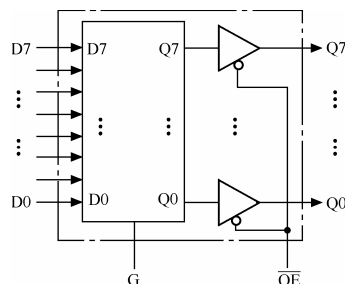


图9-7 74LS373的内部结构

### 74LS373引脚说明：

- D7~D0—8位数据输入线。
- Q7~Q0—8位数据输出线。
- G—数据输入锁存选通信号。当加到该引脚的信号为高电平时，外部数据选通到内部锁存器，负跳变时，数据锁存到锁存器中。
- OE\*—数据输出允许信号，低电平有效。当该信号为低电平时，三态门打开，锁存器中数据输出到数据输出线。当该信号为高电平时，输出线为高阻态。

74LS373锁存器功能见表9-4。



表9-4 74LS373功能表

| $\overline{OE}$ | G | D | Q   |  |
|-----------------|---|---|-----|--|
| 0               | 1 | 1 | 1   |  |
| 0               | 1 | 0 | 0   |  |
| 0               | 0 | × | 不变  |  |
| 1               | × | × | 高阻态 |  |

AT89S51单片机与74LS373锁存器的连接如图9-8所示。

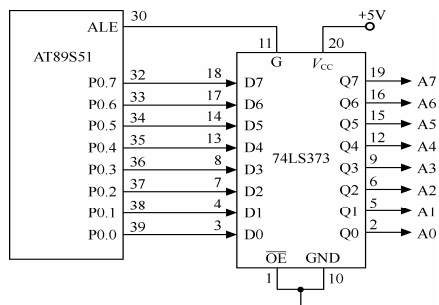


图9-8 AT89S51单片机P0口与74LS373的连接

33

## 2. 锁存器74LS573

带三态门的8D锁存器，功能及内部结构与74LS373完全一样，只是其引脚排列与74LS373不同，图9-9为74LS573引脚图。

由图9-9，与74LS373相比，74LS573输入D端和输出的Q端依次排列在芯片两侧，为绘制印制电路板提供较大方便。

74LS573引脚说明如下。

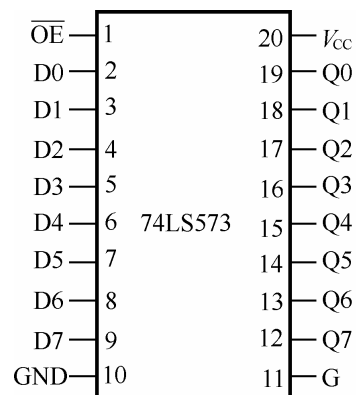


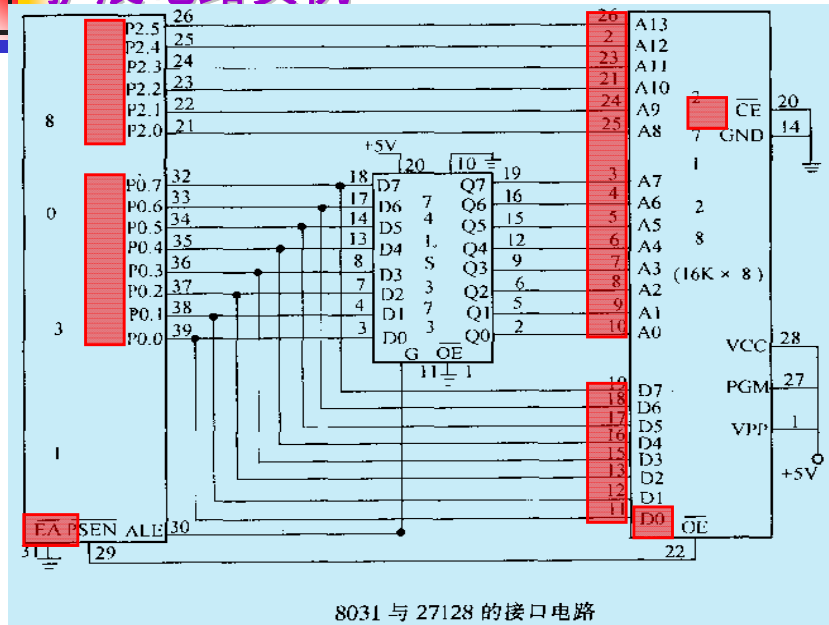
图9-9 锁存器74LS573的引脚

- D7~D0—8位数据输入线。
- Q7~Q0—8位数据输出线。
- G—数据输入锁存选通，该引脚与74LS373的G端功能相同。
- OE\*—数据输出允许，低电平有效。当低电平时，三态门打开，锁存器中数据输出到数据输出线。当该信号为高电平时，输出线为高阻态。

34

36

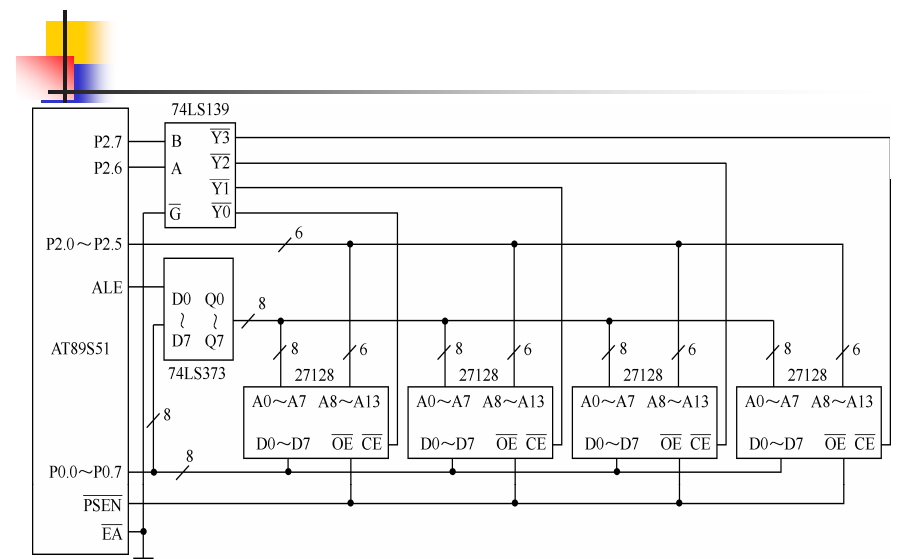
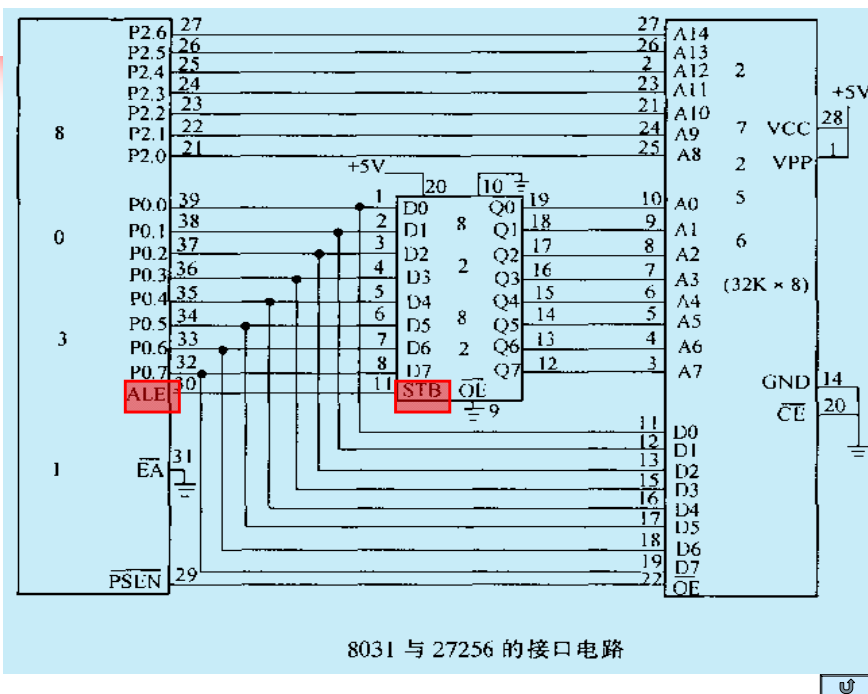
## 扩展电路实例



## 地址空间——地址段不唯一

| A15  | A14                     | A13  | A12  | A11  | A10  | A9   | A8   | A7~A0 | 存储器 |
|------|-------------------------|------|------|------|------|------|------|-------|-----|
| P2.7 | P2.6                    | P2.5 | P2.4 | P2.3 | P2.2 | P2.1 | P2.0 | P0    | IC  |
| X    | X                       | 0    | 0    | 0    | 0    | 0    | 0    | 0~0   |     |
| 00B  | 0000H、4000H、8000H、C000H |      |      |      |      |      |      |       |     |
| 01B  | 1                       | 1    | 1    | 1    | 1    | 1    | 1    | 1~1   |     |
| 10B  | 3FFFH、7FFFH、BFFFH、FFFFH |      |      |      |      |      |      |       |     |
| 11B  |                         |      |      |      |      |      |      |       |     |

片外剩2根线，每片有4段地址段



## 9.1 系统并行扩展技术

## 9.2 外部数据存储器的扩展方法

## 9.3 片内Flash存储器的编程

## 9.4 E2PROM的并行扩展

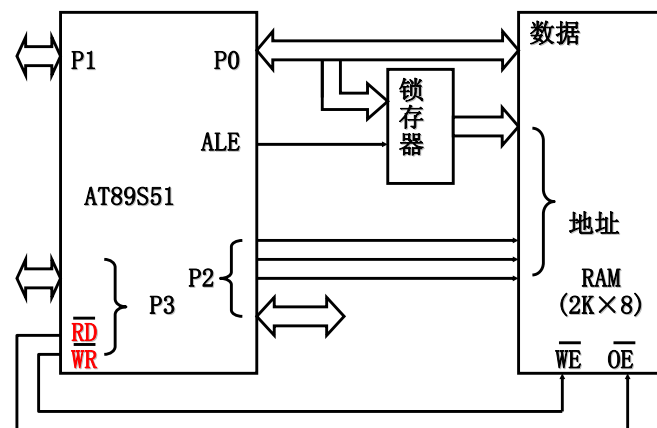
## 9.5 并行I/O芯片82C55的设计

## 9.2 外部数据存储器的扩展方法

P0口为RAM的复用地  
址/数据总线，

P2口的三  
根线用于对  
RAM进行页面  
寻址。

在对外部  
RAM读/写期  
间，CPU产生  
/RD/WR信号。



### 9.2.1 常用的静态RAM (SRAM) 芯片

单片机系统中常用RAM典型芯片有6116 (2KB)、6264 (8KB)、62128 (16KB)、62256 (32KB)。

都单一+5V电源供电，双列直插，6116为24引脚，6264、62128、62256为28引脚。RAM芯片引脚见图9-10。

各引脚功能如下。

● A0~A14—地址输入线。

● D0~D7—双向三态数据线。

● CE\*—片选信号输入线，低电平有效。对于6264芯片，当24脚 (CS) 为高电平且为低电平时才选中该片。

● OE\*—读选通信号输入线，低电平有效。

● WE\*—写允许信号输入线，低电平有效。

● VCC—工作电源+5V。

● GND—地

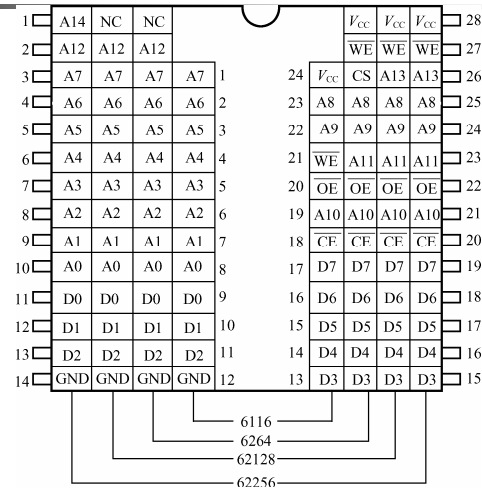


图9-10 常用RAM引脚

45

## 常用的RAM芯片

在89S51应用系统中，最常用的静态随机存取存储器RAM电路有6116(2K×8)和6264(8K×8)。

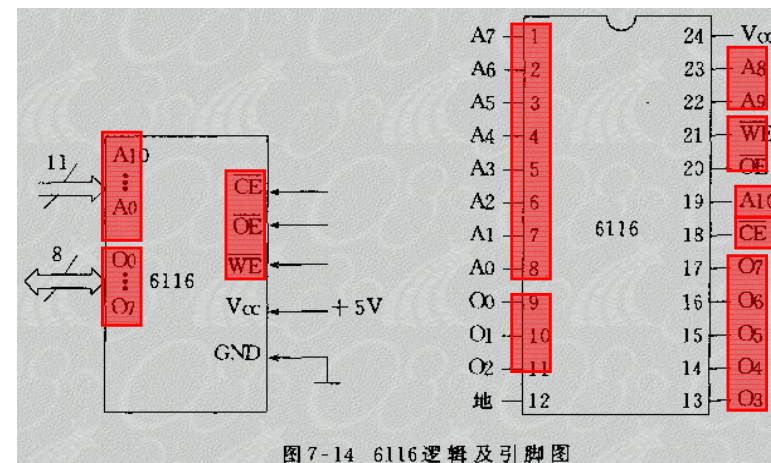


图 7-14 6116逻辑及引脚图

## 常用的RAM芯片

静态随机存取存储器6116(2K×8)和6264(8K×8)。

信号线; $\overline{OE}$ 、 $\overline{WE}$ 为读/写信号线。6116 的操作方式控制如表 7-1 所列。

表 7-1 6116 操作方式选择

| $\overline{CE}$ | $\overline{WE}$ | $\overline{OE}$ | 方 式 | 功 能                     |
|-----------------|-----------------|-----------------|-----|-------------------------|
| 0               | 0               | 1               | 写   | O0~O7 上内容写入 A0~A10 对应单元 |
| 0               | 1               | 0               | 读   | A0~A10 对应单元内容输出到 O0~O7  |
| 1               | X               | X               | 非选  | O0~O7 呈高阻抗              |

## 常用的RAM芯片

静态随机存取存储器6264(8K×8)。

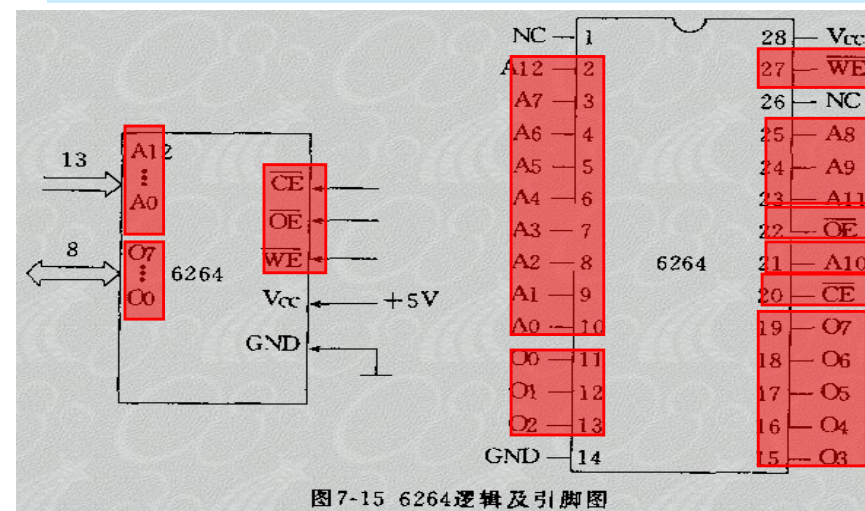


图 7-15 6264逻辑及引脚图

## 常用的RAM芯片

静态随机存取存储器6264(8K×8)。

表 7-2 6264 操作方式选择

| $\overline{CE}$ | $\overline{WE}$ | $\overline{OE}$ | 方 式 | 说 明                        |
|-----------------|-----------------|-----------------|-----|----------------------------|
| 0               | 0               | 1               | 写   | 00~07 上信息写入 A0~A12 上地址对应单元 |
| 0               | 1               | 0               | 读   | A0~A12 上地址对应单元内容输出到 00~07  |
| 1               | X               | X               | 非选  | 00~07 呈高阻抗                 |

## 9.2.2 并行扩展数据存储器的设计

访问外扩展的数据存储器，要由P2口提供高8位地址，P0口提供低8位地址和8位双向数据总线。AT89S51单片机对片外RAM的读和写由AT89S51的(P3.7)和(P3.6)信号控制，片选端由地址译码器的译码输出控制。因此，进行接口设计时，主要解决地址分配、数据线和控制信号线的连接。如果读/写速度要求较高，还要考虑单片机与RAM的读/写速度匹配问题。

图9-13所示为用线选法扩展外部数据存储器的电路。数据存储器选用6264，该芯片地址线为A0~A12，故AT89S51单片机剩余地址线为3条。用线选法可扩展3片6264，3片6264的存储器空间如表9-6所示。

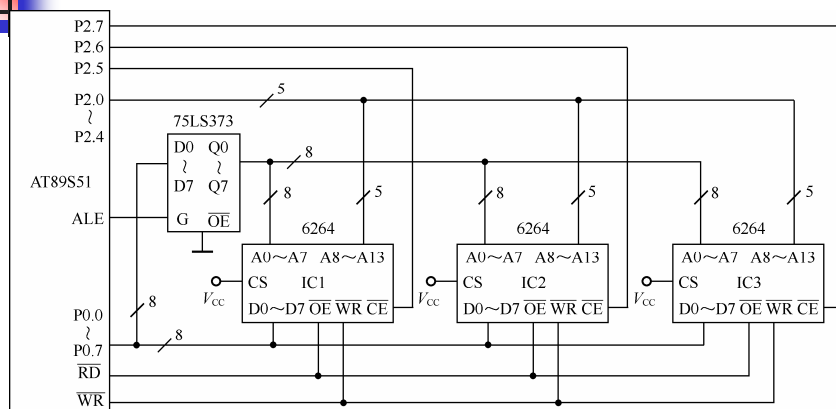


图9-13 线选法扩展外部数据存储器电路图

表 9-6 3片 6264 芯片对应的存储空间表

| P2.7 | P2.6 | P2.5 | 选中芯片 | 地址范围        | 存储容量 |
|------|------|------|------|-------------|------|
| 1    | 1    | 0    | IC1  | C000H~DFFFH | 8KB  |
| 1    | 0    | 1    | IC2  | A000H~BFFFH | 8KB  |
| 0    | 1    | 1    | IC3  | 6000H~7FFFH | 8KB  |

用译码法扩展外部数据存储器的接口电路如图9-14所示。图中数据存储器选用62128，该芯片地址线为A0~A13，这样，AT89S51剩余地址线为两条，采用2-4译码器可扩展4片62128。各62128芯片的地址范围如表9-7所示。



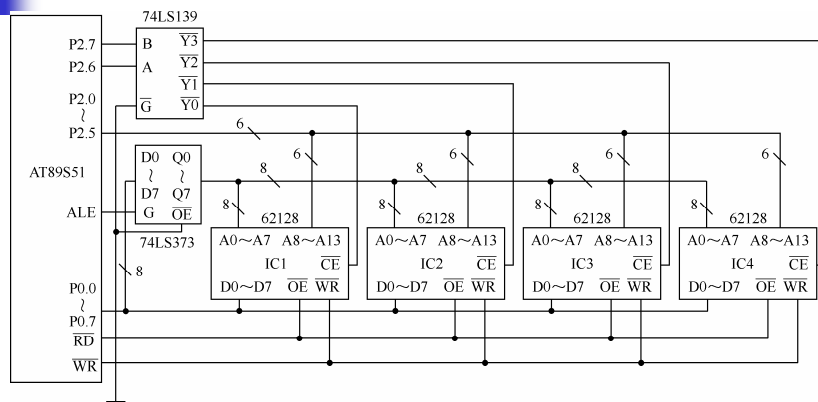


图9-14 译码法扩展外部数据存储电路图



表9-7 各62128芯片的地址空间分配

| 2-4 译码器输入<br>P2.7 P2.6 | 2-4 译码器<br>有效输出 | 选中芯片 | 地址范围        | 存储容量 |
|------------------------|-----------------|------|-------------|------|
| 0 0                    | $\overline{Y0}$ | IC1  | 0000H~3FFFH | 16KB |
| 0 1                    | $\overline{Y1}$ | IC2  | 4000H~7FFFH | 16KB |
| 1 0                    | $\overline{Y2}$ | IC3  | 8000H~BFFFH | 16KB |
| 1 1                    | $\overline{Y3}$ | IC4  | C000H~FFFFH | 16KB |

【例9-1】编写程序将片外数据存储器中的0x5000~0x50FF的256个单元全部清“0”。参考程序：

```
xdata unsigned char databuf[256] _at_ 0x5000;
void main(void)
{
    unsigned char i;
    for(i=0;i<256;i++)
    {
        databuf[i]=0;
    }
}
```

## 9.1 系统并行扩展技术

## 9.2 外部数据存储器的扩展方法

## 9.3 片内Flash存储器的编程

## 9.4 E2PROM的并行扩展

## 9.5 并行I/O芯片82C55的设计

### 9.3 片内Flash存储器的编程

程序存储器具有非易失性，在电源关断后，存储器仍能保存程序，并且程序存储器中的信息一旦写入，就不能随意更改，特别不能在程序运行过程中写入新的内容，故称为只读存储器（ROM）。

本小节只讨论如何把已调试完毕的程序代码写入到AT89S51的片内Flash存储器中，即对Flash存储器的编程问题。

AT89S51单片机片内4KB Flash存储器的基本特性如下：

- (1) 可循环写入/擦除1 000次；
- (2) 存储器数据保存时间为10年；
- (3) 具有3级加密保护。



### 9.3.1 使用通用编程器的程序写入

通用编程器一般通过串行口或USB口与PC机相连，并配有相应的驱动软件。编程器通过USB口与PC机通讯，可进行芯片型号自动判别，编程过程中的擦除、烧写、校验等各种操作。

编程器供电部分由USB端口的5V电源提供，省去笨重的外接电源并加入USB接口保护电路，即自恢复保险丝，不怕操作短路。

编程器的驱动软件界面友好，菜单、工具栏、快捷键齐全，具有编程、读取、校验、空检查、擦除、Flash存储器加密等功能。



### 9.3.2 使用下载线的ISP编程

AT89S5x系列单片机支持对片内Flash存储器在线编程（ISP），即PC机直接通过下载线向单片机片内Flash存储器写入程序代码。编程完毕的片内Flash存储器也可用ISP方式擦除或再编程。

ISP下载线按与PC机的连接分为三种类型：串口型、并口型以及USB型，可自行制作，也可在电子市场购买。常用的为USB型，并且接口标准为ATMEL公司的标准，10引脚IDC，如图9-17所示。

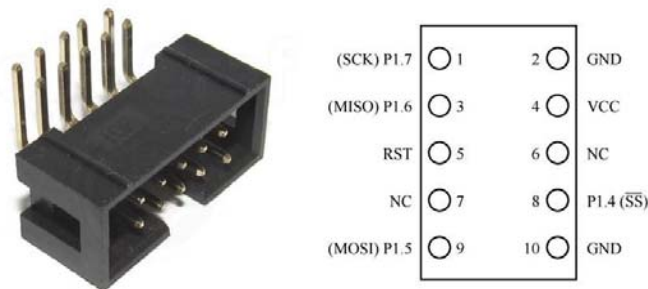


图9-17 IDC端口的实物图以及端口的定义



## 9.1 系统并行扩展技术

## 9.2 外部数据存储器的扩展方法

## 9.3 片内Flash存储器的编程

## 9.4 E2PROM的并行扩展

## 9.5 并行I/O芯片82C55的设计





## 9.4 E2PROM的并行扩展

在以单片机为核心的智能仪器仪表、工业监控等应用系统中，对某些状态参数数据，不仅要求能够在线修改保存，而且断电后能保持。断电数据的保护可采用电可擦除写入的存储器E2PROM，其突出优点是能够在线擦除和改写，并且，操作简单，可字节写入，非常适合用作运行过程中频繁改写某些非易失的小数据量的存储器。

E2PROM有并行和串行之分，并行E2PROM的速度比串行的快，容量大。串行E2PROM的扩展将在下一章介绍。本节只介绍AT89S51单片机扩展并行E2PROM芯片2864的设计。



### 9.4.1 并行E2PROM芯片简介

常见的并行E2PROM芯片有2816/2816A，2817/2817A，2864A等。这些芯片的引脚如图9-18所示。

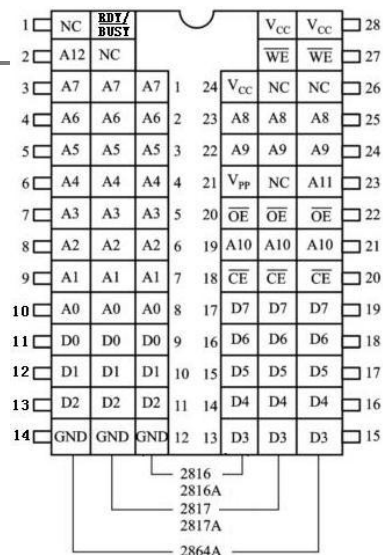


图9-18 常见的并行E2PROM引脚图



### 9.4.2 AT89S51单片机扩展E2PROM AT2864的设计

2864A与AT89S51单片机的接口电路如图9-19所示。2864A的存储容量为8K字节，与同容量的静态RAM 6264的引脚是兼容的，2864A的片选端由高位地址线P2.7（A15）来控制。

单片机对2864A的读写非常方便，在单一+5V电压下写入新数据即覆盖了旧的数据，类似于对RAM的读写操作，2864A典型的读出数据时间为200~350ns，但是字节编程写入时间为10ms~15ms，要比对RAM写入时间长许多。



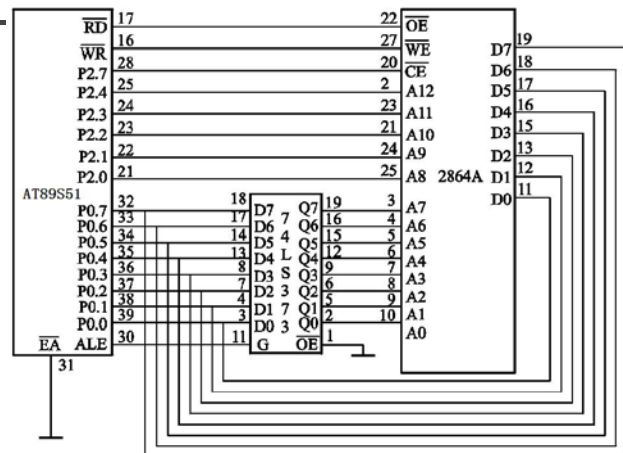


图9-19 2864A与AT89S51单片机的接口电路

## 9.1 系统并行扩展技术

## 9.2 外部数据存储器的扩展方法

## 9.3 片内Flash存储器的编程

## 9.4 E2PROM的并行扩展

## 9.5 并行I/O芯片82C55的设计

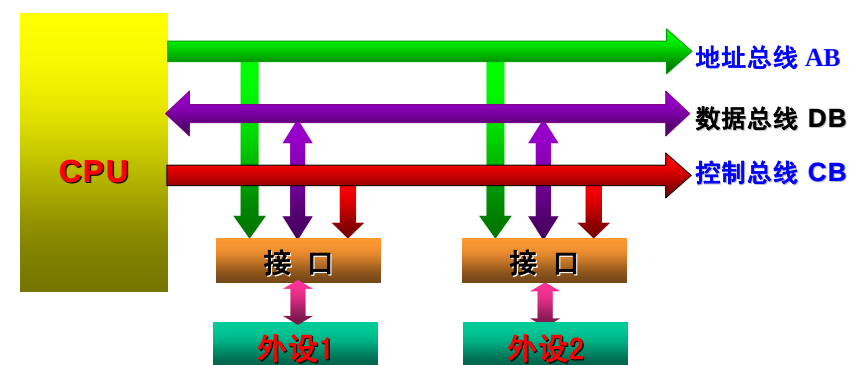
## 9.5 并行I/O芯片82C55的设计

AT89S51本身有4个通用的并行I/O口P0~P3，但真正用作通用I/O口线的只有P1口和P3口某些位线。

当AT89S51本身4个并行I/O口不够用时，需进行外部I/O接口扩展。本节介绍AT89S51单片机扩展可编程并行I/O接口芯片82C55的设计。

此外还介绍使用廉价的74LSTTL芯片扩展并行I/O接口以及使用AT89S51串行口来扩展并行I/O口的设计。

外设与CPU是通过接口连接

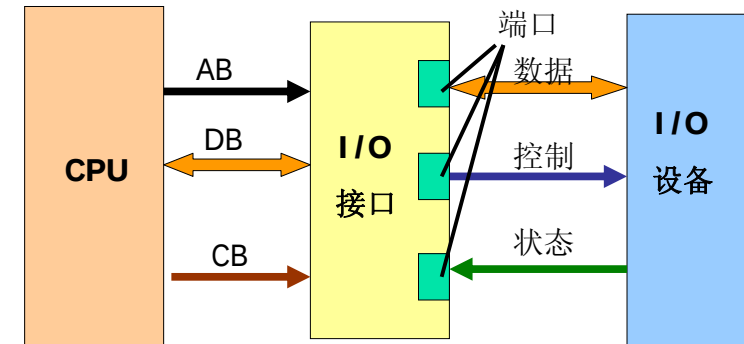


## I/O接口和 I/O端口

1. I/O接口(Interface)是一电子电路(以IC芯片或接口板形式出现), 其内有若干专用寄存器和相应的控制逻辑电路构成。它是CPU和I/O设备之间交换信息的媒介和桥梁。

2. I/O端口( Port )是 I/O接口中可通过编程实现寻址并进行读写的寄存器。CPU 与外设之间交换信息具体是通过I/O端口来进行的。

CPU与外设通过I/O接口通信示意图:



## 9.5.2 I/O端口的编址方式

### 独立编址

采用一套与存储器不同的地址, 利用/MREQ (存储器请求) 和/IORQ (输入输出请求), CPU 有专门的I/O指令 如: IN, OUT

### 统一编址

I/O端口与内存储器采用一套地址, 完全象存储器单元一样处理, 使用访问存储器的指令。

如: movx 类指令

### MCS51系列采用统一编址

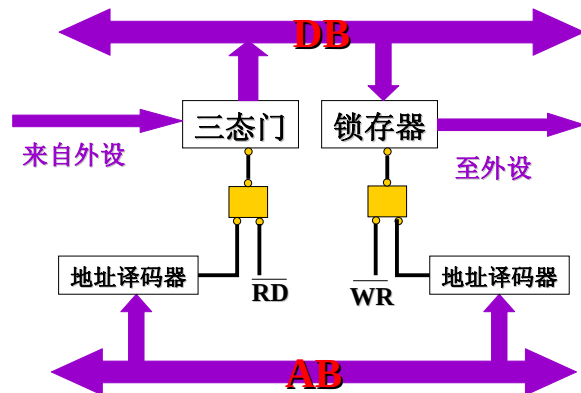
片内RAM 与P0~P3均用MOV指令

片外RAM与外扩口I/O口均用MOVX指令

## 9.5.3 I/O数据传输的方式

- 同步传输 (无条件传输)
- 查询传输
- 中断传输

## 1 无条件传送方式（又称同步传送）



- ◆ 应用于定时为已知的且固定不变的低速I/O
- ◆ 无需等待的高速I/O

## 2 查询式传送方式,又称条件传送——异步传送

- ✂ CPU查询外设状态信息(Ready, Busy), 条件满足时, 进行数据传送
- ✂ 程序简单
- ✂ 高速CPU查询低速外设
- ✂ 浪费CPU时间, 效率低

## 3 中断传送方式

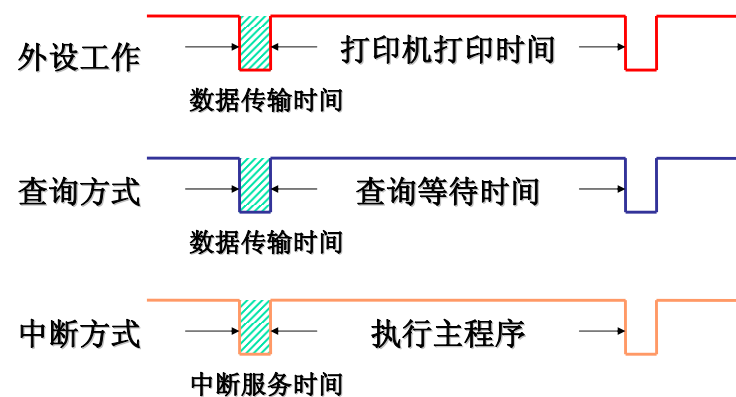
✂ **中断**: 要求进行输入、输出的外设, 发出就绪信号给CPU, 作为中断请求, 打断CPU正在进行的工作, 即中断CPU正在执行的程序。

✂ **中断过程**

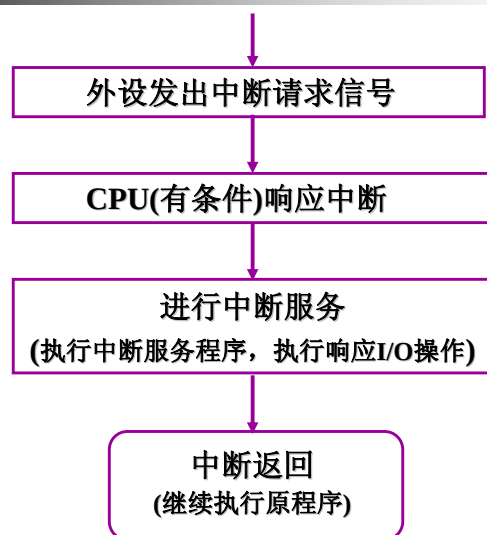
✂ **中断方式与查询方式比较**

- ✂ 提高了CPU的效率
- ✂ CPU与外设可并行工作
- ✂ CPU可及时响应外部事件

## 中断方式与查询方式CPU占用时间比较



## 中断过程



## 9.5.4 并行接口8255A

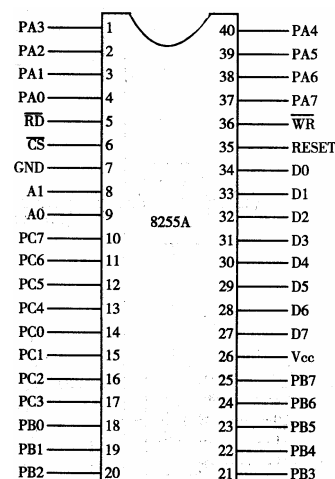
- Intel公司生产的可编程并行I/O接口芯片
- 含3个独立的8位并行输入/输出端口，各端口均具有数据的控制和锁存能力
- 可通过编程，设置各端口工作在某一确定状态下。



## 1. 8255芯片介绍

### (1) 引脚

- D7~D0: 与51双向数据传送
- $\overline{CS}$ : 片选信号
- $\overline{RD}$ : 读选通
- $\overline{WR}$ : 写选通
- PA7 ~ PA0: A口 } 并行 I/O
- PB7 ~ PB0: B口 }
- PC7 ~ PC0: C口 }
- A1、A0: 端口选择



## 8255A并行接口结构图

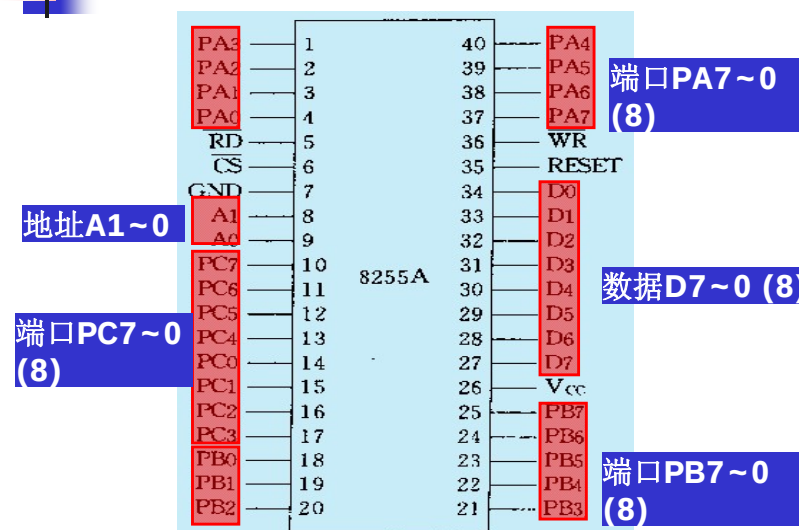
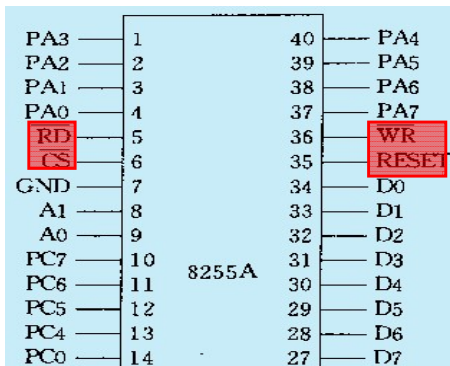


图 8-1 8255A引脚图

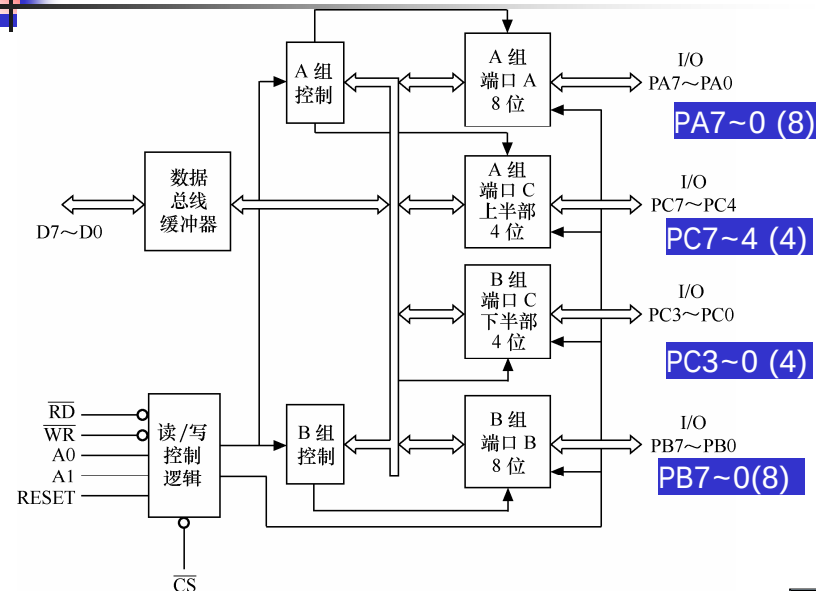


## 8255A并行接口结构图



**/CS:** 片选信号线，低电平有效，表示芯片被选中；  
**/RD:** 读信号线，低电平有效，控制数据读出；  
**/WR:** 写信号线，低电平有效，控制数据写入；  
**RESET:** 复位信号线，高电平有效；

## 8255A并行接口结构图

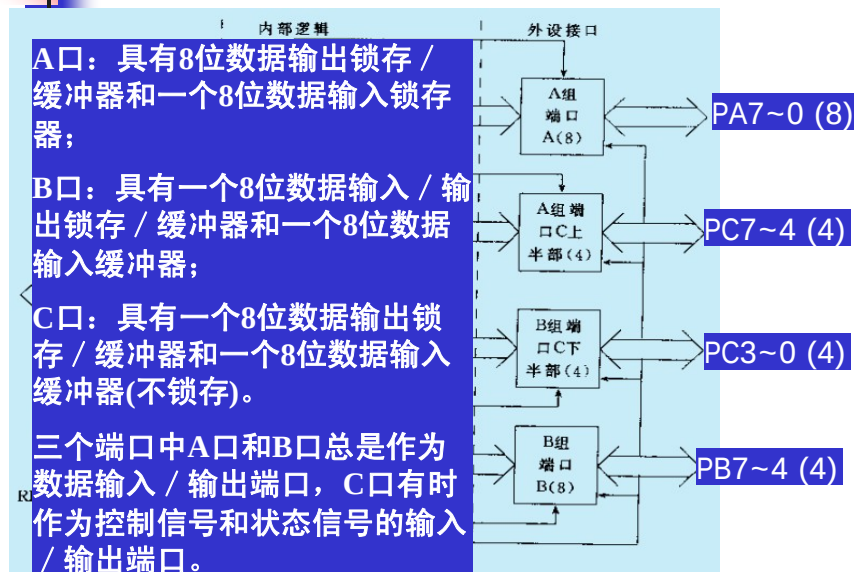


## 结构

**A组** { 端口A  
端口C的高4位

**B组** { 端口B  
端口C的低4位

## 8255A并行接口结构图





## 8255A的读/写控制

读 / 写控制逻辑电路输入的控制信号有 /RD、/WR、RESET和A1、A0。它根据这些信号控制I / O口及控制寄存器的读 / 写操作。

其中地址线A1、A0用来选择I / O口和控制字寄存器，与读 / 写控制信号 /RD和/WR构成各种工作状态，如表9-1所示。

## 8255A I/O口工作状态

表 8-1 8255A 的 I/O 口工作状态

| A1 | A0 | $\overline{RD}$ | $\overline{WR}$ | $\overline{CS}$ | 工作状态        |         |
|----|----|-----------------|-----------------|-----------------|-------------|---------|
| 0  | 0  | 0               | 1               | 0               | A 口数据→数据总线  | Read 读  |
| 0  | 1  | 0               | 1               | 0               | B 口数据→数据总线  |         |
| 1  | 0  | 0               | 1               | 0               | C 口数据→数据总线  |         |
| 0  | 0  | 1               | 0               | 0               | 总线数据→A 口    | Write 写 |
| 0  | 1  | 1               | 0               | 0               | 总线数据→B 口    |         |
| 1  | 0  | 1               | 0               | 0               | 总线数据→C 口    |         |
| 1  | 1  | 1               | 0               | 0               | 总线数据→控制字寄存器 |         |
| X  | X  | X               | X               | 1               | 数据总线→三态     | 禁止      |
| 1  | 1  | 0               | 1               | 0               | 非法状态        |         |
| X  | X  | 1               | 1               | 0               | 数据总线→三态     |         |

## 8255A I/O口工作状态

表 8-1 8255A 的 I/O 口工作状态

| A1 | A0 | $\overline{RD}$           | $\overline{WR}$ | $\overline{CS}$ | 工作状态       |         |
|----|----|---------------------------|-----------------|-----------------|------------|---------|
| 0  | 0  | 0                         | 1               | 0               | A 口数据→数据总线 | Read 读  |
| 0  | 1  | 0                         | 1               | 0               | B 口数据→数据总线 |         |
| 1  | 0  | 0                         | 1               | 0               | C 口数据→数据总线 |         |
| 0  | 0  | 1                         | 0               | 0               | 总线数据→A 口   | Write 写 |
| 0  | 1  | 1                         | 0               | 0               | 总线数据→B 口   |         |
| 1  | 0  | 1                         | 0               | 0               | 总线数据→C 口   |         |
| 1  | 1  | A1 A0<br>0 0              |                 |                 | 寄存器        |         |
| ×  | ×  | 00是对A口进行输入/输出操作，可计算PA口的地址 |                 |                 | 禁止         |         |
| 1  | 1  |                           |                 |                 |            |         |
| ×  | ×  |                           |                 |                 |            |         |

## 8255A I/O口工作状态

表 8-1 8255A 的 I/O 口工作状态

| A1 | A0 | $\overline{RD}$           | $\overline{WR}$ | $\overline{CS}$ | 工作状态      |         |
|----|----|---------------------------|-----------------|-----------------|-----------|---------|
| 0  | 0  | 0                         | 1               | 0               | A口数据→数据总线 | Read 读  |
| 0  | 1  | 0                         | 1               | 0               | B口数据→数据总线 |         |
| 1  | 0  | 0                         | 1               | 0               | C口数据→数据总线 |         |
| 0  | 0  | 1                         | 0               | 0               | 总线数据→A口   | Write 写 |
| 0  | 1  | 1                         | 0               | 0               | 总线数据→B口   |         |
| 1  | 0  | 1                         | 0               | 0               | 总线数据→C口   |         |
| 1  | 1  | A1 A0<br>0 1              |                 |                 | 字寄存器      | 禁止      |
| X  | X  | 01是对B口进行输入/输出操作，可计算PB口的地址 |                 |                 |           |         |
| 1  | 1  |                           |                 |                 |           |         |
| X  | X  |                           |                 |                 |           |         |
|    |    |                           |                 |                 | 数据总线→二态   |         |



## 8255A I/O口工作状态

表 8-1 8255A 的 I/O 口工作状态

| A1 | A0 | $\overline{RD}$            | $\overline{WR}$ | $\overline{CS}$ | 工作状态       |         |
|----|----|----------------------------|-----------------|-----------------|------------|---------|
| 0  | 0  | 0                          | 1               | 0               | A 口数据→数据总线 | Read 读  |
| 0  | 1  | 0                          | 1               | 0               | B 口数据→数据总线 |         |
| 1  | 0  | 0                          | 1               | 0               | C 口数据→数据总线 |         |
| 0  | 0  | 1                          | 0               | 0               | 总线数据→A 口   | Write 写 |
| 0  | 1  | 1                          | 0               | 0               | 总线数据→B 口   |         |
| 1  | 0  | 1                          | 0               | 0               | 总线数据→C 口   |         |
| 1  | 1  | A1 A0<br>1 0               |                 |                 | 寄存器        | 禁止      |
| X  | X  | 01是对B口进行输入/输出操作, 可计算PB口的地址 |                 |                 |            |         |
| 1  | 1  |                            |                 |                 |            |         |
| X  | X  |                            |                 |                 |            |         |

## 8255A I/O口工作状态

表 8-1 8255A 的 I/O 口工作状态

| A1 | A0 | $\overline{RD}$ | $\overline{WR}$ | $\overline{CS}$ | 工作状态        |         |
|----|----|-----------------|-----------------|-----------------|-------------|---------|
| 0  | 0  | 0               | 1               | 0               | A 口数据→数据总线  | Read 读  |
| 0  | 1  | A1 A0<br>1 1    |                 |                 | 数据总线        |         |
| 1  | 0  |                 |                 |                 | 数据总线        |         |
| 0  | 0  | 1 1             | 0 0             | 0               | A 口         | Write 写 |
| 0  | 1  |                 |                 |                 | B 口         |         |
| 1  | 0  |                 |                 |                 | 总线数据→C 口    |         |
| 1  | 1  | 1               | 0               | 0               | 总线数据→控制字寄存器 | 禁止      |
| X  | X  | X               | X               | 1               | 数据总线→三态     |         |
| 1  | 1  | 0               | 1               | 0               | 非法状态        |         |
| X  | X  | 1               | 1               | 0               | 数据总线→三态     |         |

### 9.5.5 工作方式控制字及C口控制字

#### (1) 基本工作方式

- **方式0, 基本输入输出:**
  - 没有固定的信号联线, 仅完成无条件的数据传输
- **方式1, 选通输入输出:**
  - A、B为单纯的数据口, C口为联络信号的传输。
- **方式2, 双向传送(仅PA口):**
  - 双向的I/O总线
- 工作方式的选择由CPU送出的控制字寄存器选择

#### 方式0:

- 相当于三个独立的8位简单接口
- 各端口既可设置为输入口, 也可设置为输出口, 但不能同时实现输入及输出
- C端口可以是一个8位的简单接口, 也可以分为两个独立的4位端口
- 常用于连接简单外设 (适于无条件或查询方式)

## 方式1:

- 利用一组选通控制信号控制A端口和B端口的数据输入输出
- A口、B口作输入或输出口，C口的部分位用作选通控制信号
- A口、B口在作为输入和输出时的选通信号不同



## 方式1的应用:

- 方式1主要用于中断控制方式下的输入输出
- C口的8位除用作选通信号外，其余位可工作于方式0下，作为输入或输出口



## 方式2:

- 双向输入输出方式-----可以既作为输入口，又作为输出口
- 只有A端口可工作在方式2下



## 方式2的应用:

- 可使A端口作为双向端口所有
- 用于中断控制方式
- 当A口工作于方式2时，B口可工作于方式1（此时C口的所有位都用作选通控制信号的输入输出），也可工作于方式0（此时C口的剩余位也可工作于方式0）



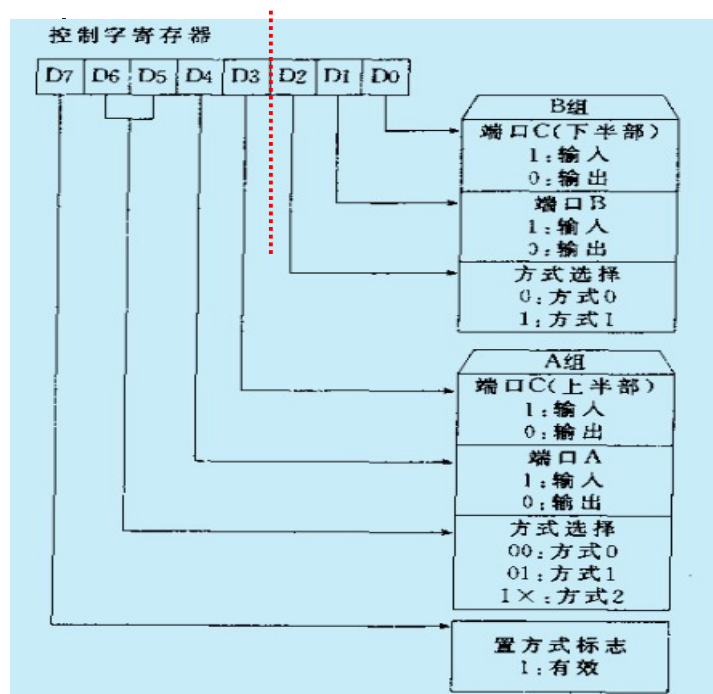
## (2) 方式控制字及状态字

- 利用软件编程确定3个端口工作于何种方式下；
- **C端口可以按位操作。**当其工作于方式**0**下且作为输出口时，一般需要对作为输出的位设置初始状态（即初始化）



## (2) 方式控制字与状态字格式

- 控制字-----确定3个端口的工作方式
- 状态字-----确定C口某一位的初始状态



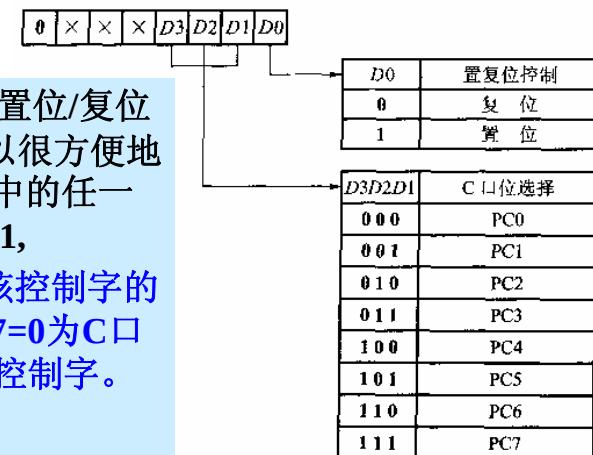
控制字寄存器

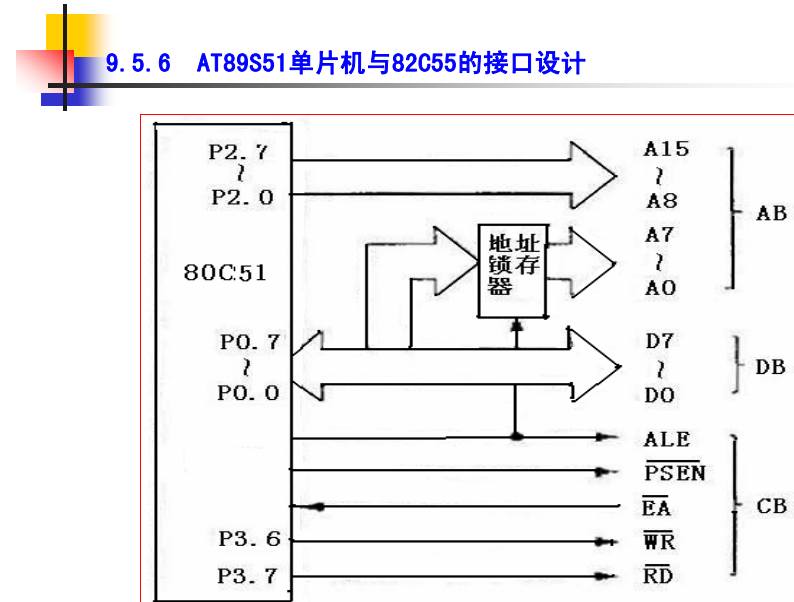
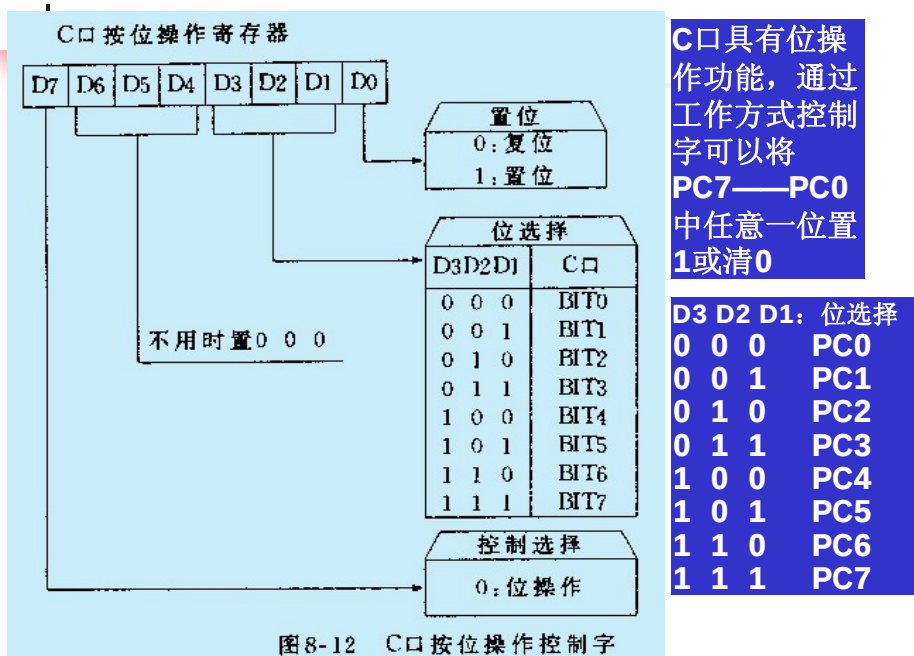
**D7=1: 命令控制字有效。**  
当**D7=0**，通过控制口对C口进行位操作。



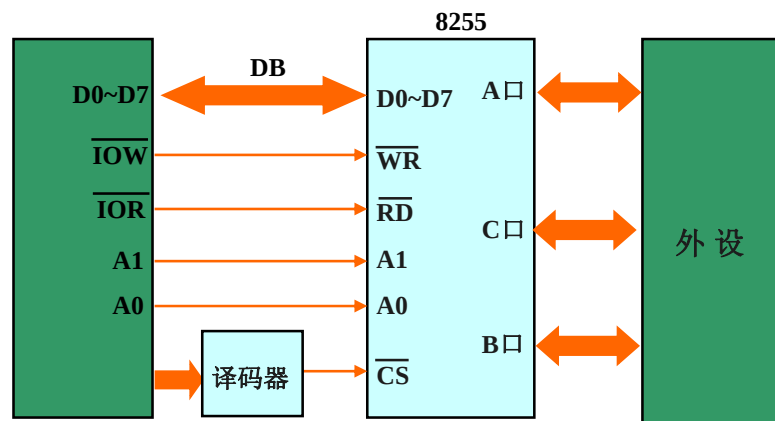
## (3) C口按位置位/复位控制字

- 利用C口置位/复位控制字可以很方便地使C口8位中的任一位清0或置1,
- **D7位为该控制字的标志位, D7=0为C口置位/复位控制字。**





## 8255与系统的连接示意图



## 编程要点

- 初始化时要设置8255的工作方式及端口的输入输出
- 端口地址的确定(根据电路原理图)





```
extern void delay_1000us ( );
void init8255(void)
{
    COM8255=0x85;    //工作方式控制字写入控制寄存器
}
void main(void)
{
    init8255(void)
    for(;;)
    {
        COM8255=0x0b;    //PC5脚为高电平
        delay_1000us ( );    //高电平持续1000s
        COM8255=0x0a;    //PC5脚为低电平
        delay_1000us ( );    //低电平持续1000s
    }
}
```