

# AT89C52单片机原理

## 第8章 单片机并行扩展技术



# 1 单片机内部的并行I/O口及应用

---

单片机内部有四个8位并行I/O口：

 P0、P1、P2和P3

 P0~P3这4个并行I/O口都可以作准双向通用I/O口

 P0、P2和P3口还有复用的第二功能

每个口都包含：

 一个（8位）锁存器（即特殊功能寄存器P0~P3）

 一个输出驱动器

 和两个三态缓冲器

功能:

# 1.1 P0口

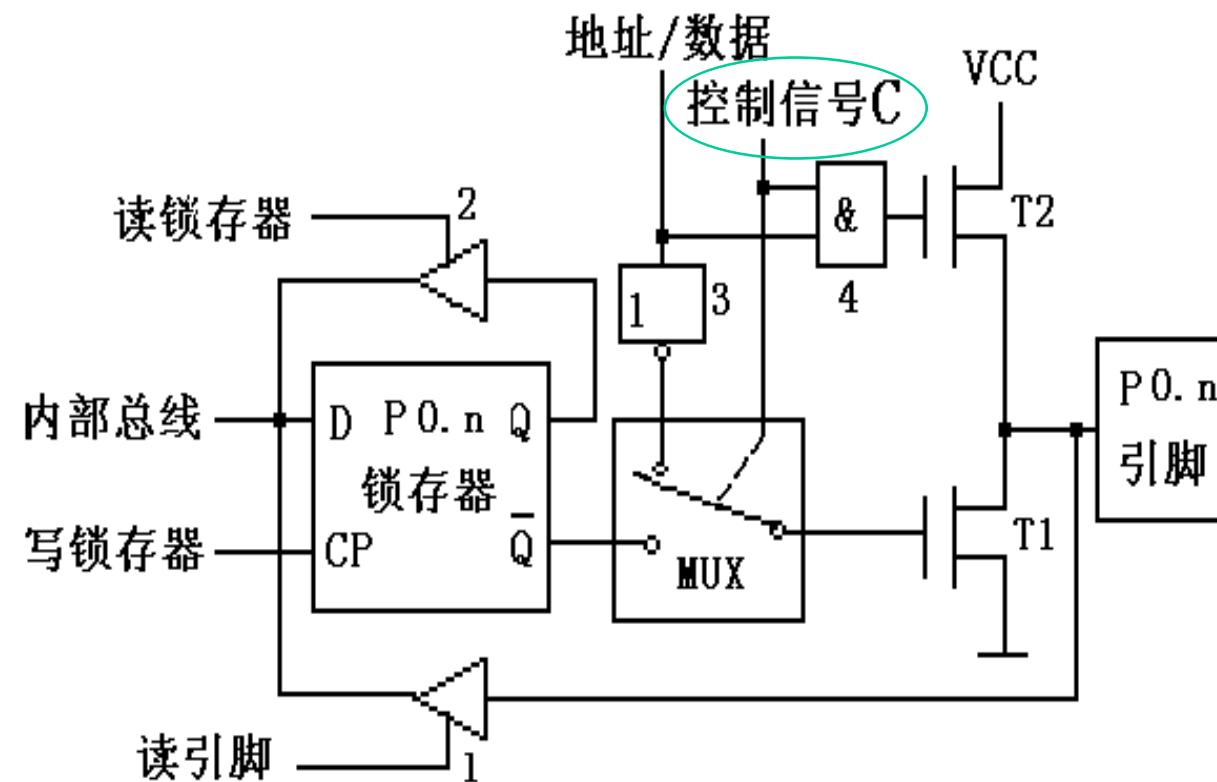
控制=0:

❓ P0口可作通用I/O口，是漏级开路的。因此必须外接上拉电阻，以保证“1”信号正常输出。

控制=1:

❓ P0口分时作为地址/数据总线使用。

内部结构: 如图所示



当外部扩展存储器或I/O口时，由EA、PC指针从外部程序存储器取指令码、执行指令MOVC、MOVX时，控制信号=1,使MUX与反向器3连通，作分时地址/数据总线。

特点:

- 通用I / O口:输出时接上拉电阻，输入时先写入“1”;
- 20P0口的每位输出可驱动8个LSTTL负载。

## 功能： P1口作通用I/O口

## 1.2 P1口

P1口也是一个“准双向”口，作输入口时要先将输出驱动管截止。

即先执行一条指令

**MOV P1,#0FFH**

内部结构： 如图所示

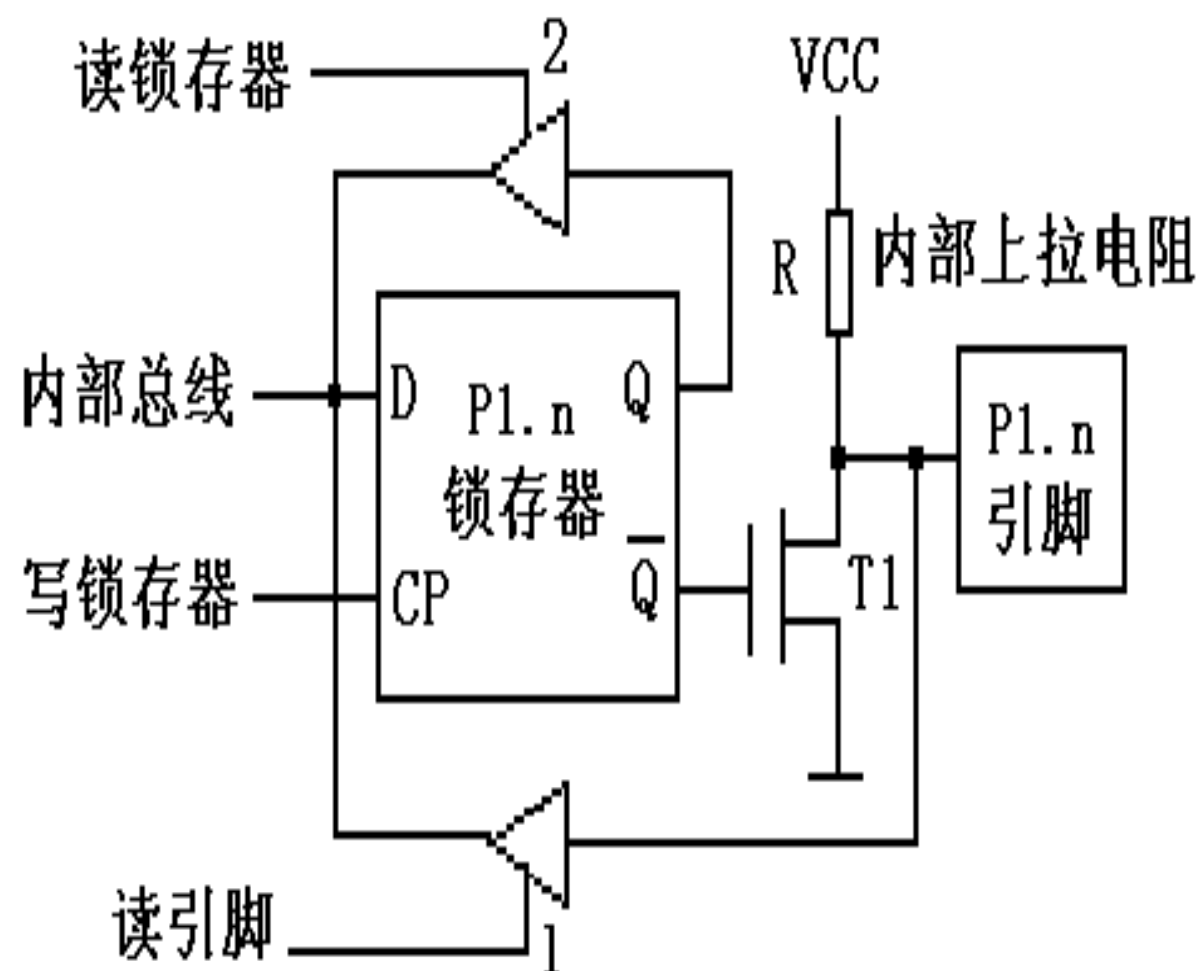
举例： 从P1口的低四位输入数据

**MOV P1,#00001111b ;先给P1口低四位写1**

**MOV A,P1 ; 再读P1口的低四位**

特点：

- 只作通用I / O口:输入时先向对应的锁存器写入“1”;
- P1口的每位输出可驱动4个LSTTL负载。



功能:

## 1.3 P2口

控制=0:

P2口用作通用I/O口

与P1口类似

控制=1:

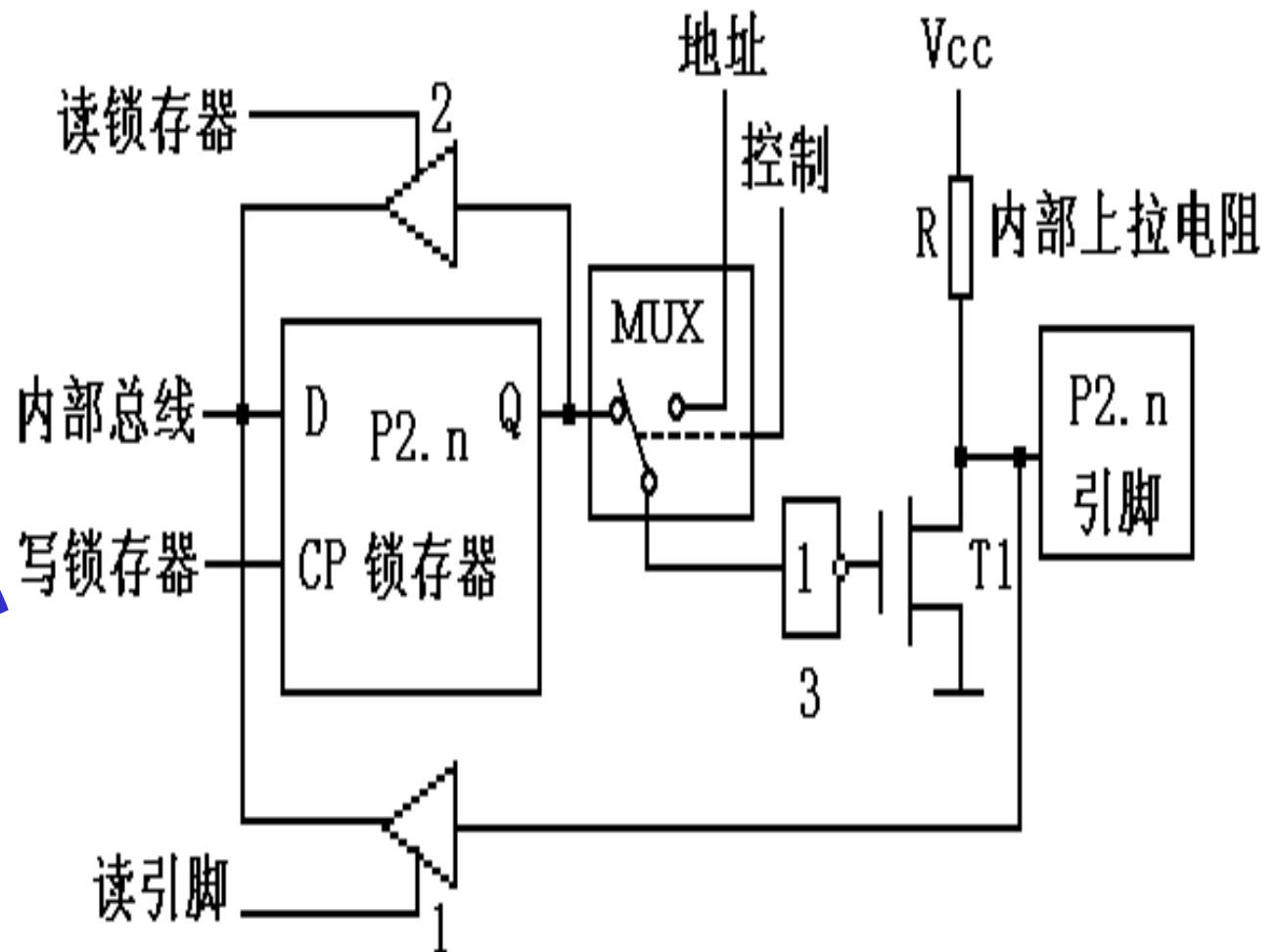
P2口用作高8位地址总线

线

内部结构: 如图所示

特点:

- P2口作高8位地址输出线时, 与P0口的低8位地址一起构成16位的地址总线;
- 作通用I/O口使用时, 功能与P1口一样;
- P2口的每位输出可驱动4个LSTTL负载。



功能:

❓ 第二输出功能端=1:

P3口用作通用I/O口

❓ P3口用作第二功能

❓ 三根第二功能输出引脚

TXD、 $\overline{WR}$ 和 $\overline{RD}$

❓ 五根第二功能输入引脚

$\overline{INT0}$ 、 $\overline{INT1}$ 、

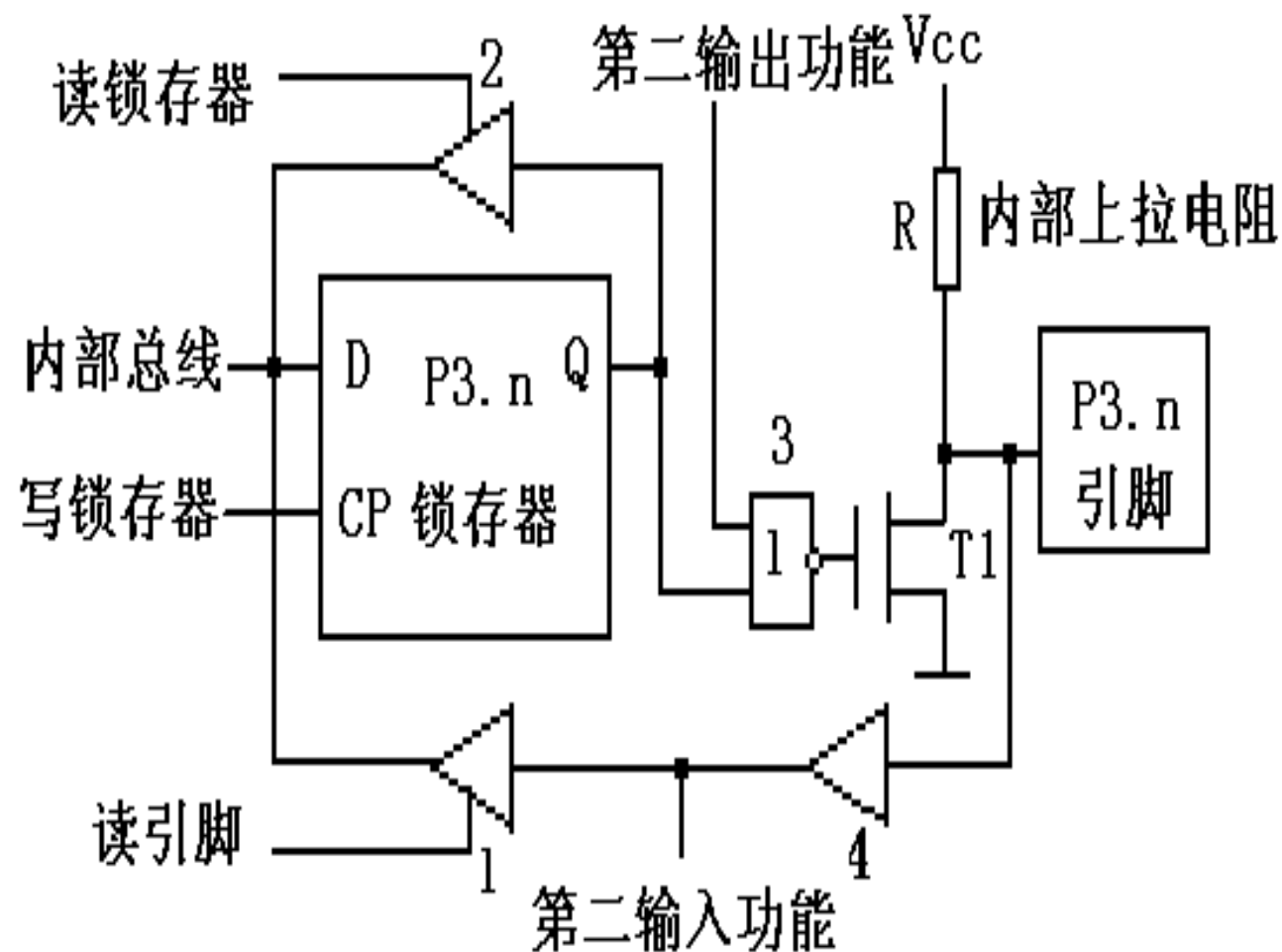
T0、T1 和RXD

内部结构:如图所示

特点:

➤ P3口的输出可驱动4个LSTTL负载。

## 1.4 P3口



# 1.5 MCS-51内部并行I/O口的应用

“读锁存器”和“读引脚”的指令

P0、P2、P3口复用作片外扩展系统的系统总线

直接输入/输出

# P0~P3编程和使用的方法

## 一、“读锁存器”和“读引脚”的指令

### 1. “读锁存器”的指令

P0~P3作目的操作数的指令，能实现“读—修改—写”口锁存器的操作，这类指令有：

#### (1) 字节操作指令

**ANL Px, —**

**ORL Px, —**

**XRL Px, —**

**DEC Px**

**DJNZ Px, rel**

**INC Px**

#### (2) 位操作指令

**JBC PX.Y,rel**

**CPL PX.Y**

**CLR PX.Y**

**SETB PX.Y**

**MOV PX.Y,C**

其中：X是口的序号0~3，Y是位的序号0~7



## 2. “读引脚”的指令

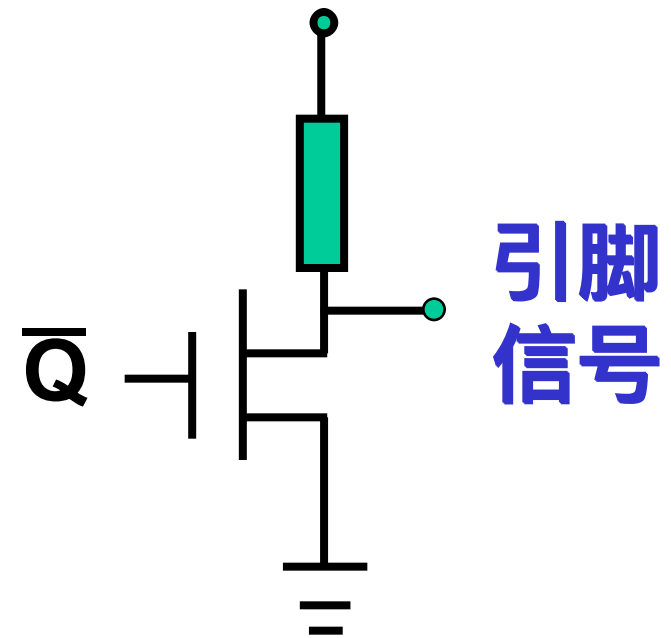
“读引脚”指令的特点是：P0~P3 作为源操作数出现在指令中，但在读引脚数据之前，必须对所读的口或口位的D锁存器写入“1”。

要正确读引脚数据，必须先写一条**MOV PX, #0FFH**指令或能对所读的口或口位的D锁存器写入“1”的指令，而后紧接着写下面的指令：

### (1) 字节操作指令

MOV	A, Px
MOV	direct, Px
XCH	A, PX
PUSH	PX
ADD	A, PX
ADDC	A, PX
SUBB	A, PX

ANL	A, PX
ORL	A, PX
XRL	A, PX
ANL	direct, PX
ORL	direct, PX
XRL	direct, PX



## (2) 位操作指令

同理，必须先写一条SETB PX.Y或能对所读口位的D锁存器写入“1”的指令，在其后再紧跟如下指令。

MOV C, PX.Y

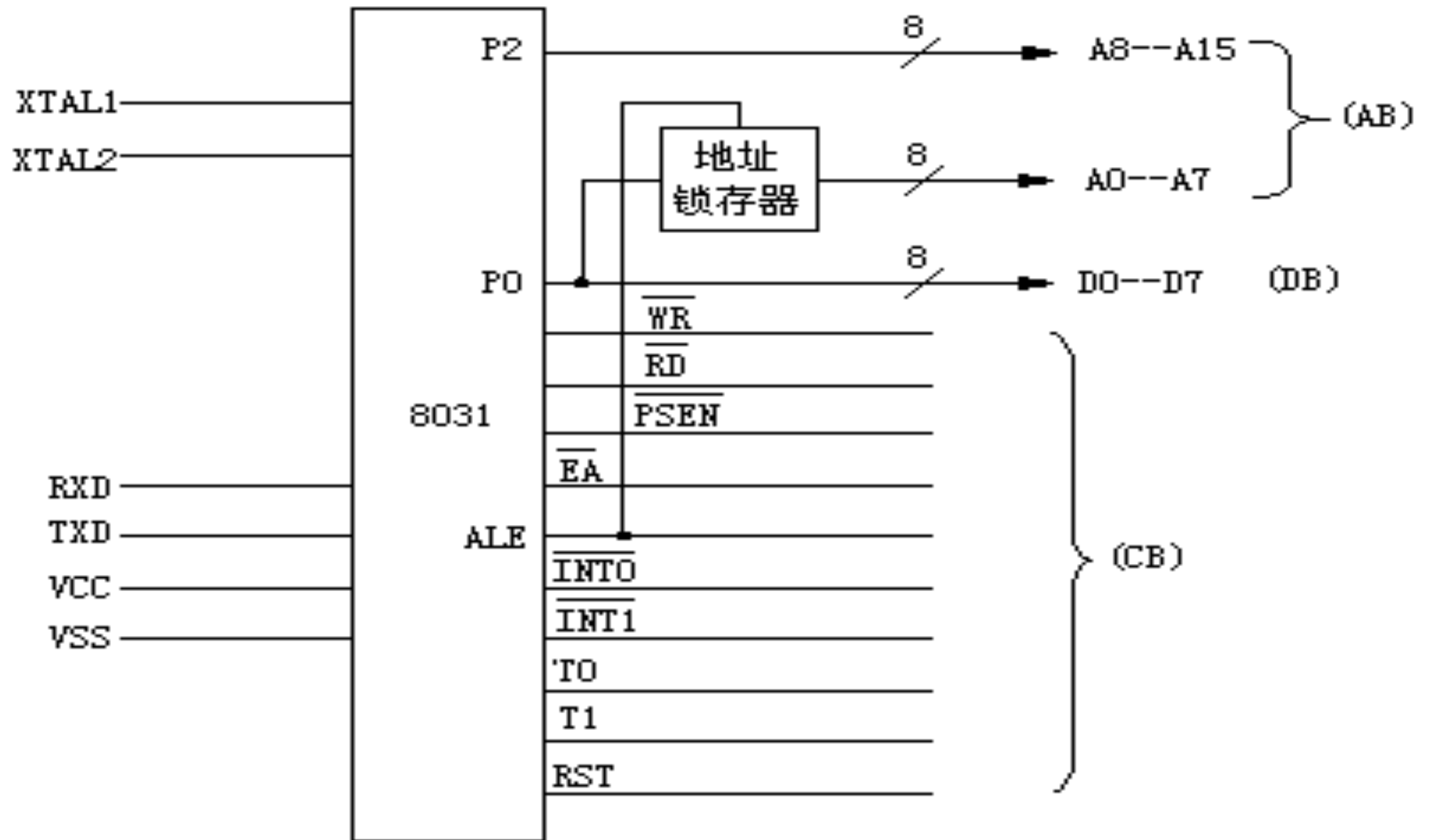
ANL C, PX.Y

ORL C, PX.Y

JNB PX.Y, rel

JB PX.Y, rel

## 二、P0、P2、P3口复用作片外扩展系统的系统总线



MCS-51单片机没有提供专门的外扩三总线的引脚，而是由P0、P2和P3口引脚构成三总线结构，外部芯片通过这三总线进行扩展。

# 8. 1 概 述

在组成单片机应用系统时:

**❓ 首先遇到的问题就是存储器的扩展。**单片机内部设置的存储器一般容量较小，满足不了实际需要，因此需要配置外部存储器，包括程序存储器和数据存储器。

**❓ 其次要解决的问题是I/O口的扩展。**

当外围设备较多时，仅有的几个内部I/O接口就不够用，在大多数应用系统中，MCS-51单片机都需要扩展输入/输出接口芯片，有时还需要扩展定时器、串行口、数模转换器(D/A)和模数转换器(A/D)以满足实际需要。

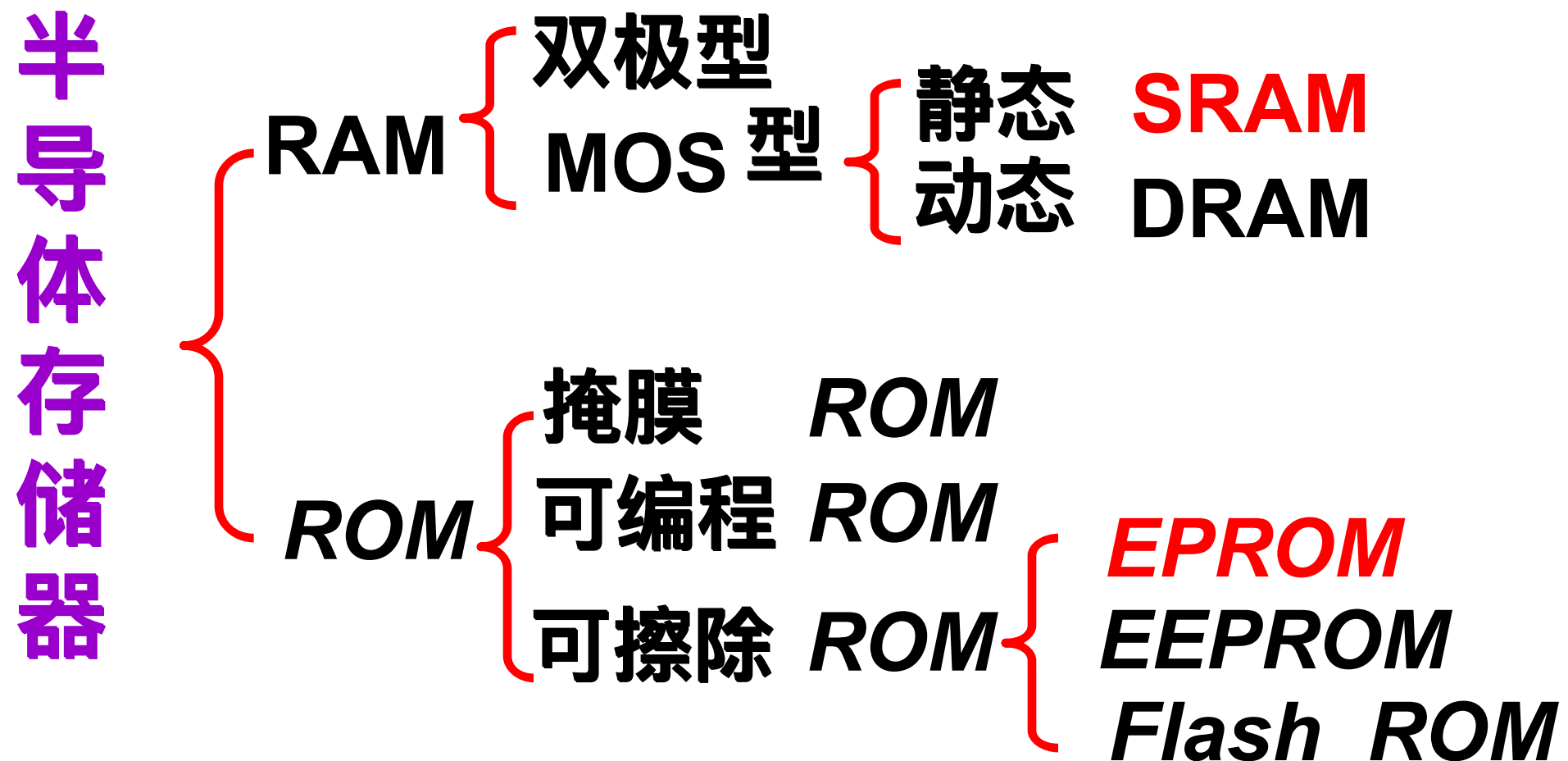
# 扩展 方法

---

- 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？

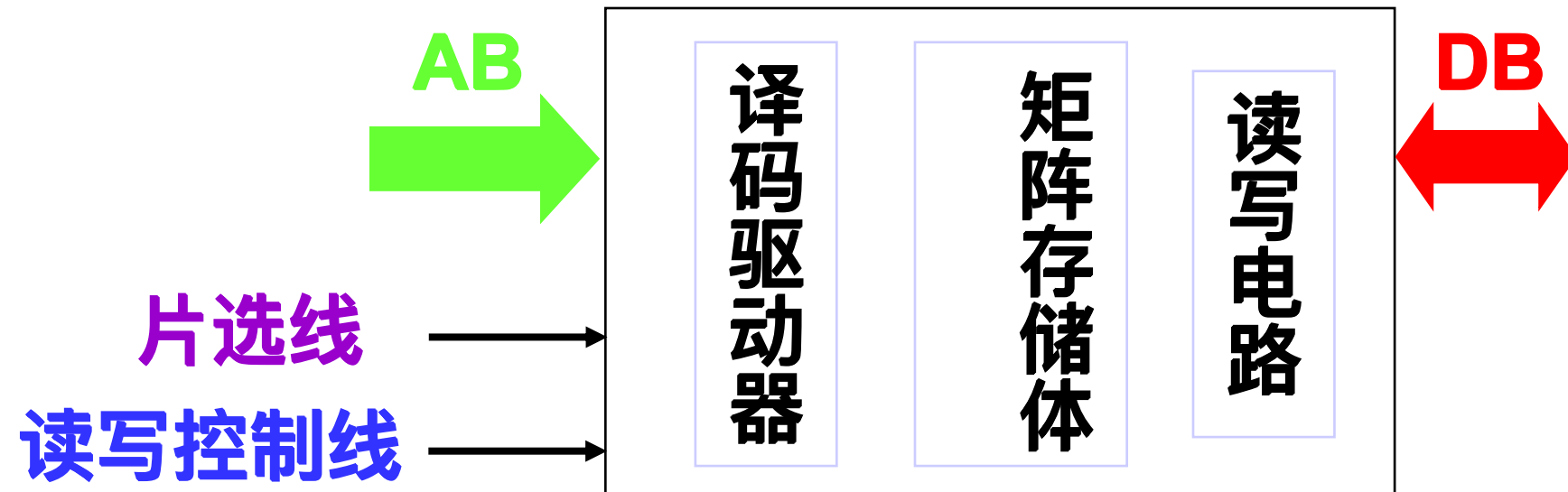
# 半导体存储器的分类

## 按读写方式分类

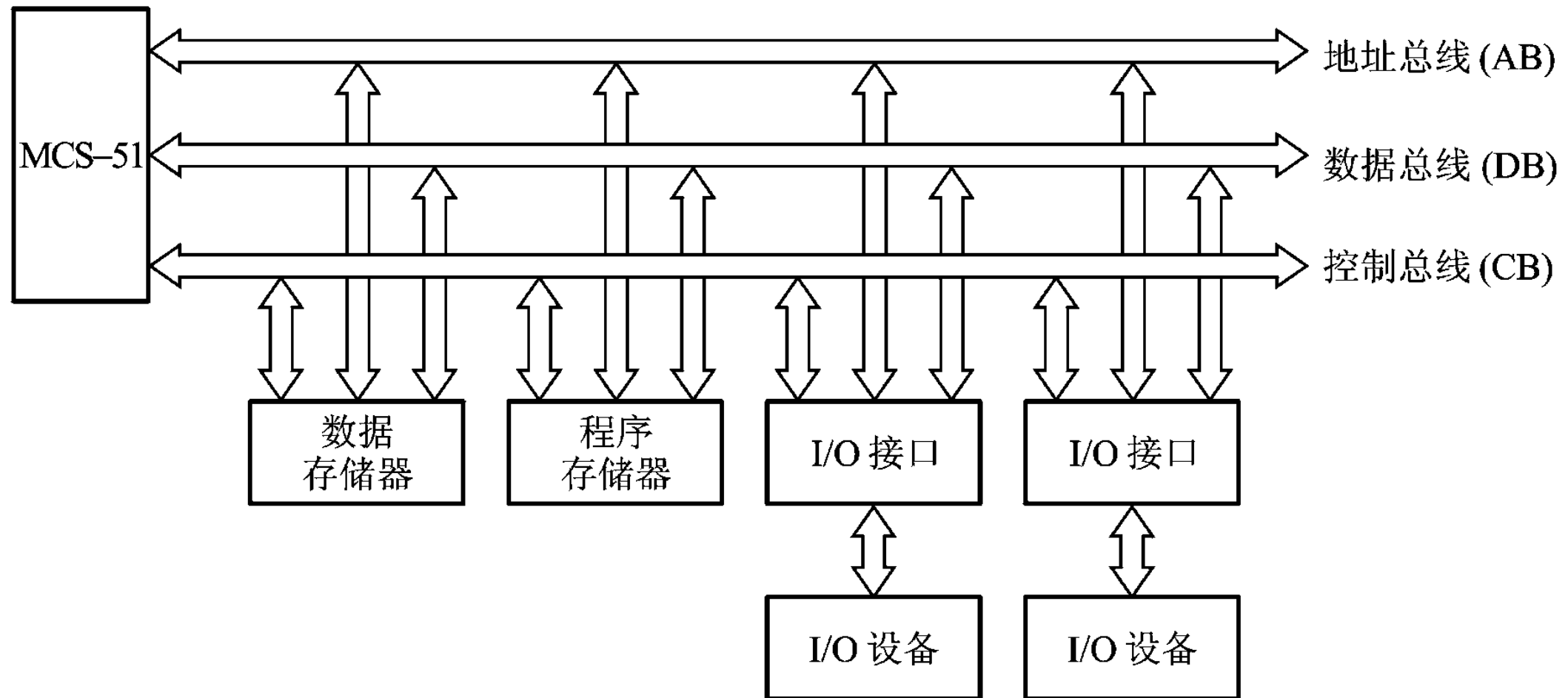


# 半导体存储器的基本结构

- E 矩阵存储体
- E 译码驱动器
- E 读写电路
- E 地址总线
- E 数据总线
- E 控制总线
- E 电源线



# 单片机对外扩展三总线



8-1

与计算机扩展连接芯片的外部引脚线都可以归属为三总线结构，扩展连接的一般方法实际上是三总线对接。要保证单片机和扩展芯片协调一致地工作，要共同满足其工作时序。





## 8.1.2 地址锁存器和译码器

---



**地址锁存器**

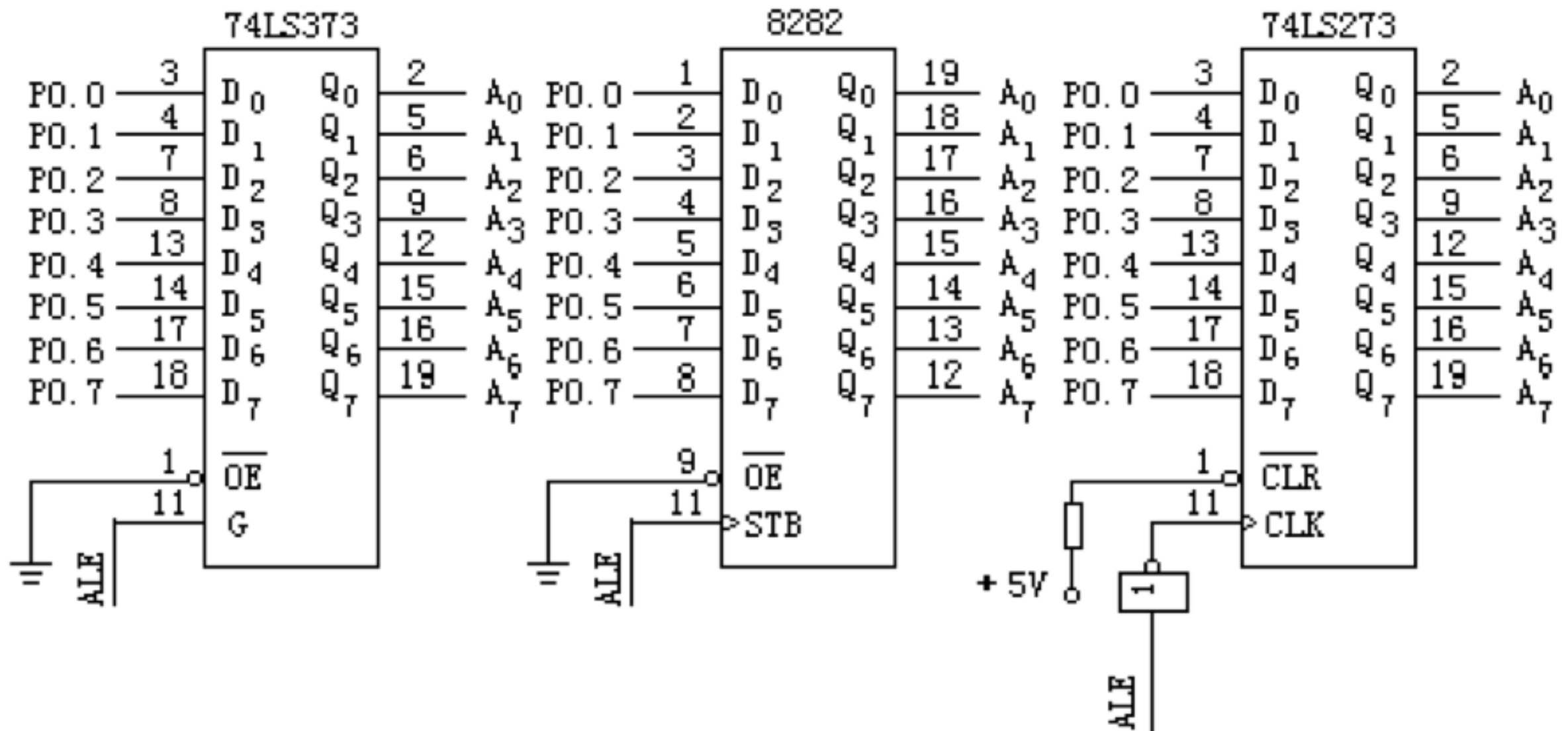


**译码器**



# 地址锁存器

MCS-51单片机的P0口是分时复用的地址/数据总线，因此在进行程序存储器扩展时，必须利用地址锁存器将地址信号锁存起来。



## 带三态缓冲输出的8D锁存器74LS373或8282

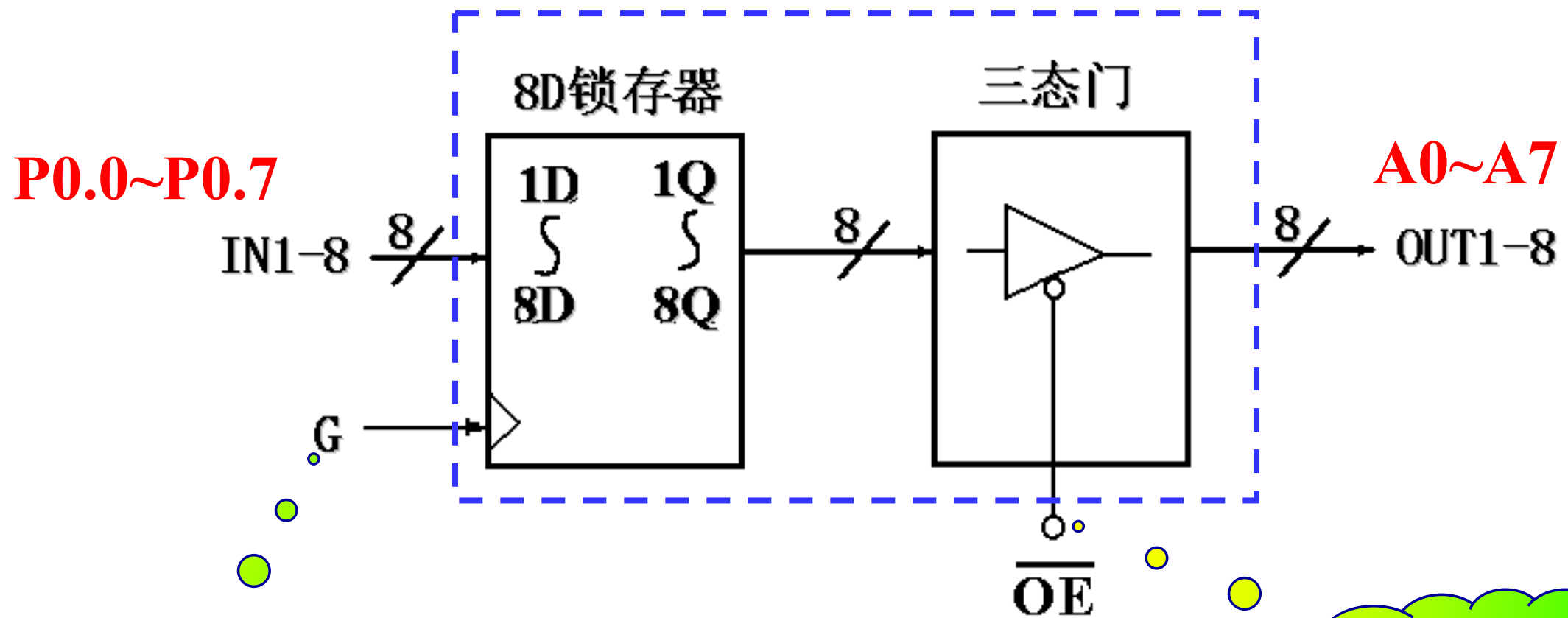


图7—4 74LS373和8282结构图

地址锁存器---74LS373 和8282 (8位三态同相锁存器)

引脚功能: 1D~8D: 8位并行数据输入端

1Q~8Q: 8位并行数据输出端

**G:** 数据输入信号为1时, Q 端数据 = D端数据, 为0时Q端数据保持。

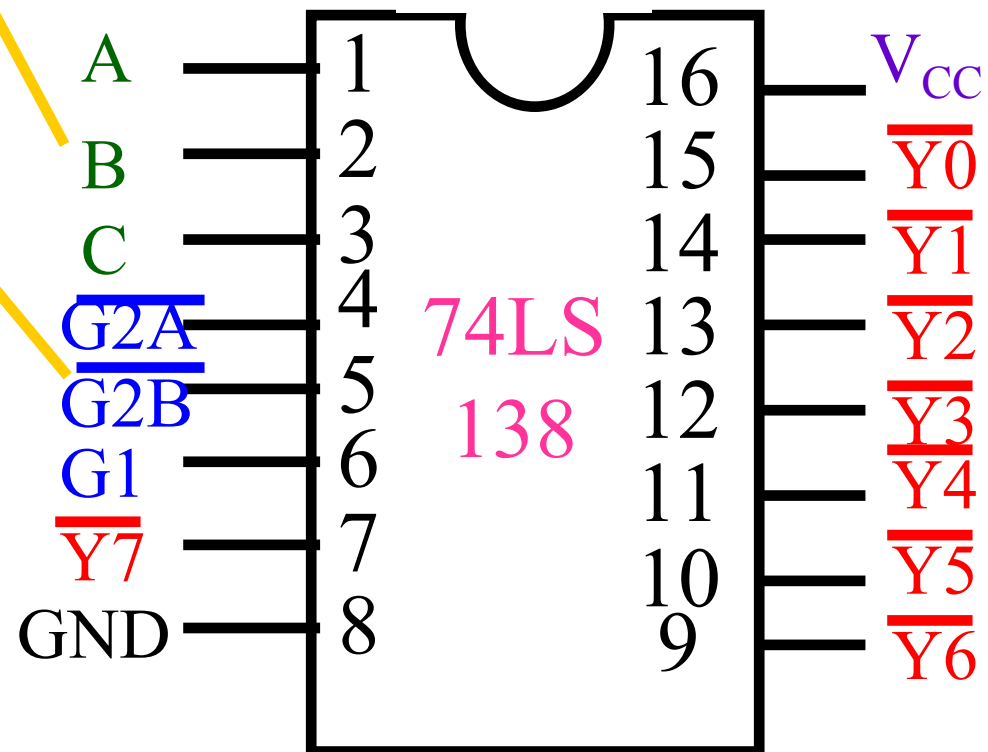
$\overline{OE}$  输出使能端(片选端), 低电平有效



# 3—8译码器 74LS138

3个选择输入端

3个允许输入端



8个输出端

# 74LS138真值表

$G_1$ $\overline{G_{2A}}$ $\overline{G_{2B}}$	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 1 0 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



# 存储器扩展的基本方法

## 1、存储器扩展的基本问题。

**1) 扩展容量：**16根地址线最大可扩展到64K

**2) 扩展要解决的问题：**地址线、扩展芯片在64K范围所占的地址范围

**3) 存储器扩展的编址：**存储芯片的选择、片内单元的编址

**4) 选择芯片的方法：**片选技术

## 2、存储器扩展的片选技术

一般产生片选有两种方法：**线选法和译码法。**

## (1) 线选法

线选法用**低位地址线对片内的存储单元进行寻址**，所需的地址线由片内地址线决定，用余下的**高位地址线分别接至芯片的片选端**，以区分各芯片的地址范围。例如要扩展8K容量的外RAM，地址线和片选如下：

**地址线：** $\log_2(8K) = \log_2(2^{13}) = 13$ 条( $A_{12} \sim A_0$ )

**片选线：**余下的 $A_{15} \sim A_{13}$ 分别接至芯片的片选端。 $A_{15} \sim A_{13}$ 轮流出现低电平，可保证一次只选一片。

用线选法扩展存储器的缺点

- ① 各芯片间地址不连续，而习惯上使用连续地址。
- ② 有相当数量的地址不能使用,否则造成片选混乱。

## (2) 译码法

译码法将低位地址总线直接连至各芯片的地址线,将高位地址总线经地址译码器译码后作为各芯片的片选信号。

一般使用2/4译码器、3/8译码器,对P2口高位地址线进行译码,适用于大规模扩展。



## 8.2 外部程序存储器的扩展

硬件电路如图所示。

单片机访问外部程序存储器所使用的控制信号有：

**ALE**：低8位地址锁存控制信号

**PSEN**：外部程序存储器“读取”控制信号

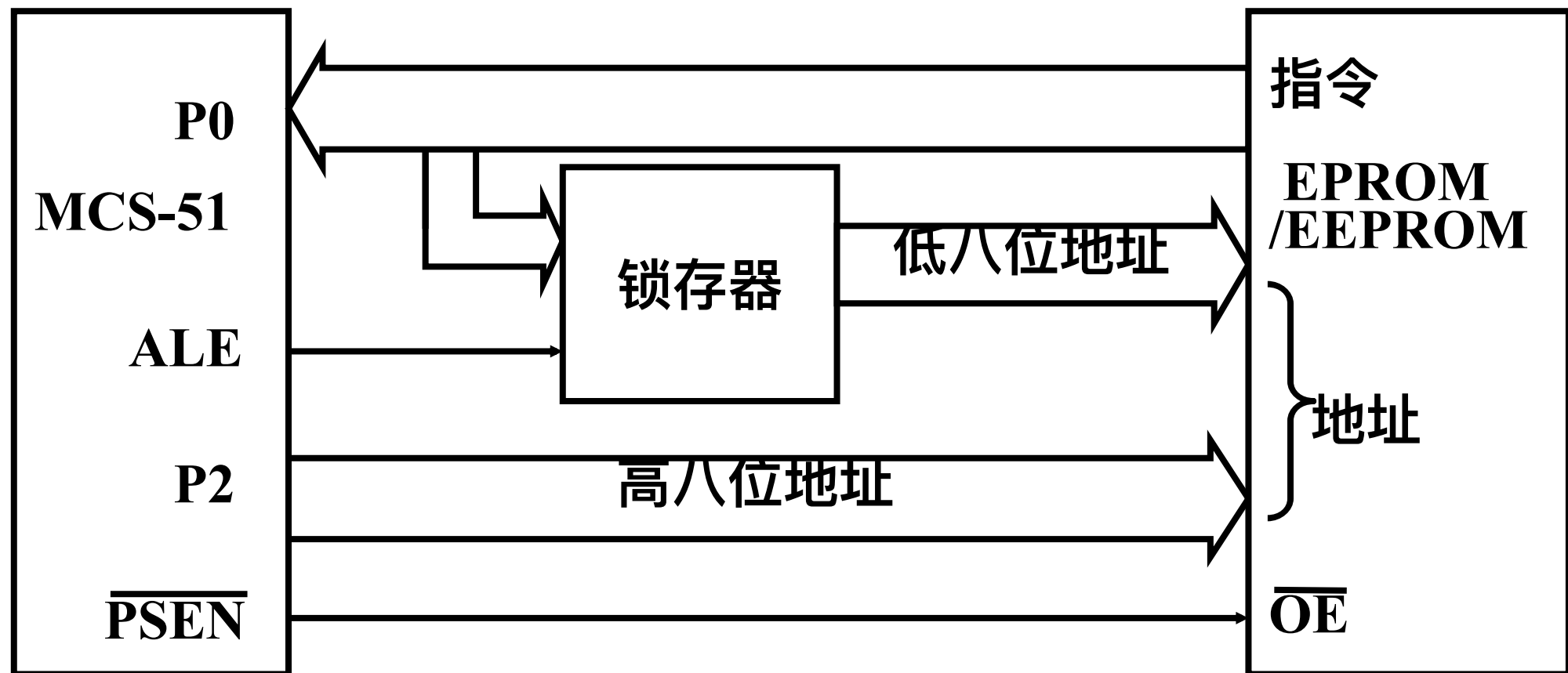
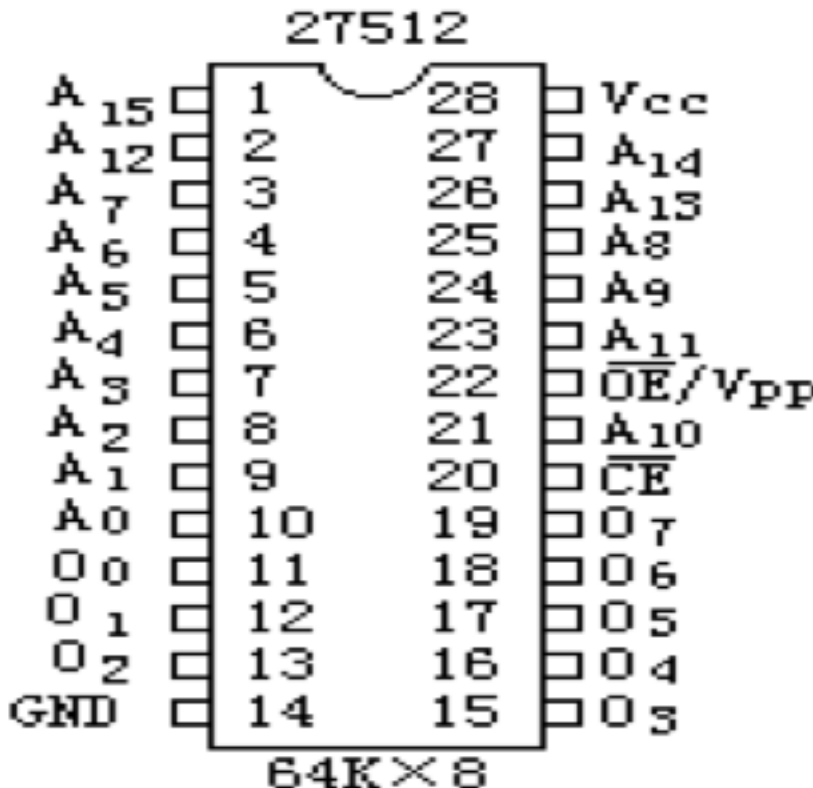
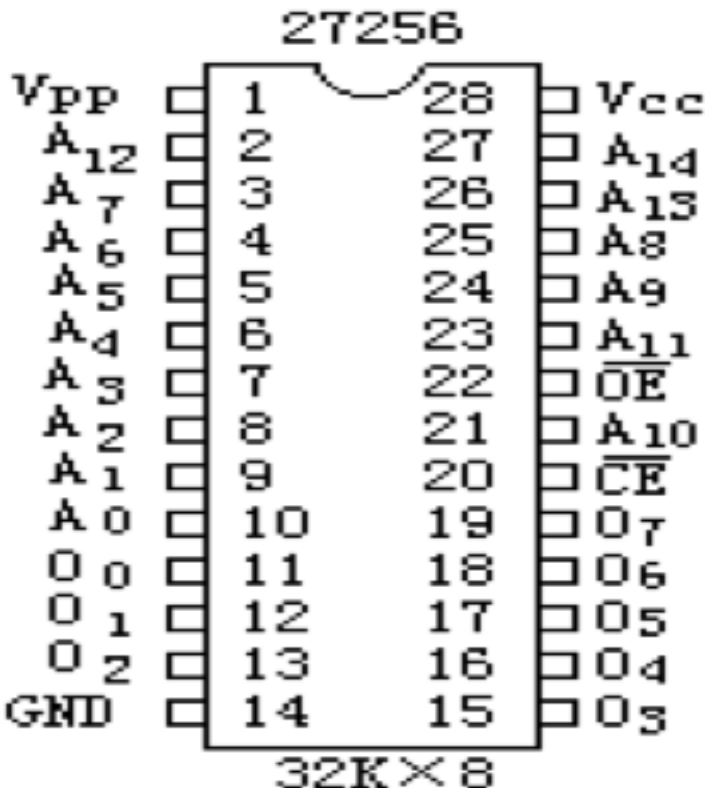
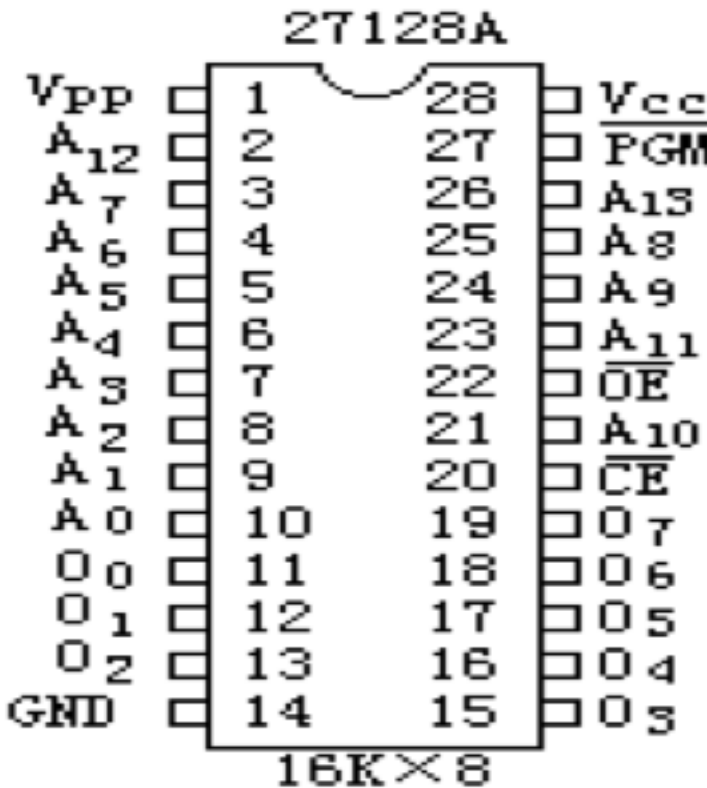
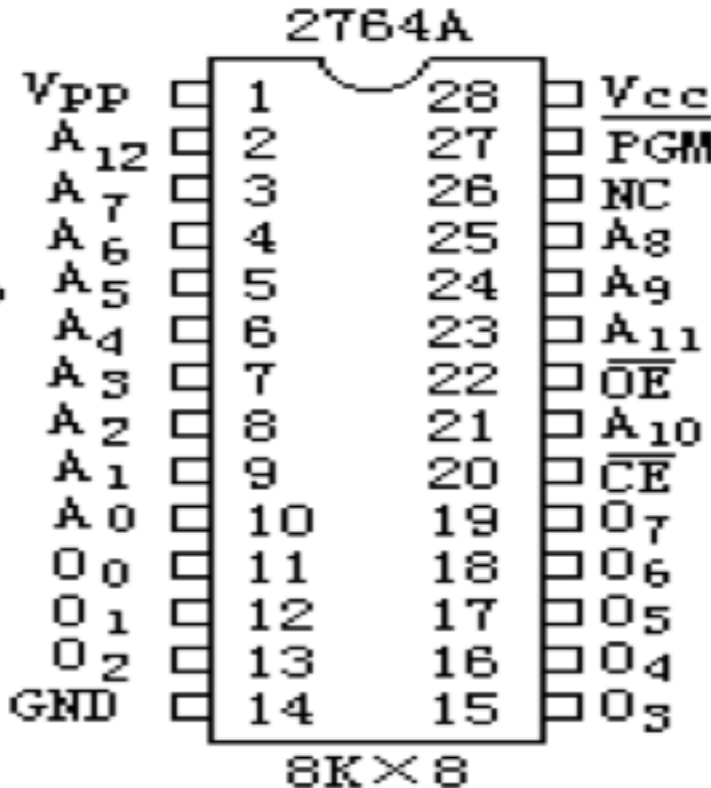
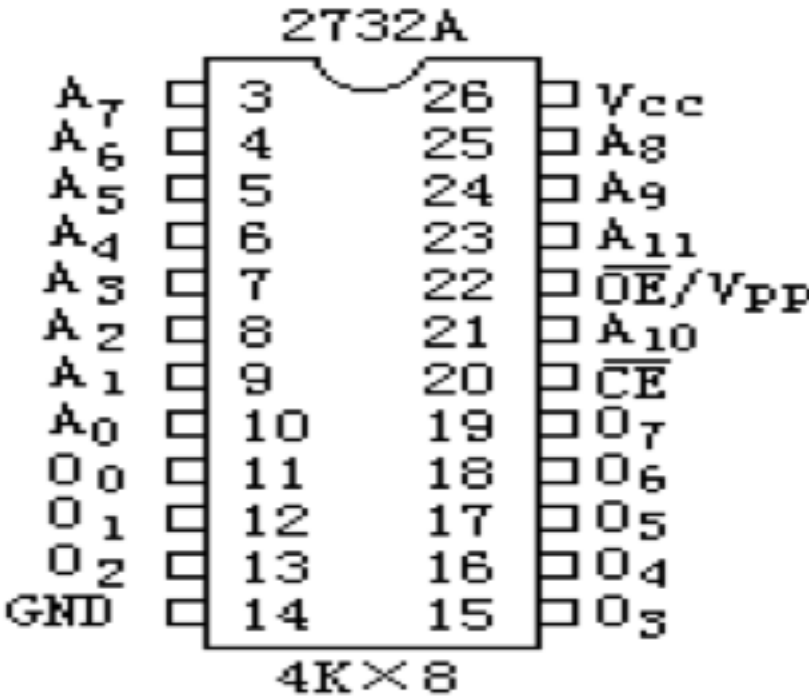
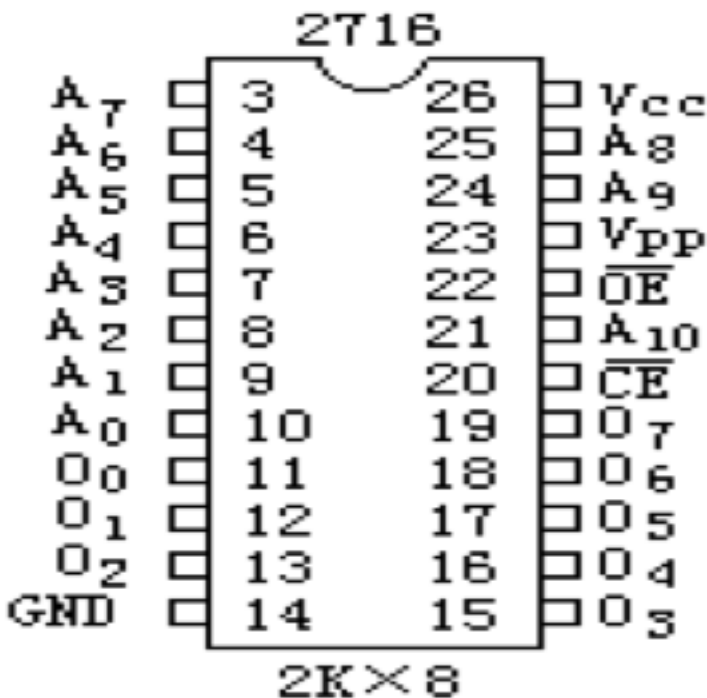


图8—1 MCS-51单片机程序存储器的扩展

# 1. 常用EPROM芯片介绍 (名称/8)



## 1、EPROM电路

常用的EPROM芯片为：2764、27128、27256、27512等

编程电源线

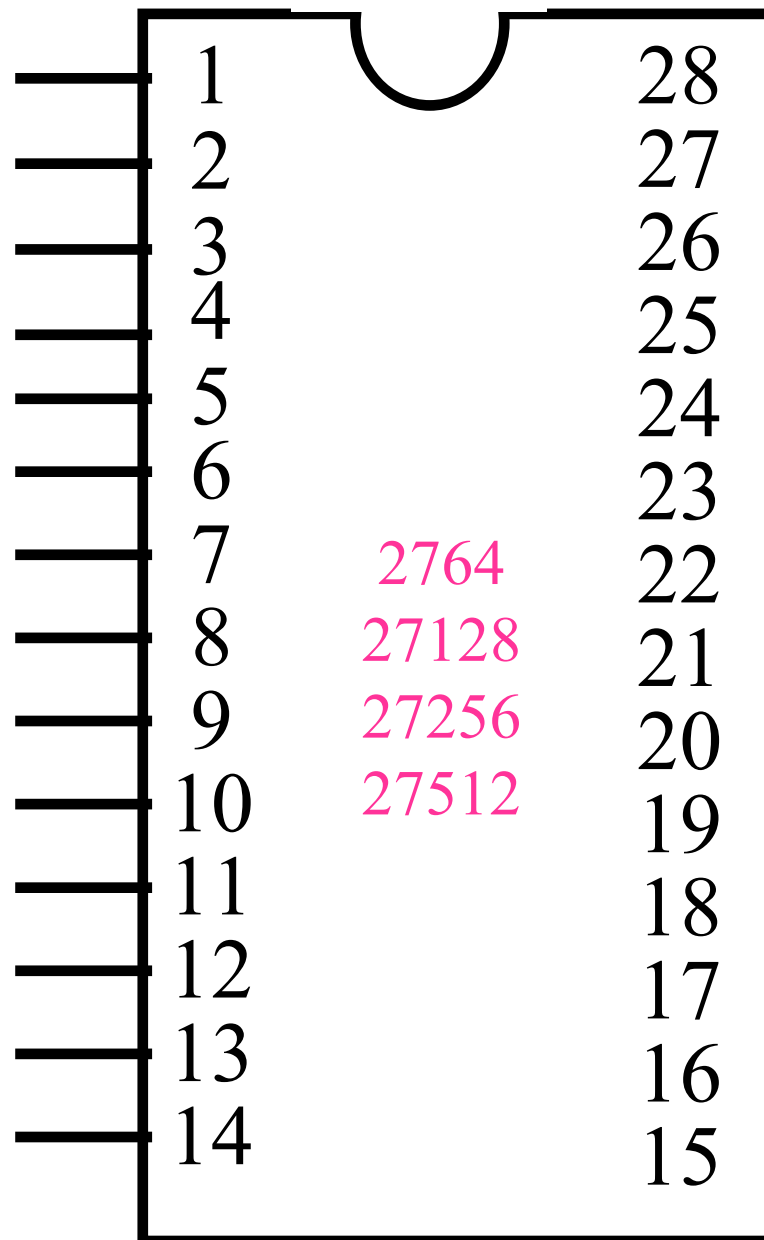
$V_{PP}$

地址输入线

$A_0 \sim A_i$

地线

GND



2764  
27128  
27256  
27512

工作电源线

$V_{CC}$

编程脉冲  
输入线

PGM

NC

$A_8$

$A_9$

$A_{11}$

$\overline{OE}$

$A_{10}$

$\overline{CE}$

$Q_7$

$Q_6$

$Q_5$

$Q_4$

$Q_3$

读出选通线

片选线

三态数据线

$D_0 \sim D_7$

## 2、扩展电路实例

---

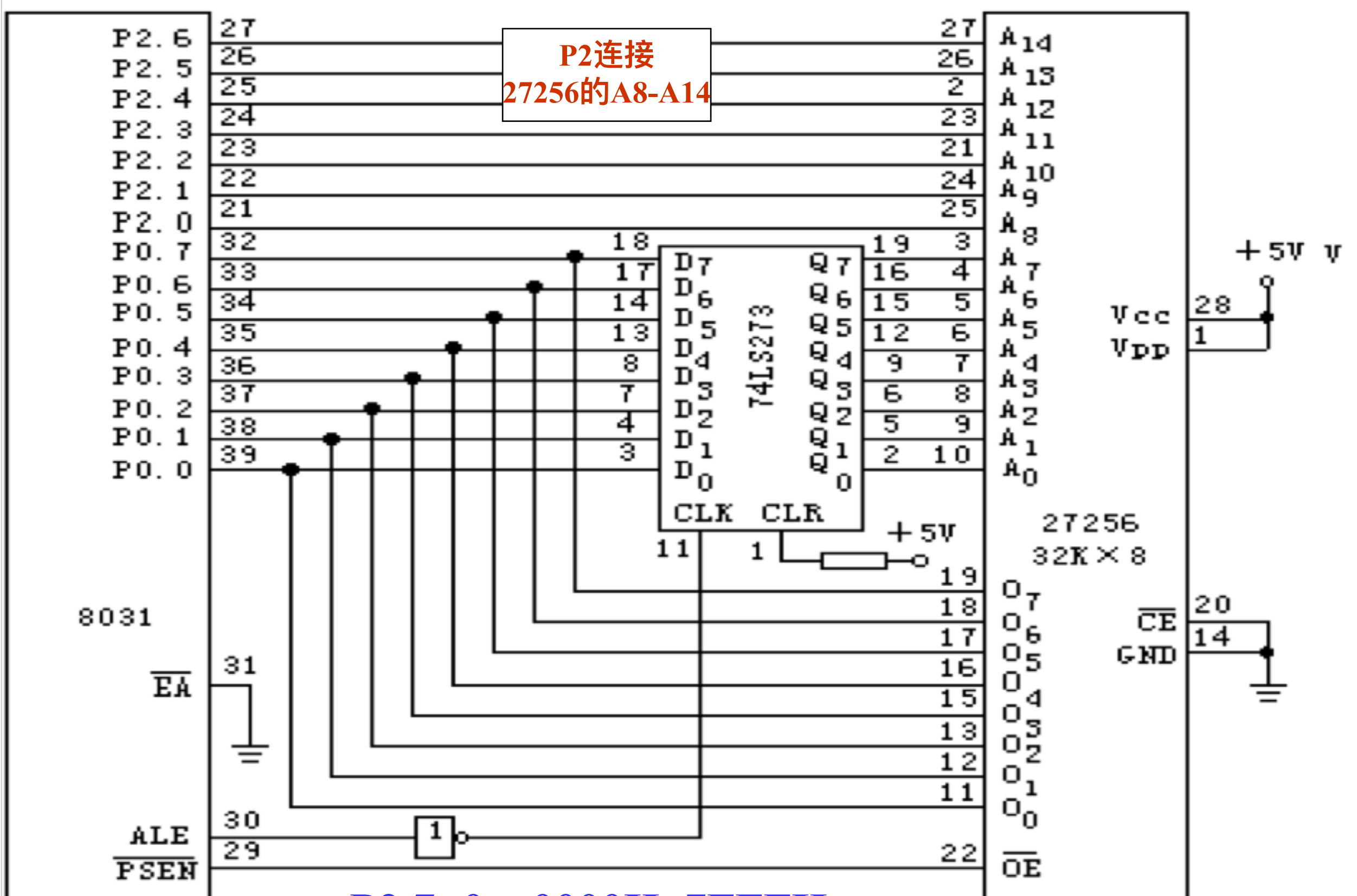
扩展一片EPROM

27256

扩展两片EPROM

2764





**P2.7=0    0000H~7FFFH**

**P2.7=1    8000H~FFFFH**

**$\overline{PSEN}$ 接 $\overline{OE}$**   
 **$\overline{CE}$ 接地**



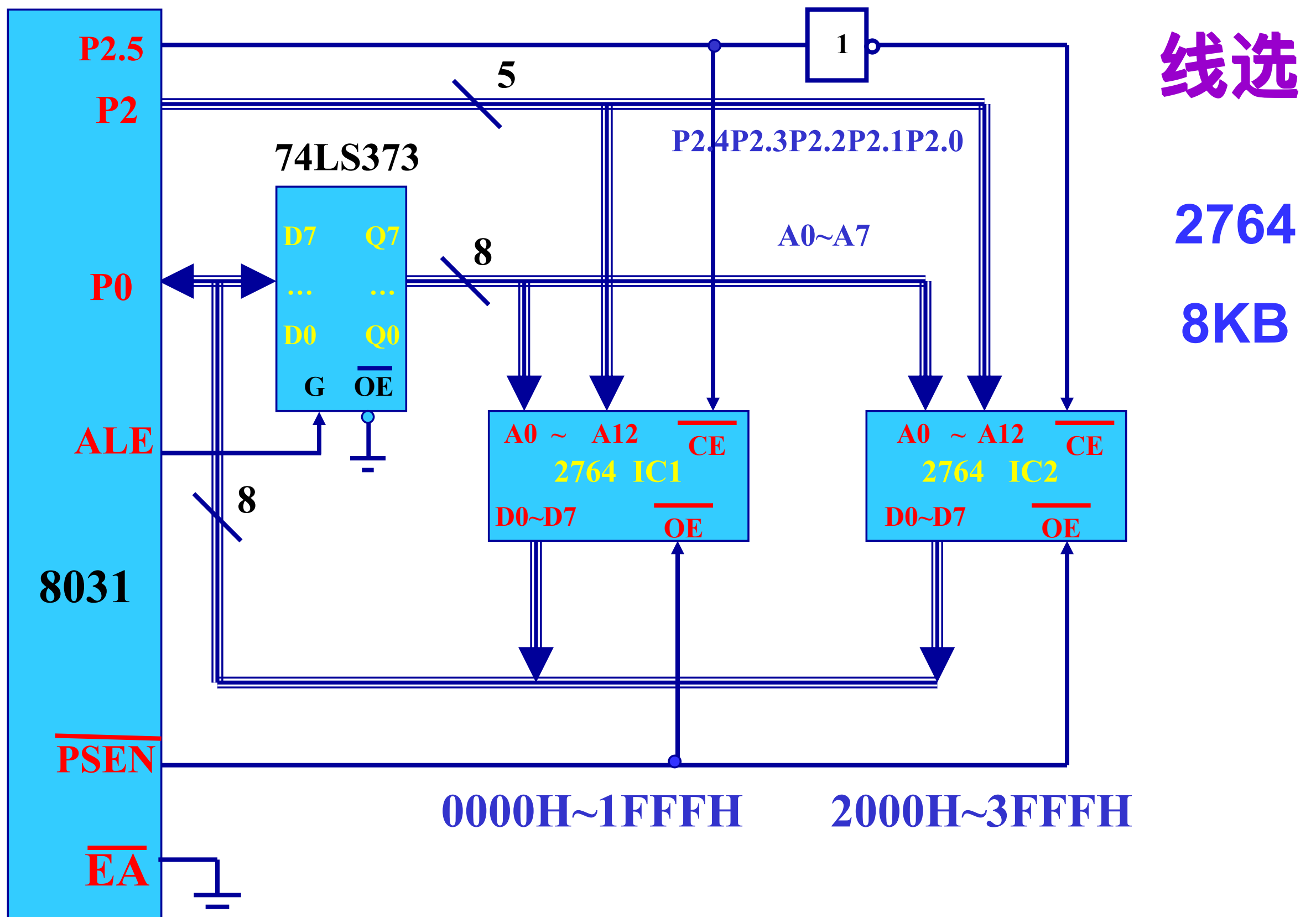
# EPROM 2764——程序存储器

## 2764 工作方式真值表

/CE	/OE	PGM	VPP	D0~D7	方式
0	0	1	+5V	输出	读
1	×	×	+5V	高阻	维持
0	1	0	VPP	输入	编程
0	0	1	VPP	输出	编程校验
1	×	×	VPP	高阻	编程禁止

注：VPP为编程脉冲，可以为 +5V,+12.5v,+21V,+25V等

PGM是编程脉冲输入线



# 地址空间——地址段不惟一

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
X	X	0	0	0	0	0	0	0~0	
000B 010B 100B 110B			0000H 、 4000H、 8000H 、 C000H						
			1	1	1	1	1	1~1	
			1FFFH、 5FFFH、 9FFFH、 DFFFH						
X	X	1	0	0	0	0	0	0~0	IC2
001B 011B 101B 111B			2000H、 6000H、 A000H、 E000H						
			1	1	1	1	1	1~1	
			3FFFH、 7FFFH、 BFFFH、 FFFFH						



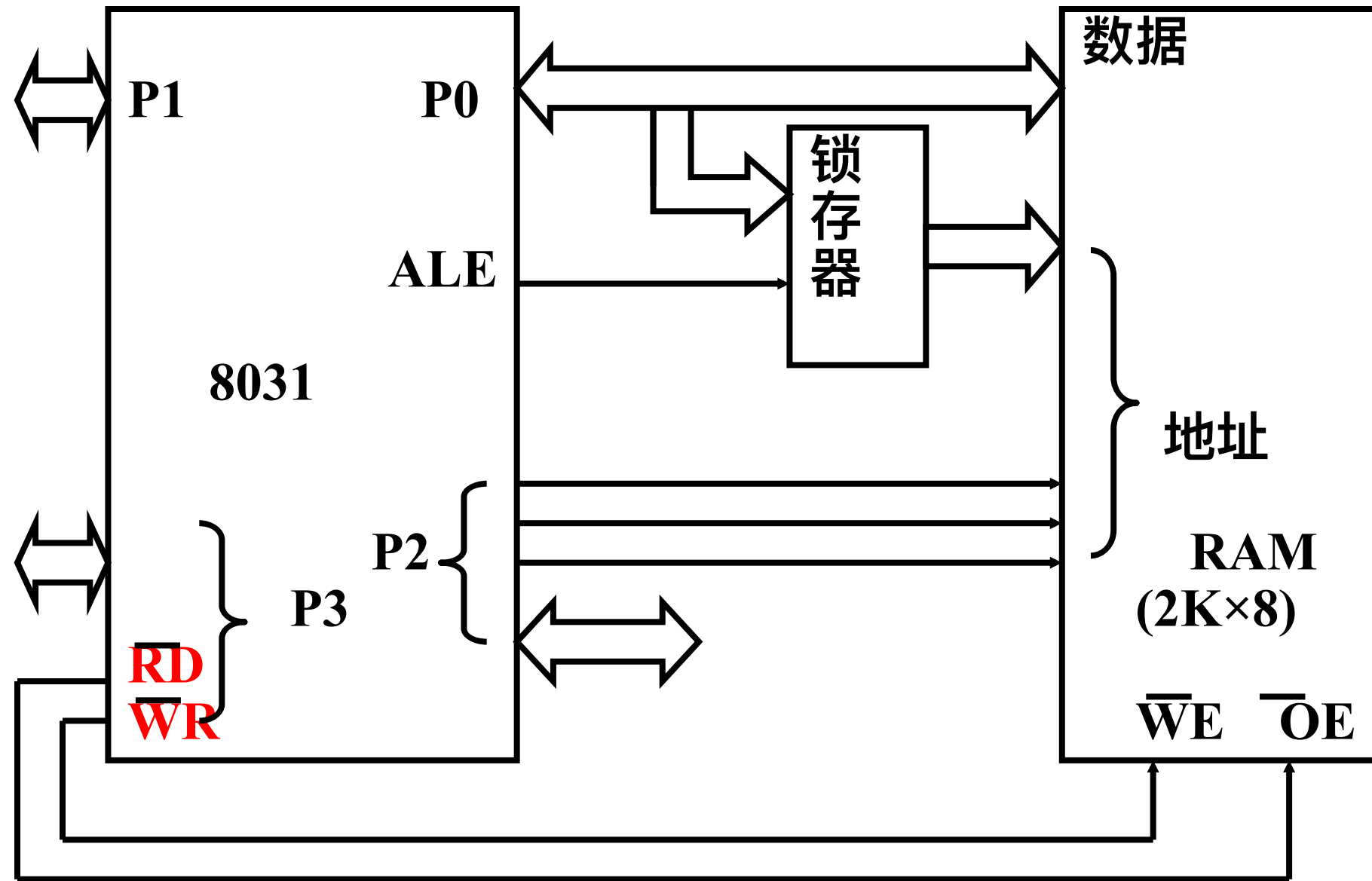


## 8.3 外部数据存储器的扩展

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。



# 静态RAM扩展

静态RAM在应用时存取速度快、使用方便并且价格比较的低廉。但它在掉电时，内部的数据会丢失。典型6116(2K)、6264(8K)、62128(16K)、62256(32K)等芯片。为了避免掉电数据丢失，自动保护的静态RAM，如：DS1225、DS1235。

各重要的引脚含义如下：

**A0-Ai**：地址输入线，决定存储器的容量；

**I/O0-I/O7**：双向三态数据线；

**$\overline{\text{CE}}$** ：片选信号输入线；

**$\overline{\text{OE}}$** ：读选通输入信号线；

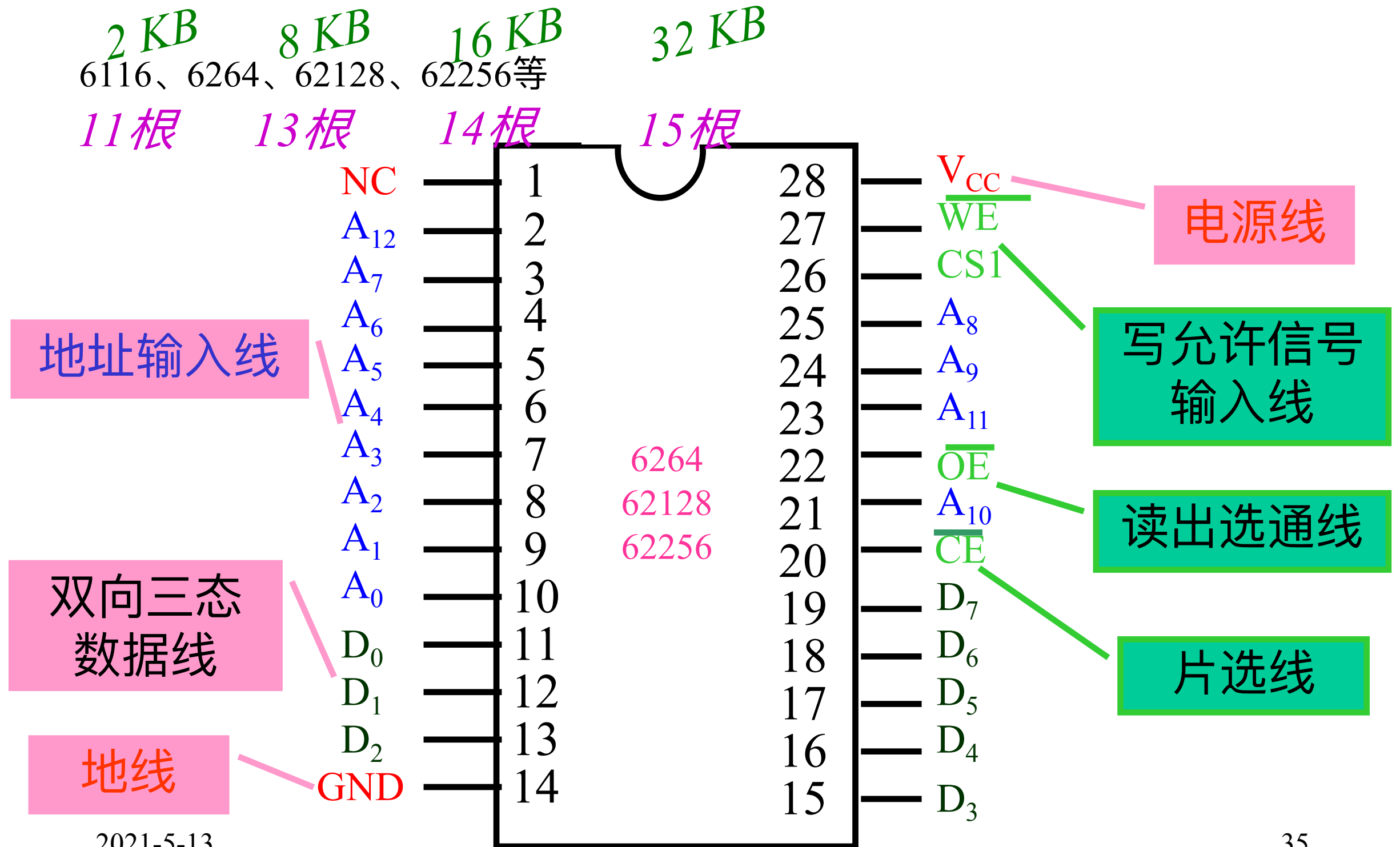
**$\overline{\text{WE}}$** ：写选通输入信号；

**V<sub>PP</sub>**：编程电源输入线；

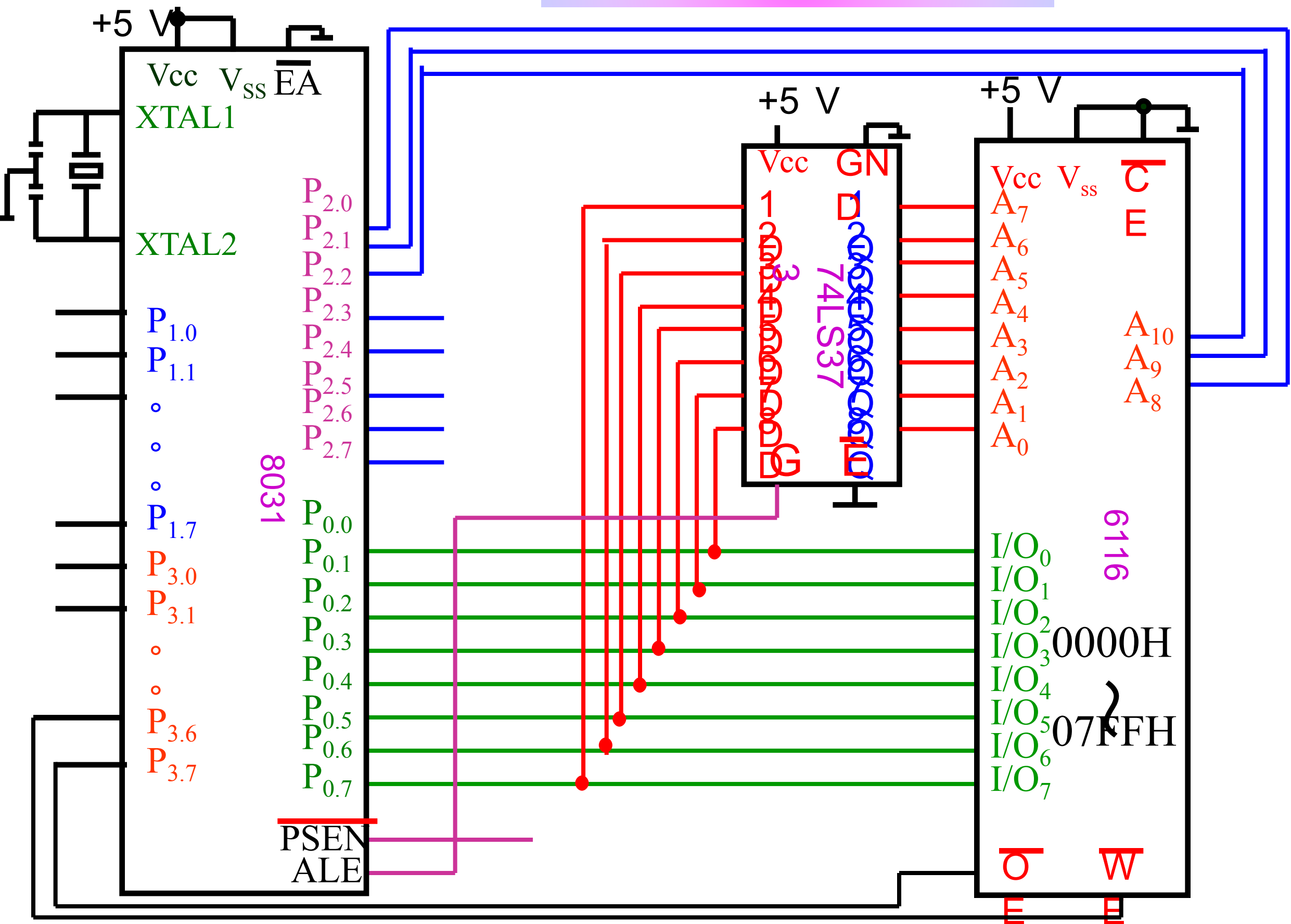
**V<sub>CC</sub>**：工作电源输入线（常为+5V）；

**GND**：工作时接地线。

# 静态RAM芯片



# 8031扩展2KBRAM



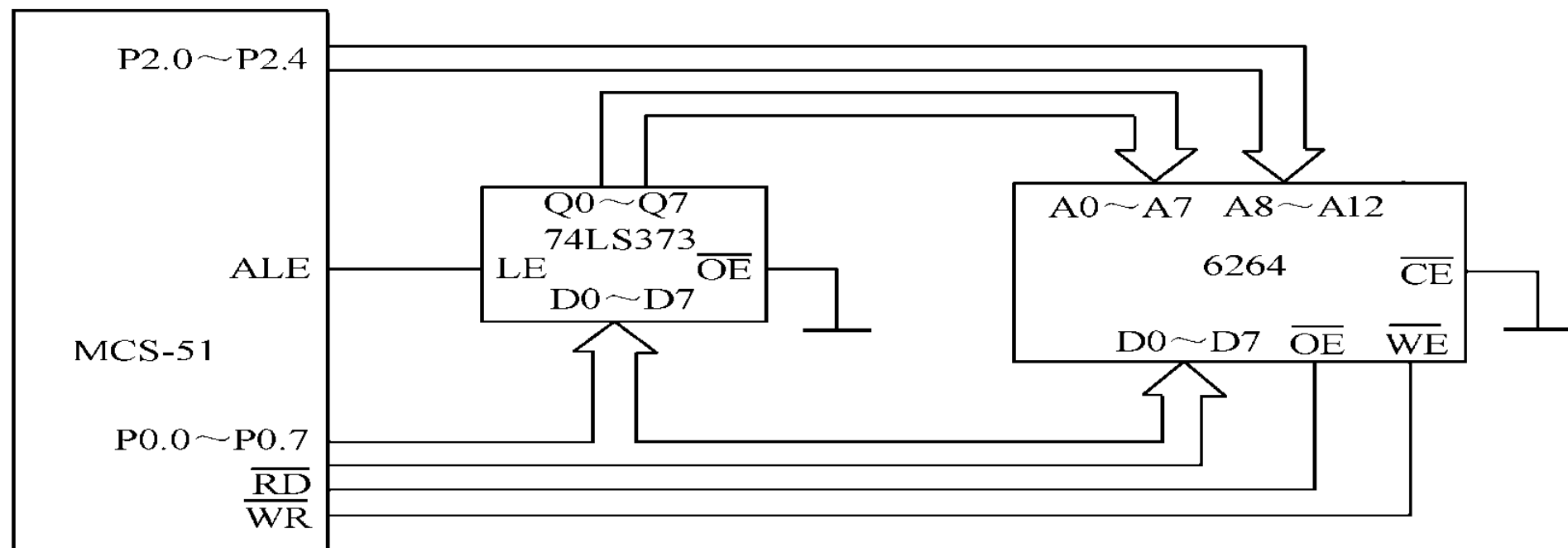
# 数据存储器典型扩展电路

## MCS-51扩展6264的电路连接方法：

**数据线：** P0口接RAM的D<sub>0</sub>~D<sub>7</sub>；

**地址线：** 6264容量为8KB， $2^{13}=8\text{KB}$ ，需要A<sub>0</sub>~A<sub>12</sub>共13根地址线。P0口经地址锁存器后接RAM的A<sub>0</sub>~A<sub>7</sub>； P2.0~P2.4接RAM的A<sub>8</sub>~A<sub>12</sub>。

**控制线：** ALE接373的LE， $\overline{\text{RD}}$ 接RAM的 $\overline{\text{OE}}$ 、 $\overline{\text{WR}}$ 接RAM的 $\overline{\text{WE}}$ ，只有一片RAM，且系统无其他I/O接口及外围设备扩展，片选 $\overline{\text{CE}}$ 可以接地。扩展电路如图所示。



6264的地址范围为：0000H~1FFFH。

注意 74LS373 的 $\overline{\text{OE}}$ 和RAM6264的 $\overline{\text{CE}}$ 。

**[例]** 在上页图的数据存储器扩展电路中，将片内RAM 以50H单元开始的16个数据，传送到片外数据存储器0000H开始的单元中。程序如下：

```
ORG 1000H
DMV: MOV R0, #50H          ; 数据指针指向片内50H单元
      MOV R7, #16          ; 待传送数据个数送计数寄存器
      MOV DPTR, #0000H    ; 数据指针指向数据存储器6264的0000H单元
AGN:  MOV A, @R0           ; 片内待输出的数据送累加器A
      MOVX @DPTR, A       ; 数据输出至数据存储器6264
      INC R0
      INC DPTR            ; 修改数据指针
      DJNZ R7, AGN        ; 判断数据是否传送完成
      SJMP $
      END
```

# 扩展两片SRAM

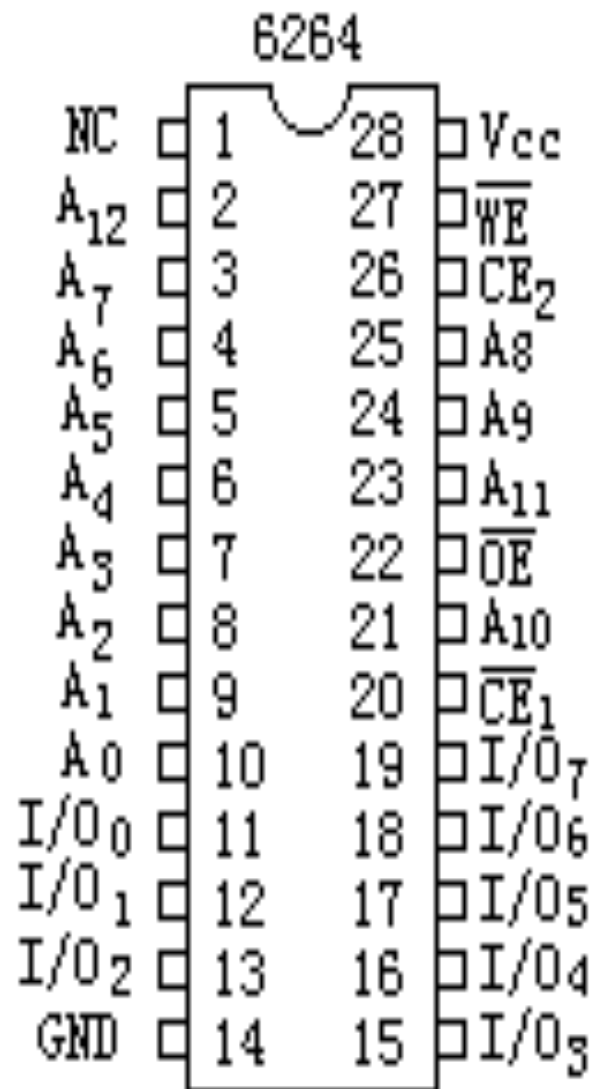
6264

线选法

译码法



# 6264引脚图和工作方式

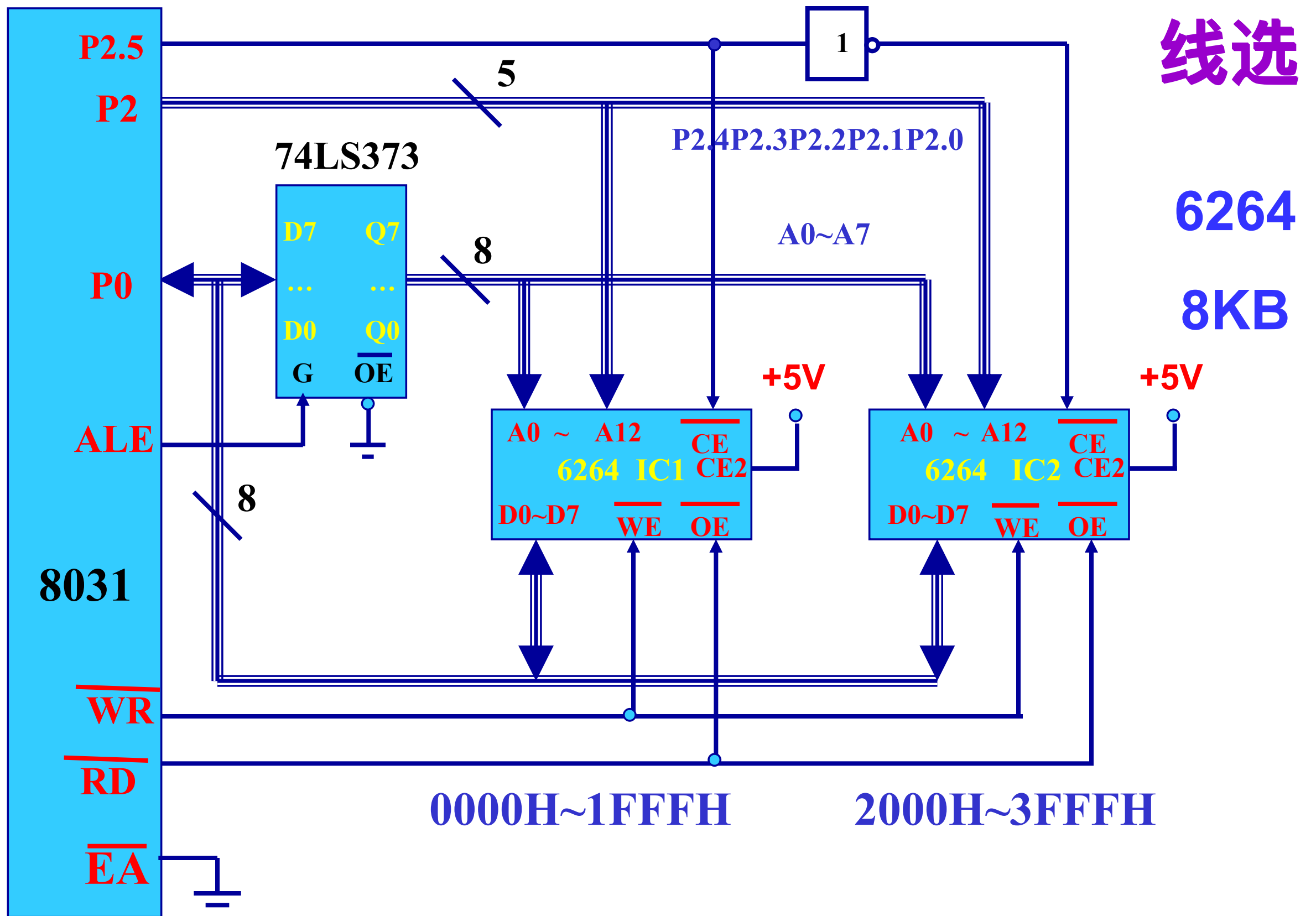


$\overline{WE}$	$\overline{CE}_1$	$CE_2$	$\overline{OE}$	方式	$D_0 \sim D_7$
X	H	X	X	未选中(断电)	高阻
X	X	L	X	未选中(断电)	高阻
H	L	H	H	输出禁止	高阻
H	L	H	L	读	$D_{OUT}$
L	L	H	H	写	$D_{IN}$

6264有2个片选端只须用其一个，一般用 $CE_1$ ， $CE_2$ 直接接 $V_{CC}$ 。







# 地址空间——地址段不惟一

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
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									

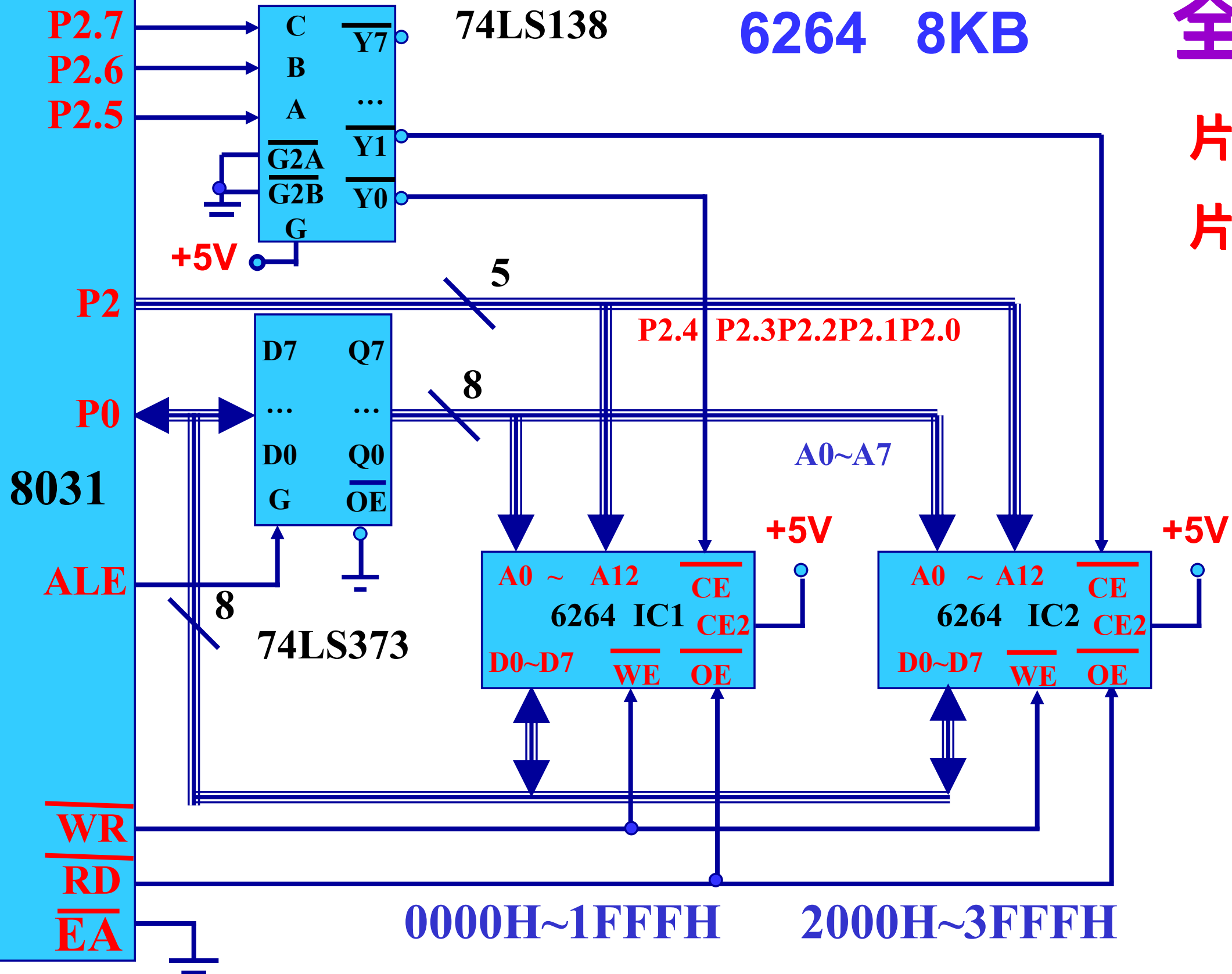


# 全译码

6264 8KB

片内13根

片外3根



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							
000B $\overline{Y0}—\overline{CE}$			0000H						
			1	1	1	1	1	1~1	
			1FFFH						
0	0	1	0	0	0	0	0	0~0	IC2
001B $\overline{Y1}—\overline{CE}$			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

---

## ❓ 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

## ❓ 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，地址惟一



# 补1：同时扩展程序存储器和数据存储器的方法

## SRAM 6116/6264——数据存储器

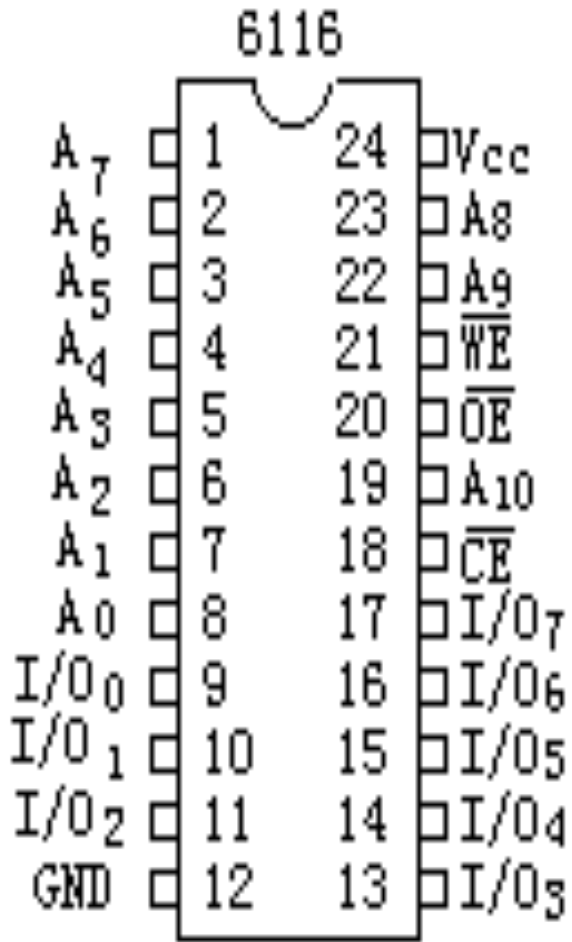


表 7—3 6116 的操作方式

$\overline{CE}$	$\overline{OE}$	$\overline{WE}$	方式	$D_0 \sim D_7$
H	X	X	未选中	高阻
L	L	H	读	$D_{OUT}$
L	H	L	写	$D_{IN}$

# EPROM 2732——程序存储器

## 2732 工作方式真值表

/CE	/OE    -VPP	D0~D7	方式
0	0	输出	读
1	×	高阻	维持
0	VPP	输入	编程
0	0	输出	编程校验
1	VPP	高阻	编程禁止

注：VPP为编程脉冲，可以为 +5V,+12.5v,+21V,+25V等

焦点：片选信号的产生

2732

6116

EPROM 与SRAM扩展

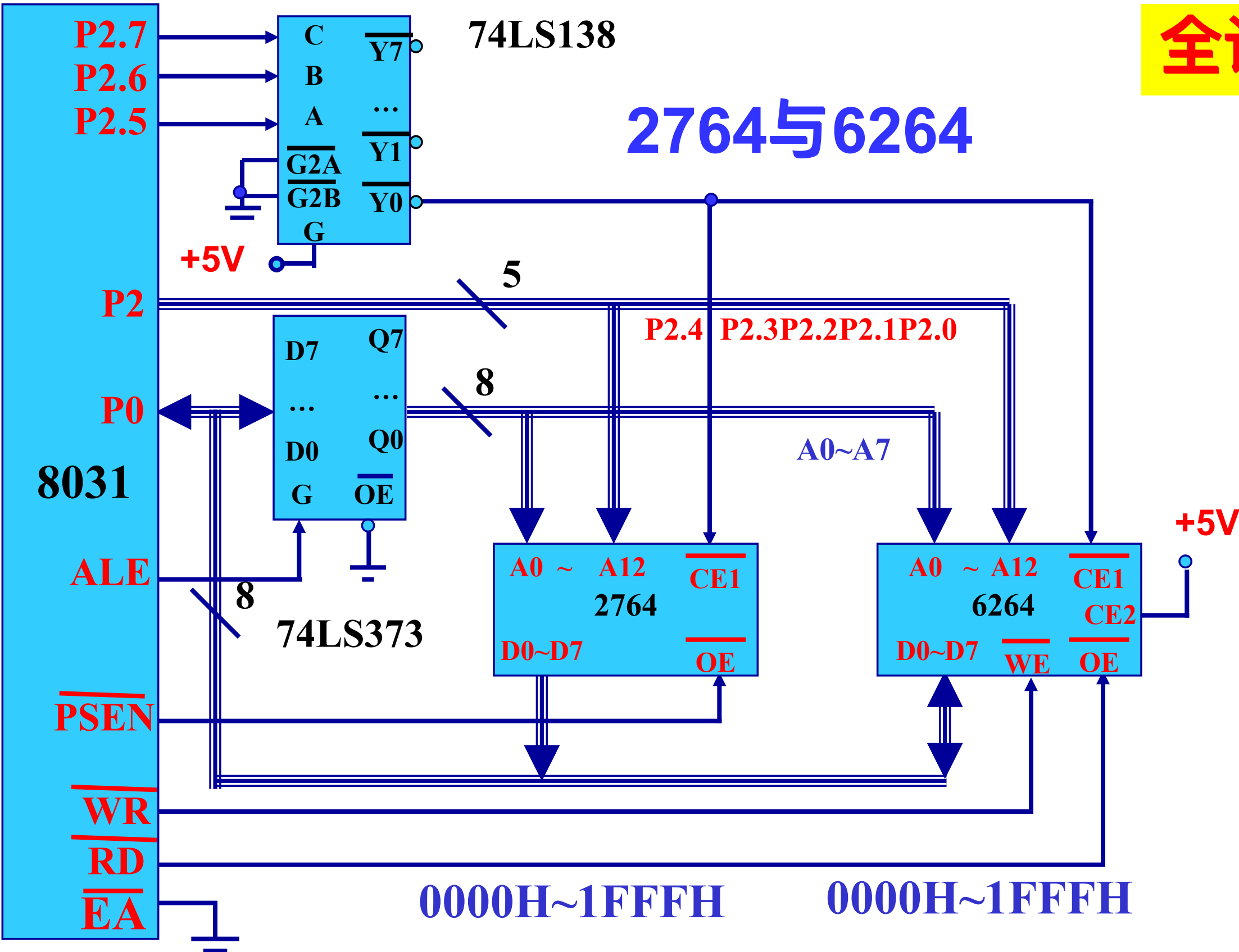
6264

2764





# 2764与6264



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 2764
C	B	A							
0	0	0							
000B $\overline{Y0}—\overline{CE}$			0000H						
			1	1	1	1	1	1~1	
			1FFFH						
0	0	0	0	0	0	0	0	0~0	IC2 6264
000B $\overline{Y0}—\overline{CE}$			0000H						
			1	1	1	1	1	1~1	
			1FFFH						

## 8.4 I/O地址译码技术

---

### 1. 线选法

### 2. 全地址译码法

51或52系列提供给用户的 I/O口只有P1或P3口的部分口线。所接的外设较多时,就必须扩展I/O接口。

扩展的I/O口和外部数据存储器统一编址、采用相同的控制信号、相同的寻址方式和相同的指令。



# 扩展中的译码技术

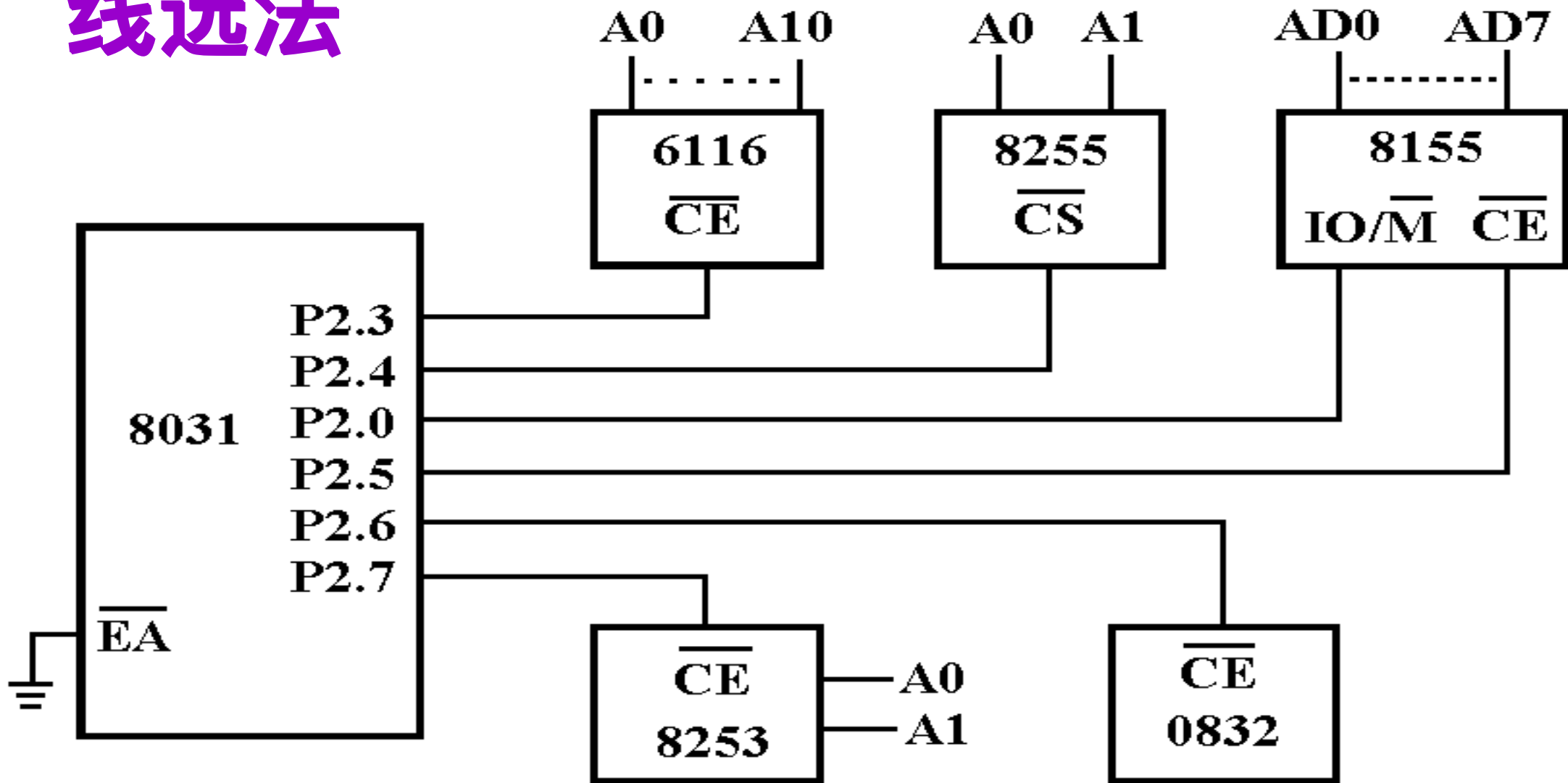
---

- 分清片内地址线与片外地址线（容量）
- 要扩展的芯片个数，统筹考虑分配地址
- 分清程序存储器和数据存储器或I/O口
- 同类存储器防止地址重合



# 扩展RAM和I/O口

## 线选法



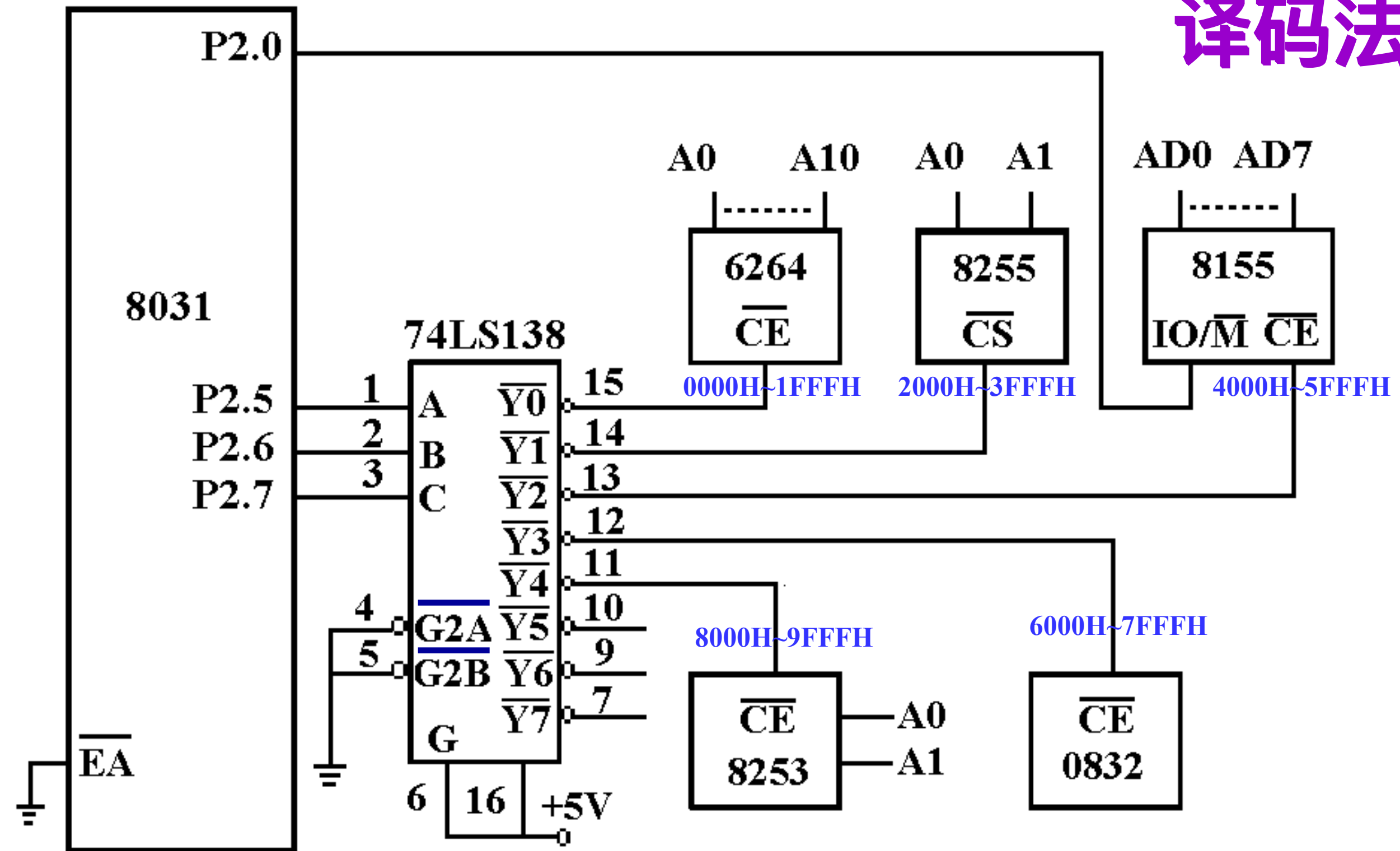
把单独的地址线（通常是P2口的某一根线）接到外围芯片的片选端上。只要该地址线为低电平，就选中该芯片。



外围器件		地址选择线 ( $A_{15} \sim A_0$ )	片内地址单元数	地址编码
6116		1 1 1 1 0 × × × × × × × × × × × × × ×	2K	F000 ~ F7FFH
8255		1 1 1 0 1 1 1 1 1 1 1 1 1 1 × ×	4	EFFC ~ EFFFH
8155	RAM	1 1 0 1 1 1 1 0 × × × × × × × × × ×	256	DE00 ~ DEFFH
	I/O	1 1 0 1 1 1 1 1 1 1 1 1 1 × × ×	6	DFF8 ~ DFFDH
D/AC 0832		1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1	1	BFFFH
定时/计数器 8253		0 1 1 1 1 1 1 1 1 1 1 1 1 1 × ×	4	7FFC ~ 7FFFH

## 数据存储与I/O口同时扩展

# 译码法



当芯片所需的片选信号多于可利用的地址线时，采用全地址译码法。将低位地址作为芯片的片内地址，用译码器对高位地址线译码，译出的信号作为片选线。



外围器件		地址选择线 ( $A_{15} \sim A_0$ )	片内地址单元数	地址编码
6264		0 0 0 × × × × × × × × × × × × × ×	8K	0000~1FFFH
8255		0 0 1 1 1 1 1 1 1 1 1 1 × ×	4	3FFC~3FFFH
8155	RAM	0 1 0 1 1 1 1 0 × × × × × × × ×	256	5E00~5EFFH
	I/O	0 1 0 1 1 1 1 1 1 1 1 1 × × ×	6	5FF8~5FFDH
0832		0 1 1 1 1 1 1 1 1 1 1 1 1 1	1	7FFFH
8253		1 0 0 1 1 1 1 1 1 1 1 1 1 1 × ×	4	9FFC~9FFFH

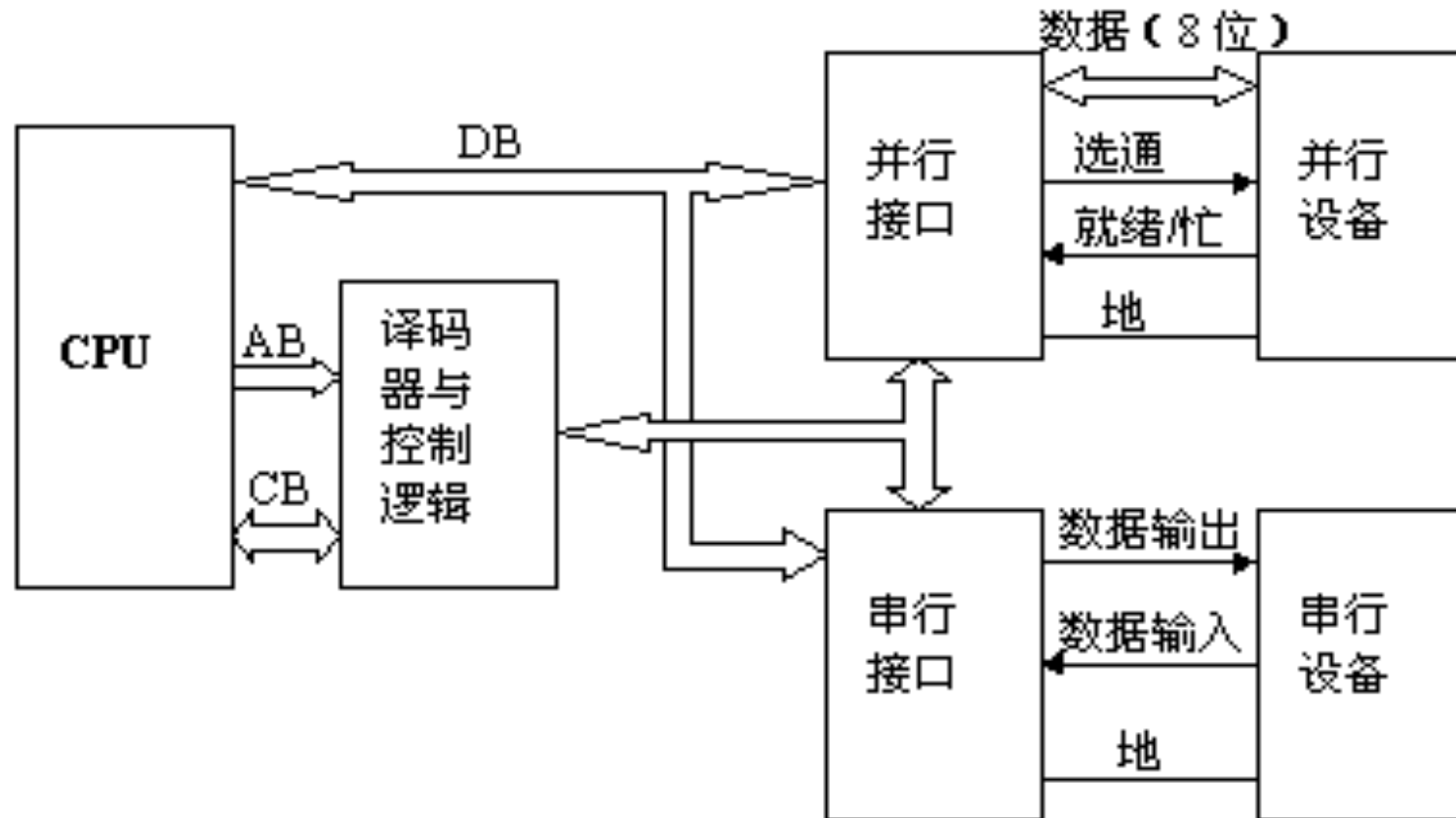
## 数据存储与I/O口同时扩展



# 8.5 并行接口

## 8.5.1 并行接口概述

### 并行接口与串行接口



### 可编程并行I/O接口芯片扩展

可编程I/O接口芯片的特点：适应多种功能需求，使用灵活，可扩展多个并行I/O口，可以编程设定为输入或输出口，应用非常广泛。

## 8.5.2 可编程芯片8255A

### 一、8255A的基本功能

8255A是一个通用可编程接口电路。其具有的资源为：

三个可编程的8位并行I/O口PA、PB和PC口，

PC口分高4位和低4位，高4位可与PA口合为一组(A组)，低4位可与PB口合为一组(B组)，PC口可按位置位/复位；

A口工作方式：方式0、1、2；

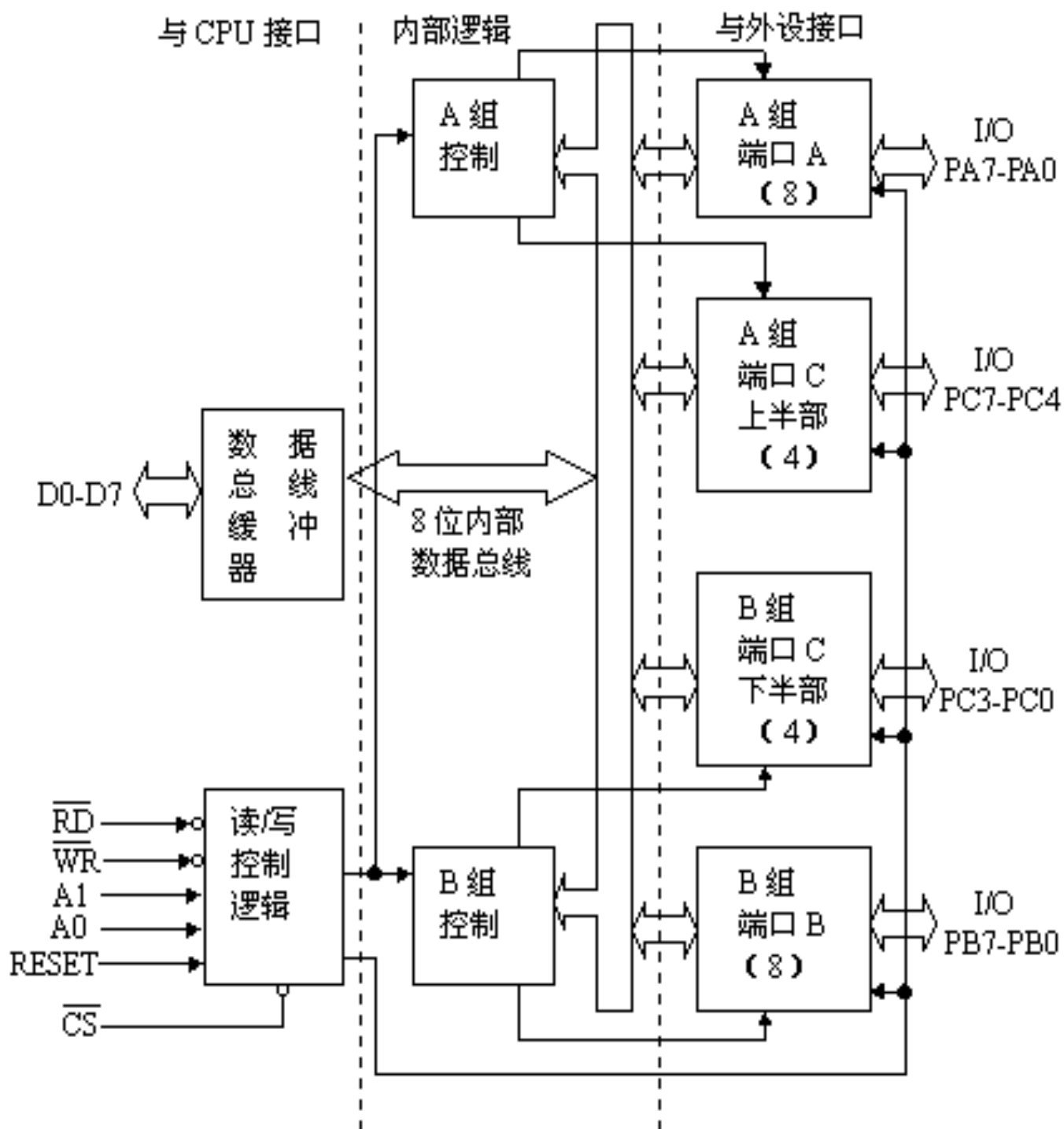
B口工作方式：方式0、1。

一个8位的数据口D0~D7。

能支持无条件、查询、中断控制方式。

### 二、8255A的内部结构和外部引脚

# 1、内部结构



**数据总线缓冲器:** 双向三态驱动器, 与单片机的数据总线相连。

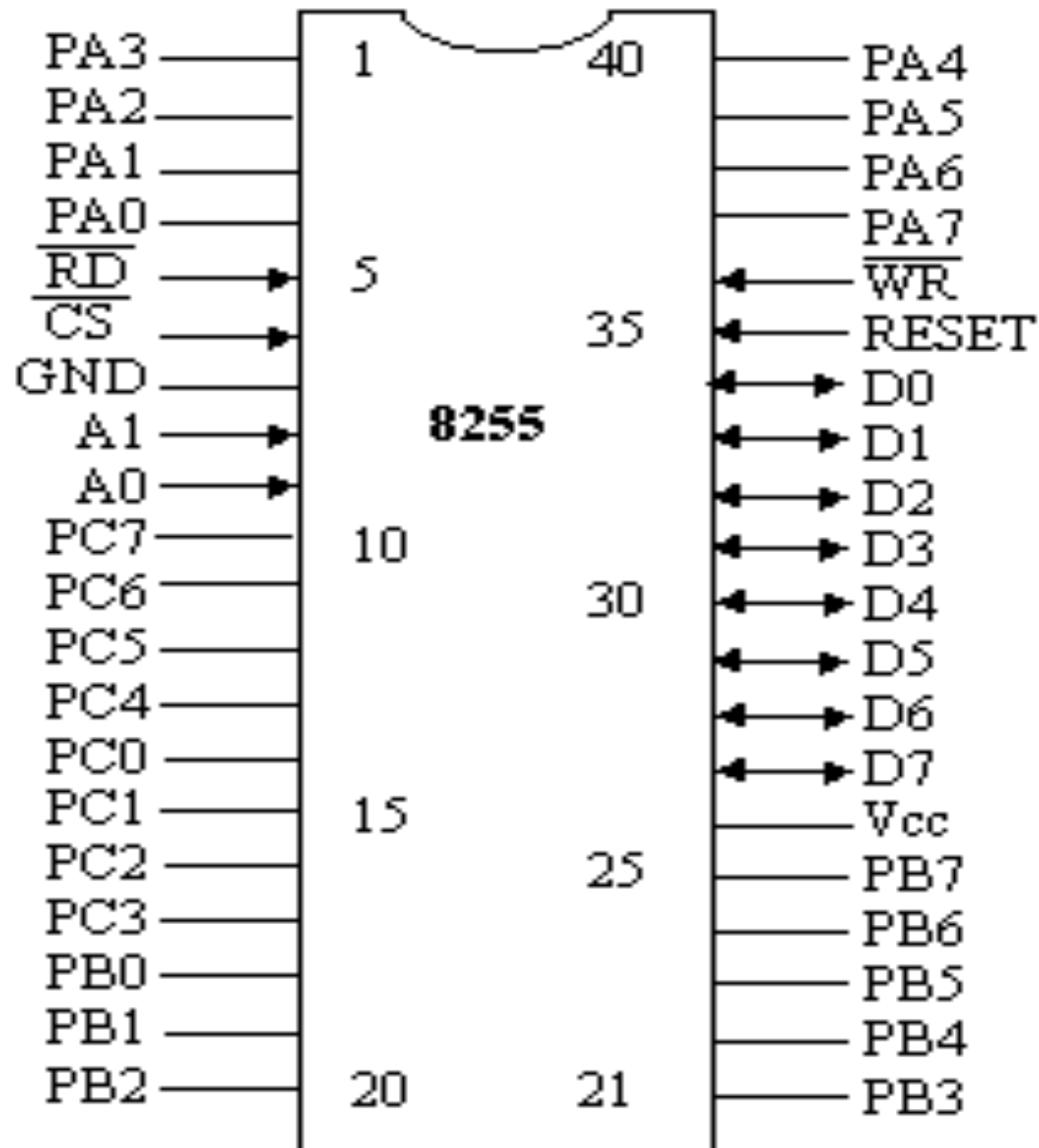
**读/写控制逻辑:** 根据单片机的地址信息(A1、A0)与控制信息( $\overline{RD}$ 、 $\overline{WR}$ 、RESET), 控制片内数据、CPU控制字、外设状态信息的传送。

**控制电路:** 根据CPU送来的控制字使所管I/O口按一定方式工作。对C口可按位实现"置位"或"复位"。控制电路分为两组: A组控制电路控制A口及C口的高4位(PC7~PC4), B组控制电路控制B口及C口的低4位(PC3~PC0)。

**三个并行I/O端口:** A口可编程为8位输入, 或8位输出, 或双向传送; B口可编程为8位输入, 或8位输出, 但不能双向传送; C口分为两个4位口, 用于输入或输出, 也可用作A口、B口的状态控制信号。

## 2、外部引脚

40个引脚，  
DIP封装形式。



**CPU侧:**

**D0-D7, A1A0,  
CS, RD、WR**

**外设侧:**

**PA0-PA7,  
PB0-PB7,  
PC0-PC3,  
PC4-PC7**

# • 引脚

- 1) **数据线(8条)**: D0~D7为数据总线, 用于传送CPU和8255之间的数据、命令和状态字。
- 2) **控制线和寻址线(6条)**:
  - ☐ **RESET**: 复位信号, 输入, 高电平有效。一般和单片机的复位相连。**复位后, 8255所有内部寄存器清0, 所有口都为输入方式。**
  - ☐ **/RD和/WR**: 读/写信号线, 输入, 低电平有效。当/RD为0时, 8255送出信息到CPU; 反之亦然。
  - ☐ **/CS**: 片选线, 输入, 低电平有效。
  - ☐ **A0、A1**: 地址输入线。**4种组合00、01、10、11分别用于选择A、B、C口和控制寄存器。**
- 3) **I/O口线(24条)**: PA0~PA7、PB0~PB7、PC0~PC7为24条双向三态I/O总线, 分别与A、B、C口相对应, 用于8255和外设之间传送数据。
- 4) **电源线(2条)**: VCC为+5 V, GND为地线。

**I/O端口地址：**8255A提供4个端口(使用A1A0)；

**命令：**初始化—设置工作方式；

**操 作**—向PA、PB、PC口写数据，  
从PA、PB、PC口读数据。

**8255A的端口与命令关系表：**

A1	A0	$\overline{\text{RD}}$	$\overline{\text{WR}}$	$\overline{\text{CS}}$	操作	
0	0	0	1	0	A□→数据总线 B□→数据总线 C□→数据总线	输入
0	1	0	1	0		
1	0	0	1	0		
0	0	1	0	0	数据总线→A□ 数据总线→B□ 数据总线→C□ 数据总线→控制口或C□	输出
0	1	1	0	0		
1	0	1	0	0		
1	1	1	0	0		
X	X	X	X	1	端口输出为高阻 非法 端口输出为高阻	禁止
1	1	0	1	0		
X	X	1	1	0		

### 三、 8255A的控制字与初始化编程

8255A有两个控制字：

#### ➤ 方式控制字

用于设定PA口、PB口和PC口的工作方式。

(控制寄存器只能写入不能读出)

#### ➤ 置位/复位控制字

置位/复位控制字用于对PC口按位进行操作。

# 1、工作方式控制字

1	D6	D5	D4	D3	D2	D1	D0
标志位	A 组方式选择 00=方式 0 01=方式 1 1X=方式 2		A 口 0=输出 1=输入	C 口 (高 4 位) 0=输出 1=输入	B 组方式选择 0=方式 0 1=方式 1	B 口 0=输出 1=输入	C 口 (低 4 位) 0=输出 1=输入

工作方式控制字与 C 口按位置/复位冲突控制：

工作方式控制字——D7=1；

C 口按位置/复位——D7=0。



**例1：**要求使用8255A的A口工作于方式1作输入，B口工作于方式0作输出，C口上半部输入，下半部输出，8255A端口地址为7FF0H-7FF3H。请写出初始化程序。

**控制字为：10111000B=0B8H**

**初始化程序为：**

**MOV A, #0B8H**

**MOV DPTR, #7FF3H**

**MOVX @DPTR, A**

## 2、C口按位置/复位控制字

0	D6	D5	D4	D3	D2	D1	D0
标志位	不用 (一般置0)			C口的位选择 000=C口位0 001=C口位1 ..... 111=C口位7			1=置位 0=复位

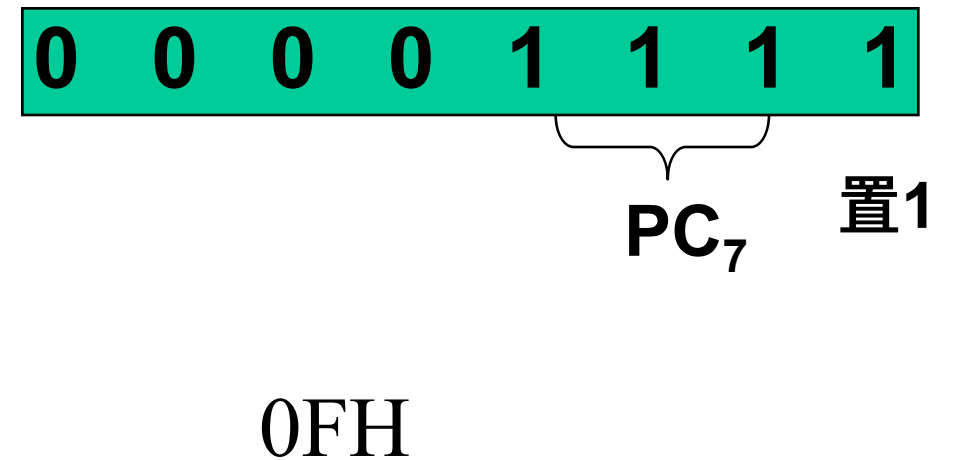
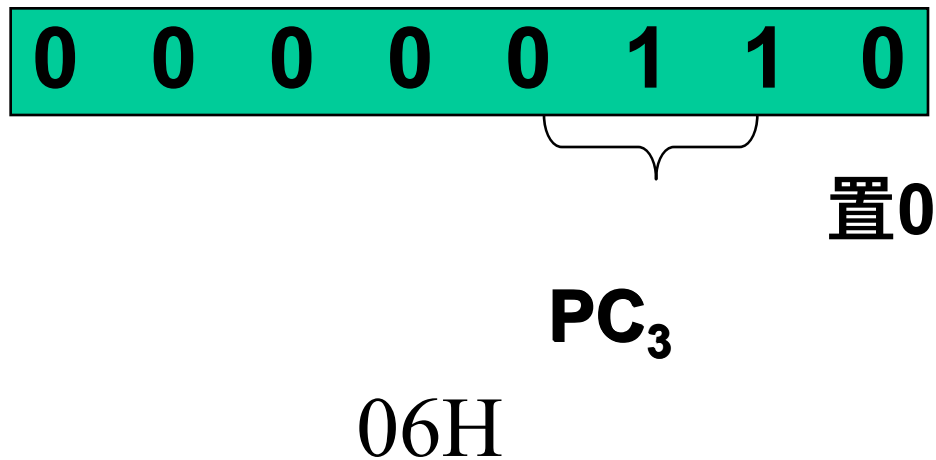
**例2：**如上例，若A口工作于方式1作输入，在使用中断传送方式过程中，要写PC4的按位置位字：

**MOV A, #00001001B**

**MOV DPTR, #7FF3H**

**MOVX @DPTR, A**

例如：要将C端口的 $PC_3$ 置0， $PC_7$ 置1，可用下列程序段实现。



```
MOV  A, #06H      ;  $PC_3$ 置0控制字送A
MOX  DPTR, #PortCtr ; 控制口地址 PortCtr送DPTR
MOVX @DPTR, A      ; 对 $PC_3$  完成置0操作
MOV  A, #0FH      ;  $PC_7$ 置1控制字送A
MOVX @DPTR, A      ; 对 $PC_7$ 完成置1操作
```

# 四、8255A的工作方式

## 三种工作方式：

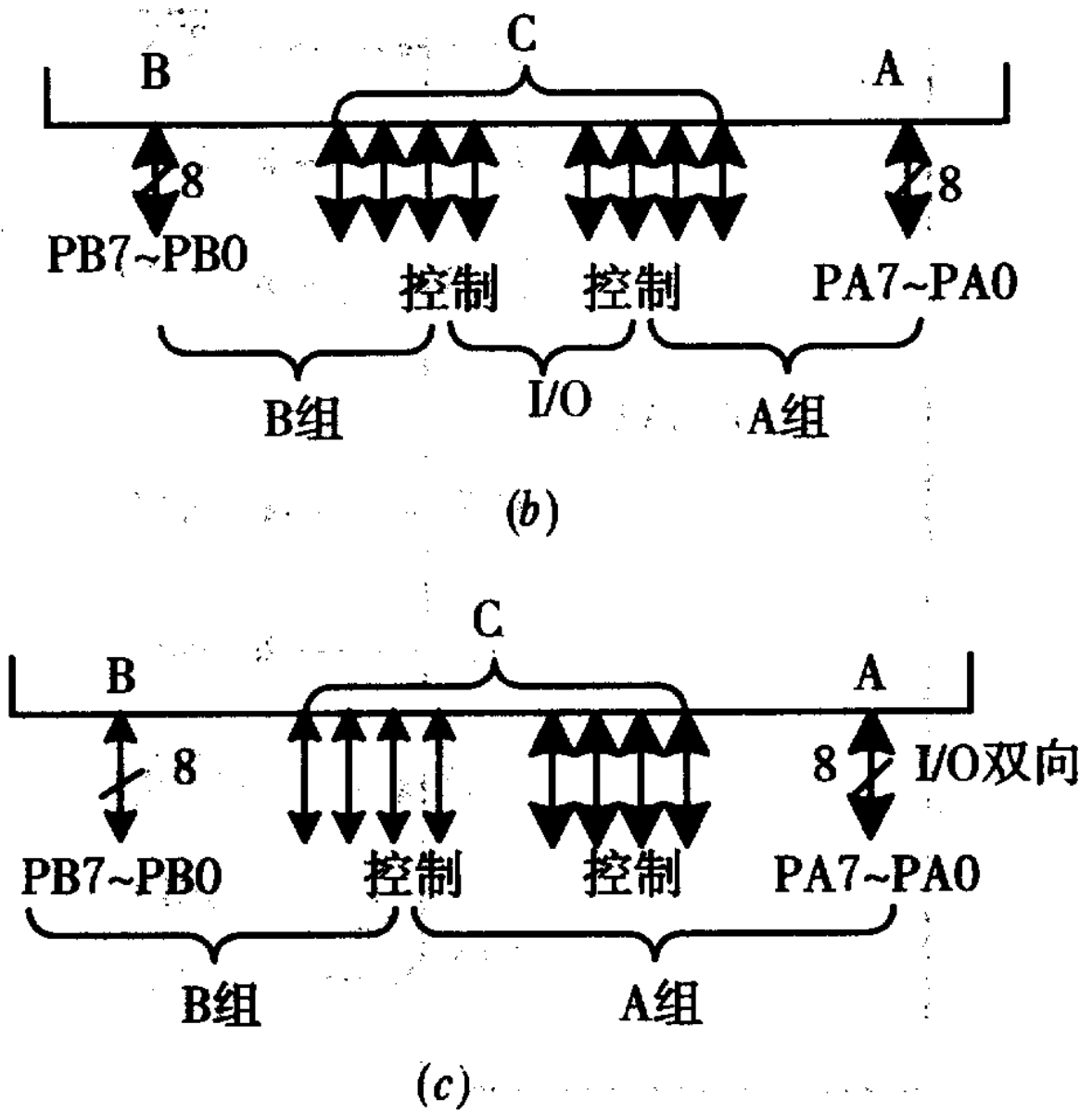
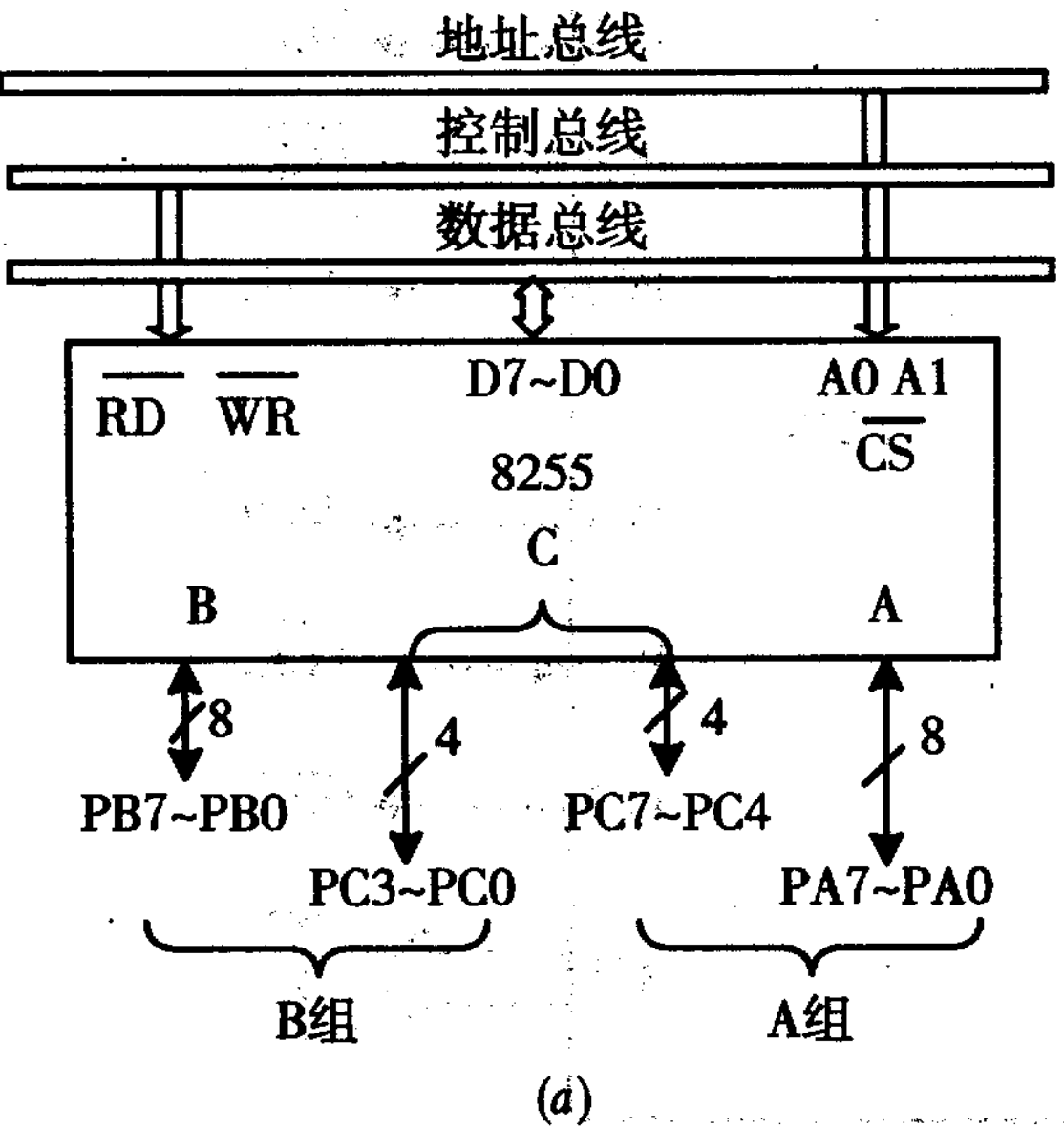


图3 8255A的3种工作方式

(a)方式0基本输入/输出方式； (b)方式1选通输入/输出方式； (c)方式2选通双向传输方式

8255A有三种工作方式：方式0、方式1、方式2。方式的选择是通过写控制字的方法来完成的。

### **方式0：基本输入/输出方式**

输入输出都无需握手信号，端口A、B、C都可作为输入或输出口使用。**该方式适用于无条件与查询方式数据传送。**

### **方式1：选通输入/输出方式**

输入、输出都通过应答关系实现，这时端口A或B用作数据口，端口C的一部分引脚用作握手信号线与中断请求线。**该方式适用于查询与中断方式数据传送。**

### **方式2：选通双向传输方式**

**仅适用于端口A**，这时A口的PA7-PA0作为双向的数据总线，端口C有5条引脚用作A的握手信号线和中断请求线，而B口和C口余下的3位仍可工作于方式0或1。**该方式适用于查询与中断方式数据传送。**

## 四、8255A的工作方式

### 1、方式0—基本I/O方式

**特点：**

**基本的输入/输出方式**

**3个口都可以提供简单的输入/输出操作，  
无须“选通”和“状态”信号。**

**2个8位口，2个4位口**

不需要任何选通信号，适合于无条件传输数据的设备，  
数据输出有锁存功能，数据输入有缓冲（无锁存）功能。

**例3：**用8255A（起始地址为7FF0H）PA.1口发出一个脉冲，编写相应程序。

**MOVA, #80H ; PA口方式0输出, PB口方式0输出**

**MOV DPTR, #7FF3H ; PC口高、低4位均输出**

**MOVX @DPTR, A**

**MOV DPTR, #7FF0H**

**MOV A, #00H**

**MOVX @DPTR, A**

**MOV A, #02H ; PA1产生一个正向脉**

**冲**

**MOVX @DPTR, A**

2021-5-13 **MOV A, #00H**

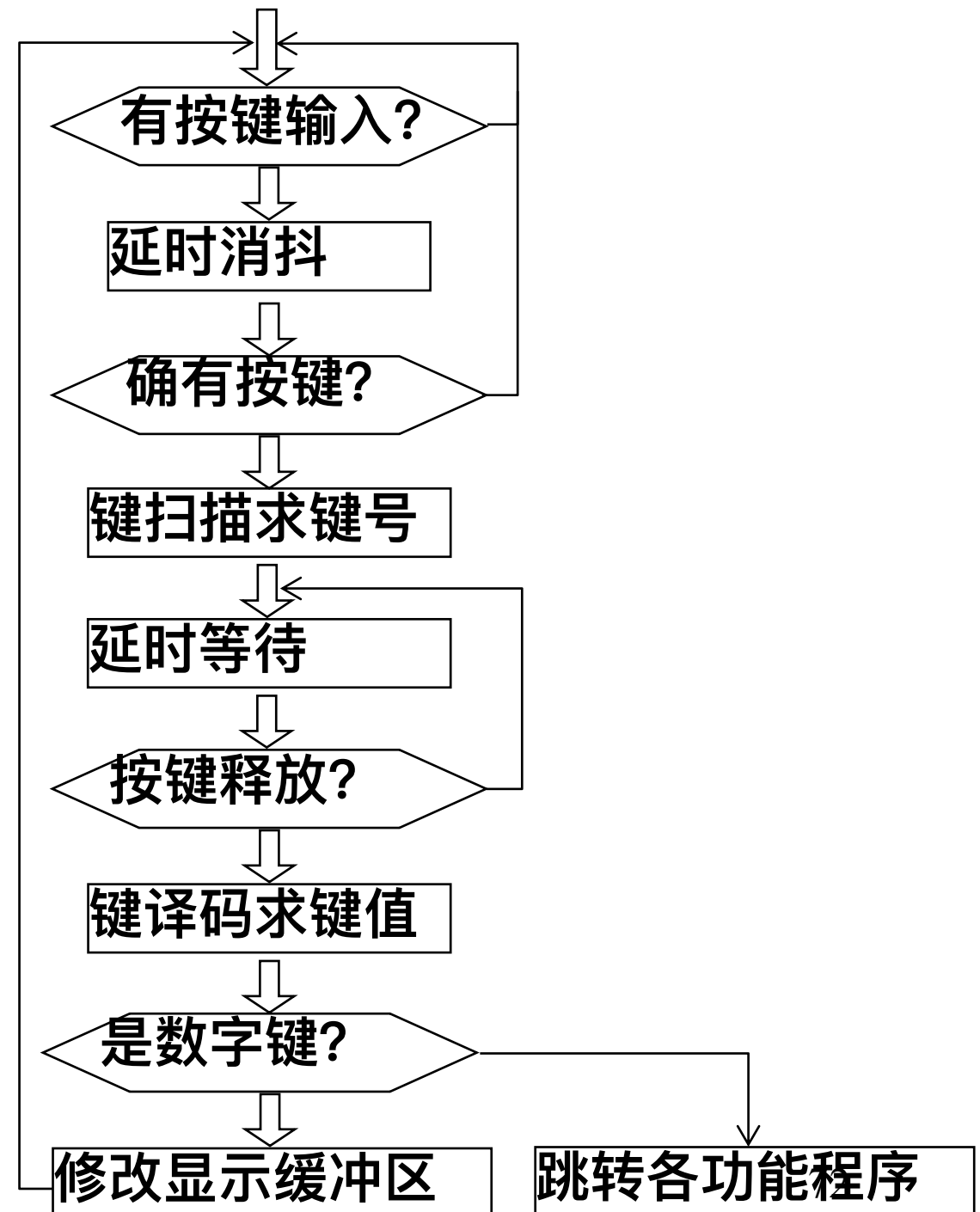
**MOVX @DPTR, A**

# 键盘与单片机接口

## 单个按键的连接与应用

常用的按键，都存在两种状态：断开和闭合。当某一键被按下，则为闭合状态；键释放，则为断开状态。键盘电路的功能就是将键的闭合和断开状态用“0”和“1”来表示，然后通过数据总线送到CPU内部进行键的识别。

### 按键处理程序：





# 键盘接口

## 按键值编码方式:编码键盘与非编码键盘

**编码键盘:** 采用专用的编码/译码器件,被按下的键由该器件译码输出相应的键码/键值。

**特点:** 增加了硬件开销,编码因选用器件而异,编码固定,但编程简单。适用于规模大的键盘。

**非编码键盘:** 单片机系统多采用此类键盘

采用软件编码/译码的方式,通过扫描对每个被按下的键判别输出相应的键码/键值。

**特点:** 不增加硬件开销,编码灵活,适用于小规模键盘,特别是单片机系统。但编程较复杂,占CPU时间,还须软件“消颤”。

# 按键组连接方式:独立式键盘与矩阵式键盘

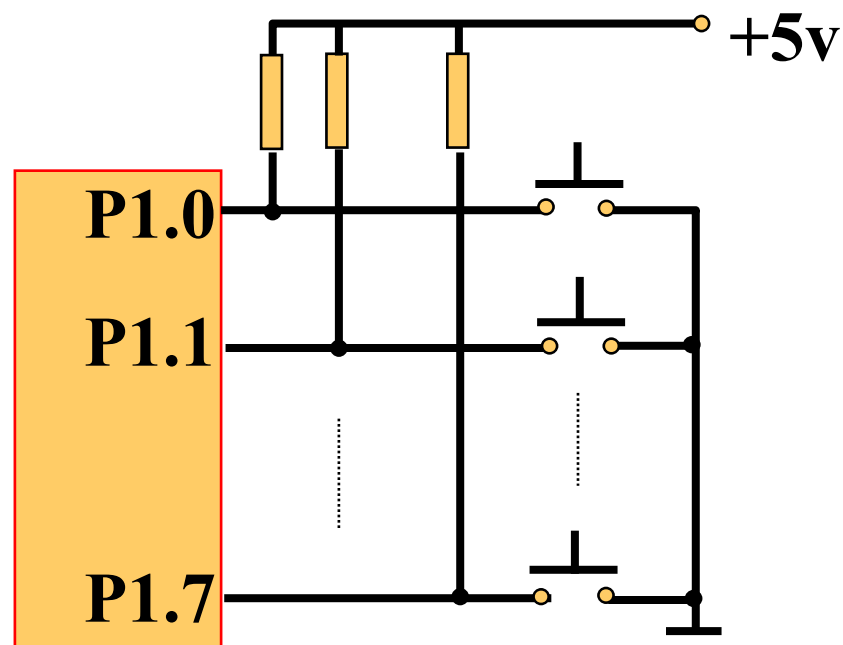
**独立式键盘:** 每键相互独立,各自与一条I/O线相连,CPU可直接读取该I/O线的高/低电平状态。

**特点:** 占I/O口线多,但判键速度快,多用于设置控制键、功能键。适用于键数少的场合。

**矩阵式键盘:** 键按矩阵排列,各键处于矩阵行/列的结点处,CPU通过对连在行(列)的I/O线送已知电平的信号,然后读取列(行)线的状态信息,逐线扫描,得出键码。

**特点:** 键多时占用I/O口线少,但判键速度慢,多用于设置数字键。适用于键数多的场合。

## 独立式键盘电路



每个按键单独占有一根I/O接口引线。

### 1.扫描法

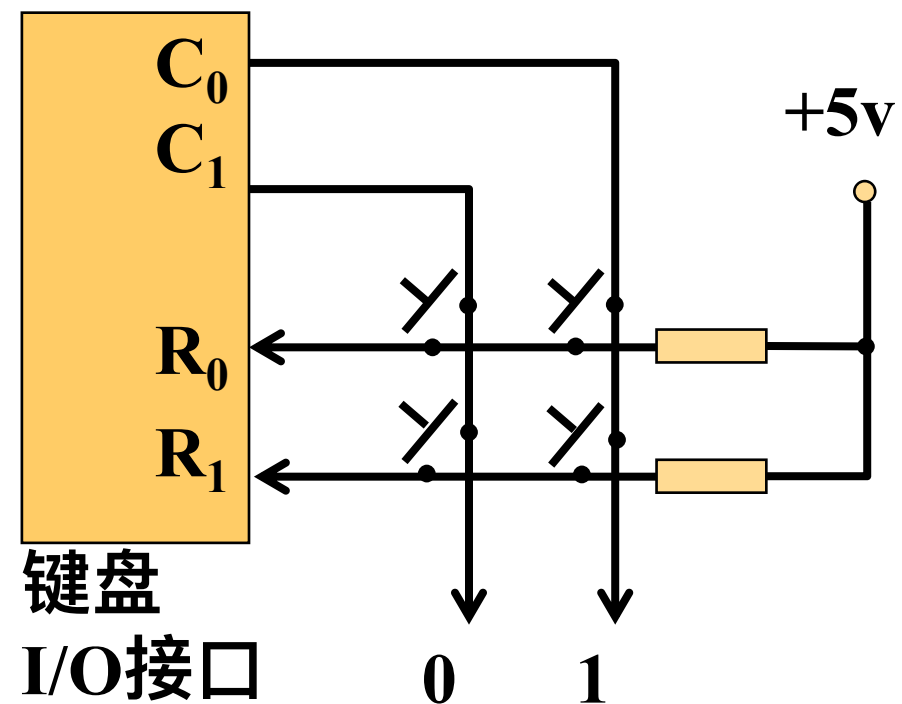
列线输出，行线输入。

列线逐行输出0，某行有按键，行线输入有0，若无按键，行线输入全部为1。

### 2.反转法

行列线交换输入、输出，两步获取按键键号。

## 矩阵式键盘

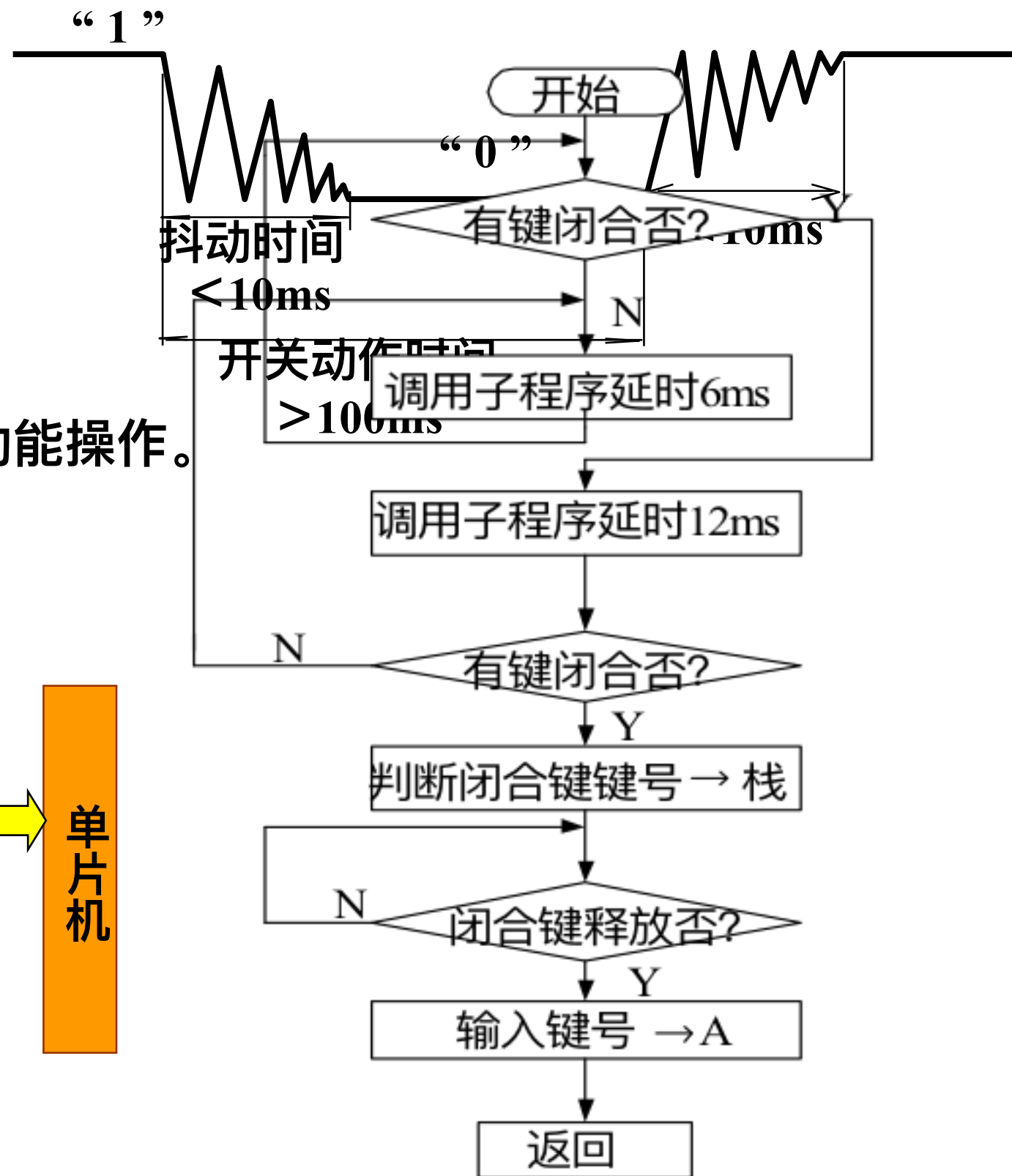
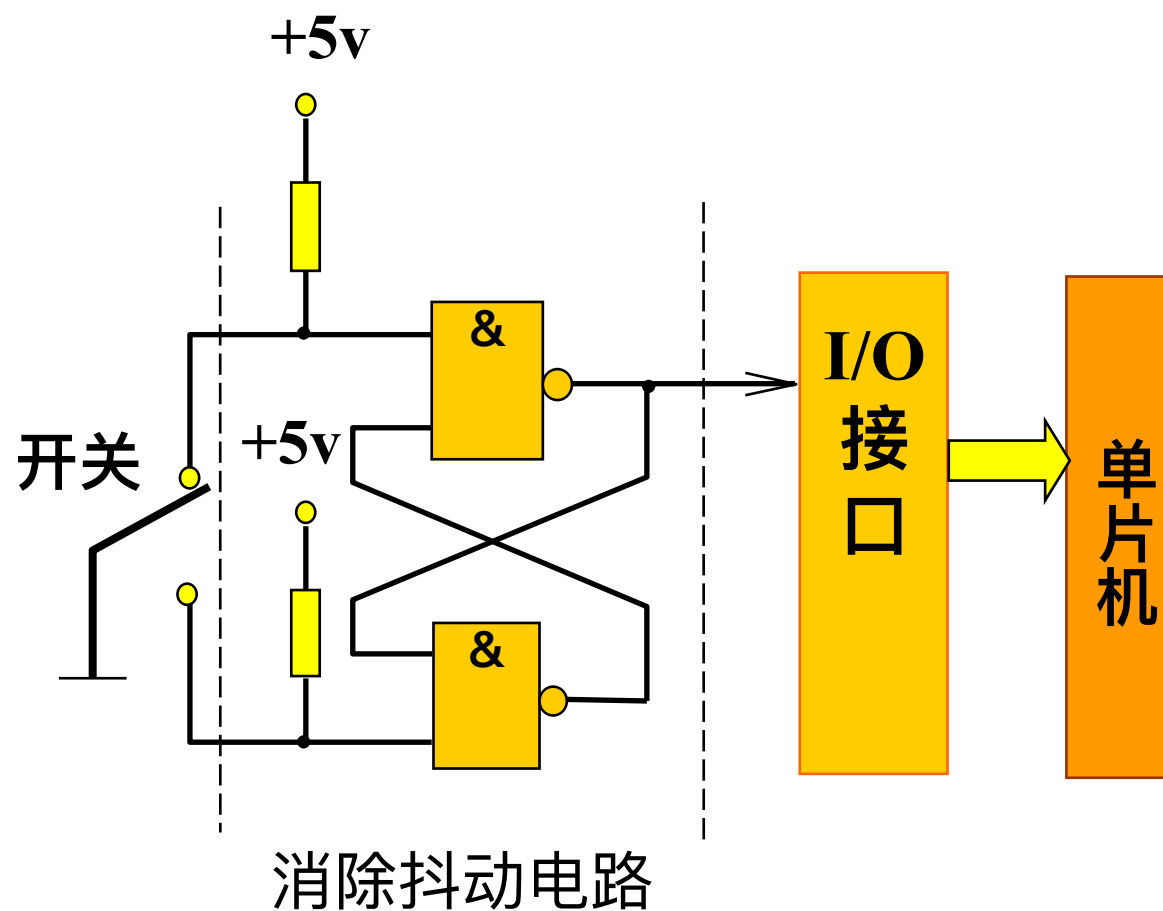


用I/O口线组成行、列结构，按键设置在行列的交点上。

# 键盘的工作方式

## 键盘扫描子程序的功能：

- ① 判断键盘上有无键按下；
- ② 去除键的机械抖动影响；
- ③ 求按下键的键号；
- ④ 键闭合一次仅进行一次键功能操作。



# 键盘处理程序任务

## 1.键输入

检查键盘是否有键被按下，消除按键抖动。确定被按键的键号，获取键号。

## 2.键译码

键号为键盘位置码，根据键号查表得出被按键的键值。键值：数字键0~9、字符键0AH~0FH、功能键10H~ 。

## 3.键处理

根据键值转移到不同程序段。

若键值属于数字、字符键，则调用显示数字和字符的子程序。

若键值属于功能键，则进行多分支转移，执行各个功能程序段。

## 2、方式1—选通输入/输出方式

A组包括A口和C口的高四位（PC7~PC4），A口可由程序设定为输入口或输出口，C口的高四位则用来作为输入/输出操作的控制和同步信号；B组包括B口和C口的低四位（PC3~PC0），功能和A组相同。

### (1) 方式1输入

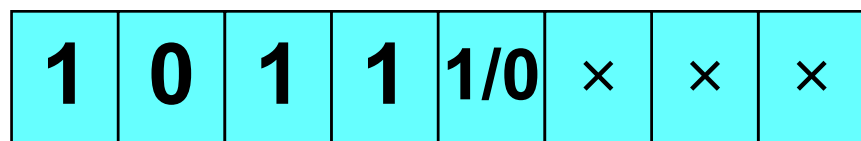
**当任一端口工作于方式1输入时，/STB与/IBF构成了一对应答联络信号。**

### (2) 方式1输出

**当任一端口工作于方式1输出时，/OBF与/ACK构成了一对应答联络信号。**

# (1)选通的输入方式

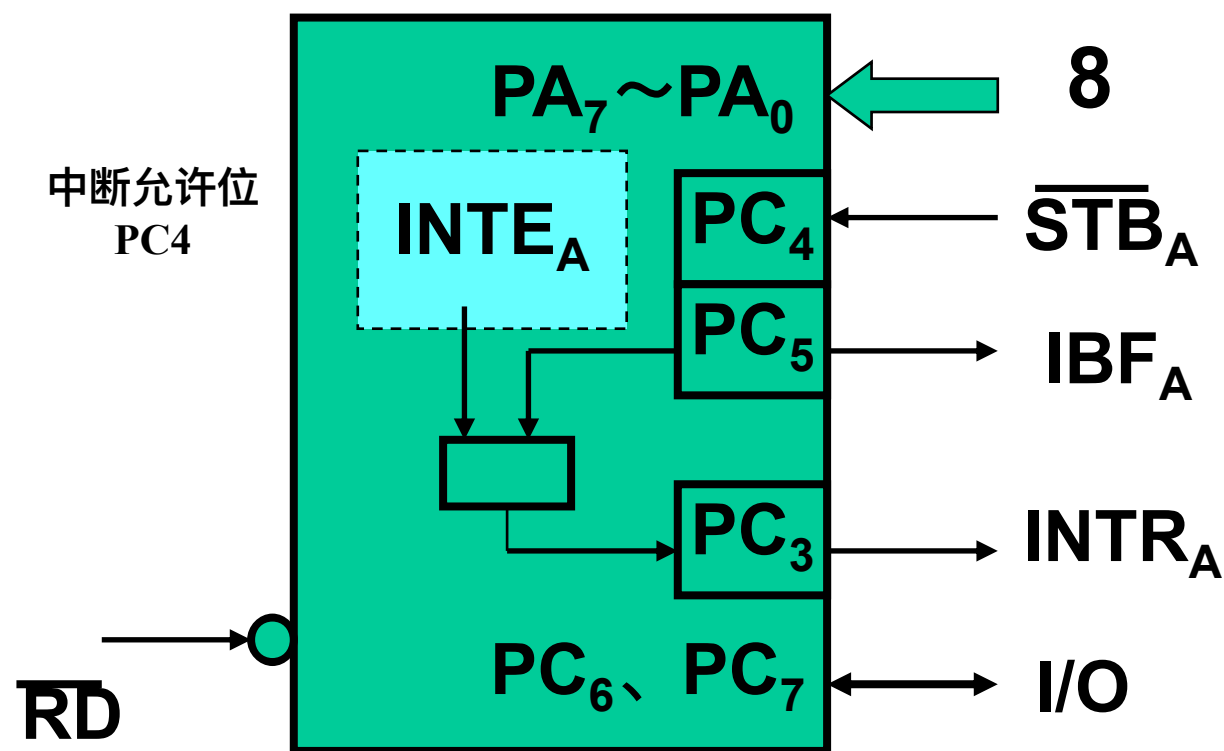
A组工作于方式1输入的控制字



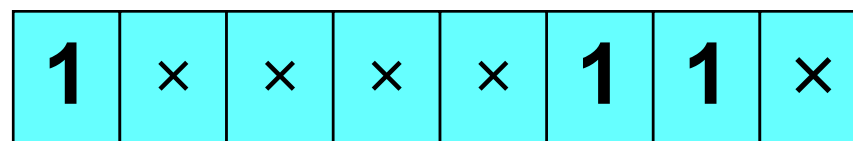
方式1

PC<sub>7</sub>、PC<sub>6</sub>  
1输入, 0输出

端口A输入

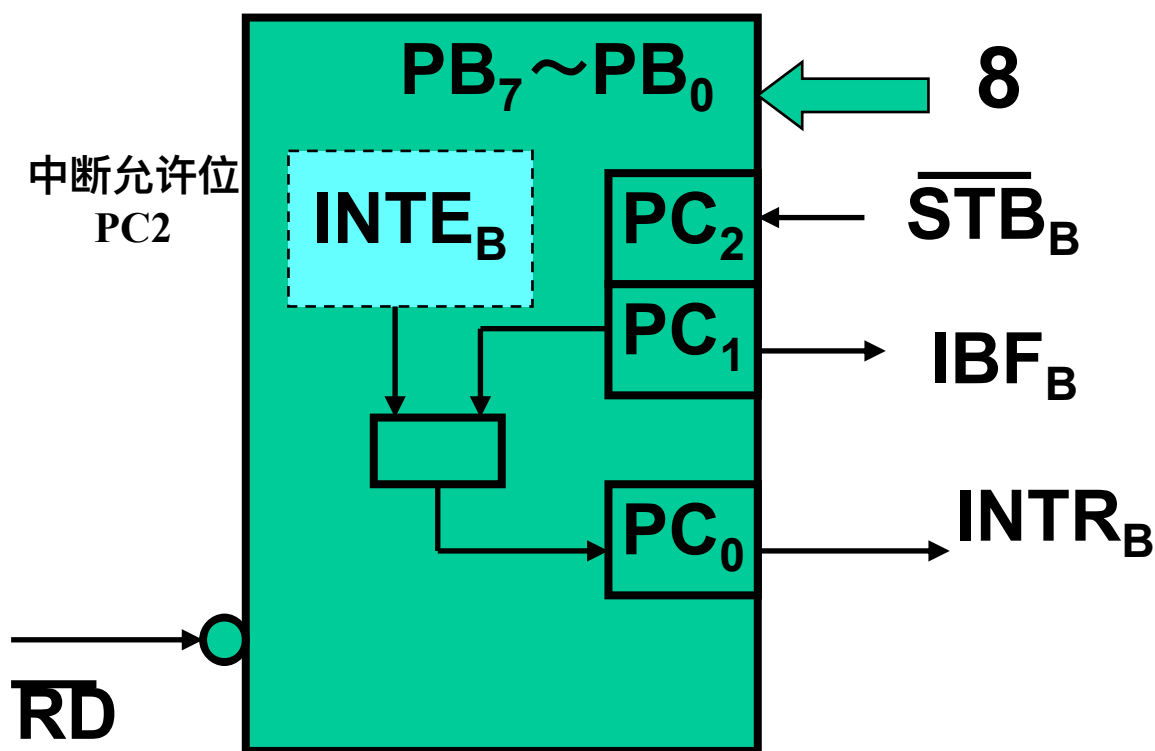


B组工作于方式1输入的控制字



方式1

端口B输入



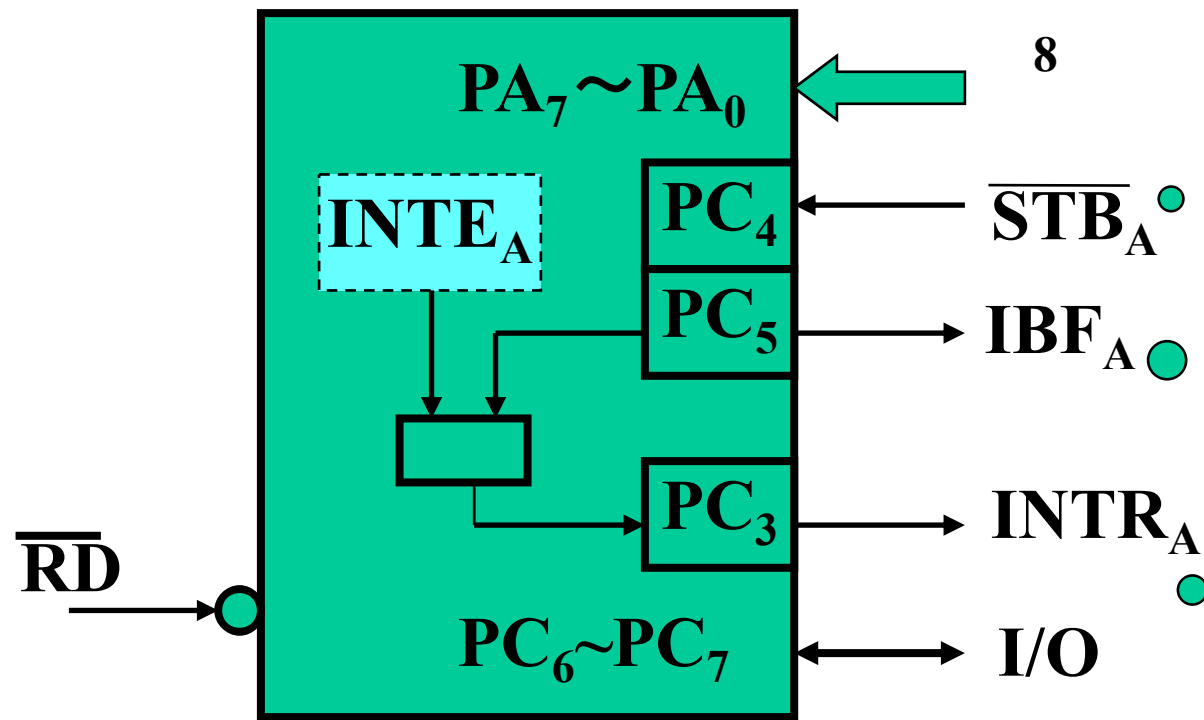
/STB —— 输入选通信号, 外设发来, 表示数据已送入输入缓冲器。

IBF —— 输入缓冲器满信号, 发给外设 (通知外设数据未被取走, 暂不能接收新数据)

INTR —— 中断请求信号, 8255A发给单片机

INTE A, B —— 中断允许触发器

## A口在**选通输入**方式下的工作过程



选通信号输入端，低电平有效，由**外设**提供。当外设输入一个数据到  $PA_7 \sim PA_0$  时，该信号有效。

输入缓冲器满信号，高电平有效，由**8255输出至外设**。当  $IBF=1$  时，表示当前有一个新数据在输入缓冲器中，可作为状态信号供CPU查询用。

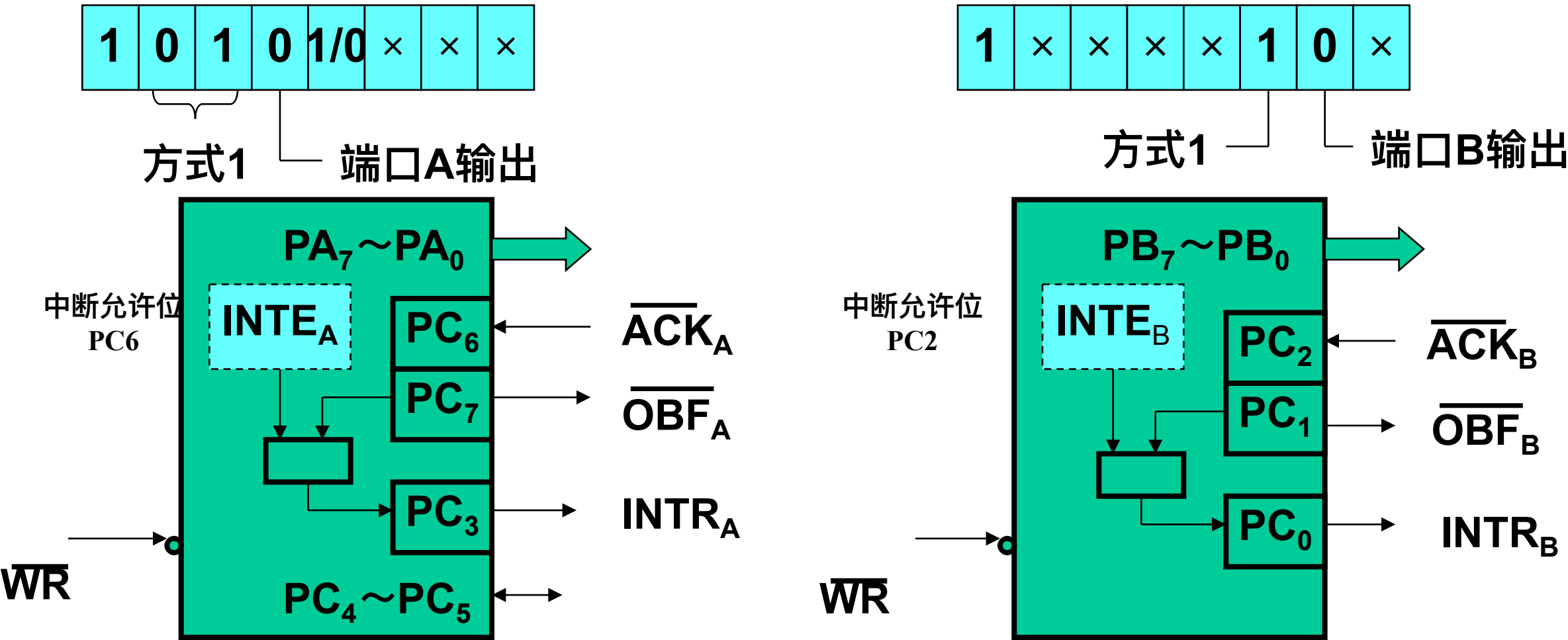
**8255送往CPU**的中断请求信号，高电平有效。CPU响应中断后，可通过中断服务程序从A口的数据输入缓冲/锁存器读取输入设备送来的数据。

- $\overline{STB}_A$  有效
- 8255收到  $\overline{STB}_A$  后自动做两件事：一是把  $PA_7 \sim PA_0$  上的输入数据存入A口的数据输入缓冲/锁存器；
- 二是使  $IBF_A$  有效，阻止外设输入新的数据
- 在  $INTE_A$  和  $IBF_A$  共同有效下发出中断请求
- 中断响应



# (2)选通的输出方式

方式1在选通输出情况下对应的控制信号：



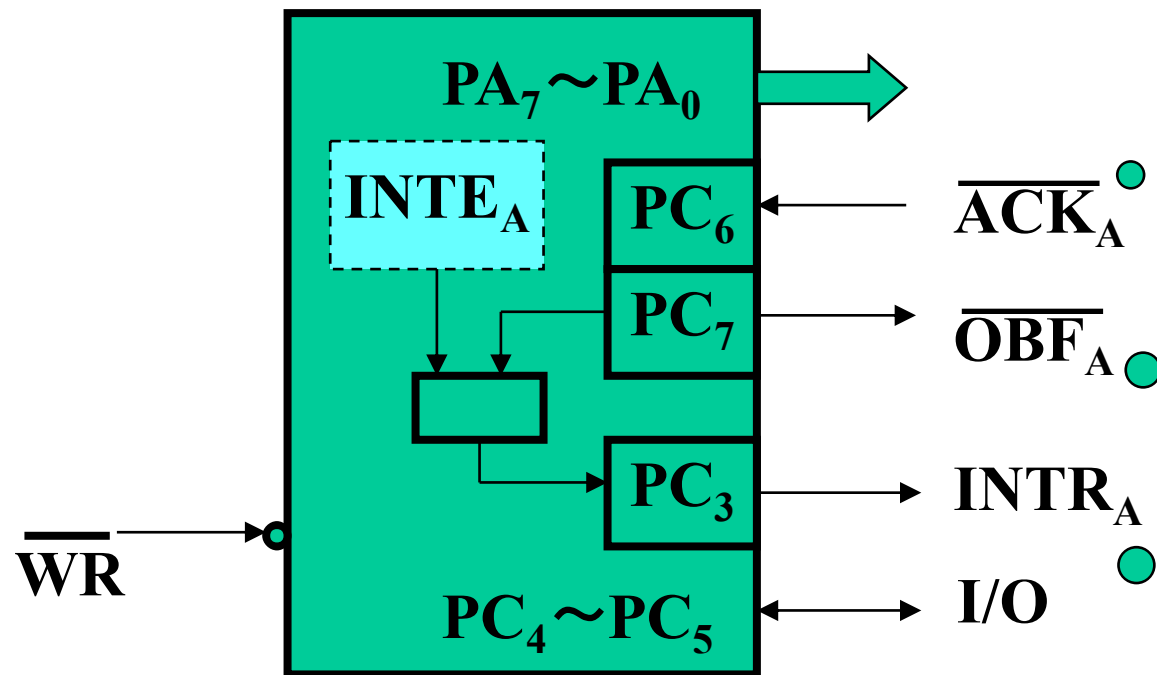
/OBF —— 输出缓冲器满信号，发给外设（单片机将数据已送到指定口，外部设备可以取走）

/ACK —— 外设响应信号，由外部设备发来（数据已被外设取走）

INTR —— 中断请求信号

INTE<sub>A</sub>、INTE<sub>B</sub> —— 中断允许触发器

## A口在选通输出方式下的工作过程



- 单片机通过MOVX指令把输出数据送到A口的数据输出缓冲/锁存器，8255收到后使/OBF<sub>A</sub>有效
- 外设收到/OBF<sub>A</sub>后自动做两件事：  
一是从PA7~PA0上取走输出数据；  
二是使/ACK<sub>A</sub>有效
- 发出中断请求，让CPU输出新数据
- 中断响应

外设响应输入信号，低电平有效，由**外设送给8255**。有效时表示外设已经取走CPU通过8255输出的数据。

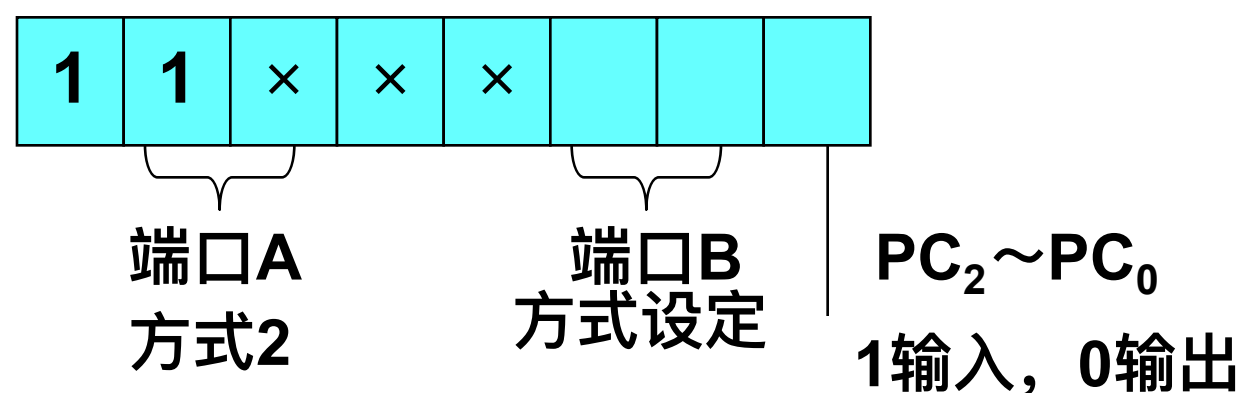
输出缓冲器满，低电平有效，由8255送给外设。有效时表示CPU已经把数据写入8255指定的端口，通知外设可以把数据取走。

**8255送往CPU的中断请求信号，高电平有效。当外设已经接收了CPU通过8255输出的数据后，INTR=1，向CPU申请中断，要求CPU继续输出数据，CPU在中断程序中把数据写入8255。**

### 3. 方式2：带选通的双向传输方式

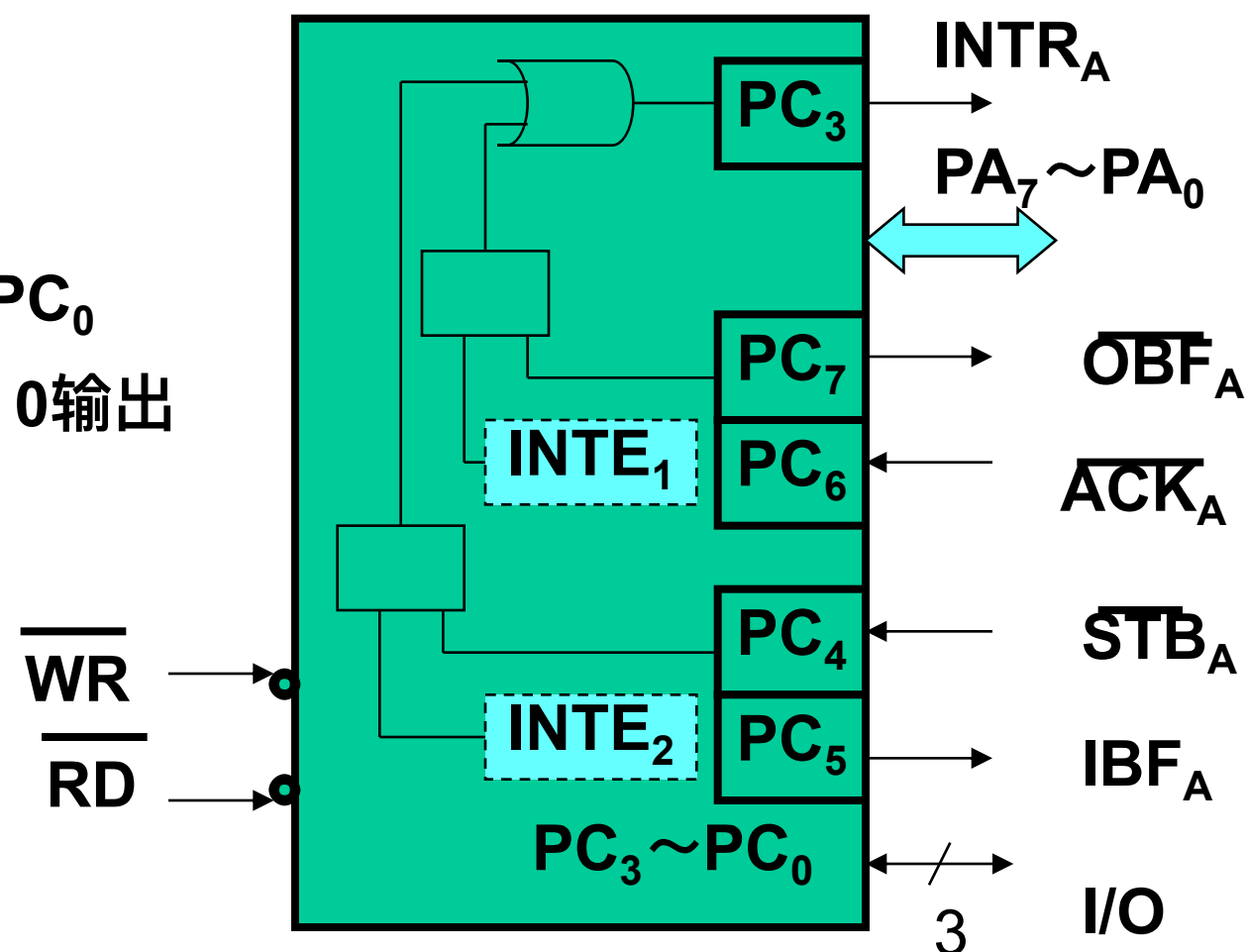
**双向的传输方式：可以向外设发送数据；又可以接收从外设发来的数据(此时B口可以工作在方式0或方式1 )。**

**方式2的基本定义如下：只能适用于A端口。一个8位的双向端口(A端口)和1个5位的控制端口(C端口PC7~PC3)。**

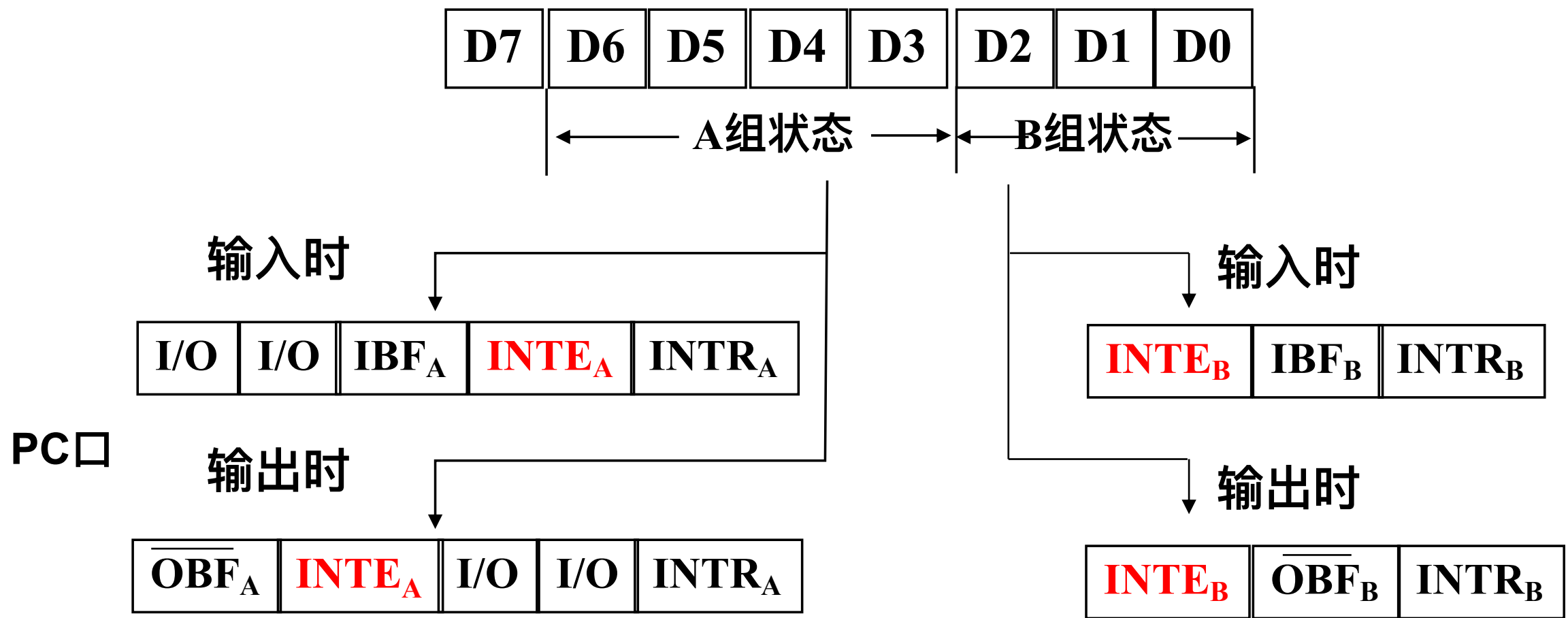


### 方式2时各控制信号：

**/STB**: 选通信号, 低电平有效。  
**IBF**: 输入缓冲器满信号, 高电平有效。  
**/OBF**: 输出缓冲器满信号, 低电平有效。  
**INTR**: 中断请求信号, 高电平有效。  
**/ACK**: 数据接收应答信号, 低电平有效。  
**INTEI**: 输出中断允许信号。



# 8255状态字



## 方式1下的状态字



**例5：在8255与单片机的连接电路中，使用置位/复位控制字实现在8255的PC7输出方波。**

**解：8255的控制寄存器(7FFH)的设定**

**D7：恒为0；**

**D6-D4：未使用；**

**D3-D1：PC口的位选择；**

**D0：置位时为1，复位时为0。**

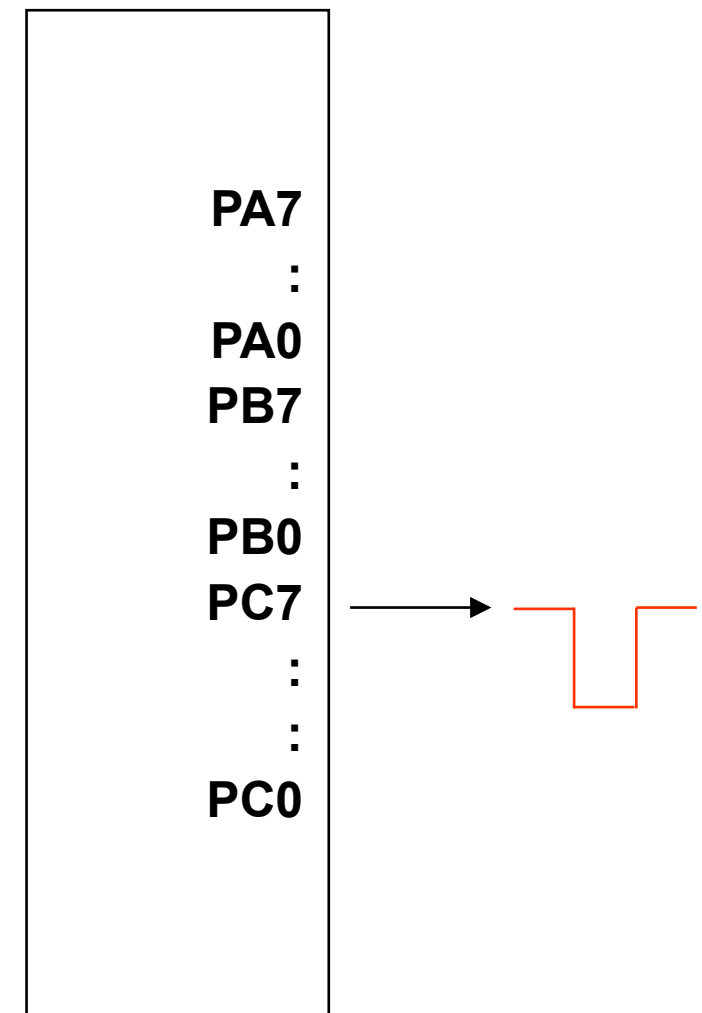
**所以：**

**当PC7输出高电平时，**

**控制寄存器应该为：0FH；**

**PC7输出低电平时，**

**控制寄存器应该为：0EH。**



# 程序清单：

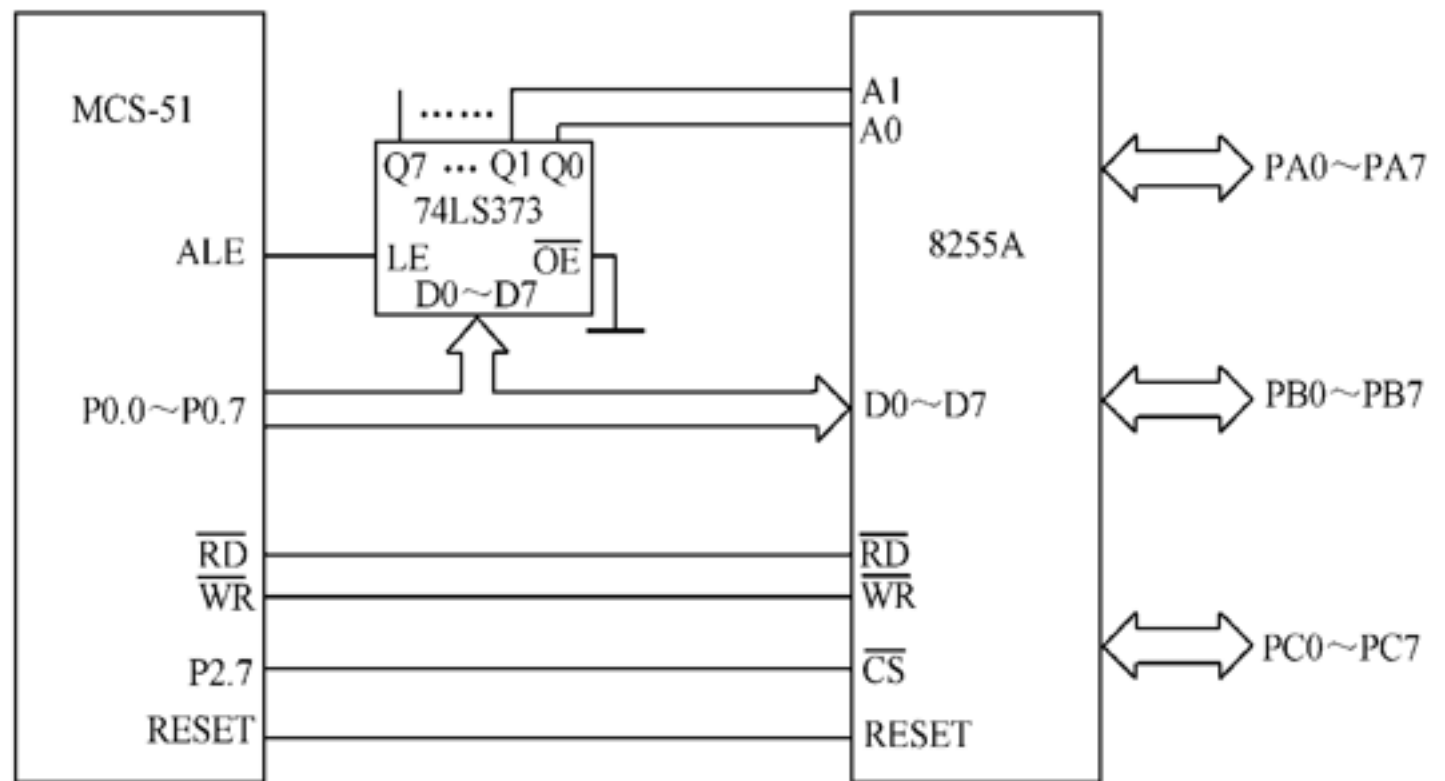
•	<b>ORG</b>	<b>0000H</b>	
	<b>MOV</b>	<b>DPTR,#7FFFH</b>	
	<b>LOOP: MOV</b>	<b>A,#0FH</b>	} 写PC7=1的命令字
	<b>MOVX</b>	<b>@DPTR,A</b>	
	<b>MOV</b>	<b>R2, #80H</b>	
	<b>DJNZ</b>	<b>R2, \$</b>	} 延时
	<b>MOV</b>	<b>R2,#80H</b>	
	<b>MOV</b>	<b>A,#0EH</b>	} 写PC7=0的命令字
	<b>MOVX</b>	<b>@DPTR,A</b>	
	<b>DJNZ</b>	<b>R2, \$</b>	
	<b>SJMP</b>	<b>LOOP</b>	← 循环执行
	<b>END</b>		

## 五、8255应用举例

MCS-51单片机外扩8255A芯片的电路原理图如下图所示。8255A芯片内部已有数据总线驱动器，可以直接与MCS-51单片机总线相连接（P0口接D0~D7）。8255A的RESET， $\overline{RD}$ ， $\overline{WR}$  分别与MCS-51单片机的RESET， $\overline{RD}$ ， $\overline{WR}$  相连，

$\overline{CS}$  接P2.7，

单片机地址线最低2位分别接8255A芯片的A1，A0。PA，PB，PC及控制寄存器的地址分别是7FFCH，7FFDH，7FFE H和7FFFH。



假设图中8255的PA口接一组开关，PB口接一组指示灯，如果，要将MCS-51的寄存器R2的内容送指示灯显示，将开关状态读入MCS-51的累加器A，则8255初始化和输入/输出程序如下：

```
ORG 1000H  
R8255:  MOV DPTR, #7FFFH;  
        MOV A, #98H; 10011000B  
        MOVX @DPTR,A  
        MOV DPTR,#7FFDH; B口输出  
        MOV A,R2  
        MOVX @DPTR, A  
        MOV DPTR,#7FFCH; A口输入  
        MOVX A,@DPTR  
        RET
```



# 第八章结束

# 课程小节

- ▶ 计算机运算基础（数制、编码、运算、运算结果处理）
- ▶ 基本概念（单片机内部资源、存储器结构、寻址方式、中断、输入输出方式、存储器和接口扩展、串行异步通信）
- ▶ 指令系统和汇编语言程序设计（程序结构、处理过程、指令和寻址方式的正确运用）
- ▶ 存储器扩展方式（程序存储器、数据存储器 and 接口；线选方法、译码方法）
- ▶ 单片机内串口和定时计数器的初始化编程和应用
- ▶ 外扩接口8255的初始化编程和应用

# 考试题型

- 填空题
- 选择题(单项)
- 判断题
- 程序分析和设计
- 简答题
- 存储器扩展
- 综合应用