

# 第8章 I/O端口地址译码技术

主讲：许万茹





# 本章教学内容

1

I/O地址空间和I/O端口

2

I/O端口地址分配及选用的原则

3

地址译码技术

**重点：I/O空间的地址译码技术，集成电路或译码器构成的译码电路**



## 8.1 I/O地址空间

- 与存储器空间一样，I/O地址空间也是一片**连续的地址单元**，可供各种外设作为与CPU交换信息时存放数据、状态和命令代码之用
- 一个I/O地址对应的是接口电路中的一个寄存器或控制器，称为“**端口**”
- I/O地址分配给某个外设后就成为该外设的**固有端口地址**，其它外设不能使用，否则会发生冲突



- I/O端口与存储器单元一样，以**字节**组织数据
- 实模式下，I/O端口地址线是16位的，范围是**0000H-0FFFFH**，共**64K**个可单独寻址的**8位端口**，2个相邻的8位端口可构成1个16位端口，4个相邻的8位端口可构成1个32位端口；16位端口对应于偶数地址，32位端口对应于被4整除的地址，8位端口的地址可以从任意地址开始
- IBM公司在规划微机**接口卡**时采用**非完全译码方式**，只考虑**低10位**地址线A0-A9，所以端口的地址范围是**0000H-03FFH**

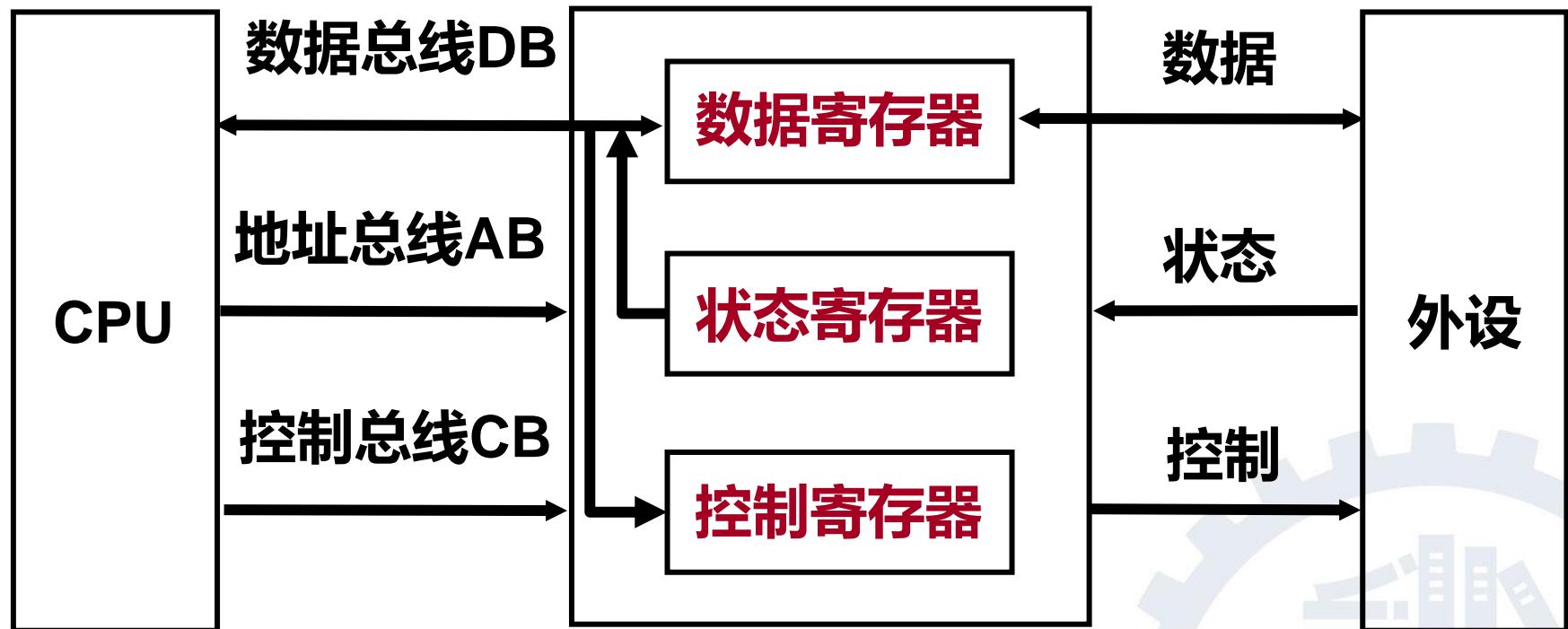


## 8.2 I/O端口

### 8.2.1 什么是I/O端口

I/O端口 (port) 是接口电路中能被CPU直接访问的寄存器的地址。CPU通过这些端口向接口中的寄存器发送命令、读取状态和传送数据。因此，一个接口可以有几个端口，如命令端口、状态端口和数据端口，分别对应于命令(控制)寄存器、状态寄存器和数据寄存器。

I/O操作是对I/O端口的操作，而不是对I/O设备的操作。  
(同一I/O端口可能对应不同的I/O设备 (共享的情况下)  
)，中间有一个选择过程-译码电路)。

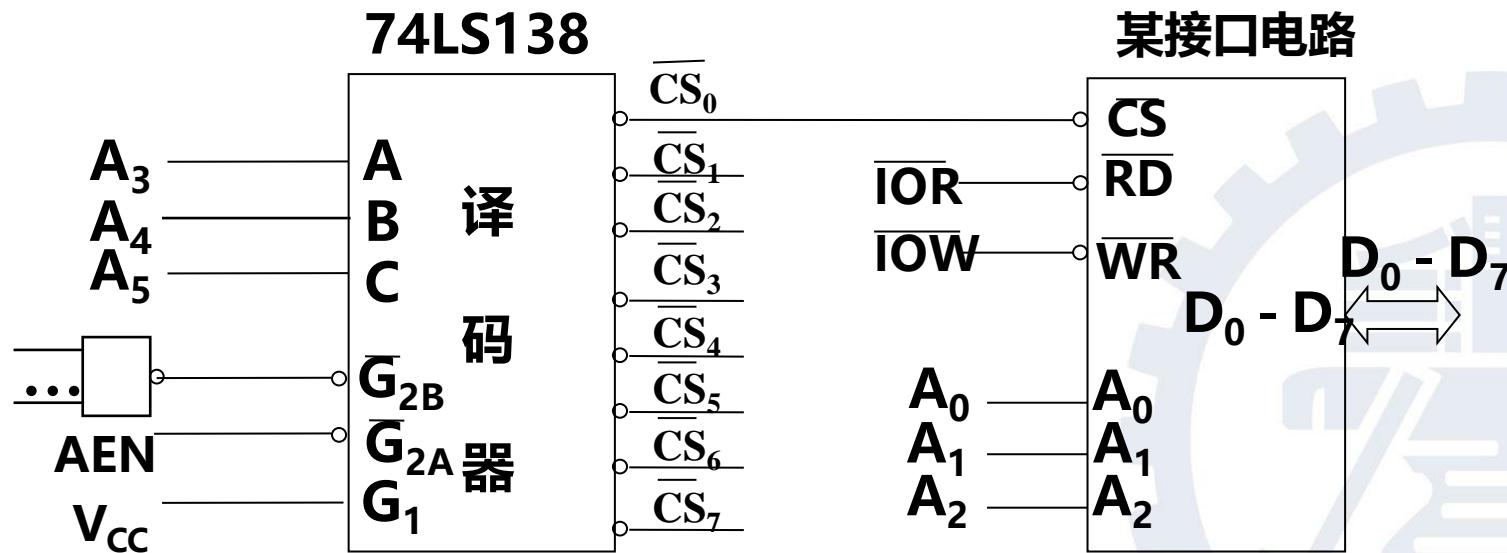


I/O接口的典型结构



从CPU到接口电路：中间有一个地址译码电路；地址译码的结果就是产生一个片选信号/CS。/CS是一个低电平脉冲，不持续有效，**只在脉冲期间CPU才可以读/写接口内的寄存器，其它时刻CPU并不控制接口电路。**

另一方面，接口电路可以根据CPU之前写入的数值来**独立工作，不依赖于/CS信号是否有效。**





## 8.2.2 I/O端口共用技术

一般情况下，一个端口只允许接纳一种信息，但有些接口芯片中，一个端口即可作命令口又可作状态口使用，或允许同一个命令口写多个命令字，这就产生了端口的共用。

缓解端口资源紧张情况



## 8.2.3 I/O端口地址编址方式

### 基本概念

- 接口内部包含一个或多个CPU可进行读写的寄存器，这些寄存器称为I/O端口。数据端口、状态端口和控制端口。
- CPU对外设的各种操作，最终都归结为对接口中各端口的读/写操作，而不是直接对外设进行操作。
- 为了区分不同的端口，需为每个端口分配一个地址编号，称为I/O端口地址。
- 由于CPU地址总线既连接到存储器又连接到I/O端口，因此，需要一种机制来区分和寻址要操作的内存单元或I/O端口，这种机制称为编址方式。
- 常用的两种I/O编址方式：  
I/O端口与内存独立编址；I/O端口与内存统一编址。



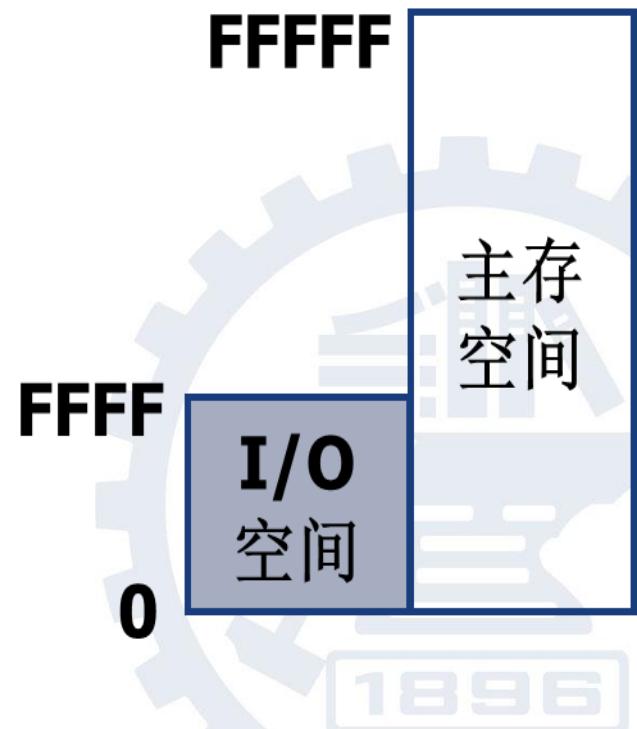
## 1、独立编址

端口地址单独编址，而不和存储空间合在一起，访问I/O端口使用专门的I/O指令。IBM-PC系列采用该方式。

### 优点：

- I/O端口地址不占用存储器空间。
- I/O指令短，执行速度快。
- 由于专门I/O指令与存储器访问指令有明显的区别，使程序中I/O操作和存储器操作层次清晰，程序的可读性强。

缺点：需要专门访问I/O端口的指令。





## 2、统一编址

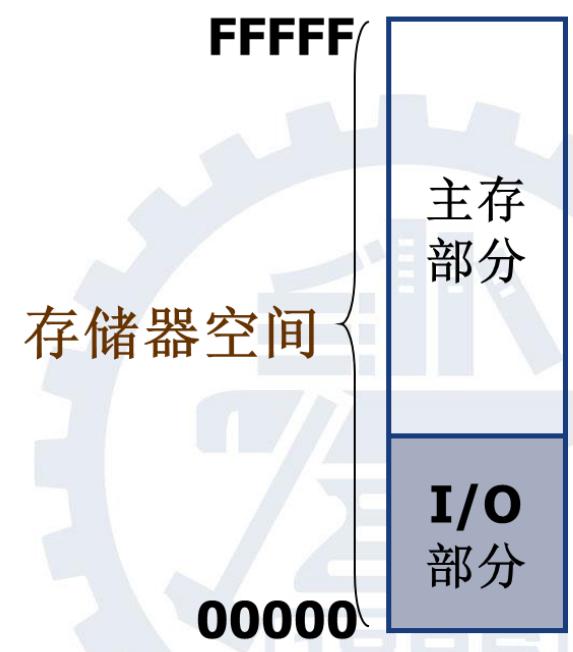
从存储器空间中划出一部分地址空间给I/O设备，把I/O接口中的端口当作存储器单元一样进行访问，不设置专门的I/O指令。

**优点：**

指令类型多，功能齐全。

**缺点：**

端口占用了存储器的地址空间，使存储器容量减小，另外指令长度比专门I/O指令要长，因而执行速度较慢。





## 8.2.4 独立编址方式的I/O端口访问

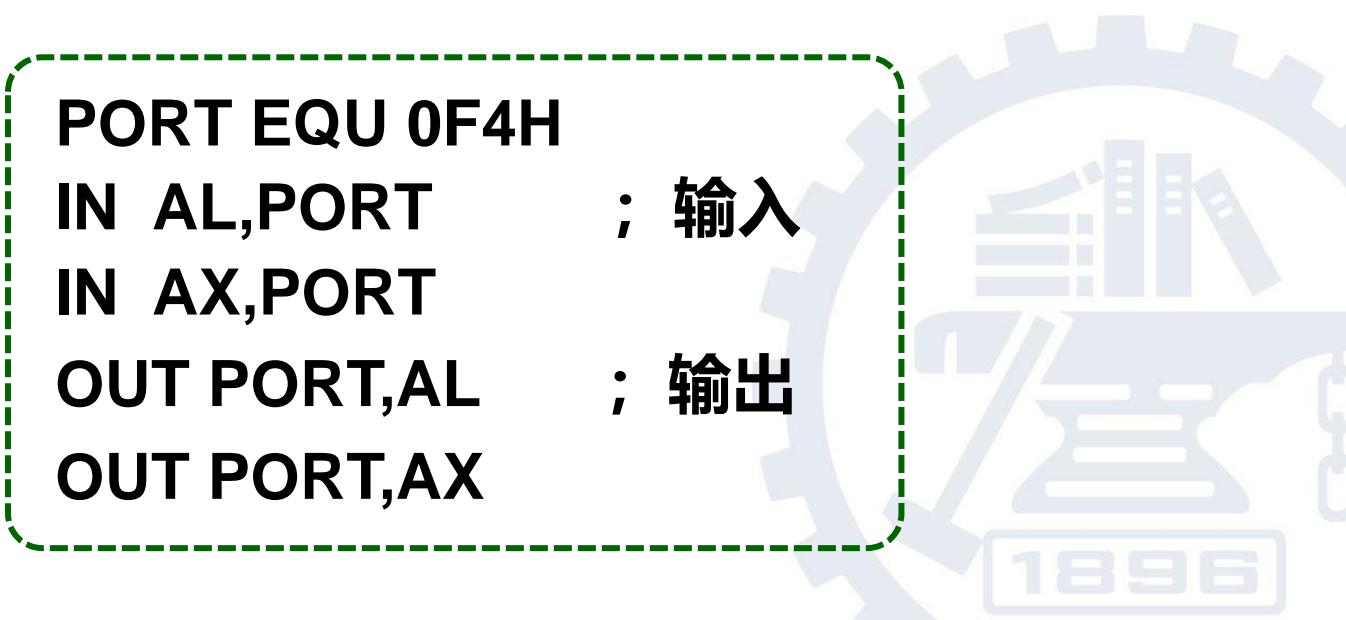
### 1、I/O指令：IN和OUT

**IN指令**从I/O端口向AL(AX)输入字节(字)

**OUT指令**从AL(AX)向I/O端口输出字节(字)

指令格式为：

```
PORT EQU 0F4H
IN AL,PORT      ; 输入
IN AX,PORT
OUT PORT,AL     ; 输出
OUT PORT,AX
```





**通常所说的对端口的访问就是CPU对端口的读/写操作。  
即指I/O端口与CPU的累加器之间的数据传送，并不涉及数据是否传送到存储器的问题。**

**例如：输入时**

<b>MOV DX,300H</b>	<b>;I/O端口</b>
<b>IN AL,DX</b>	<b>;从端口读数据到AL</b>
<b>MOV [DI],AL</b>	<b>;将数据从AL→存储器</b>

**例如：输出时**

<b>MOV DX,301H</b>	<b>;I/O端口</b>
<b>MOV AL,[SI]</b>	<b>;从内存取数到AL</b>
<b>OUT DX,AL</b>	<b>;数据从AL→端口</b>



## 2、I/O端口寻址方式

- **直接寻址**
  - I/O指令直接提供8位I/O地址
  - 只能寻址256个I/O地址 (00 ~ FFH)
  - 用i8表示I/O地址, 表达形式上与立即数一样
- **DX间接寻址**
  - 用DX寄存器保存访问的I/O地址
  - 可寻址全部I/O地址 (0000 ~ FFFFH)
  - 直接书写成DX, 表示I/O地址



例如：输入时

**IN AX,0E0H ;直接寻址**

**MOV DX,300H ;端口300H**

**IN AX,DX ;从端口300H读数据到AX**

例如：输出时

**OUT 0E0H, AX ;直接寻址**

**MOV DX,300H**

**OUT DX,AX ;间接寻址**





### 3、独立编址方式的端口操作

#### (1) 端口地址范围：最多寻址多少个I/O端口

直接寻址：端口地址8位，可寻址256个端口

间接寻址：端口地址16位，可寻址 $2^{16}=64K$ 个端口

例如：MOV DX, XXXXH

IN AL, DX ; 间接寻址

IN AX, 60H ; 直接寻址

#### (2) 传送数据的宽度：通过累加器所传输的数据位数

IN AL, DX ; 输入8位数据

IN AX, DX ; 输入16位数据

原则：传输字节数据时使用AL；传输字数据时使用AX

注意：与指令中的累加器有关，与端口地址范围无关。



## 4. I/O指令与I/O读写控制信号的关系

- 程序执行IN指令时会在系统总线上产生有效的IOR信号（低电平有效），完成读(输入)操作；
- 执行OUT指令时会在系统总线上产生有效的IOW信号（低电平有效），完成写(输出)操作



## 8.3 I/O端口地址分配

I/O地址划分成两部分：

0000H-03FFH留给微机系统和ISA总线；

0400H-0FFFFH给用户应用、主板功能和PCI总线使用。

前者作为专用端口分配给相应的设备，后者由PNP即插即用机制和操作系统动态分配和管理。



## 8.3 I/O端口地址分配及选用的原则

### 8.3.1 实模式下I/O地址的分配

- PC微机把I/O空间分成两部分：系统板上的可编程接口芯片端口地址和常规外设接口控制卡端口地址。
- PC微机采用独立编址方式，I/O地址线有16根，对应的I/O端口编址可达64K。而IBM只用了低10位地址线（即A<sub>0</sub>-A<sub>9</sub>），所以I/O端口的地址范围是：0000H-03FFH，总共只有1024个端口。



## 表8.1 系统板上芯片的端口地址

I/O芯片名称	端口地址
DMA控制器1	000-01FH
DMA控制器2	0C0-0DFH
DMA页面寄存器	080-09FH
中断控制器1	020-03FH
中断控制器2	0A0-0BFH
定时器	040-05FH
并行接口芯片 (键盘接口)	060-06FH
RT/CMOS RAM	070-07FH
协处理器	0F0-0FFH



## 表8.2 系统外设接口控制卡端口地址

I/O接口名称	端口地址
游戏控制卡	200 ~ 20FH
并行口控制卡1	370 ~ 37FH
并行口控制卡2	270 ~ 27FH
串行口控制卡1	3F8 ~ 3FFH
串行口控制卡2	2F0 ~ 2FFH
<b>原型插件板 (用户可用)</b>	<b>300 ~ 31FH</b>
同步通信卡1	3A0 ~ 3AFH
同步通信卡2	380 ~ 38FH
单显MDA	3B0 ~ 3BFH
彩显CGA	3D0 ~ 3DFH
彩显EGG/VGA	3C0 ~ 3CFH
硬驱控制卡	1F0 ~ 1FFH
软驱控制卡	3F0 ~ 3F7H
PC网卡	360 ~ 36FH



### 8.3.2 现代微机I/O地址的分配

Windows操作系统具有即插即用的资源配置机制，因此I/O端口地址的分配是动态的

### 8.3.3 I/O端口地址选用的原则

- ① 凡是被系统配置占用了的地址一律不能使用
- ② 未被占用的地址，用户可以使用，但对计算机厂家申明保留的地址，不要使用。
- ③ 用户可使用的地址为：300H-31FH

可通过“附件”→“系统信息”→“硬件资源”→“I/O”选项来查看I/O端口的地址分配状况



## 8.4 I/O端口地址译码

### ● 基本概念

- ▲ 当执行I/O指令时，CPU首先在总线上发出要访问的端口地址和必要的控制信号，然后通过一个转换电路将这些信号转换为相应的I/O端口选通信号。
- ▲ 这个转换过程就是I/O端口地址译码，完成这个过程的转换电路称为I/O端口地址译码电路。

### ● 译码信号

- ▲ 译码的实质：对地址线和控制信号进行逻辑组合，产生相应的选通信号 $\overline{CS}$ ，低电平有效。
- ▲ 参与译码的主要信号：地址信号线 $A_9 \sim A_0$ ，控制信号 $\overline{IOR}$ 、 $\overline{IOW}$ 、AEN（非DMA传送）。



## 8.4 I/O端口地址译码

### ● 原理

把地址线分为高低两部分，分别用于片间寻址和片内寻址。

- ▲ **片间寻址**：高位地址线与CPU的控制信号相结合，经译码电路产生I/O接口芯片的片选信号 $\overline{CS}$ ，实现片间寻址。
  - 通常都由接口芯片**外部**的译码电路来完成。
- ▲ **片内寻址**：低位地址线不参加片间寻址译码，而是直接连接到I/O接口芯片，实现芯片内部的片内端口寻址。
  - 通常由接口芯片**内部**的地址译码电路来完成。
  - 用于片内寻址的**低位地址线条数**取决于接口内端口的数目N。  
一般地，低位地址线数 =  $\lceil \log_2 N \rceil$ 。



## 8.4 I/O端口地址译码

### 8.4.1 I/O地址译码方法

#### (1) 全译码方法

地址线A0-A9全部参加译码，10根地址线的译码结果只是产生一个片选信号 $\overline{CS}$ ，一般用在单端口的I/O设备上。

#### (2) 部分译码方法

只有高位地址线参加译码，产生片选信号 $\overline{CS}$ ；而低位地址线直接接芯片，作为片内寻址。一般用于连接多端口I/O设备上。

#### (3) 地址开关译码方法

在部分译码方法的基础上，加上地址开关（如PC主板上的超频开关）来改变端口地址。



## 8.4.2 I/O地址译码电路的输入输出信号

### 1、译码电路的输入信号

I/O地址译码电路不仅仅与地址信号有关，而且与控制信号有关。例如：

**AEN**信号：当**AEN = 1**时，为DMA方式

**IOR**和**IOW**信号：控制对端口的读/写等

在32位机中，I/O读写信号是**IORC**和**IOWC**

### 2、译码电路的输出信号

产生一个片选信号**CS**。**CS = 0**有效，产生唯一低电平信号，选中芯片内的寄存器端口；否则芯片未被选中



## 8.5 I/O端口地址译码电路设计

### I/O端口地址译码电路设计的几个问题

- **遵循I/O端口地址的选用原则**

选用合适的地址段，避免发生冲突。

- **正确选用地址译码方法**

一般情况下，单端口地址译码采用全译码法，多端口译码采用部分译码法。

- **灵活设计I/O地址译码电路**

产生同样端口地址的译码电路不是唯一的，电路的组成可以采用不同的元器件，参加译码的地址信号和控制信号之间的逻辑组合也可以不同。



# I/O端口地址的固定式译码

## ● 特 点：

接口中的端口地址固定，不能修改。一般接口卡中大多采用该形式。

## ● 单端口地址译码

- 接口中只有一个端口地址。
- 多采用与、或、非门等基本门电路来实现。

## ● 多端口地址译码

- 系统中有多个接口芯片，或有多个端口（连续范围）
- 多采用集成的译码器来实现，如74LS138, 74LS154等



## 1. 固定式单端口地址译码电路设计

例：设计一个地址为2FBH的地址译码电路

分析：这是一个单端口的地址译码电路，采用全译码方法

表8.4 单端口地址2F8H的地址线取值

地址线	0	0	$A_9$	$A_8$	$A_7$	$A_6$	$A_5$	$A_4$	$A_3$	$A_2$	$A_1$	$A_0$
二进制	0	0	1	0	1	1	1	1	1	0	1	1
十六进制	2				F				B			



# 1. 固定式单端口地址译码电路设计

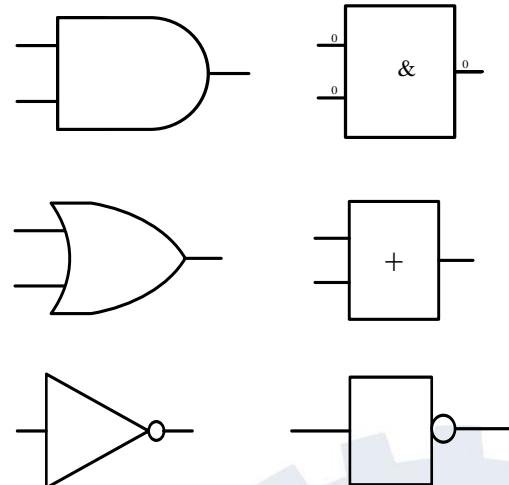
## ● 门电路译码

### ▲ 常用的逻辑门符号

- 与门
- 或门
- 非门

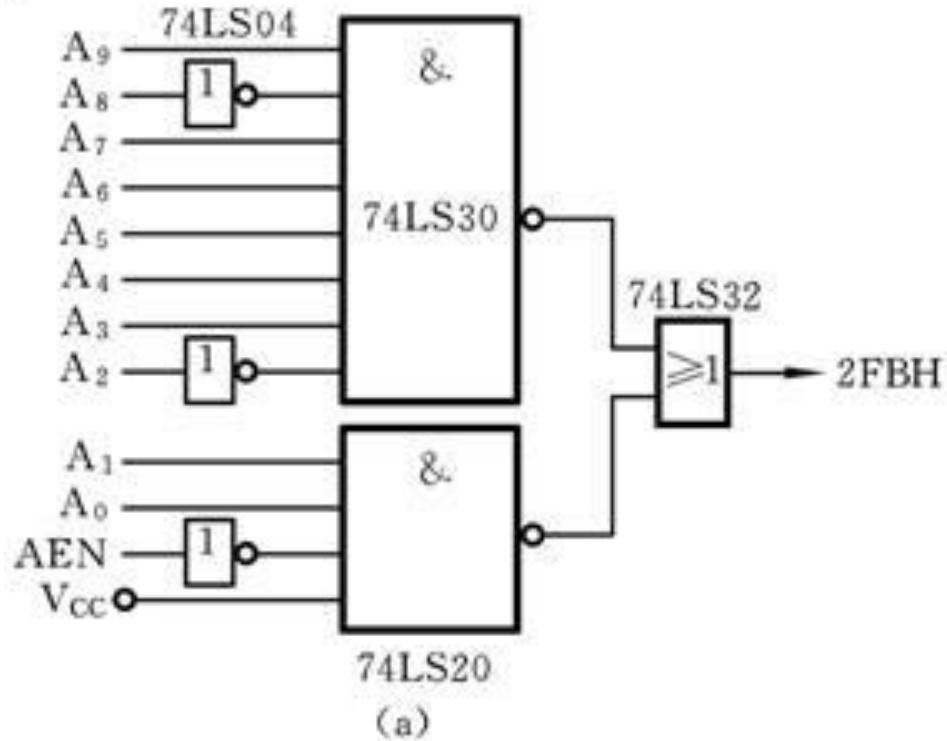
### ▲ 常用的门电路集成芯片

- 74LS04: 反相器 (6门集成)
- 74LS08: 与门 (4门集成)
- 74LS20: 4输入与非门 (2门集成)
- 74LS30: 8输入与非门 (1门集成)
- 74LS32: 2输入或门 (4门集成)

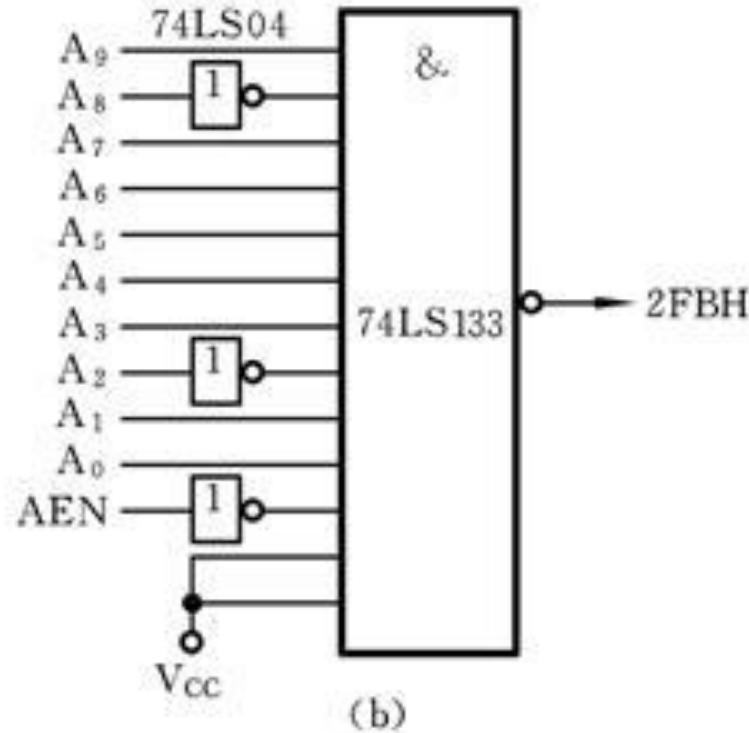




1



(a)



(b)

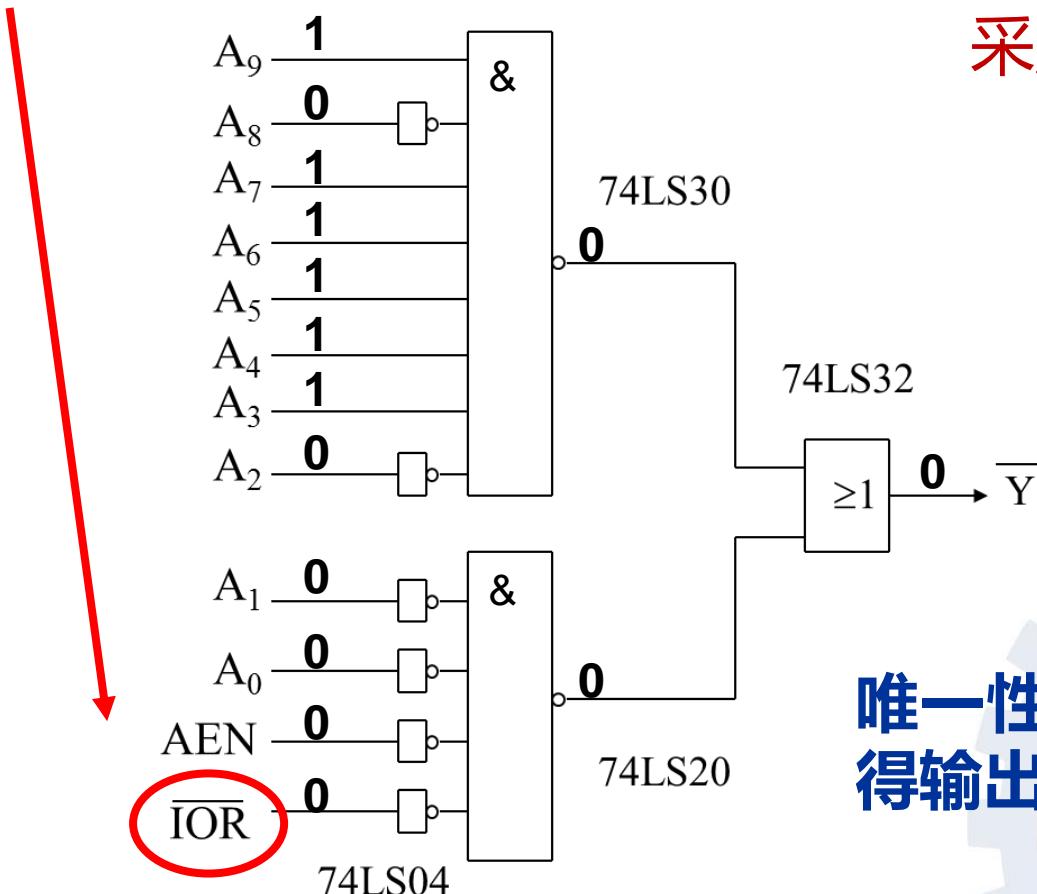
$$A_9 \cdot \bar{A}_8 \cdot \bar{A}_7 \cdot \bar{A}_6 \cdot \bar{A}_5 \cdot \bar{A}_4 \cdot \bar{A}_3 \cdot \bar{A}_2 \cdot \bar{A}_1 \cdot A_0 =$$

$$A_9 \cdot \bar{A}_8 \cdot \bar{A}_7 \cdot \bar{A}_6 \cdot \bar{A}_5 \cdot \bar{A}_4 \cdot \bar{A}_3 \cdot \bar{A}_2 \cdot \bar{A}_1 \cdot \bar{A}_0$$

或

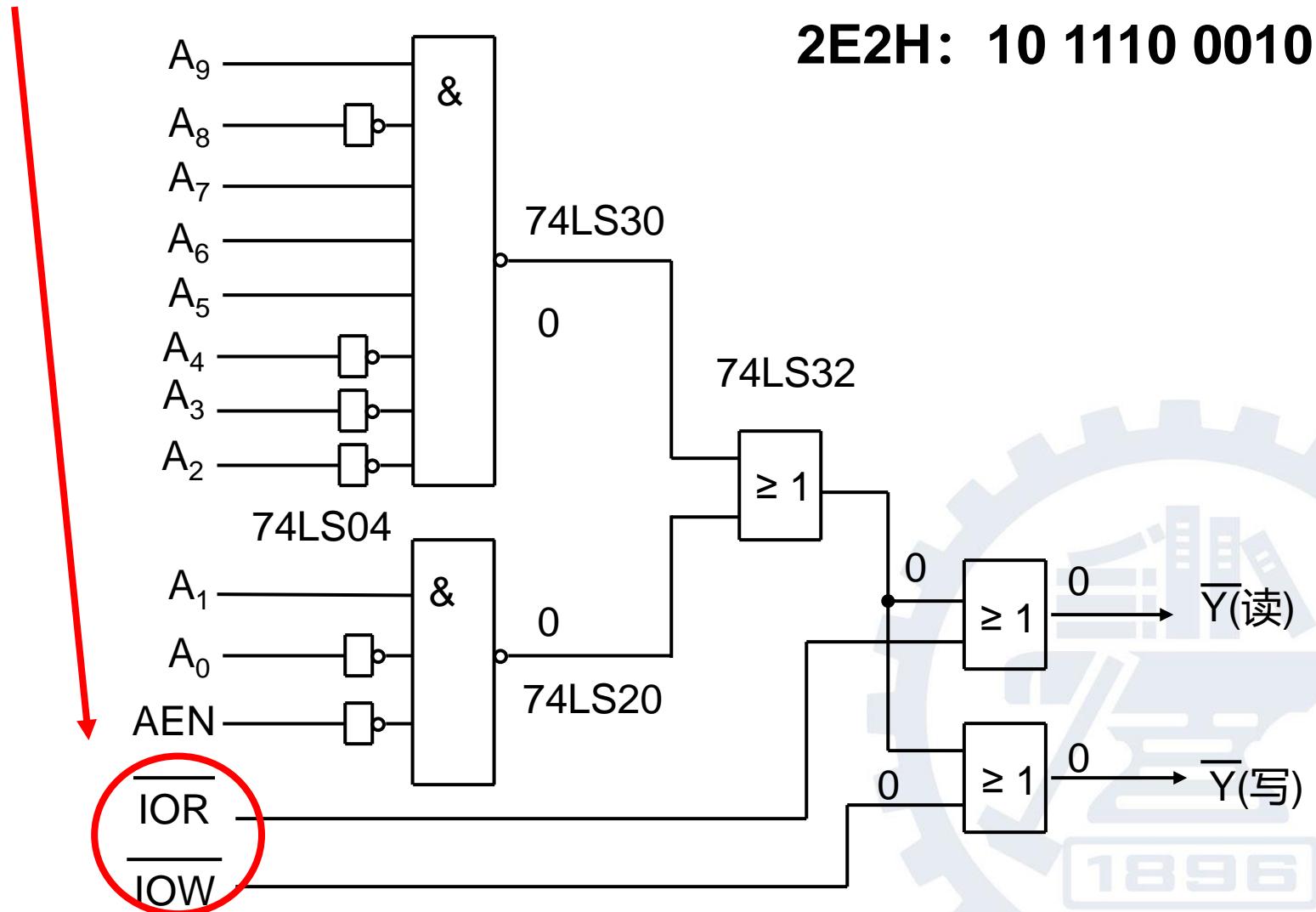


## 只读I/O端口地址 (2F8H) 的译码：





## 可读可写的I/O端口地址 (2E2H) 译码：





## 2. 固定式多端口I/O地址译码电路设计

### ● 译码器译码

▲ 多端口时 (有多个接口芯片, 或接口芯片有多个端口)  
常采用专用**译码器**译码。

### ▲ 常用的译码器

- 74LS138: 3-8译码器
- 74LS139: 双2-4译码器
- 74LS154: 4-16译码器





例：使用74LS138（3-8译码器）设计一个系统板上的地址译码电路，每个接口芯片内部的端口数目为32个，每个芯片的基址为00H、20H、40H、60H、80H、A0H、C0H、E0H。（只有A9-A0参与译码，其余地址线为0）。

分析：系统板上的I/O地址分配在000~OFFH范围内，故只使用低8位地址，则A9和A8应赋0值。为了实现每个接口芯片内部拥有32个端口，只要留出5根低位地址线不参加译码，即地址线低5位作为芯片内部端口译码 $32=2^5$ ，其余的高位地址线作为74LS138输入线参加译码，或作为控制线控制译码是否有效即可。其控制/译码线与地址线的分配如下：

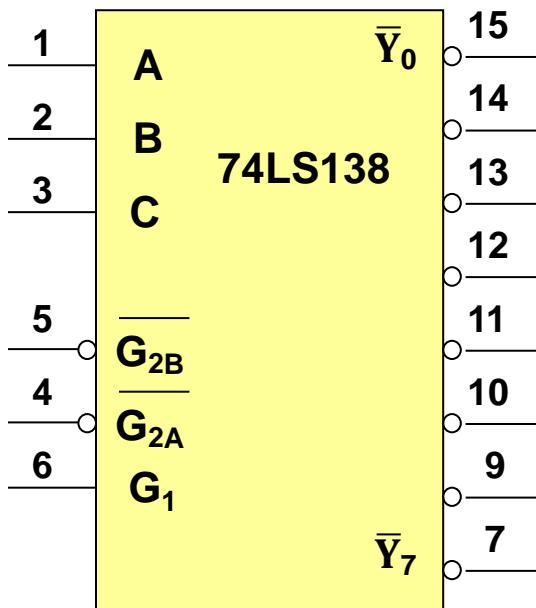
地址线	0 0 A <sub>9</sub> A <sub>8</sub>	A <sub>7</sub> A <sub>6</sub> A <sub>5</sub>	A <sub>4</sub> A <sub>3</sub> A <sub>2</sub> A <sub>1</sub> A <sub>0</sub>
用途	控制	片选	片内端口寻址
十六进制	0H	0~7H	0~1FH



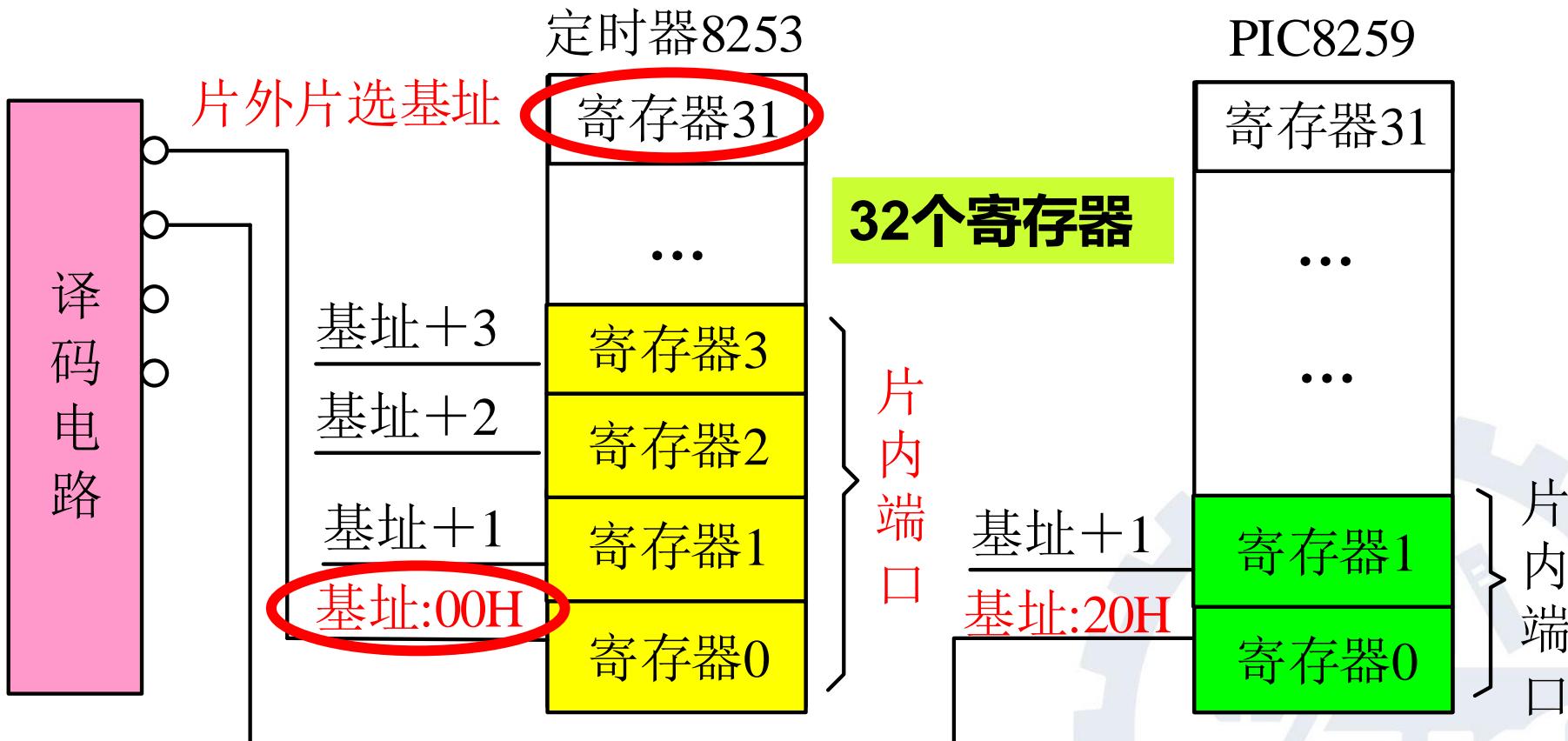
## ● 74LS138译码器工作原理

控制信号线满足时才能译码

输入与输出对应



使能输入			选择输入			Y <sub>0</sub> ~ Y <sub>7</sub> 输出	
G1	/G2A	/G2B	C	B	A		
1	0	0	0	0	0	Y <sub>0</sub> = 0, 其余为 1	
1	0	0	0	0	1	Y <sub>1</sub> = 0, 其余为 1	
1	0	0	0	1	0	Y <sub>2</sub> = 0, 其余为 1	
1	0	0	0	1	1	Y <sub>3</sub> = 0, 其余为 1	
1	0	0	1	0	0	Y <sub>4</sub> = 0, 其余为 1	
1	0	0	1	0	1	Y <sub>5</sub> = 0, 其余为 1	
1	0	0	1	1	0	Y <sub>6</sub> = 0, 其余为 1	
1	0	0	1	1	1	Y <sub>7</sub> = 0, 其余为 1	
0	x	x	x	x	x	全部为 1	
x	1	x	x	x	x	全部为 1	
x	x	1	x	x	x	全部为 1	





3-8 译码器 CBA =  
A7A6A5, 实现片选

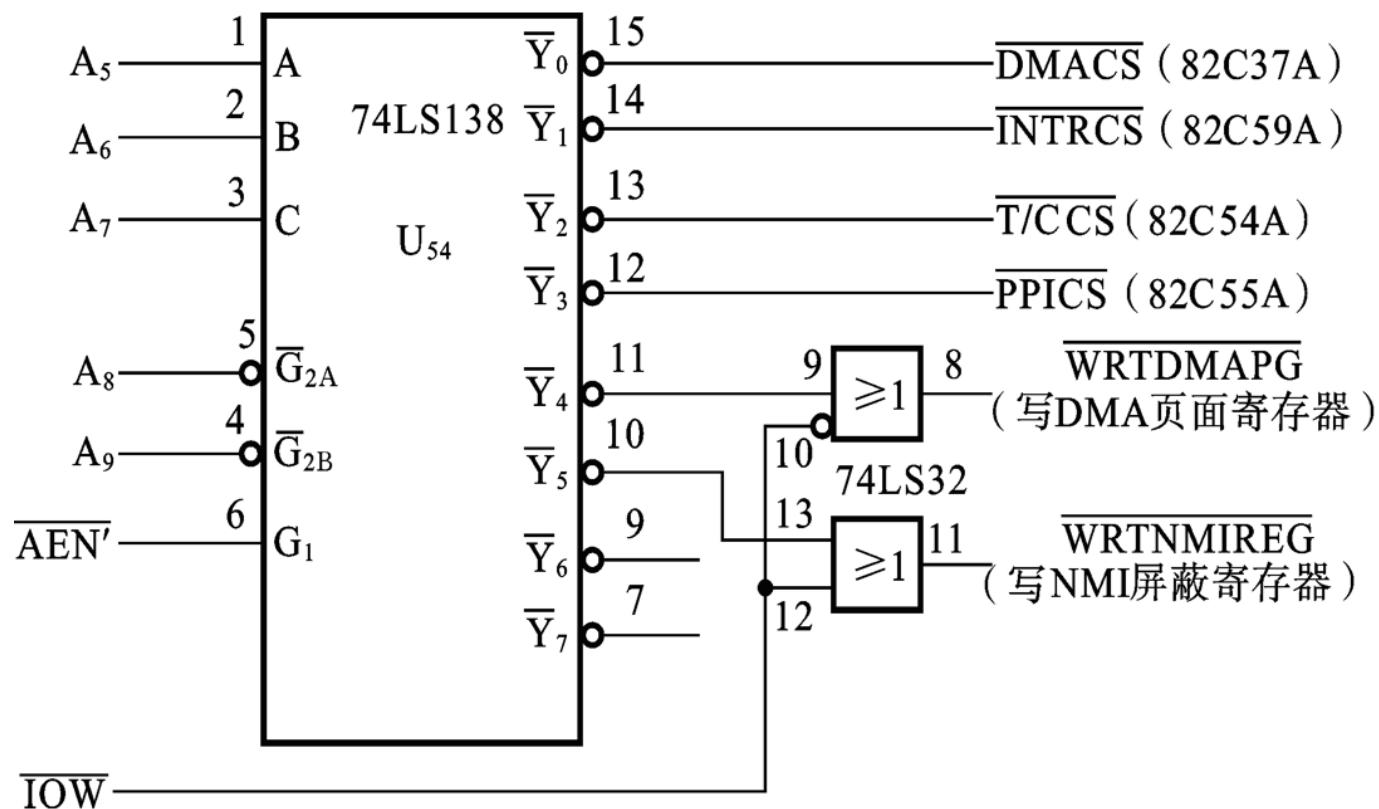
基址：3-8译码器输出地址

A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	
0	0	0	0	0	0	0						00H
0	0	0	0	0	0	1						20H
0	0	0	0	0	1	0						40H
0	0	0	0	0	1	1						60H
0	0	0	0	1	0	0						80H
0	0	0	0	1	0	1						A0H
0	0	0	0	1	1	0						C0H
0	0	0	0	1	1	1						E0H

低位地址线：  
片内地址32个  
对于3-8译码器  
每个基址,  $A0-4 = 0$



- $A_4 \sim A_0$  可用于片内寻址。
  - $A_9 \sim A_8 = 00$ ，可用于控制74LS138的  $/G_{2A}$  和  $/G_{2B}$  端。G端由组合信号  $/AEN'$  控制（非DMA操作，且为I/O操作）。
  - $A_7 \sim A_5$  可用于对8254、8255等进行片选，作为C, B, A输入。





地址线	A <sub>9</sub>	A <sub>8</sub>	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	地址范围
8237	0	0	0	0	0	x	x	x	x	x	000 ~ 01FH
8259	0	0	0	0	1	x	x	x	x	x	020 ~ 03FH
8254	0	0	0	1	0	x	x	x	x	x	040 ~ 05FH
8255	0	0	1	0	0	x	x	x	x	x	060 ~ 07FH

## 结论：

- ▲ 通过倒推分析，每个接口内部有32个端口，各接口芯片的地址范围与题目一致。
- ▲ 译码电路总的地址范围为000~0FFH (后续的自行分析)，同系统板的地址范围一致。



例：包括8个接口芯片，每个接口有4个端口，其地址分配在200H-21FH。

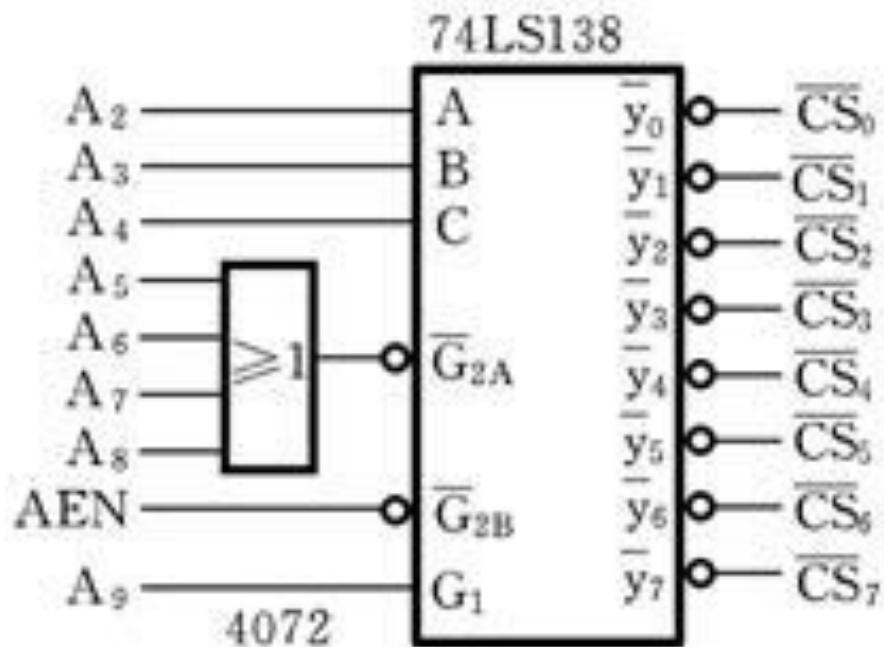
分析：8片选4端口，采用部分译码。

地址线	$A_9$	$A_8$	$A_7$	$A_6$	$A_5$	$A_4$	$A_3$	$A_2$	$A_1$	$A_0$
二进制	1	0	0	0	0	C	B	A	?	?

无需参与译码！



# 地址范围在200H-21FH区间的译码电路





### 3、开关式I/O地址译码电路设计

例：设计某微机实验平台的I/O端口地址译码电路，要求平台上每个接口芯片的内部端口数为4个，并且端口地址可选，其地址范围300H-31FH。

分析：

- ✓ 开关式译码电路可由译码器、地址开关、比较器或异或门几种元器件组成。先分析3个元器件的工作原理，然后根据题目要求进行电路设计。
- ✓ 采用部分译码方法，采用4位开关有16种不同的组合，可以改变16种地址。
- ✓ DIP开关有两种状态需要进行设定，即合(ON)和断(OFF)，本例设置ON=0，OFF=1。



分析：

✓ 74LS85比较器考虑两点：比较的对象、比较的结果

对象：A组4根输入线与地址线连接A9、A8、A7、A6

B组4根输入线与DIP开关连接S3、S2、S1、S0

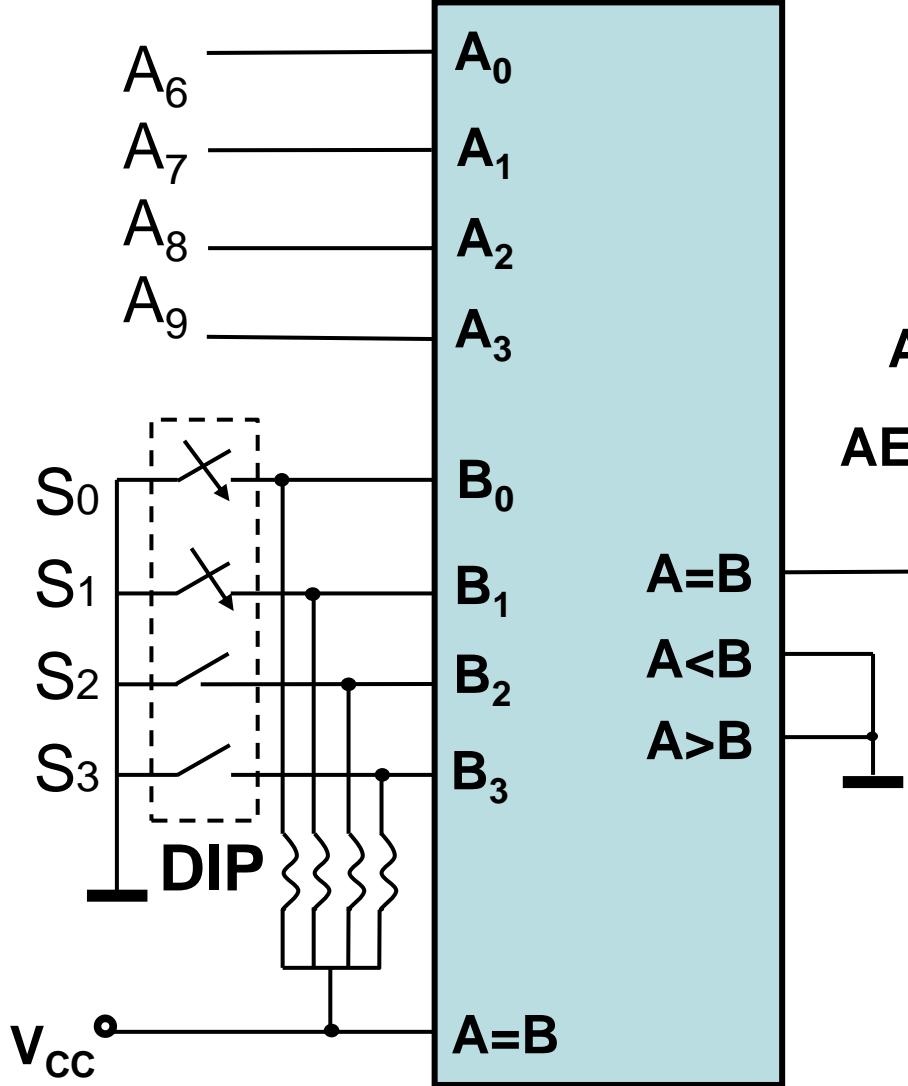
结果：3种  $A > B$ ,  $A < B$ ,  $A = B$

本例取 $A = B$ 的结果，比较器输出高电平，否则为低。

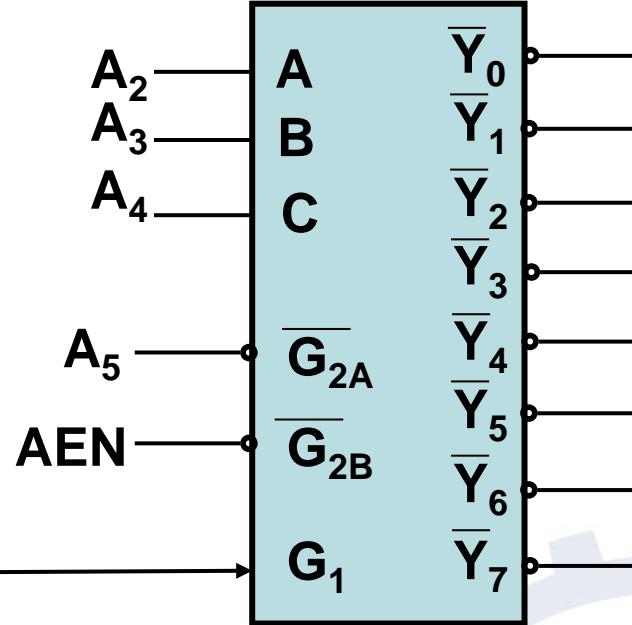
比较器的输出线连接到译码器74LS138的控制线G1上，控制译码器工作，即4位地址线与4位开关状态相等时，译码器才工作。



74LS85



74LS138



1896



## 讨 论：

- 高地址线中  $A_9 \sim A_6$  的值由DIP开关的  $S_3 \sim S_0$  状态决定，上图中  $S_3$  和  $S_4$  断开，  $S_1$  和  $S_0$  合上，故  $A_9 = A_8 = 1$ ，  $A_7 = A_6 = 0$ ，而  $A_5$  接在74LS138的/G<sub>2A</sub>上，故  $A_5 = 0$ 。
- $A_4 A_3 A_2$  三根地址线作为74LS138的输入线，经译码后可以产生8个低电平有效的选择信号/Y<sub>0</sub>~Y<sub>7</sub>，可以作为平台板上接口芯片的选择。
- 剩下的2根低地址线  $A_1$  和  $A_0$  未参加译码，作为寄存器选择，以实现每个接口芯片内部拥有4个端口。完全满足300H~31FH端口地址范围和每个接口芯片内部具有4个端口的设计要求。

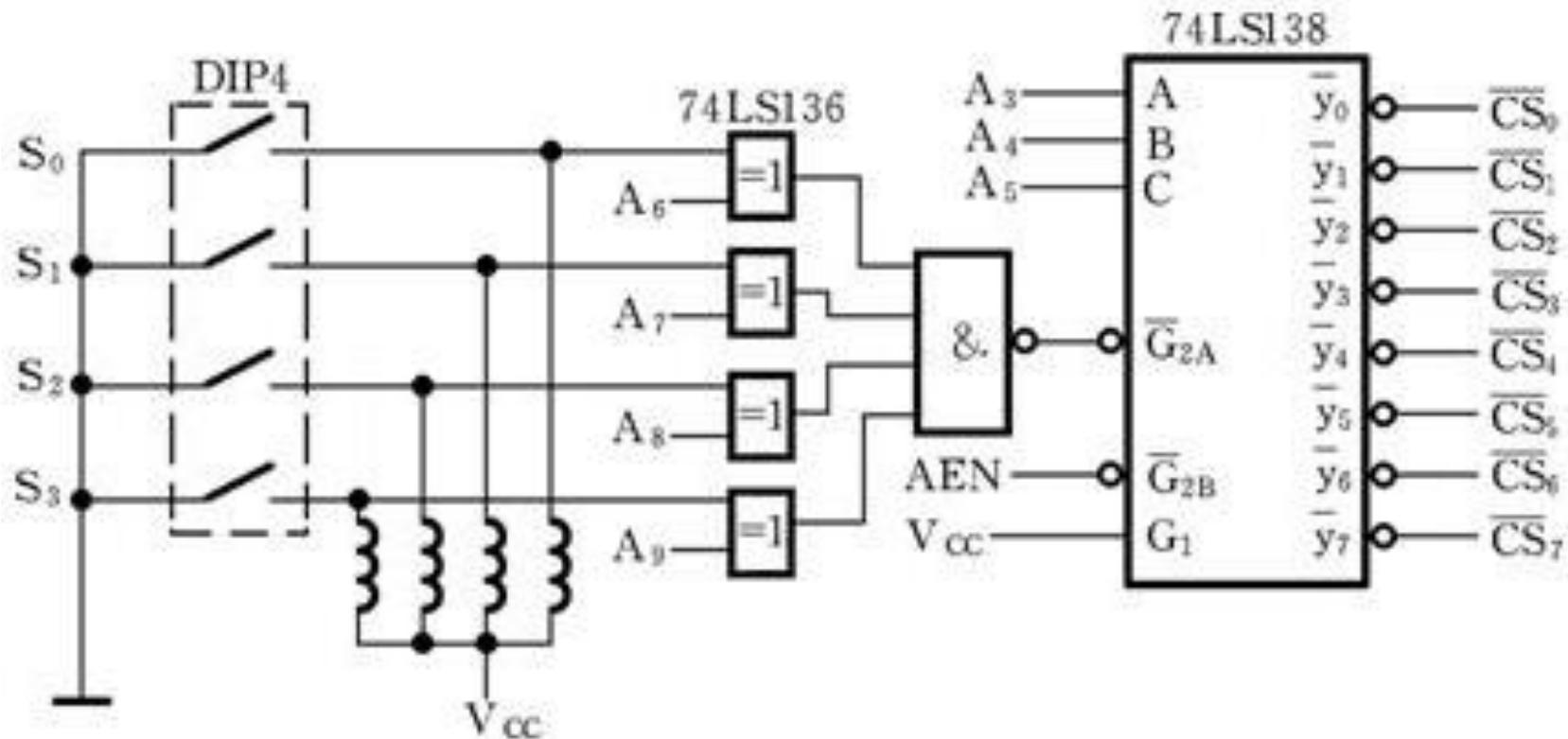


例：某微机系统的I/O空间为000H-3FFH，分成连续的16段可选地址，每个段包括8个接口芯片，每个接口芯片拥有8个端口。

分析：采用部分译码方法，采用4位开关有16种不同的组合，允许16个段可以选择。

表8.6 开关式000H-3FFH地址线取值

地址线	$A_9\ A_8\ A_7\ A_6$				$A_5\ A_4\ A_3$			$A_2\ A_1\ A_0$		
用途	S3	S2	S1	S0	C	B	A	?	?	?





## 译码电路的作用：

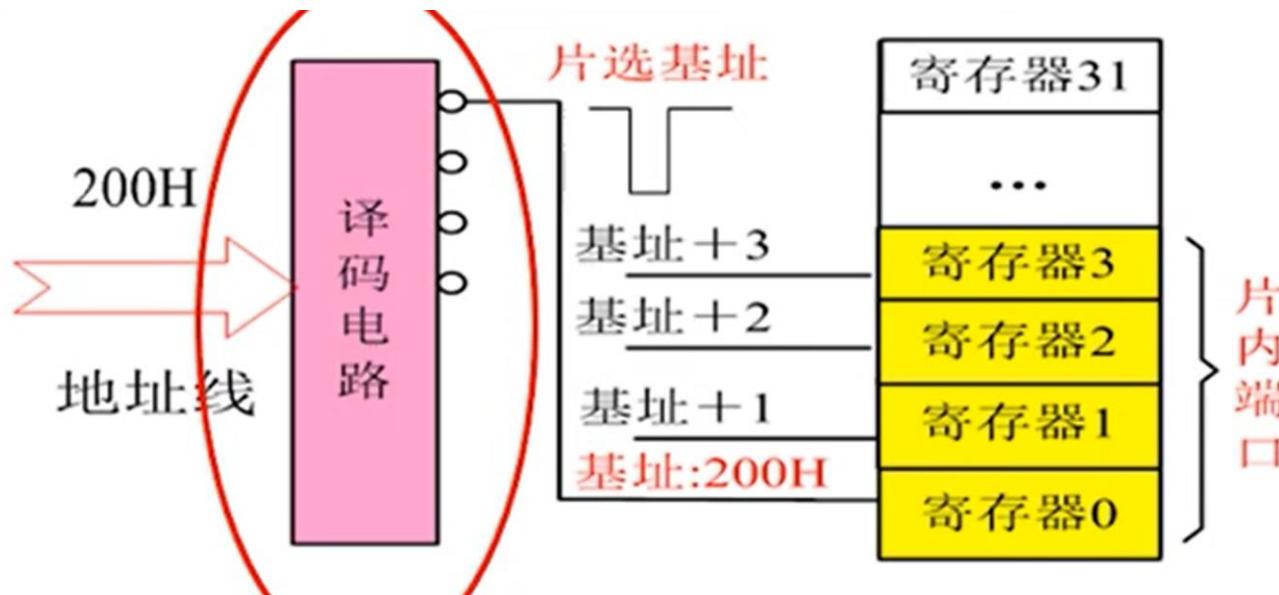
**mov dx,200H**

**mov al,55H**

**out dx,al**

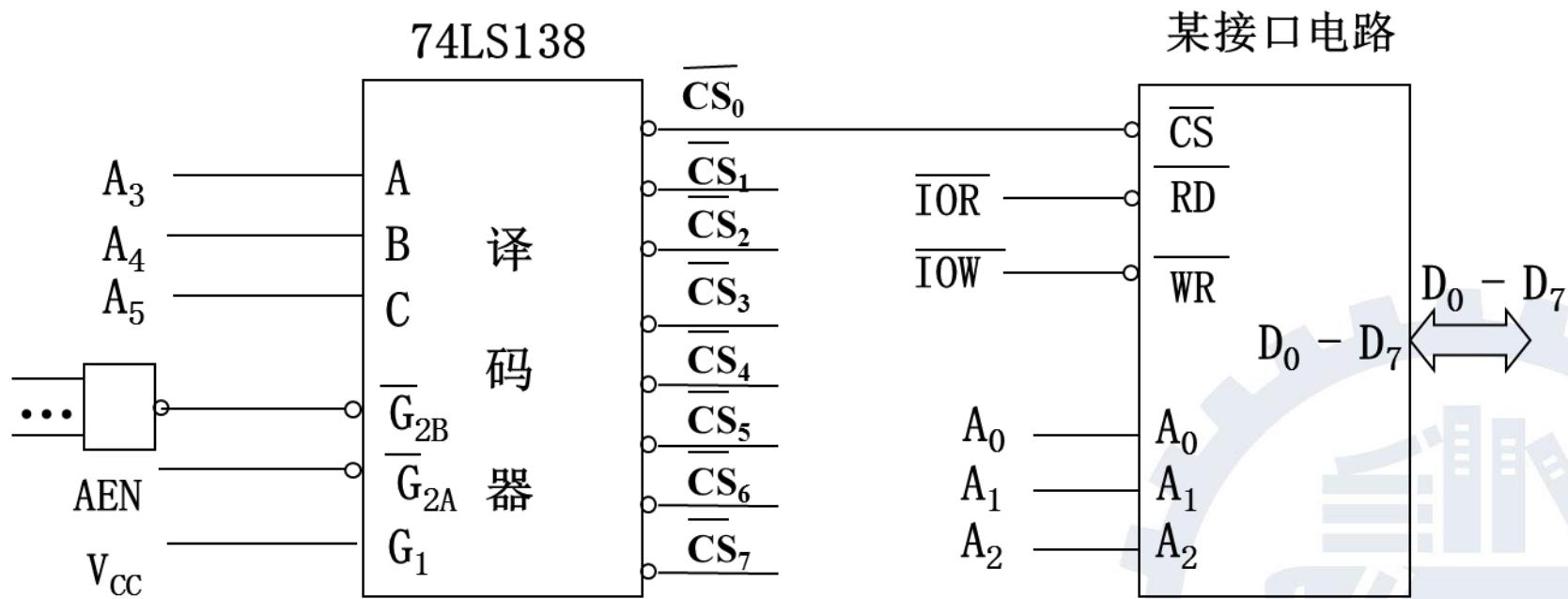
指令的执行过程是：cpu通过200H选中I/O端口，加之IOWR的配合，将数据55H送到该端口寄存器中。

指令out中的地址如何确定？----需要地址译码电路





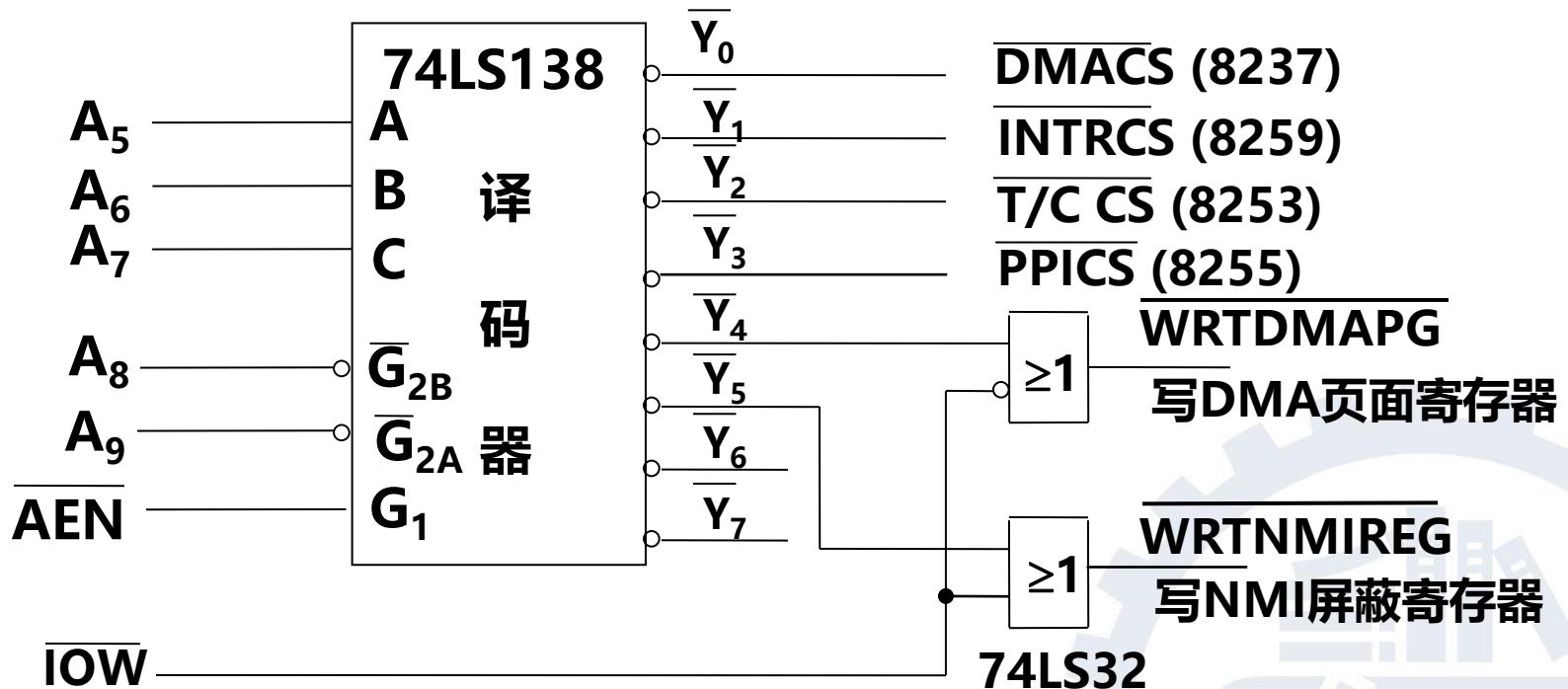
# 一般情况下，译码器和接口电路的连接方法：



地址译码电路、接口电路和CPU的连接方法



## PC机系统板上的译码电路：

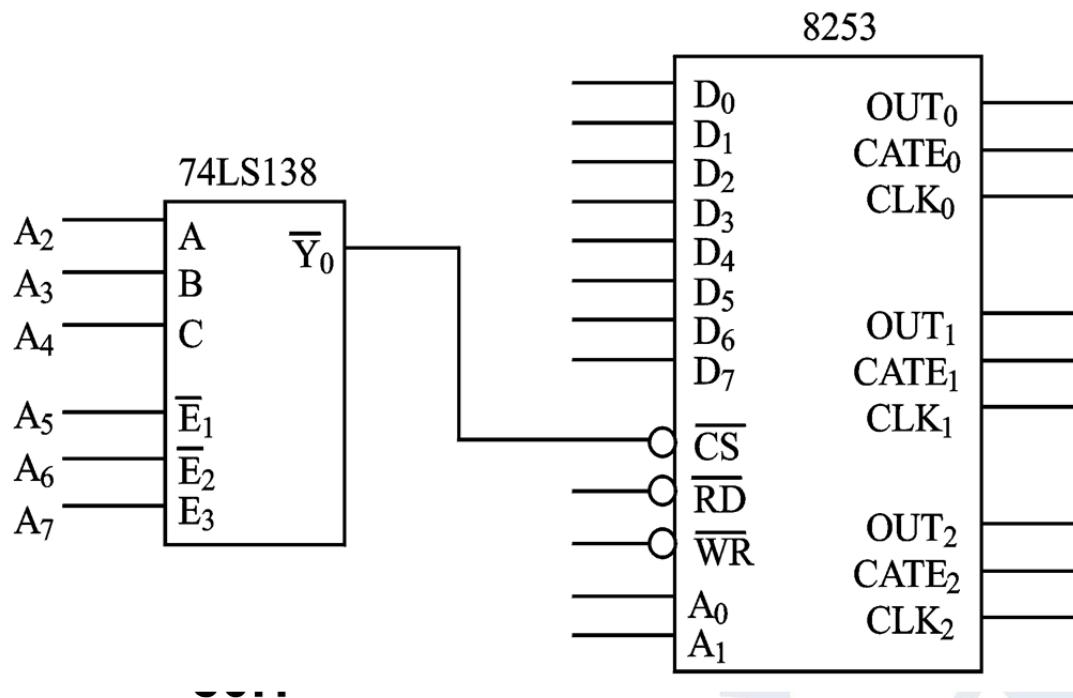


固定式多端口地址译码电路



1. 一个系统的地址译码使用74LS138实现,除图中的地址线外,其余地址线为0电平,请问8253端口地址是多少?

作业





## 2、一个系统的地址译码使用74LS138实现,除图中的地址线外,其余地址线为0电平,请问8255的端口地址是多少?

