

《微机原理与接口》

## 第3章 接口概念和原理

教师：苏曙光

华中科技大学软件学院

## ● 第三章 接口概念

- 1. 接口/端口定义
- 2. 端口访问指令
- 3. 接口/端口地址设计
- 4. 数据传输方式
- 5. 8088输入输出综合实例

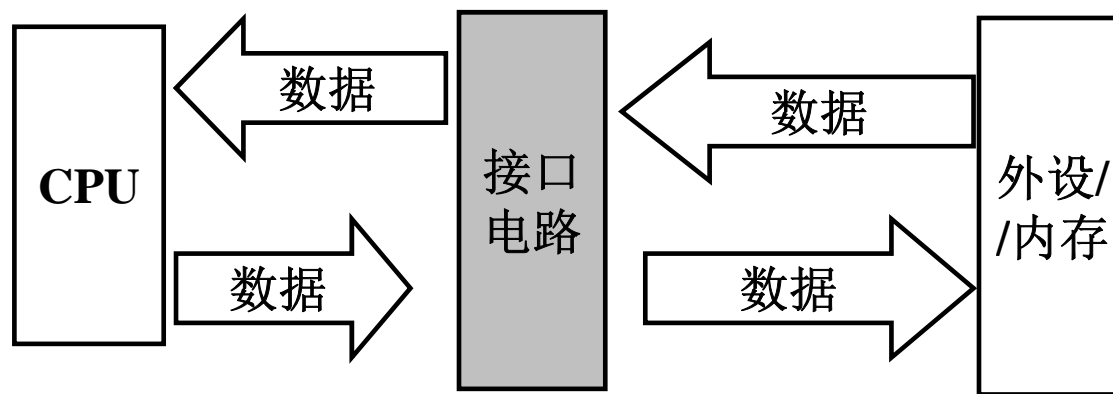


## 第1节 接口/端口的定义

## 第1节 接口定义

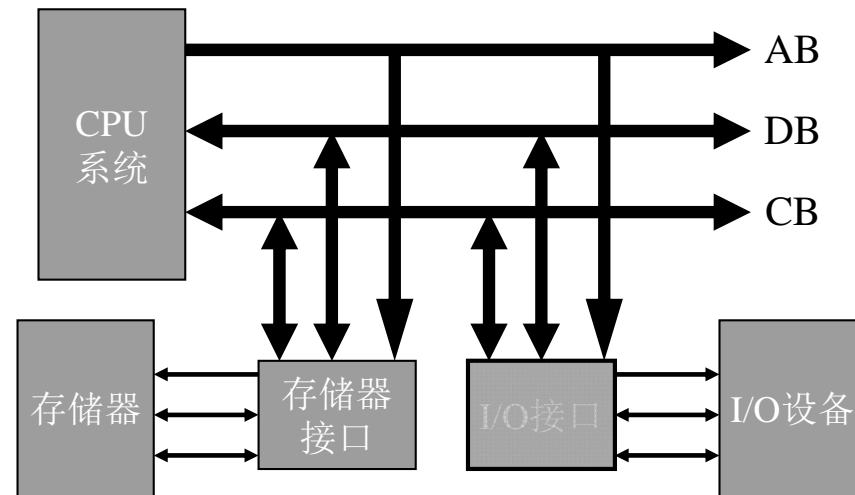
### ● 接口定义

- 接口是一组特殊控制电路，介于CPU与内存、CPU与外设之间。桥梁作用
- 任何两电路或设备间的连接电路都可称接口。



## 1. 接口定义（续）

- 各种外设都必须通过接口才能和CPU（或总线）相连

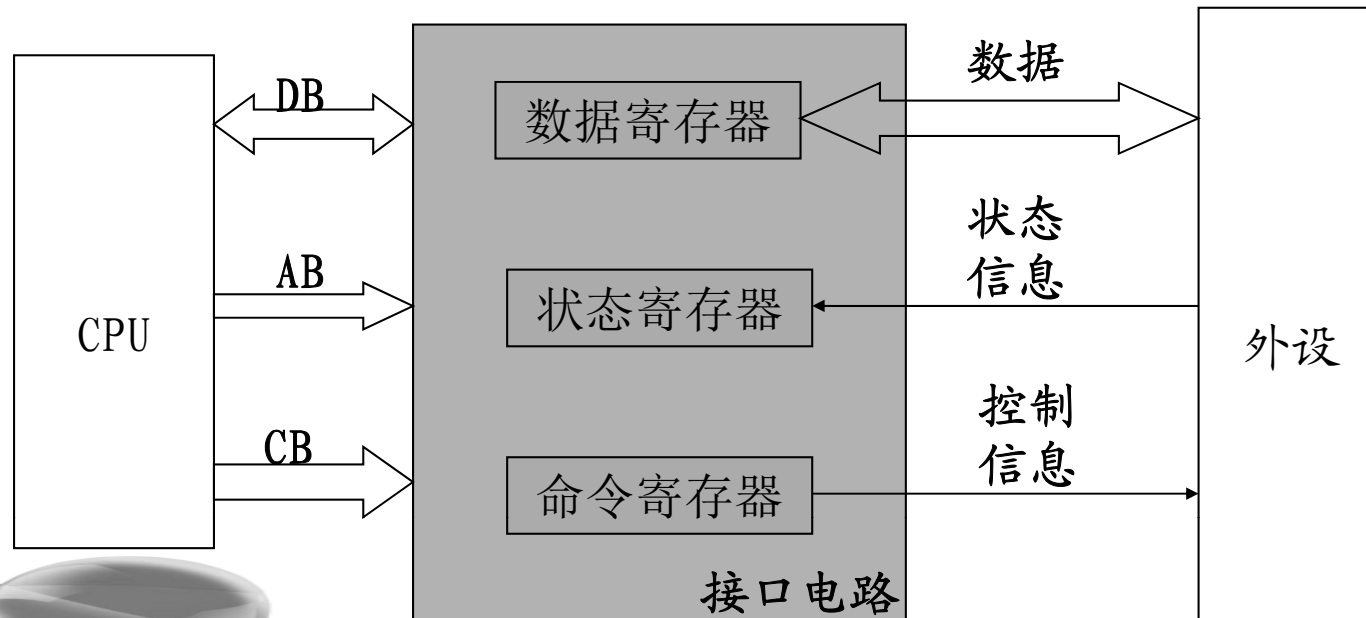


- 寻址：确定设备的地址，区分不同的设备；
- 缓冲：适配外设与CPU的工作速度；
- 转换：适配外设与CPU的信息格式、类型、幅度；
- 时序：外设与CPU的工作时序。

## 1. 接口定义（续）

- 接口电路的组成：多个寄存器构成

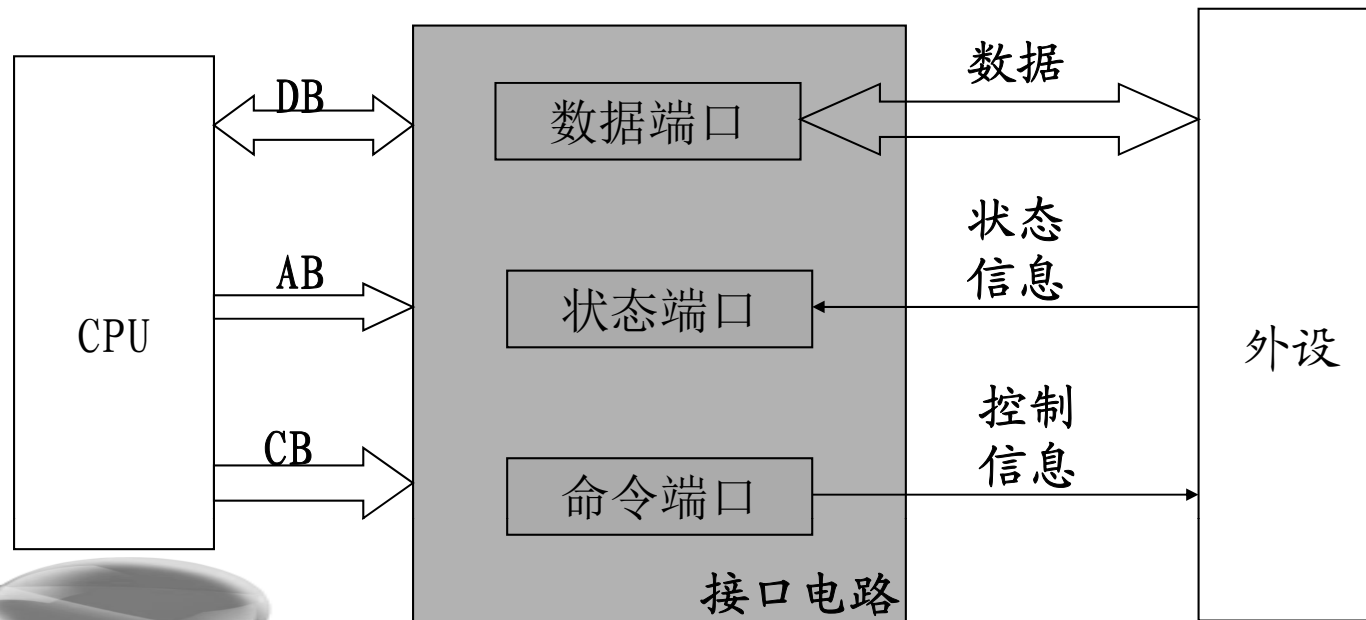
- 数据寄存器，暂存数据
- 状态寄存器，暂存状态。
- 命令寄存器，暂存命令。



## 1. 接口定义（续）

### ● 端口【PORT】

- 端口就是接口中的寄存器
- 端口：可寻址（即端口地址），CPU 与外设信息交换场所
- 接口一般含有一个或多个端口。



## 微机的端口

- 16根I/O线：地址空间 $2^{16} = 64K$
- IBM：A<sub>0</sub>~<sub>9</sub>线有效，地址空间 $2^{10} = 1K$ ：000H ~ 3FFH；
- PC系统IO端口的分配
  - 前256个端口：000h-0FFh，系统外设占用
  - 后768个端口：100h-3FFh，常规外设占用。



# 微机的端口

## ● 微机的端口分配

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	060~06FH
RT/COMS RAM	070~07FH
协处理器	0F0~0FFH

表1 系统I/O芯片的端口地址

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1	2F0~2FFH
原型插件板	300~31FH
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EGA/VGA	3C0 ~3CF
硬驱控制卡	1F0~1FFH
软驱控制卡	3F0~3F7H
PC网卡	360~36FH

表2 扩展卡的端口地址

# 微机的端口

计算机管理

操作(A) 查看(V) 工具(T)

树

- 计算机管理(本地)
  - 系统工具
    - 事件查看器
    - 系统信息
      - 系统摘要
      - 硬件资源
        - 冲突/共享
        - DMA
        - 强制硬件
        - IRQs
        - 内存
      - 组件
      - 软件环境
      - Internet Explorer
      - Office 10 Applications
      - 应用程序
    - 性能日志和警报
    - 共享文件夹
    - 设备管理器
    - 本地用户和组
    - 存储

地址范围	设备	状态
0x0800-0x0805	Motherboard resources	确定
0x0290-0x0297	Motherboard resources	确定
0x0020-0x0021	Programmable interrupt controller	确定
0x00A0-0x00A1	Programmable interrupt controller	确定
0x0080-0x0090	Direct memory access controller	确定
0x0094-0x009F	Direct memory access controller	确定
0x00C0-0x00DF	Direct memory access controller	确定
0x0040-0x0043	System timer	确定
0x0070-0x0073	System CMOS/real time clock	确定
0x0061-0x0061	System speaker	确定
0x00F0-0x00FF	Numeric data processor	确定
0x03F0-0x03F5	Standard floppy disk controller	确定
0x03F7-0x03F7	Standard floppy disk controller	确定
0x03F8-0x03FF	通讯端口 (COM1)	确定
0x02F8-0x02FF	通讯端口 (COM2)	确定
0x0378-0x037F	打印机端口 (LPT1)	确定
0x0060-0x0060	Standard 101/102-Key or Microsoft ...	确定
0x0064-0x0064	Standard 101/102-Key or Microsoft ...	确定
0x0330-0x0331	MPU-401 Compatible MIDI Device	确定
0x0201-0x0201	标准游戏端口	确定



## 第2节 端口访问指令

## 2. 端口的访问

- 端口访问

- 端口地址 (000h ~ 3FFh)
- 端口属性: 只写, 只读, 可读可写
- 端口操作: 写(OUT指令), 读(IN指令)

- 访问指令

- 写 (输出) : OUT
  - ◆ 把寄存器 (AL/AX) 的信息送往指定端口。
- 读 (输入) : IN
  - ◆ 从指定端口取信息送入寄存器 (AL/AX) 。

## 2. 端口的访问（续）

- 读（输入）指令：IN

- 四种形式【PORT指端口地址，DX寄存器】

- ◆ IN AL, PORT

- ◆ IN AX, PORT

- ◆ IN AL, DX

- ◆ IN AX, DX

## 2. 端口的访问（续）

- 写（输出）指令：OUT

- 四种形式【PORT指端口地址，DX寄存器】

- ◆OUT PORT, AL ;数据是单字节；端口地址单字节

- ◆OUT PORT, AX ;数据是双字节；端口地址单字节

- ◆OUT DX, AL ;数据是单字节；端口地址双字节

- ◆OUT DX, AX ;数据是双字节；端口地址双字节

- 例子

- ◆OUT 61H, AL ; 61H为系统板8255的PB端口地址

- ◆MOV DX, 301H ; 301H为扩展板8255的PB端口地址

- OUT DX, AL

说明：所有的I/O指令执行的结果都不影响标志位F。

## 2. 端口的访问（续）

- 端口访问和输入/输出两个概念的区别

- 对端口的访问仅仅指CPU对端口的读/写
- 输入输出一般指以内存RAM为传输目标
- 输入输出都会包括端口的访问。

- 输入的例子：

MOV DX, 300H ; I/O端口

IN AL, DX ; 从端口读数据到AL

MOV [DI], AL ; 将数据从AL输入到存储器



### 第3节接口/端口地址设计



## 端口地址编址方式

- 两种编址方式

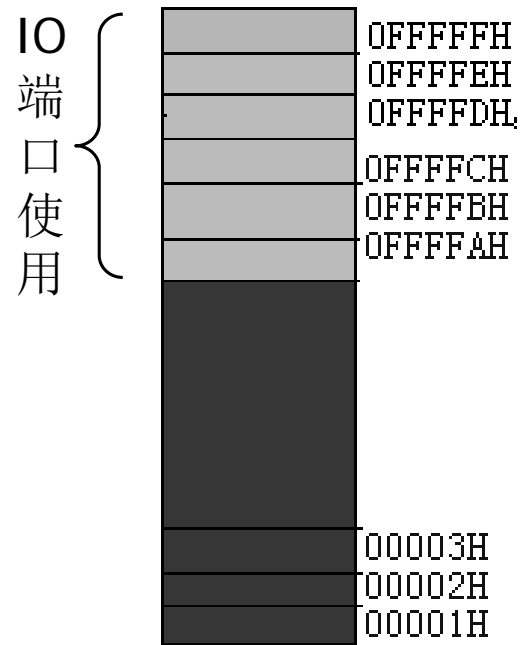
- 独立编址（I/O映射方式）

- ◆ 端口地址单独编址而不和存储器空间合在一起

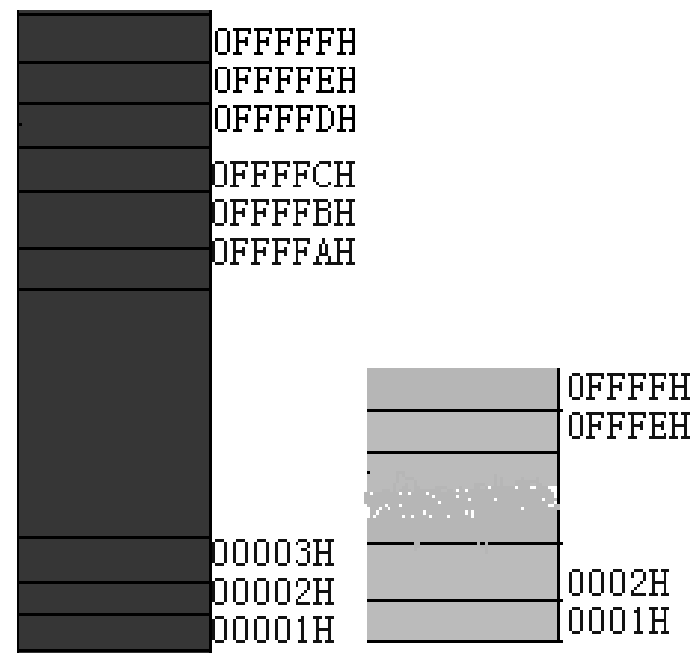
- 统一编址（存储器映射方式）

- ◆ 端口地址和存储器地址统一编址

● 两种编址方式



(a) 统一编址

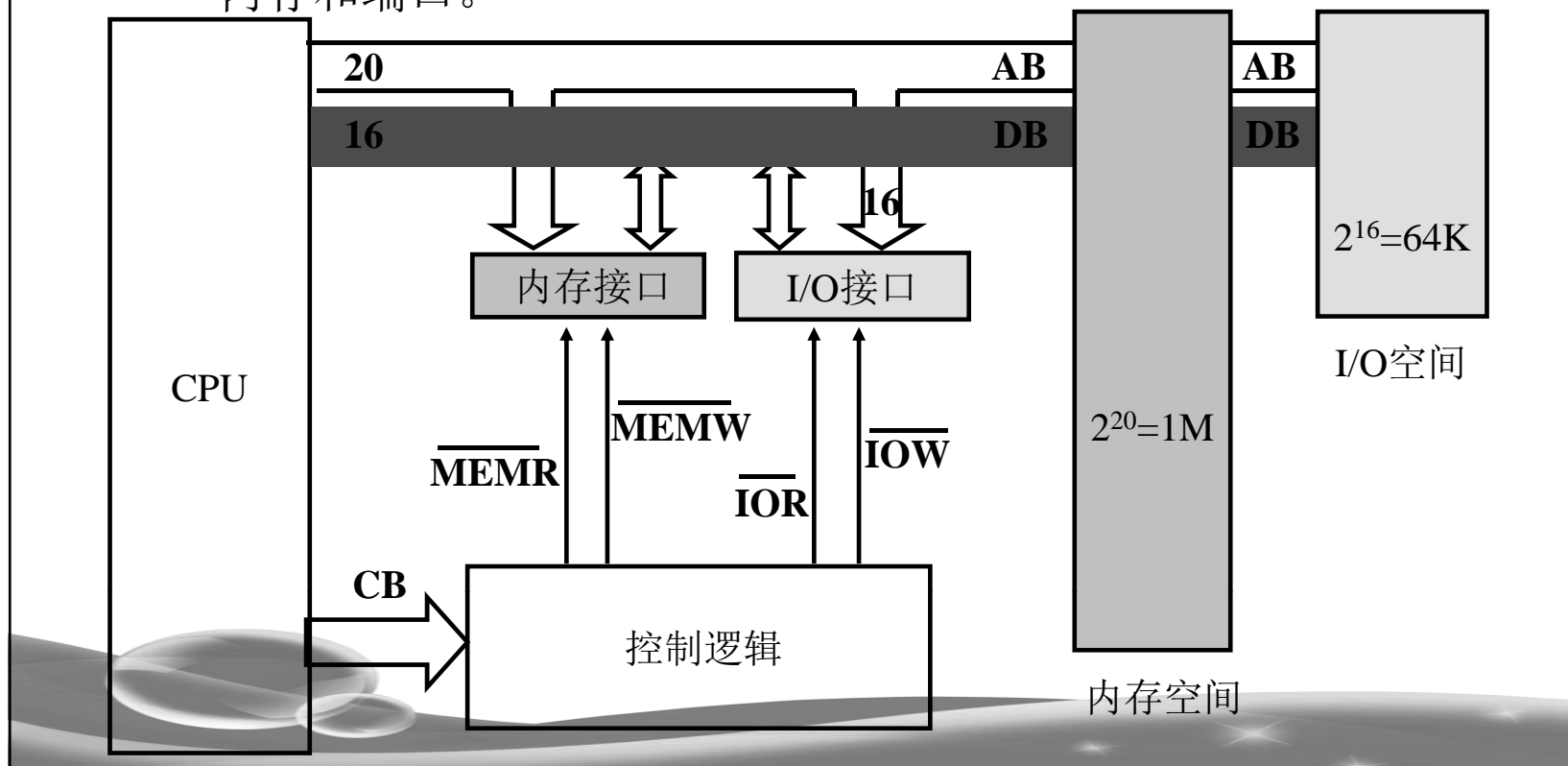


(b) 独立编址

## 端口地址编址方式（续）

- 独立编址的端口访问原理

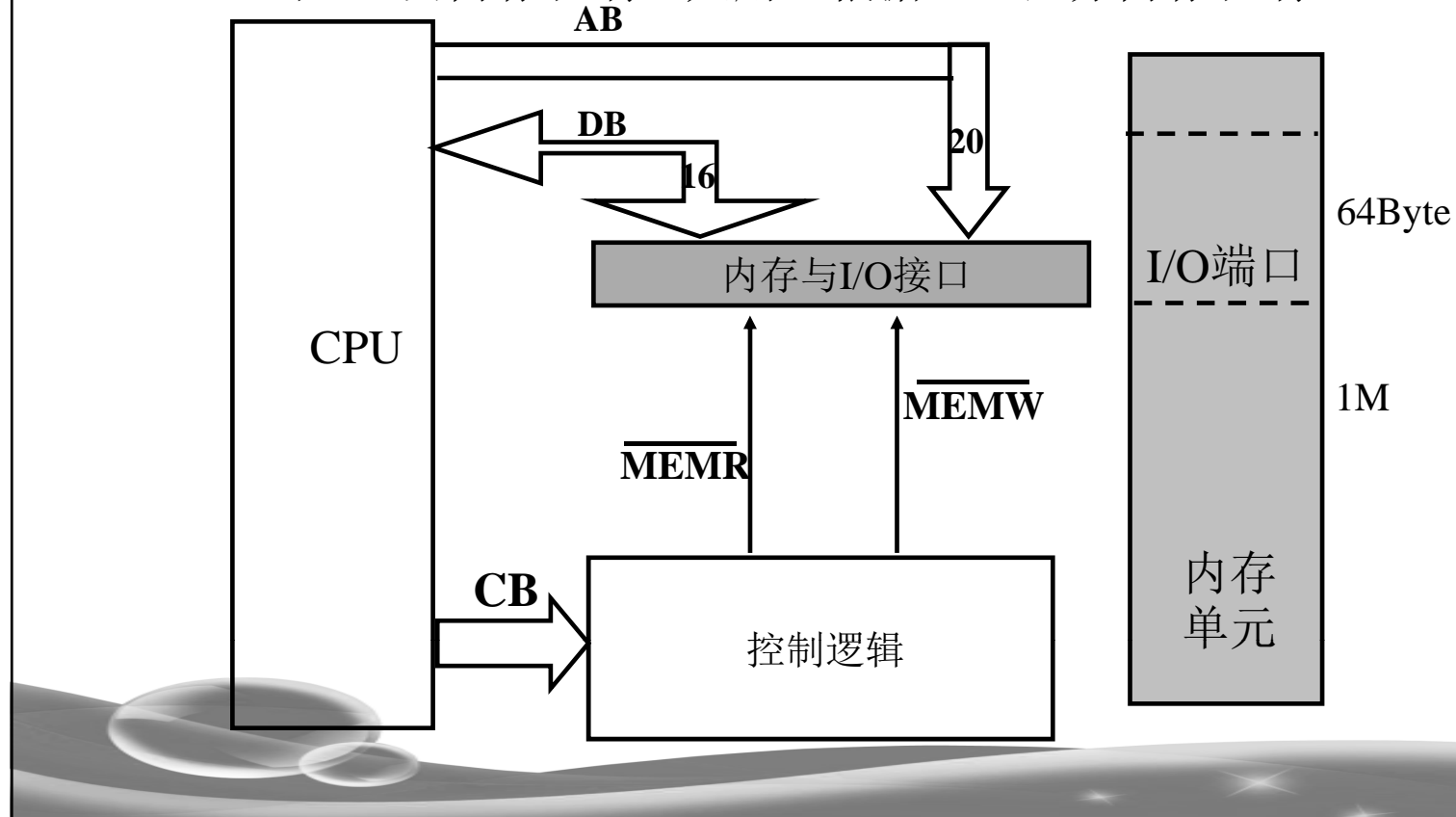
- AB和DB由内存和端口共用。根据指令（IN/OUT|MOV）区分内存和端口。



## 端口地址编址方式（续）

- 统一编址的端口访问原理

- AB和DB由内存和端口共用。根据地址区分内存和端口。

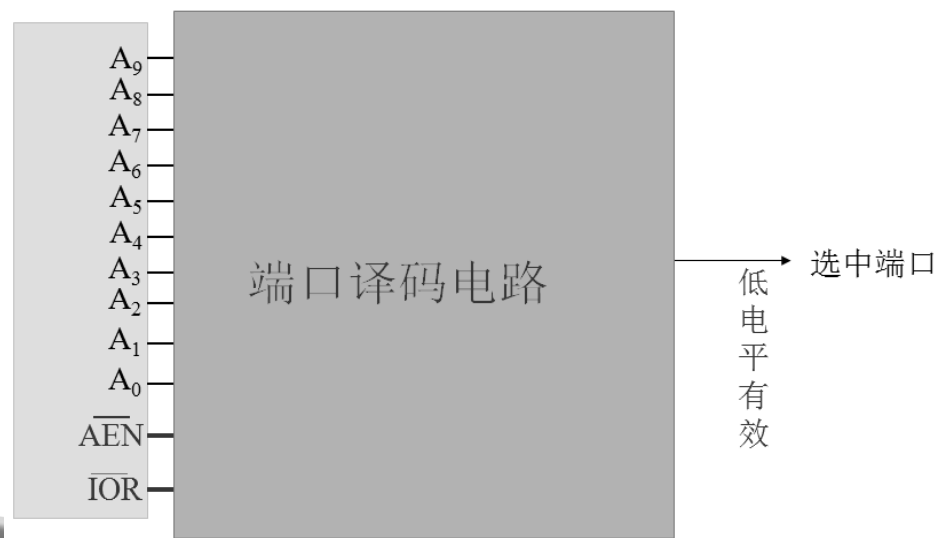


## 端口地址编址方式（续）

- 独立编址系统的端口访问的比较
  - 专用指令：IN指令和OUT指令。
  - 程序中I/O操作和存储器操作清晰可辨，程序可读性强。
  - I/O端口的读、写操作由IOR和IOW来控制
  - 微机和大型计算机通常采用这种方式。

## 端口地址设计

- 概念：给某个端口（接口/设备）设计特定的地址。
  - 即设计端口译码电路。
  - 当AB上给出该端口的地址时，该电路能选中该端口。
- ◆ 低电平有效

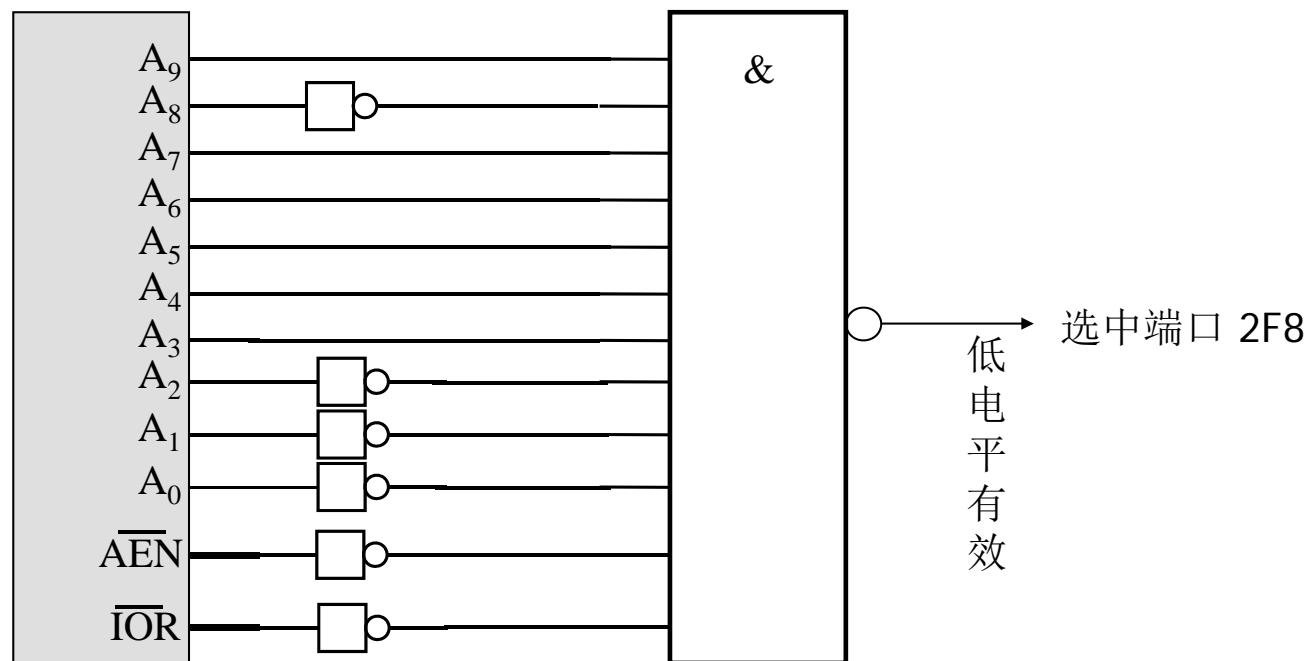


## 端口地址设计

- 设计地址译码电路三个前提
  - 有效I/O地址线10位： $A_9 \sim 0$
  - 端口读写属性（只读/只写/可读可写）
  - 考虑DMA操作：地址允许信号**AEN**（**DMAC**信号）
    - ◆  $AEN=0$ ，即非DMA操作时，端口可以访问；
    - ◆  $AEN=1$ ，即是DMA操作时，端口不能访问；

## 译码接口电路的设计例子

- 使用门电路设计端口地址**2F8H**的只读地址译码电路。





## 接口电路常用基本门和数字芯片

- 常用门电路和符号

- 与门, 非门, 与非门
- 或门

- 常用门芯片

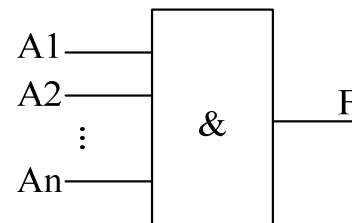
- 74LS00与非门(四2)
- 74LS02或非门(四2)
- 74LS04非门(6)
- 74LS20与非门(4)
- 74LS30与非门(8)
- 74LS133与非门(13)
- 74LS32或门(四2)

- 138译码器

- 74LS138

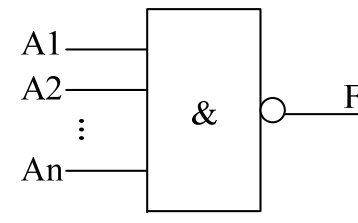
- 锁存器: 74LS373

- 缓冲器: 74LS245



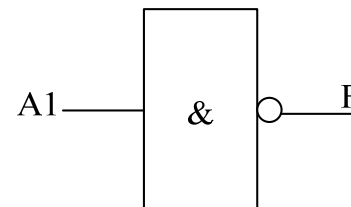
$$F = A1 \cdot A2 \cdot \dots \cdot An$$

与门



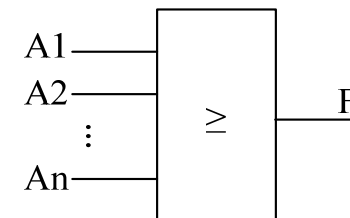
$$F = \overline{A1 \cdot A2 \cdot \dots \cdot An}$$

与非门



$$F = \overline{A1}$$

非门

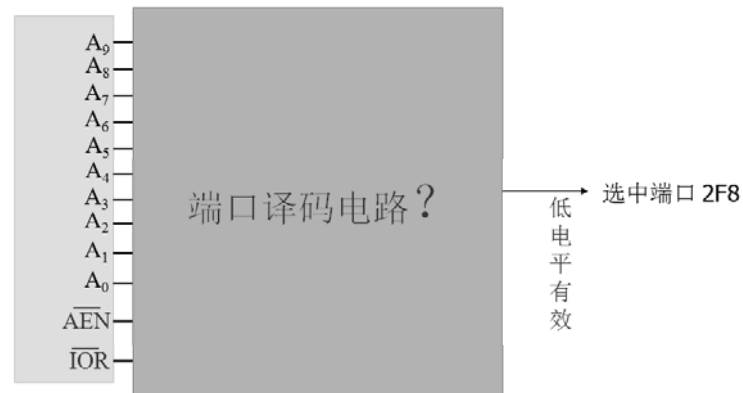


$$F = A1 + A2 + \dots + An$$

或门

## 译码接口电路的设计例子

- 使用门电路设计端口地址**2F8H**的只读地址译码电路。



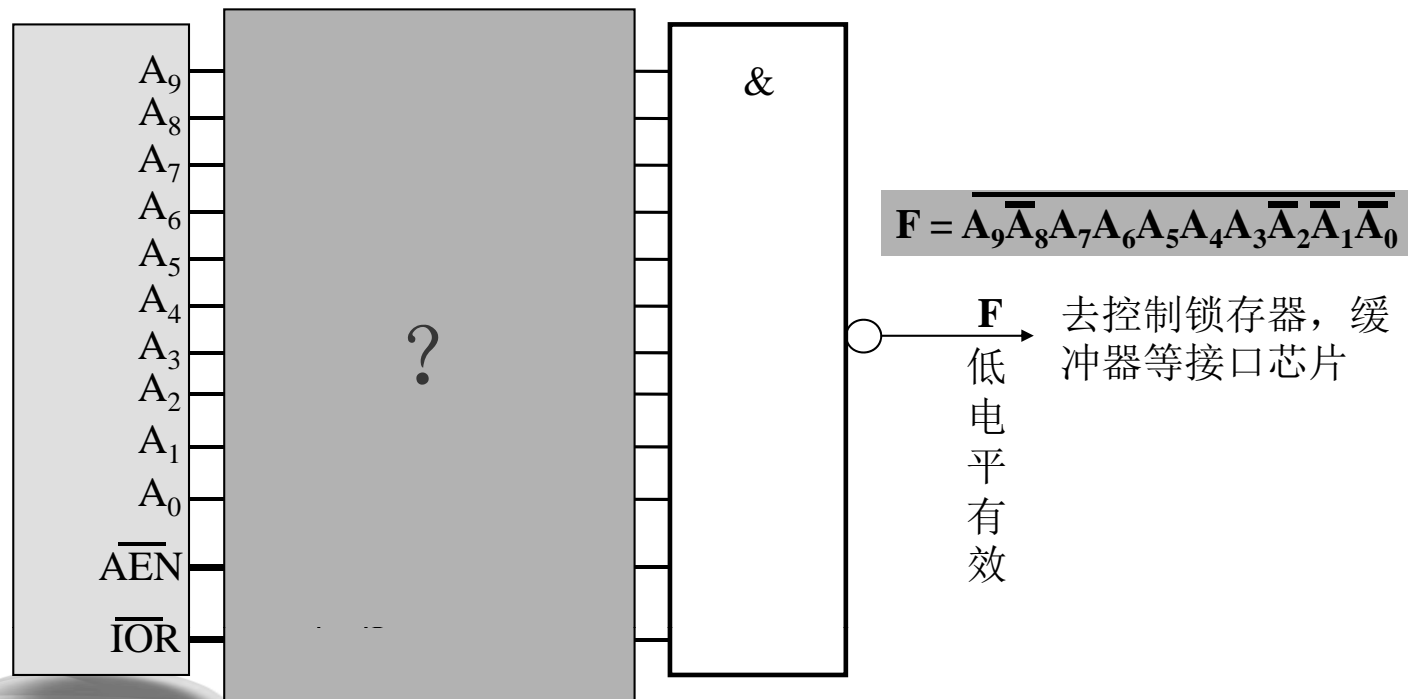
- 分析

■ 若要译码电路选中**2F8H**地址，亦即仅当地址总线输入**2F8H**时其输出有效低电平，而其它输入产生高电平。

地址线	0	0	$A_9$	$A_8$	$A_7$	$A_6$	$A_5$	$A_4$	$A_3$	$A_2$	$A_1$	$A_0$
十六进制				<b>2</b>				<b>F</b>				<b>8</b>
二进制	0	0	1	0	1	1	1	1	1	0	0	0

## 端口地址译码

地址线	0	0	$A_9$	$A_8$	$A_7$	$A_6$	$A_5$	$A_4$	$A_3$	$A_2$	$A_1$	$A_0$
十六进制				2				F				8
二进制	0	0	1	0	1	1	1	1	1	0	0	0



$$F = \overline{A_9} \overline{A_8} A_7 A_6 A_5 A_4 A_3 \overline{A_2} \overline{A_1} \overline{A_0}$$

$F$  去控制锁存器, 缓冲器等接口芯片  
低电平有效

$$F = A_9 \overline{A_8} A_7 A_6 A_5 A_4 A_3 \overline{A_2} \overline{A_1} \overline{A_0} \cdot \overline{AEN} \cdot \overline{IOR}$$

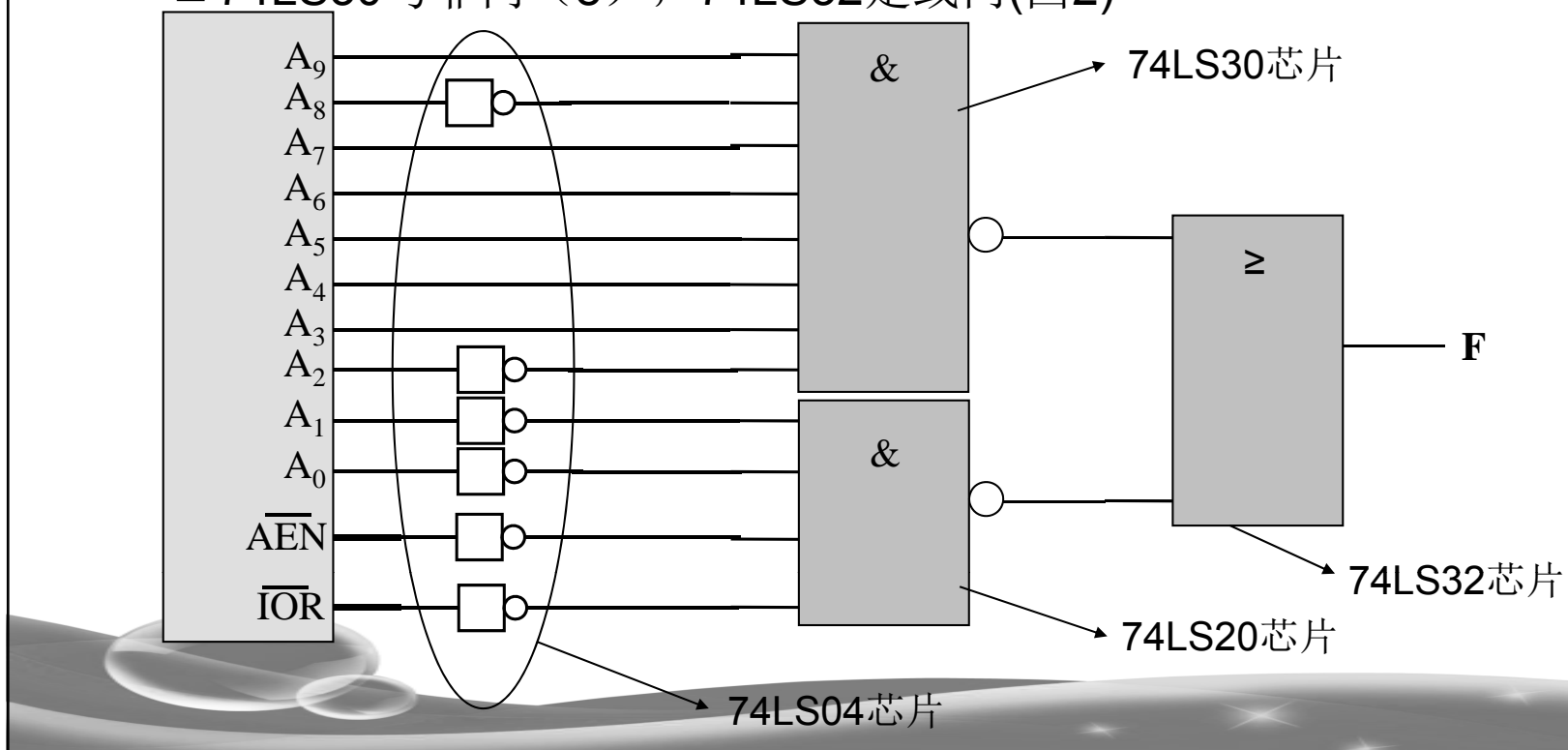
## 端口地址译码

$$F = \overline{A_9} \overline{A_8} A_7 A_6 A_5 A_4 A_3 \overline{A_2} \overline{A_1} \overline{A_0} \cdot \overline{AEN} \cdot \overline{IOR}$$

● 采用芯片设计 **2F8**  $F = (A_9 \overline{A_8} A_7 A_6 A_5 A_4 A_3 \overline{A_2}) + (\overline{A_1} \overline{A_0} \cdot \overline{AEN} \cdot \overline{IOR})$

■ 74LS04非门（6），74LS20与非门（二4）

■ 74LS30与非门（8），74LS32是或门(四2)



## ● 通过程序分析端口2F8H的访问

; 执行下面的程序，：

```
MOV  DX,  2F8H
```

```
IN   AL,  DX
```

; F = 0，对应端口被选中

执行下面的程序，能选中2F8端口吗？

```
MOV  DX,  2F8H
```

```
OUT  DX,  AL
```

## 端口地址译码(续)

- 含有多个端口的接口地址译码

- 例子：某接口有4个端口：384H~387H 。

- ◆ 1) 画出地址译码电路

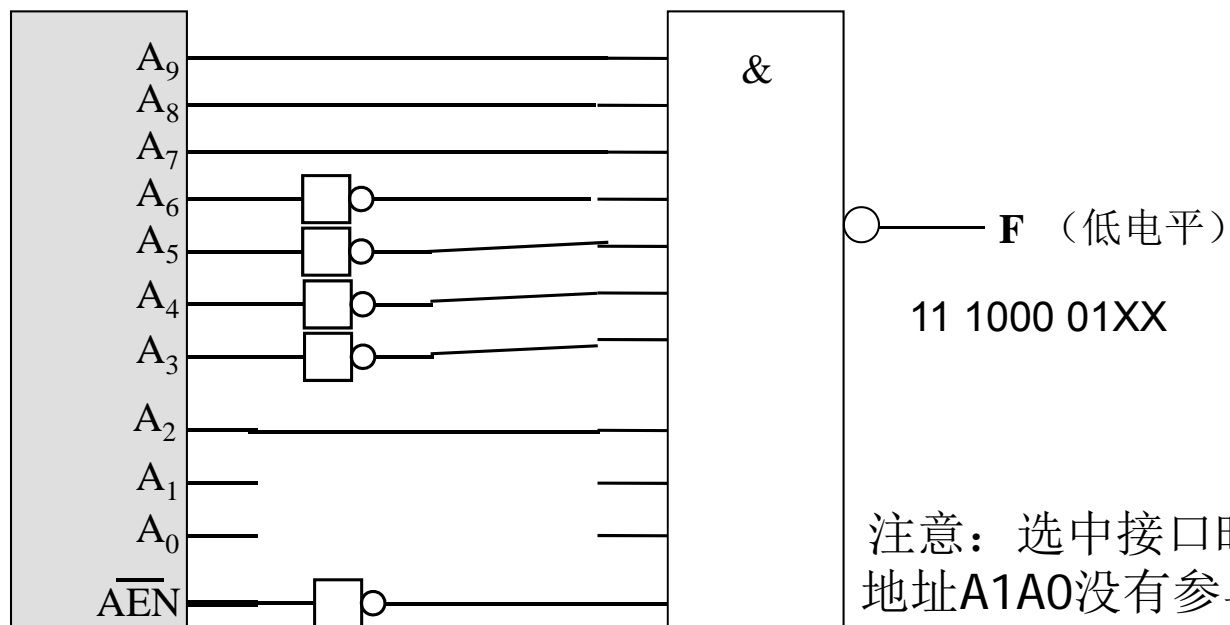
- 步骤

- ◆ 第1步：选中接口（接口译码）

- ◆ 第2步：选中接口中的某个端口（端口译码）

● 第1步：选中接口384H~387H

■ 仅当 $A_{9-0} = 11\ 1000\ 01XX$ 时译码电路输出低电平。

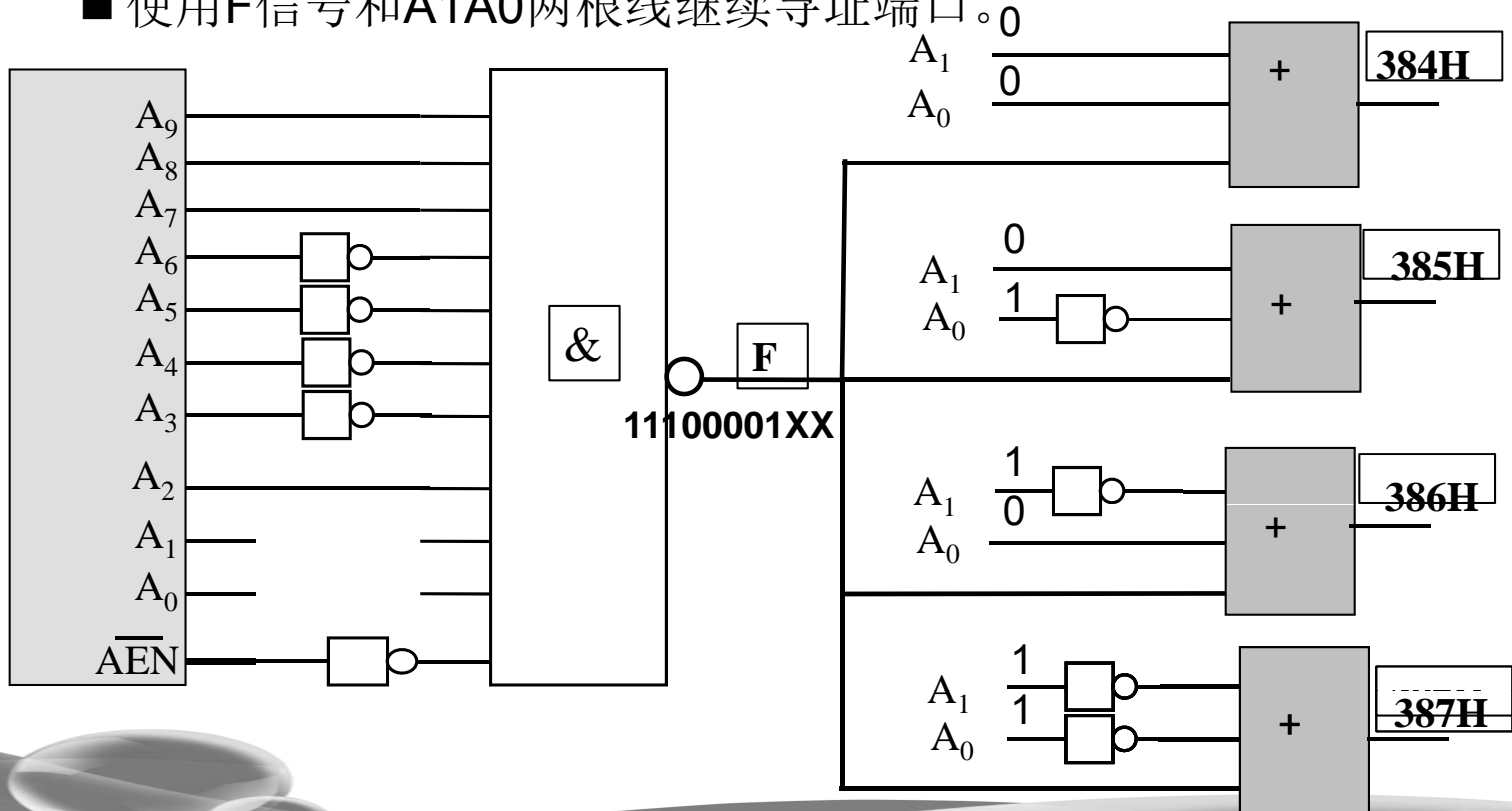


注意：选中接口时，低位地址 $A_1A_0$ 没有参与译码。

## 端口地址译码

### ● 第2步：选中特定端口

■ 使用F信号和A1A0两根线继续寻址端口。

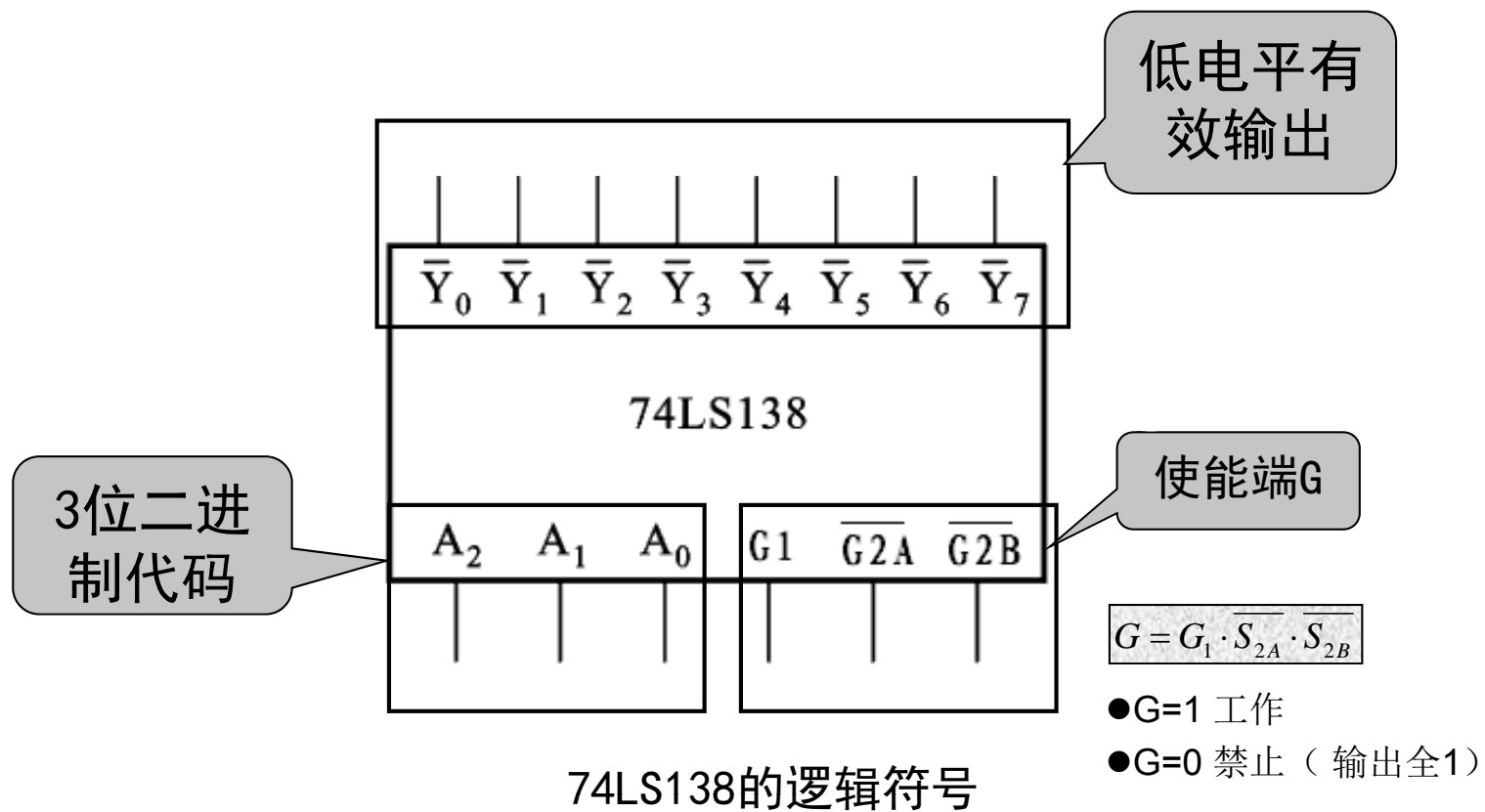




## 端口地址译码(续)

- 含有多个端口的接口地址译码
  - 例子：某接口有4个端口：384H~387H 。
  - ◆ 1) 画出地址译码电路【门电路】
  - ◆ 2) 用74LS04/20/30/32等芯片重新构成电路
  - ◆ 3) 用74LS138实现第2级译码（端口译码）

## 回顾：74LS138 —— 译码器



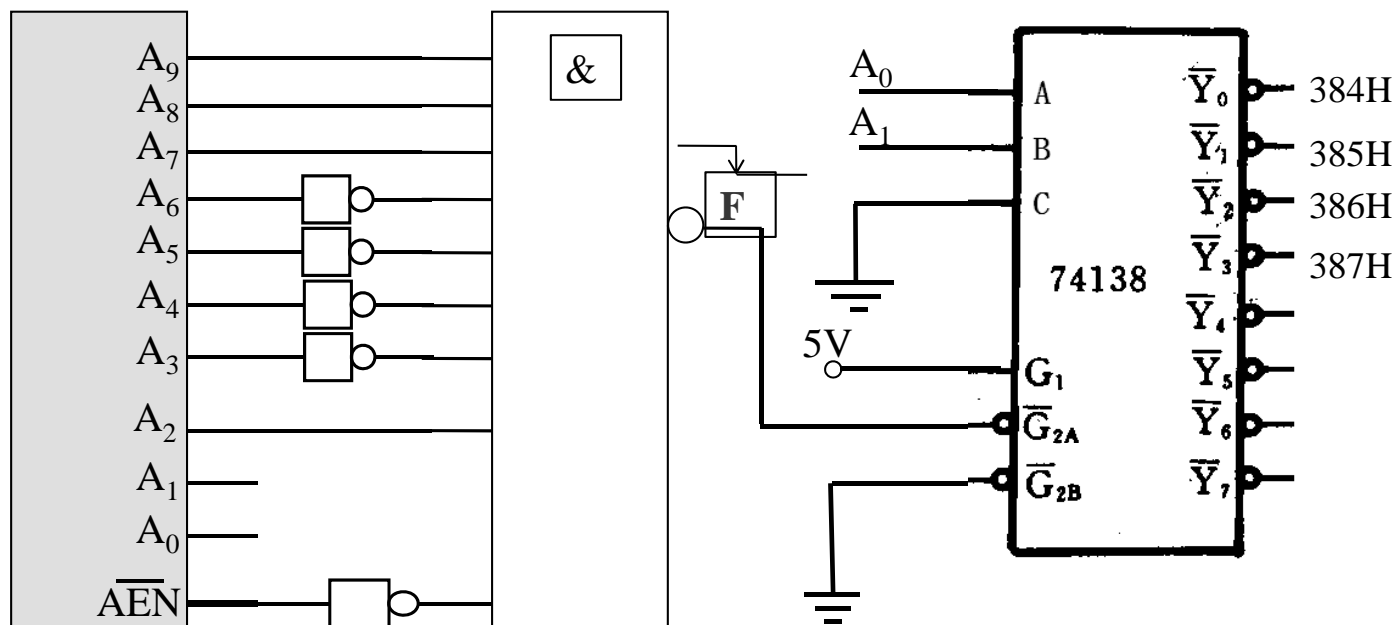
## 74LS138的功能表

输 入		输 出								
G1	$\overline{G2A+G2B}$	$A_2 A_1 A_0$	$\overline{Y}_0$	$\overline{Y}_1$	$\overline{Y}_2$	$\overline{Y}_3$	$\overline{Y}_4$	$\overline{Y}_5$	$\overline{Y}_6$	$\overline{Y}_7$
×	1	×	×	×	1	1	1	1	1	1
0	×	×	×	×	1	1	1	1	1	1
1	0	0	0	0	0	1	1	1	1	1
1	0	0	0	1	1	0	1	1	1	1
1	0	0	1	0	1	1	0	1	1	1
1	0	0	1	1	1	1	0	1	1	1
1	0	1	0	0	1	1	1	0	1	1
1	0	1	0	1	1	1	1	0	1	1
1	0	1	1	0	1	1	1	1	0	1
1	0	1	1	1	1	1	1	1	1	0

## 端口地址译码

- 第2步：选中特定端口（138译码器）

- 使用F信号和A1A0两根线继续寻址端口。



思考：如果 $A_1 \rightarrow A$ ,  $A_0 \rightarrow B$ ,  $Y_0, Y_1, Y_2, Y_3$ 的地址各是多少？

思考：如果 $C \rightarrow +5V$ ,  $Y_0, Y_1, Y_2 \dots Y_7$ 的地址各是多少？

## 端口地址译码

- 接口地址的构成形式和实现方法

- 单端口

- ◆ 门电路直接译码

- 多端口

- ◆ 两级译码

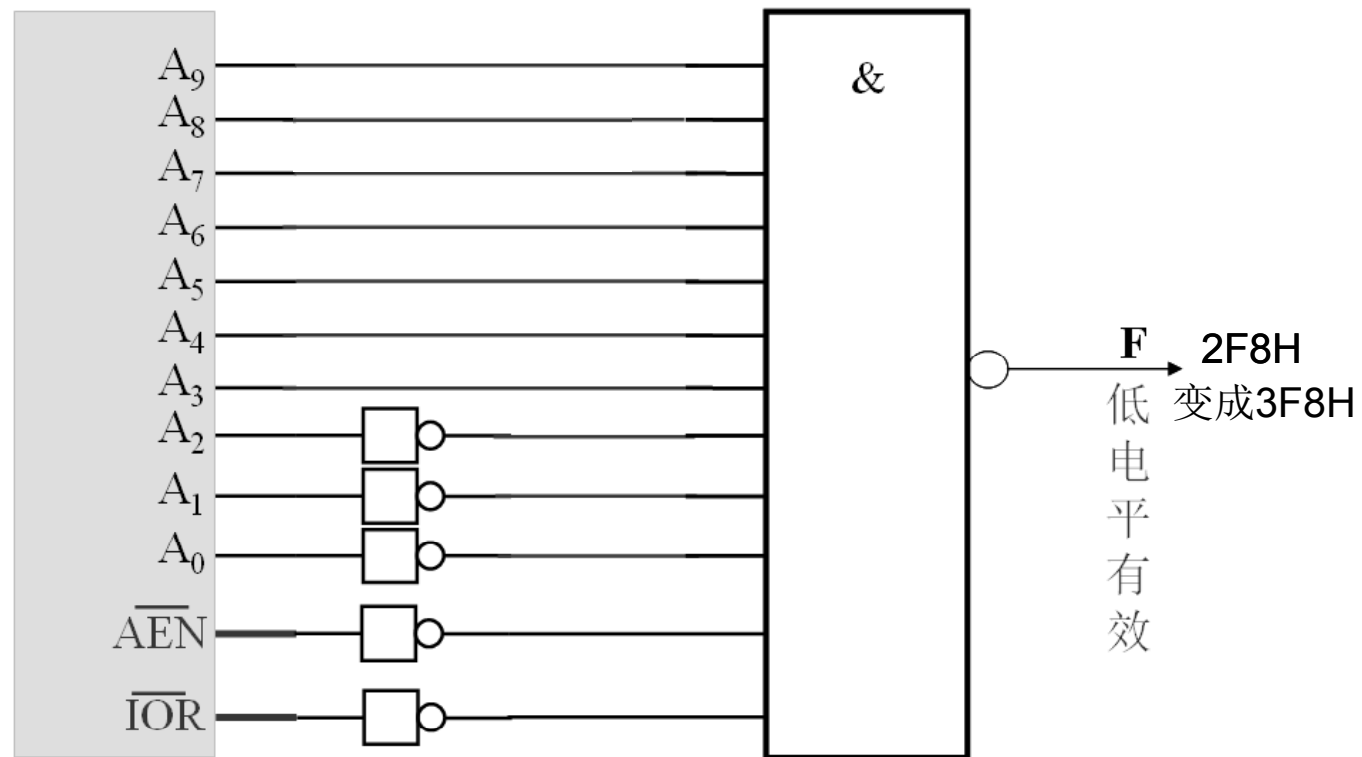
- 门电路

- 译码器（简洁，可靠）

- 地址可变

- ◆ 通过跳线或编程改变端口的地址

- 通过跳线或编程改变端口的地址(2F8h/3F8h)



## ● 选用I/O端口地址时要注意

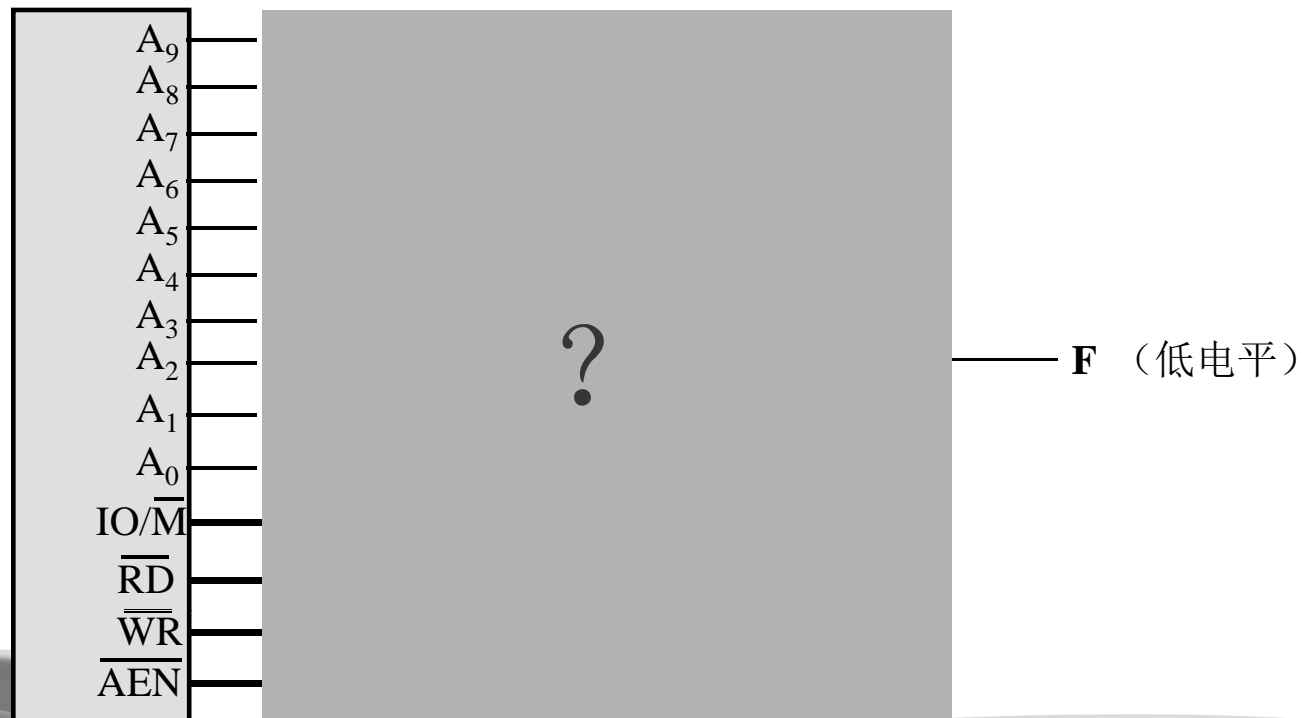
- 已占用地址不能使用；
- 保留地址不要使用；
- 为避免地址冲突，最好采用地址开关。（地址可变）
- 用户一般可使用**300~31FH**地址

## 课堂练习（1） 【 注明：姓名， 学号， 班号】

● 设计I/O端口地址为168H的只读端口地址译码电路。

■ 1) CPU的引脚:  $A_{9-0}$ ,  $\overline{IO/\overline{M}}$ ,  $\overline{RD}$ ,  $\overline{WR}$ ,  $\overline{AEN}$

■ 2) 画出原理图



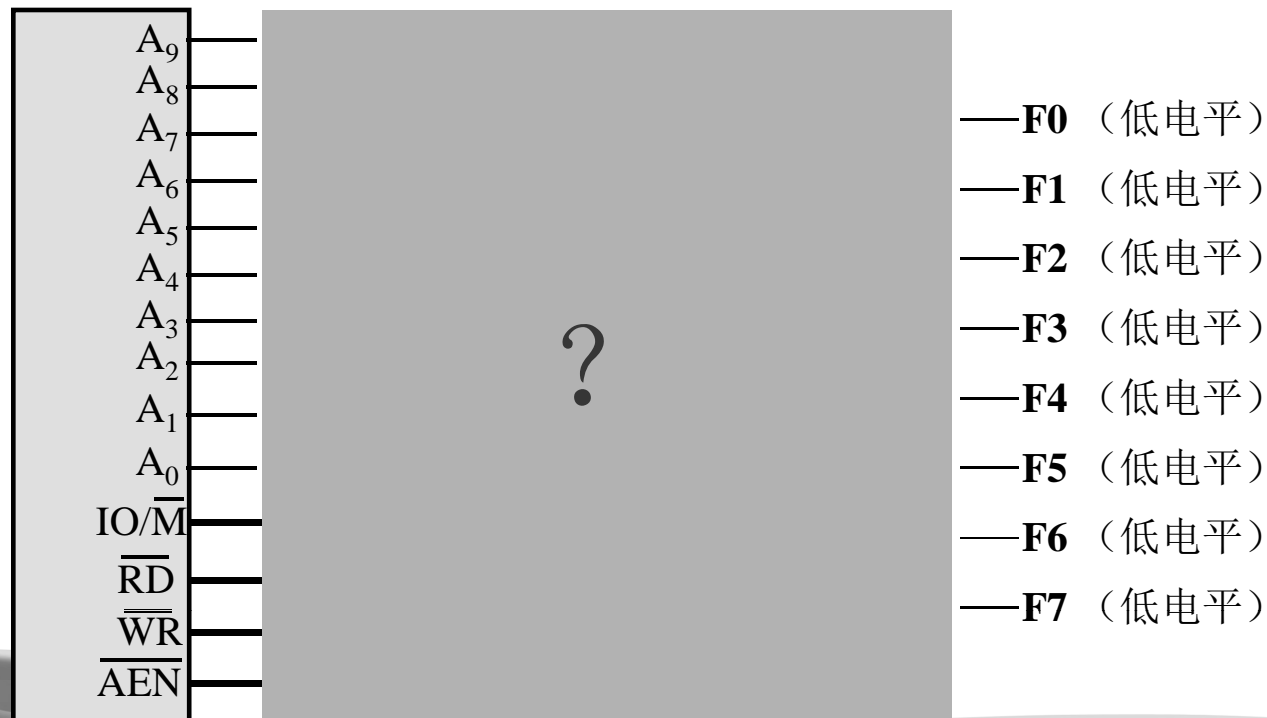


## 课堂练习（2） 【 注明：姓名，学号，班号】

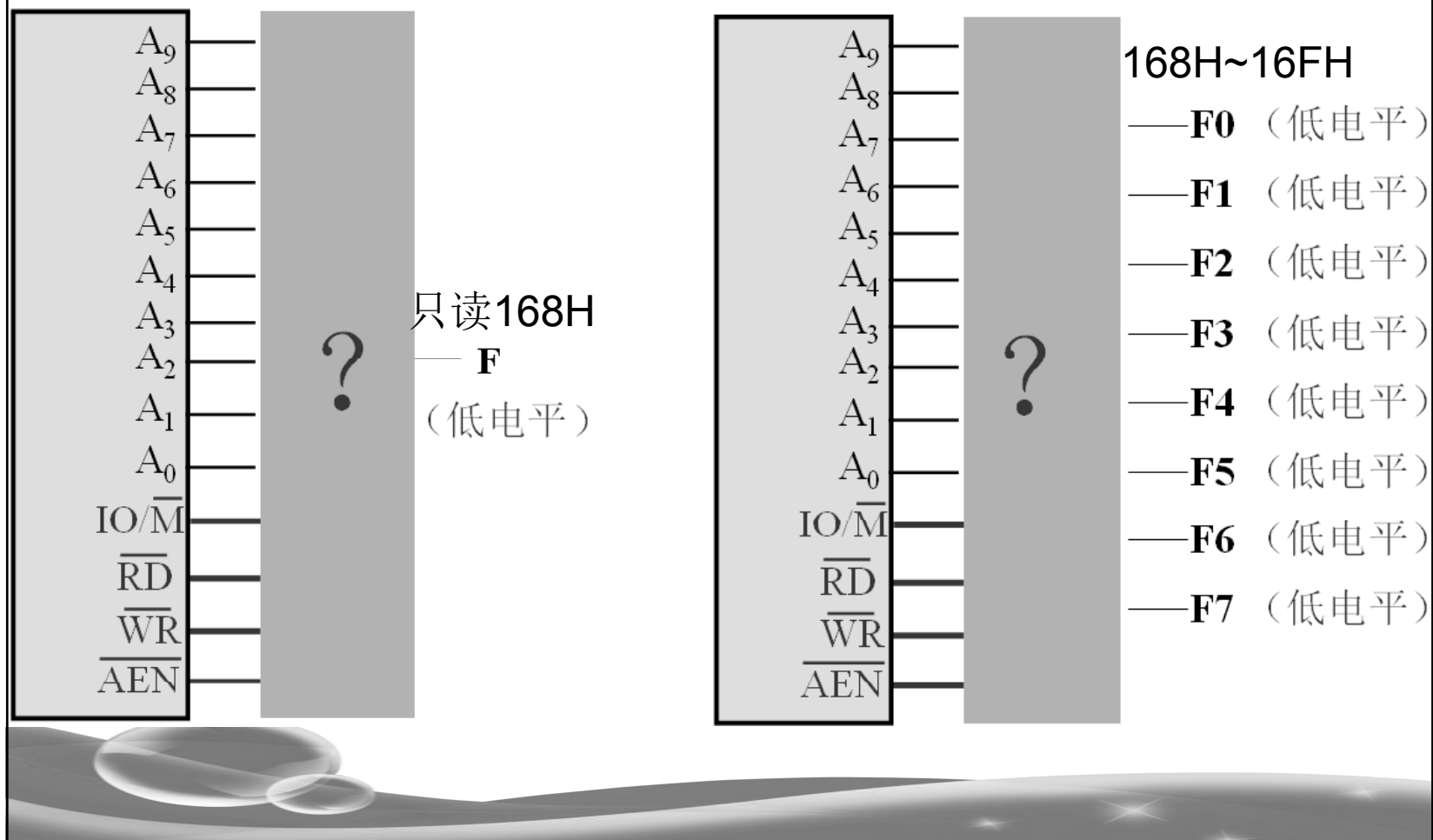
● 设计I/O端口地址为168H~16FH的端口地址译码电路。

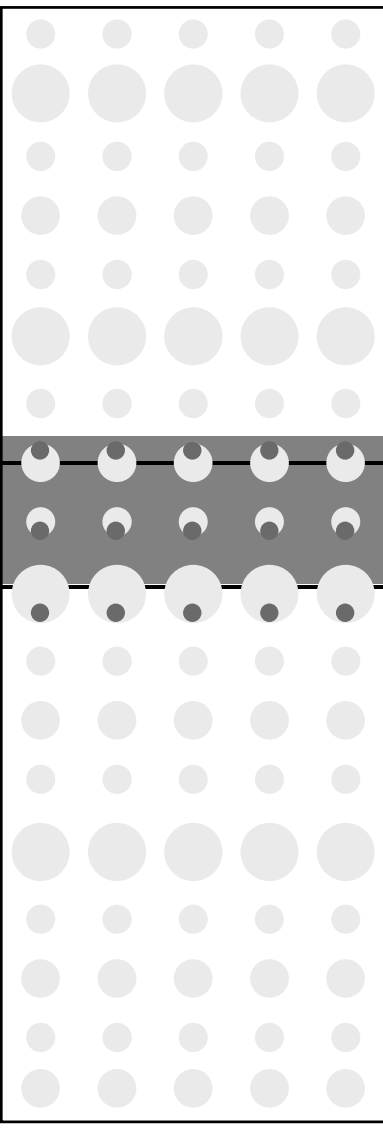
■ 1) CPU的引脚:  $A_{9-0}$ ,  $\overline{IO/\overline{M}}$ ,  $\overline{RD}$ ,  $\overline{WR}$ ,  $\overline{AEN}$

■ 2) 画出原理图【建议使用74LS138芯片】



# 课堂练习 【 注明：姓名，学号，班号 】





## 第4节 数据传输方式

- 数据传送的控制方式
  - 无条件传送方式
  - 查询传送方式（条件传送）
  - 中断传送方式
  - DMA控制方式

- 1. 无条件传送（同步传送）

- 当需要输入或输出数据时，不查询外设状态，假定外设已经准备就绪，直接使用I/O指令（IN或OUT）与外设传送数据。

- 外设准备就绪

- ◆ 输入设备：数据已经放入数据端口，CPU可以读取数据；

- ◆ 输出设备：数据端口已空，CPU可以向它写入数据。

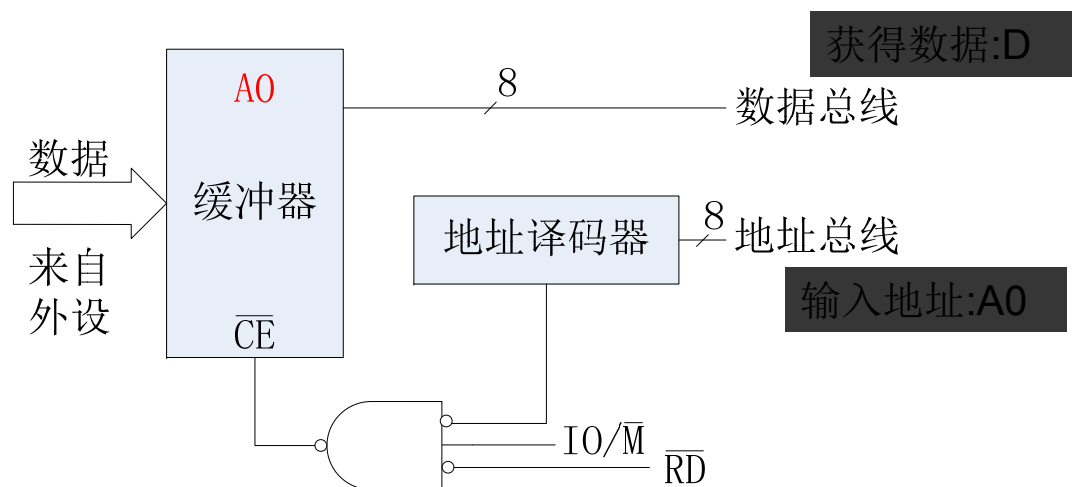
- 由于不查询外设状态，接口电路不需要状态端口

- 说明

- ◆ 通常接口硬件确保端口读写操作能同步进行。

## 无条件传送——输入

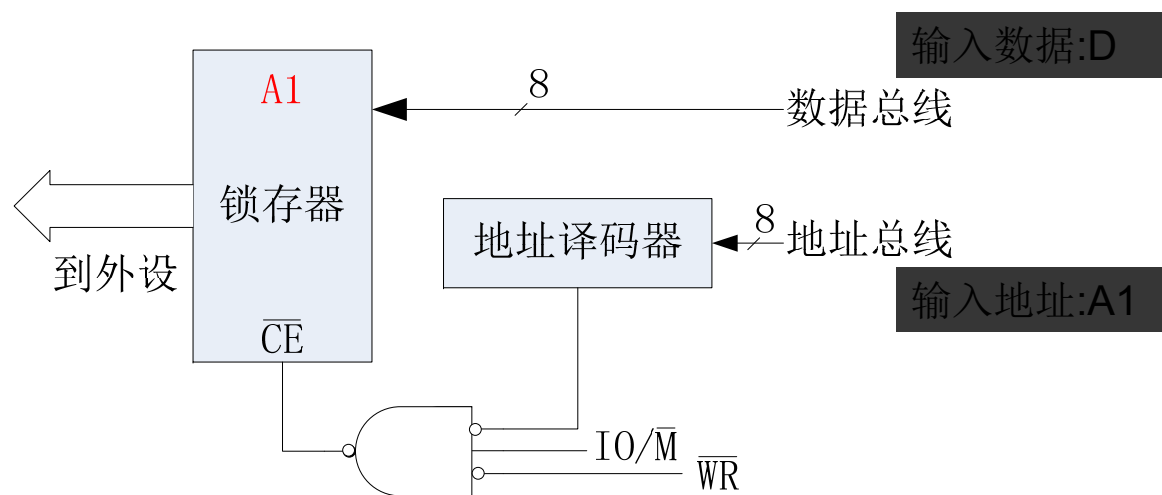
- IN AL, A0H



- 缓冲器：例如74LS244/245

## 无条件传送——输出

- OUT A1H, AL



- 锁存器：例如**74LS373**



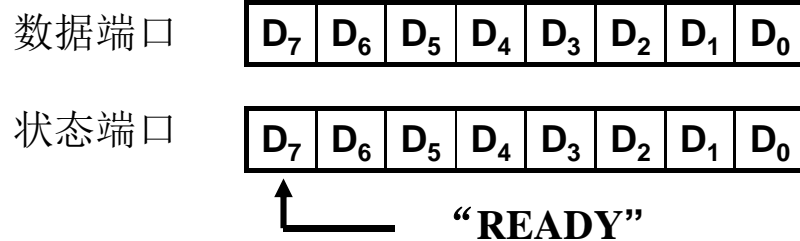


## 查询传送方式（异步传送）

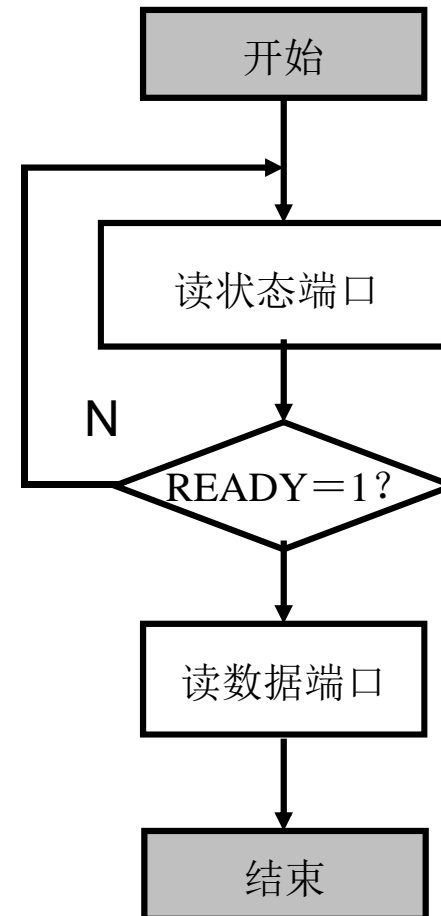
- 传送数据之前先确定外设是否准备好？！
- 传送过程
  - （1）先获取外设状态：执行IN指令读取外设状态端口
  - （2）根据外设状态判断：
    - ◆ 如果状态是“忙碌”或“未准备就绪”，则回到（1）；
    - ◆ 如果状态是“空闲”或“准备已就绪”，则continue；
  - （3）执行数据传送：对数据端口执行OUT或IN指令。
- 说明
  - 查询式接口要有数据端口和状态端口。
  - 端口一般都是8位，状态端口一般只需其中1位即可。

## 查询传送方式——输入

- 过程：读状态端口→读数据端口。



- 当READY为 1 时，表明输入数据已准备好；
- 当用IN指令完成数据输入后，READY自动变0。



● 输入过程的典型程序

```
POLL:  IN    AL,  PORT_S    ; 读状态端口: PORT_S
        TEST AL,  80H      ; 检查READY位是否为1
        JZ    POLL        ; 未准备好, 转POLL
        IN    AL,  PORT_D    ; 读数据端口: PORT_D
```

指令简介:

- (1) TEST: 类同AND指令, 不影响操作数, 仅影响标志位ZF  
若运算结果为0, 则ZF=1, 否则ZF=0。
- (2) JZ: ZF=1 (即结果为0) 则转移

● 输出过程的典型程序

```
POLL: IN      AL, PORT_S ; 输入状态信息
        TEST AL, 10H     ; 检查EMPTY位是否为1
        JZ     POLL      ; 外设不空（忙）转POLL
        MOV   AL, [DADA] ; DATA是需要输出的数据
        OUT   PORT_D, AL ; 向数据寄存器中输出数据
```

指令简介:

- (1) TEST: 类同AND指令, 不影响操作数, 仅影响标志位ZF  
若运算结果为0, 则ZF=1, 否则ZF=0。
- (2) JZ: ZF=1 (即结果为0) 则转移

## 数据传输方式（续）

- 3. 中断传送方式
- 4. 直接存储器存取方式（DMA）

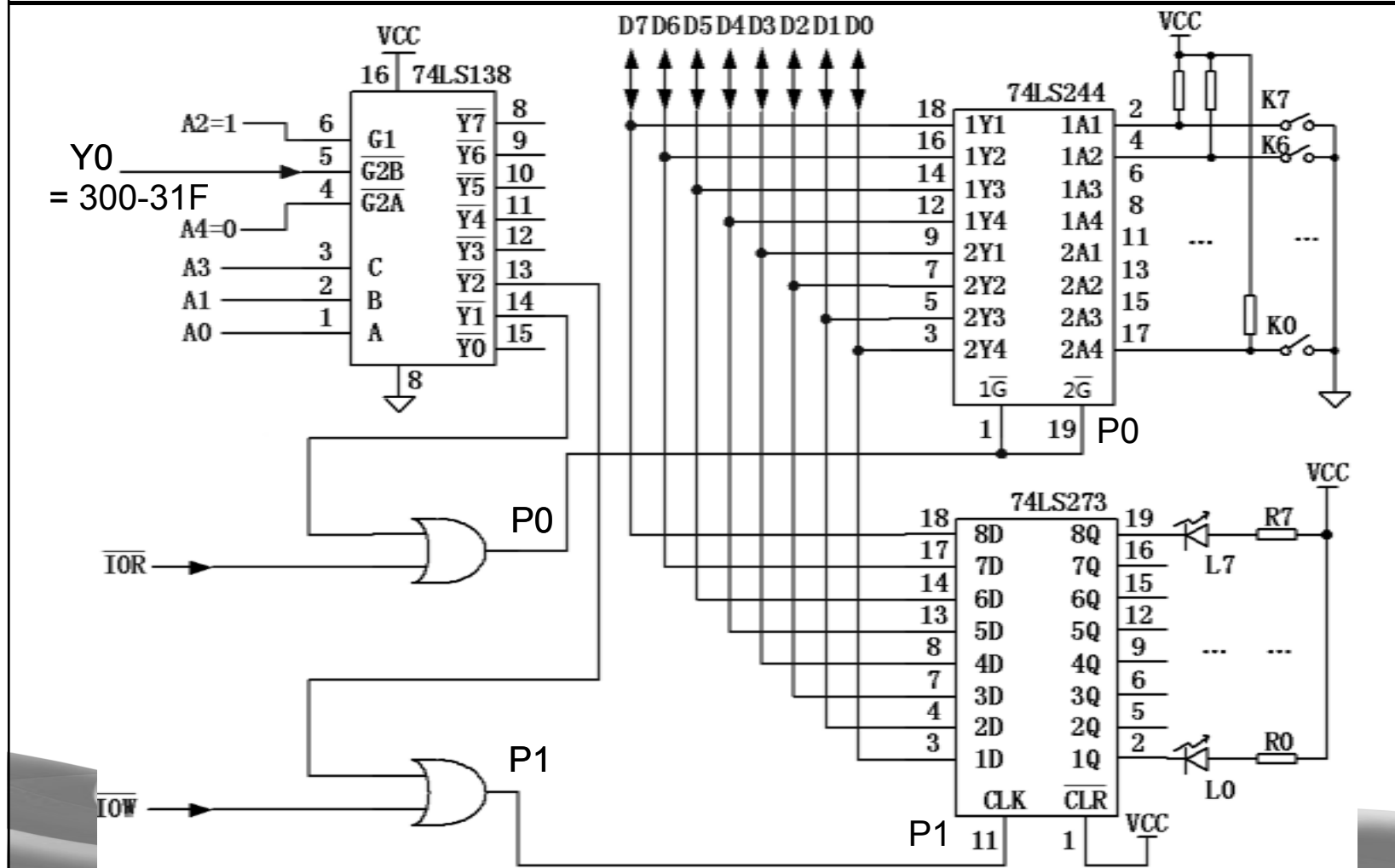


## 第5节 8088输入输出综合实例

## ● 练习的目的

- 熟悉8088外部引脚的典型连接
- 熟悉门电路，典型接口芯片的应用
- 熟悉接口和端口的概念和其访问
- 熟悉8088汇编程序的编写
- 理解软件硬件协同工作的思想

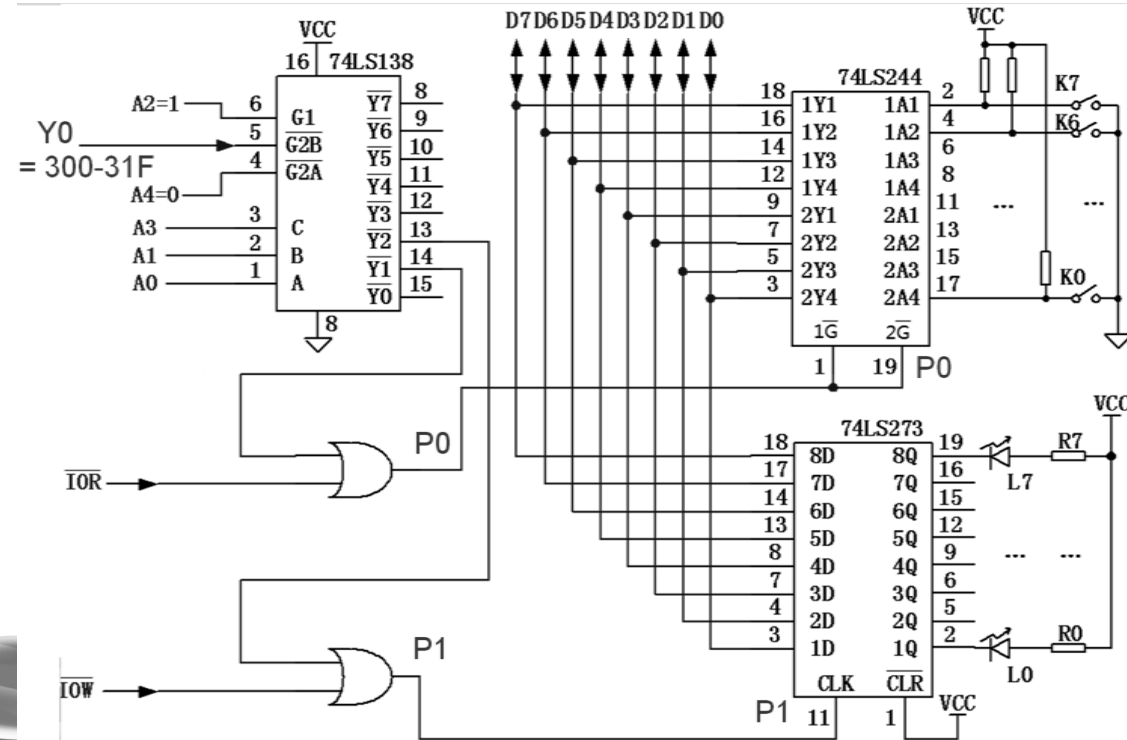
## LED(L<sub>0-7</sub>)和KEY(K<sub>0-7</sub>): 用L<sub>i</sub>(亮/灭)指示对应K<sub>i</sub>(闭/开)的状态





## 问题

- 1) 理解KEY和LED连接方式? 理解244, 273, 138的用法。
- 2) 请问按键KEY和LED的地址是多少?
- 3) 写一段I/O程序,用LED指示KEY闭合或断开状态 (亮:闭合)。



● 5) 写一段程序，实现如下功能：

■ 如果KEY<sub>i</sub>闭合，程序转去执行函数Function<sub>i</sub>，该函数点亮LED<sub>i</sub>

◆ 多个KEY同时按下时，注意优先级（假定KEY7最高）。

◆ 请先定义好：Function<sub>0</sub>， Function<sub>1</sub>， ..., Function<sub>7</sub>等8个函数。

```
function_0 proc near
    仅点亮LEDi
    ret
function_0 endp
```