

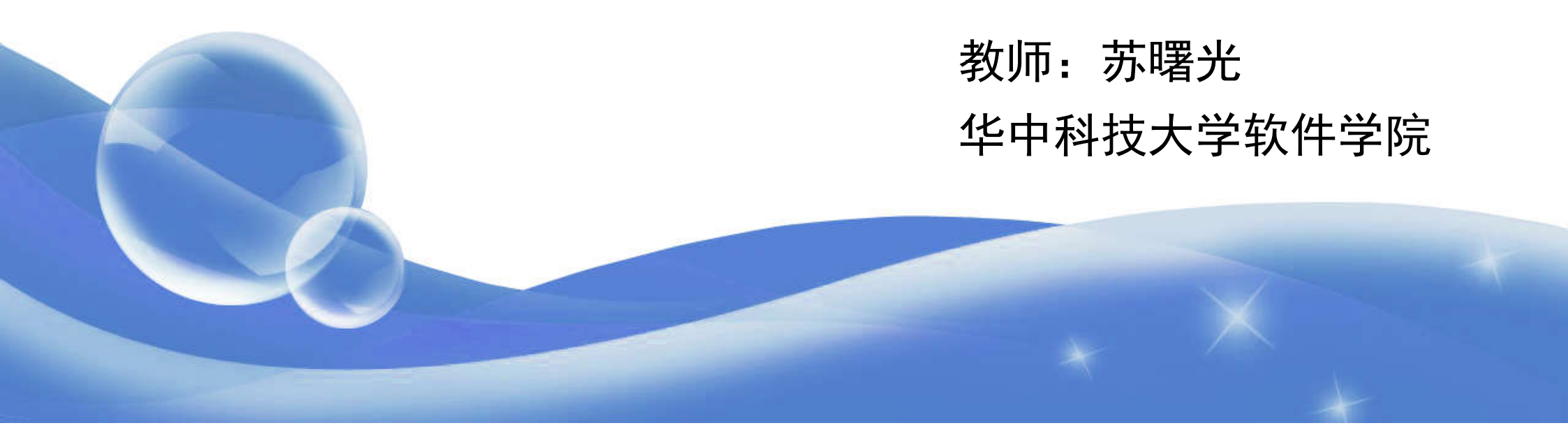


《微机原理与接口》

## 第3章 接口概念和原理

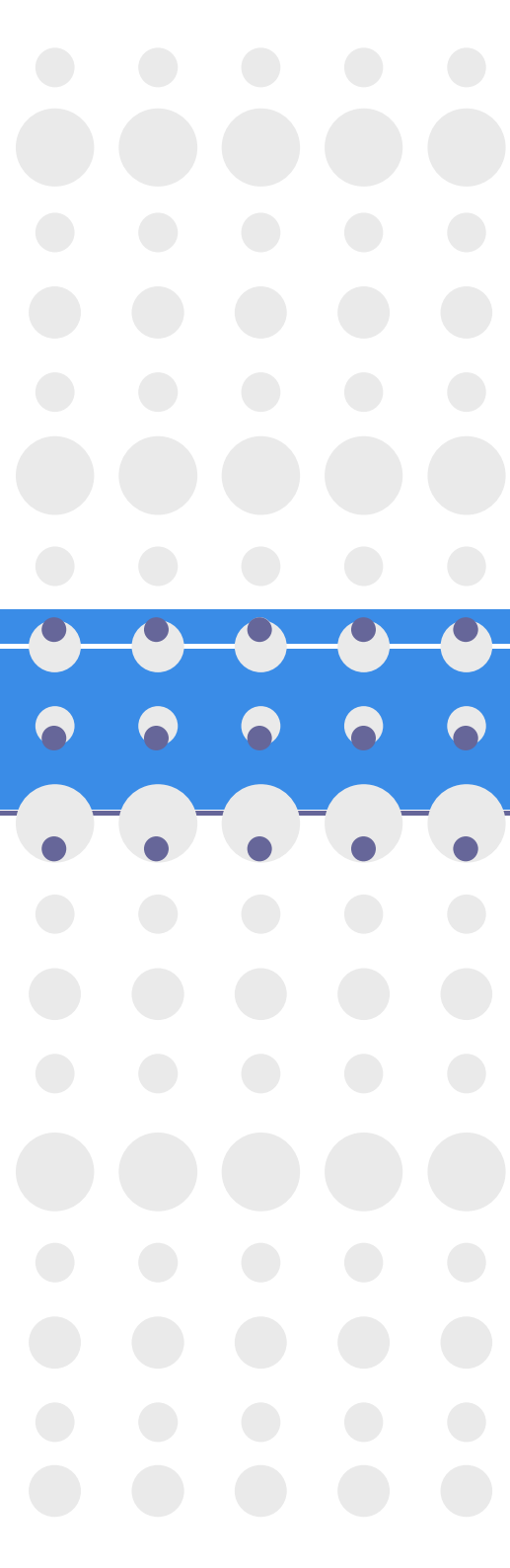
教师：苏曙光

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

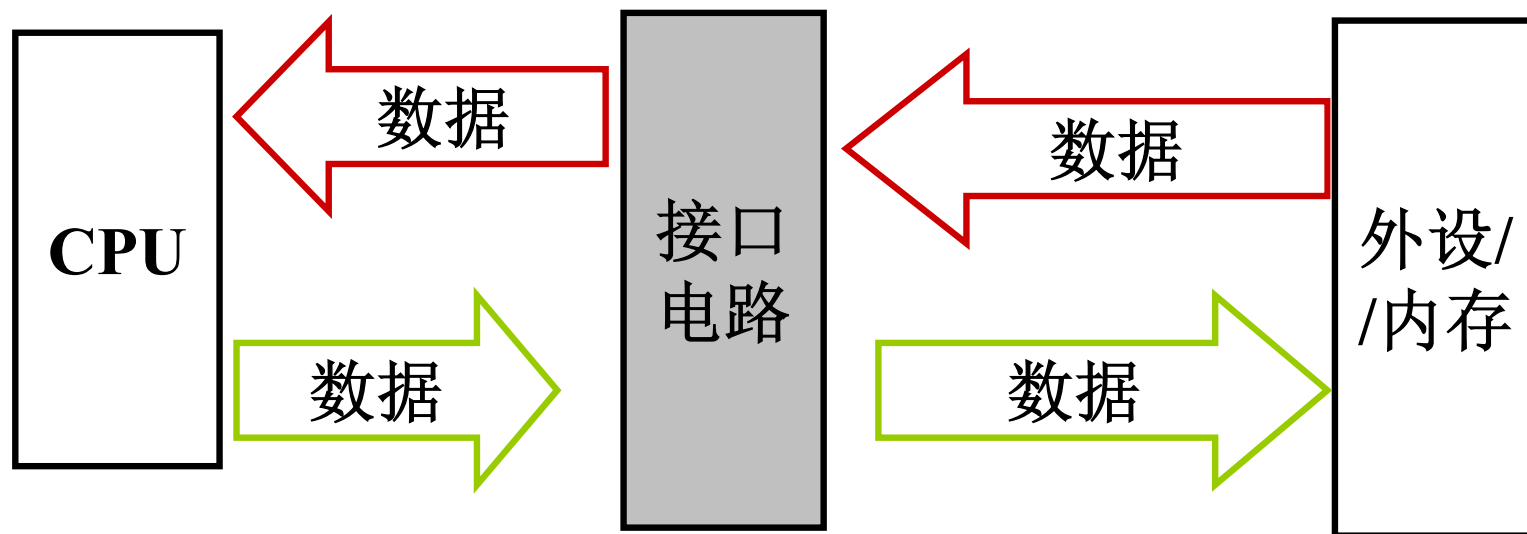


## 第1节 接口/端口的定义

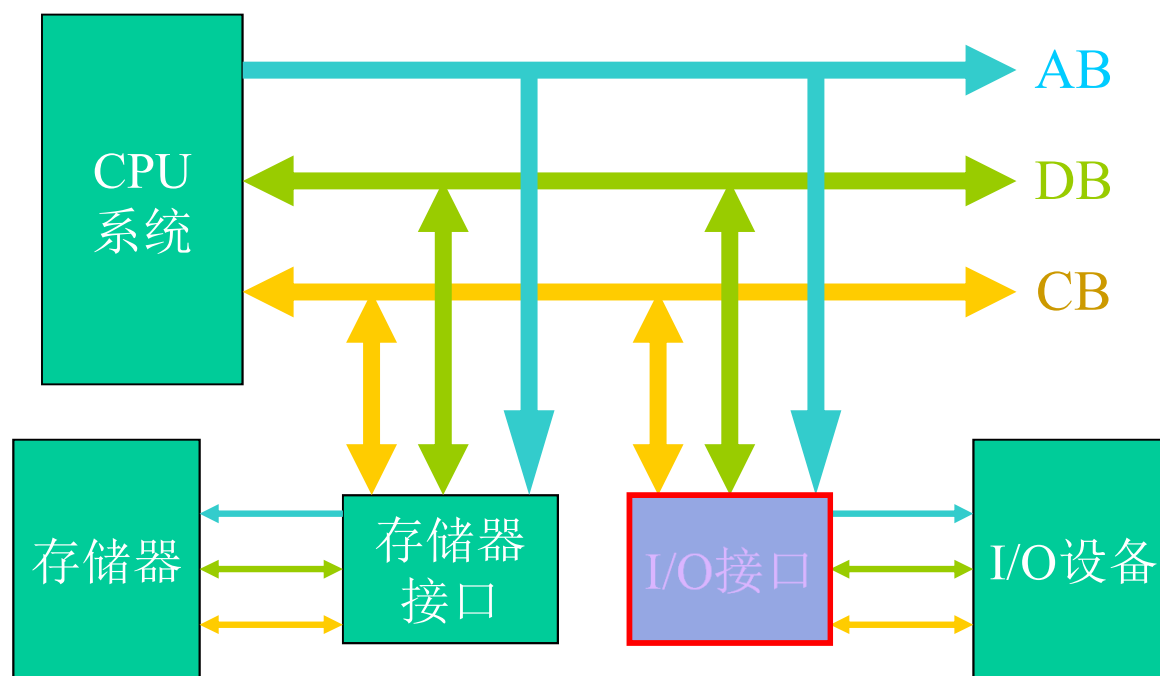
# 第1节 接口定义

## ● 接口定义

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



- 各种外设都必须通过**接口**才能和CPU（或总线）相连



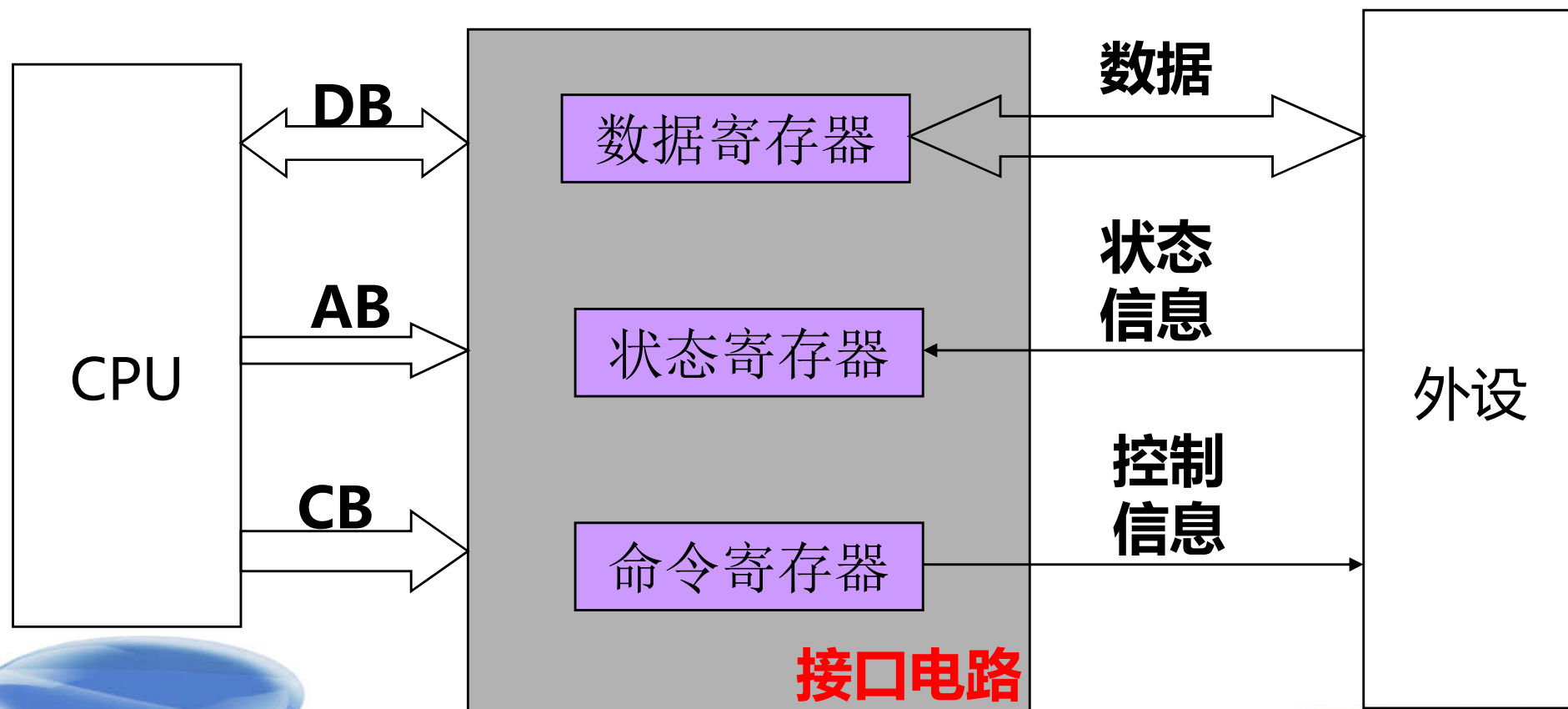
- 寻址：确定设备的**地址**，区分不同的设备；
- 缓冲：适配外设与CPU的**工作速度**；
- 转换：适配外设与CPU的信息**格式、类型、幅度**；
- 时序：外设与CPU的**工作时序**。

● 接口电路的组成：由多类/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。

定义：

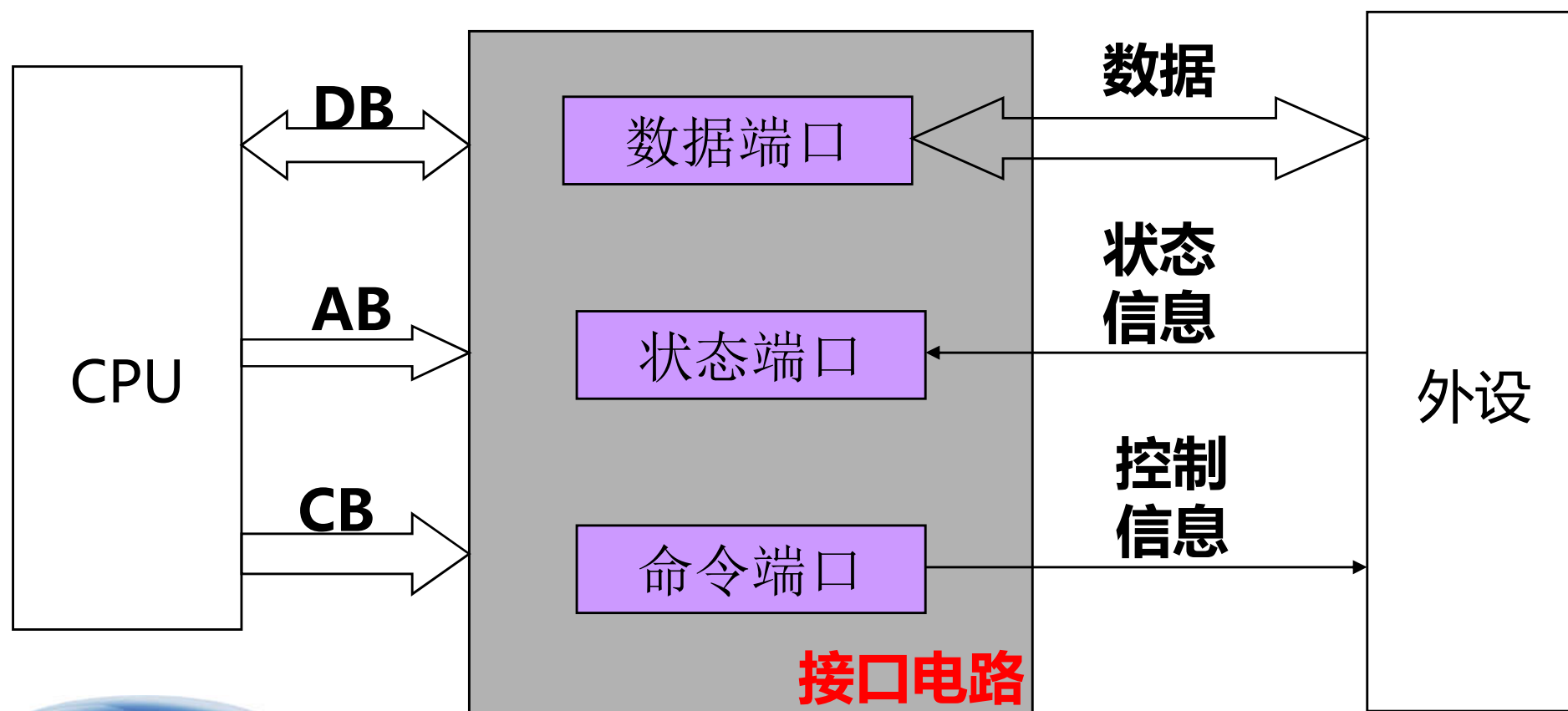
端口：寄存器的另一称呼。



## ● 端口【PORT】

■ 端口就是接口中的寄存器CPU，与外设信息交换场所

■ 端口：可寻址（端口地址）



# 微机的端口

- 16根I/O线：地址空间 $2^{16} = 64K$
- IBM:  $A_0 \sim A_9$ 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

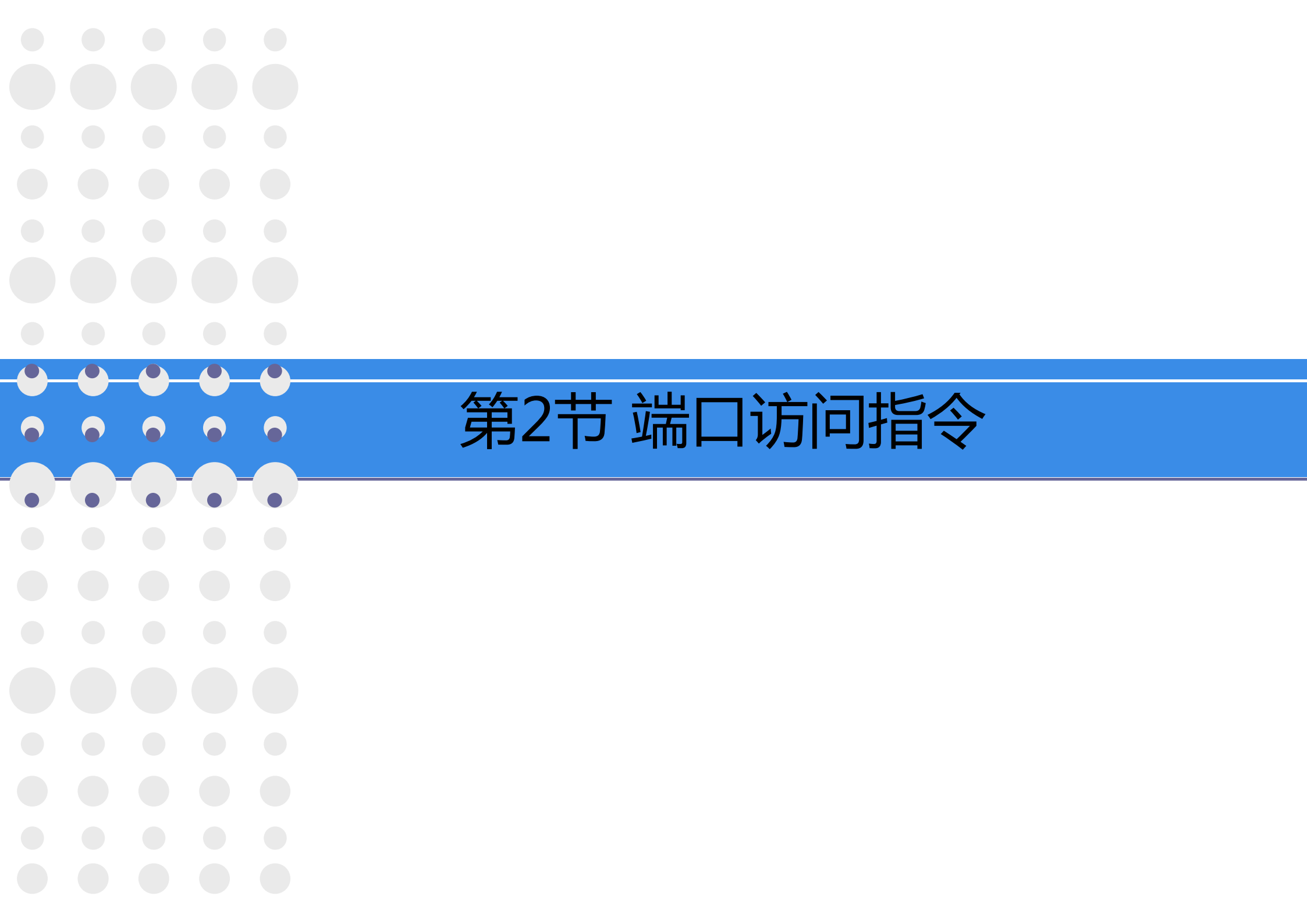
## ● 微机的端口分配

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
用户外接插件板	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



## 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

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

## 2. 端口的访问（续）

### ● 读（输入）指令：IN

■ 四种形式【 **PORT:** 具体的端口地址, **DX:** 寄存器 】

◆ IN AL, PORT ;数据是单字节; 地址单字节

◆ IN AX, PORT ;数据是双字节; 地址单字节

◆ IN AL, DX ;数据是单字节; 地址双字节

◆ IN AX, DX ;数据是双字节; 地址双字节

■ 例子

◆ IN AL, 60H ; 60H为系统板8255的PA端口地址

◆ MOV DX, 300H ; 300H为扩展板8255的PA端口地址

IN AL, DX

## 2. 端口的访问（续）

- 写（输出）指令：OUT

- 四种形式【**PORT**：端口地址，**DX**：寄存器】

- ◆ OUT PORT, AL ;数据是单字节；地址单字节

- ◆ OUT PORT, AX ;数据是双字节；地址单字节

- ◆ OUT DX, AL ;数据是单字节；地址双字节

- ◆ OUT DX, AX ;数据是双字节；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

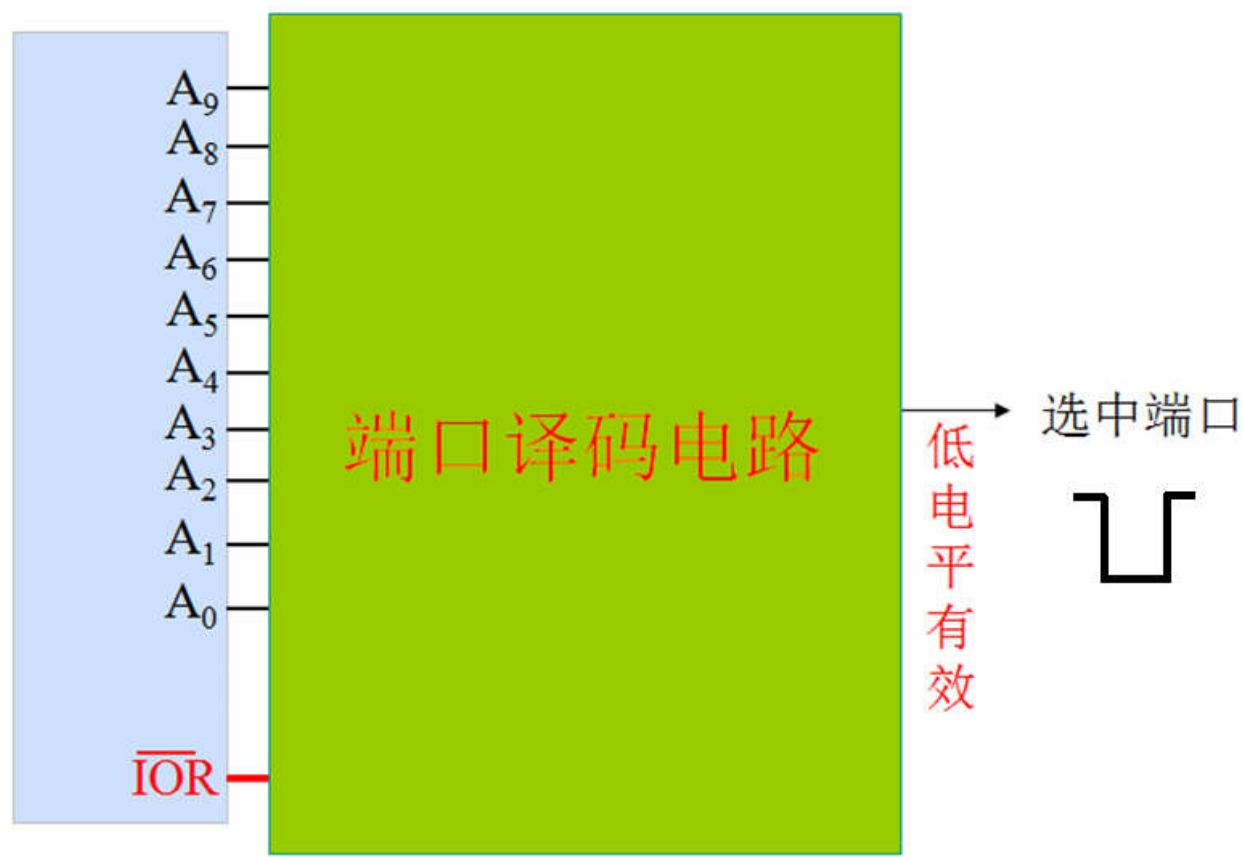
# 端口地址设计

● 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：

◆ 输入：端口地址（结合 $\overline{IO/\overline{M}}$ ,  $\overline{WR}$ ,  $\overline{RD}$ 等信号）

◆ 输出：仅当AB上出现端口地址时，输出低电平。



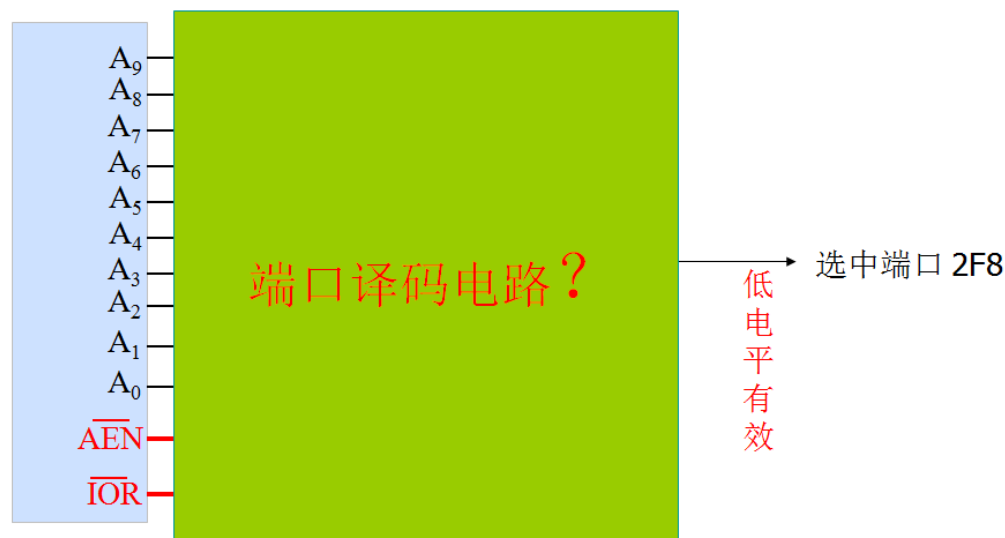


# 端口地址设计

- 设计地址译码电路三个前提
  - 有效I/O地址线10位：**A<sub>9~0</sub>**
  - 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。



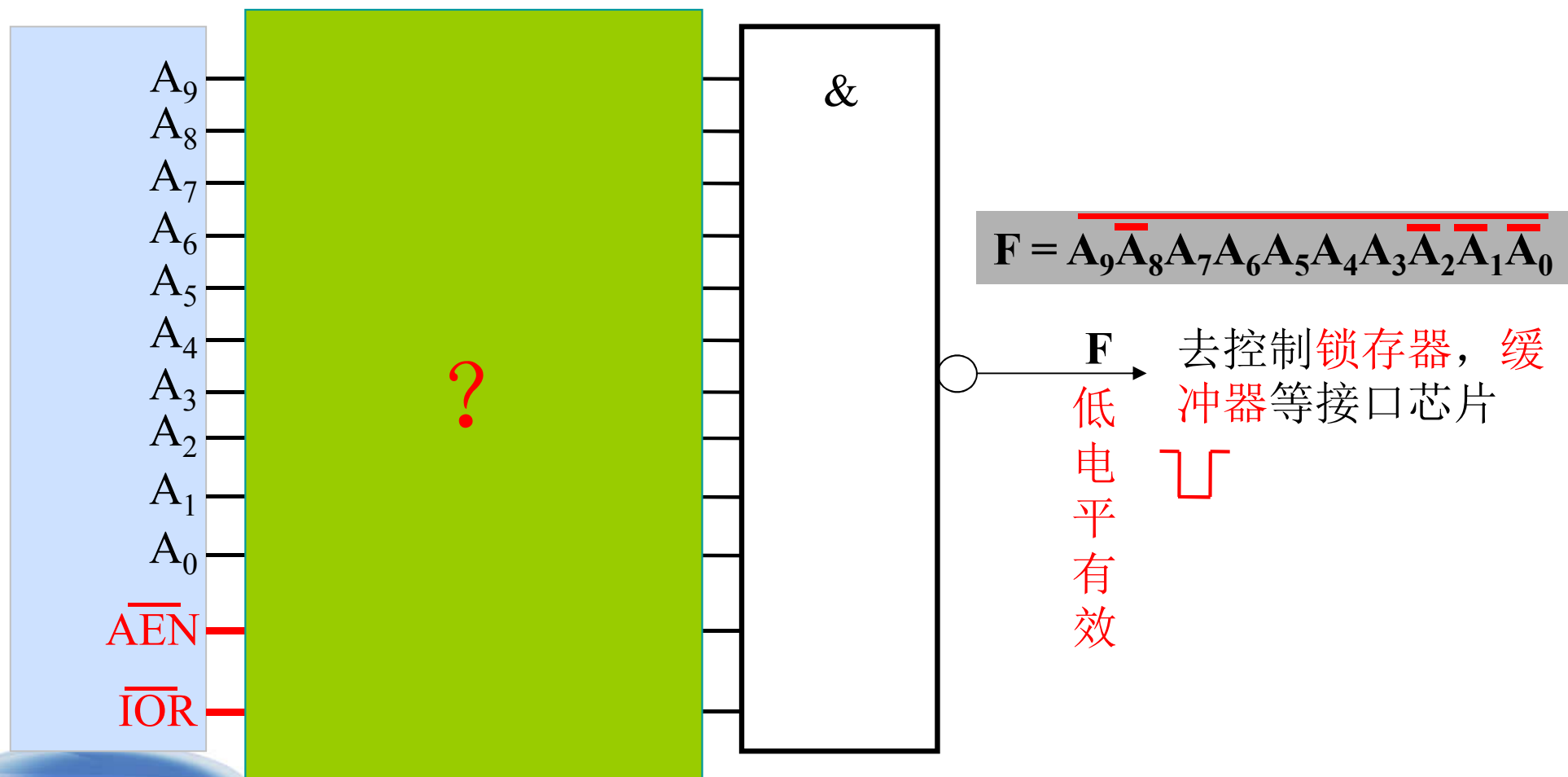
- 分析

■ 仅当AB输入**2F8H**时输出**低**电平，其它输入则输出高电平。

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>
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 <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>
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

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

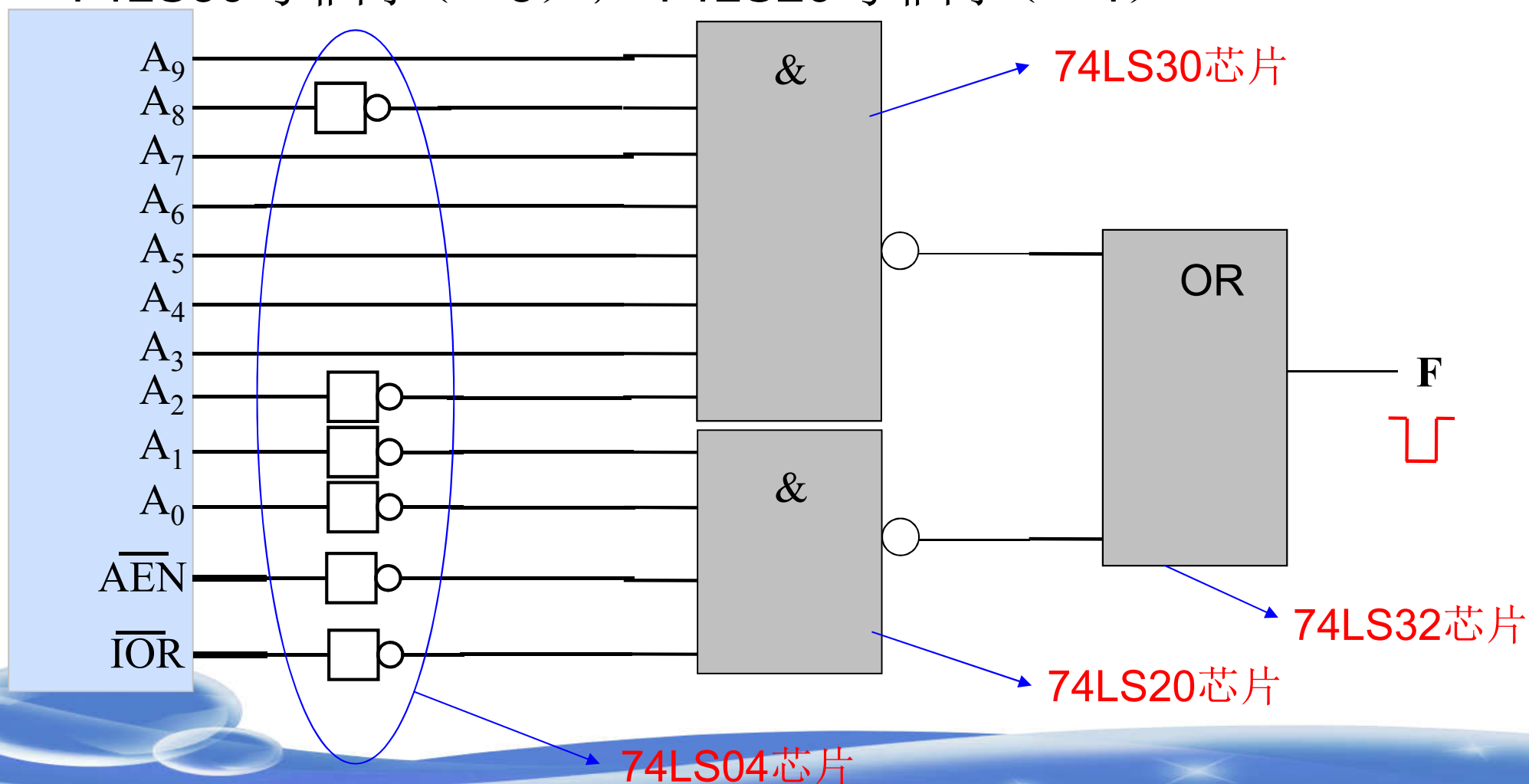
# 端口地址译码

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

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

■ 74LS32或门(四2)， 74LS04非门（六）

■ 74LS30与非门（二8）， 74LS20与非门（二4）



## ● 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

- 11 1000 0100;

- 11 1000 0101;

- 11 1000 0110;

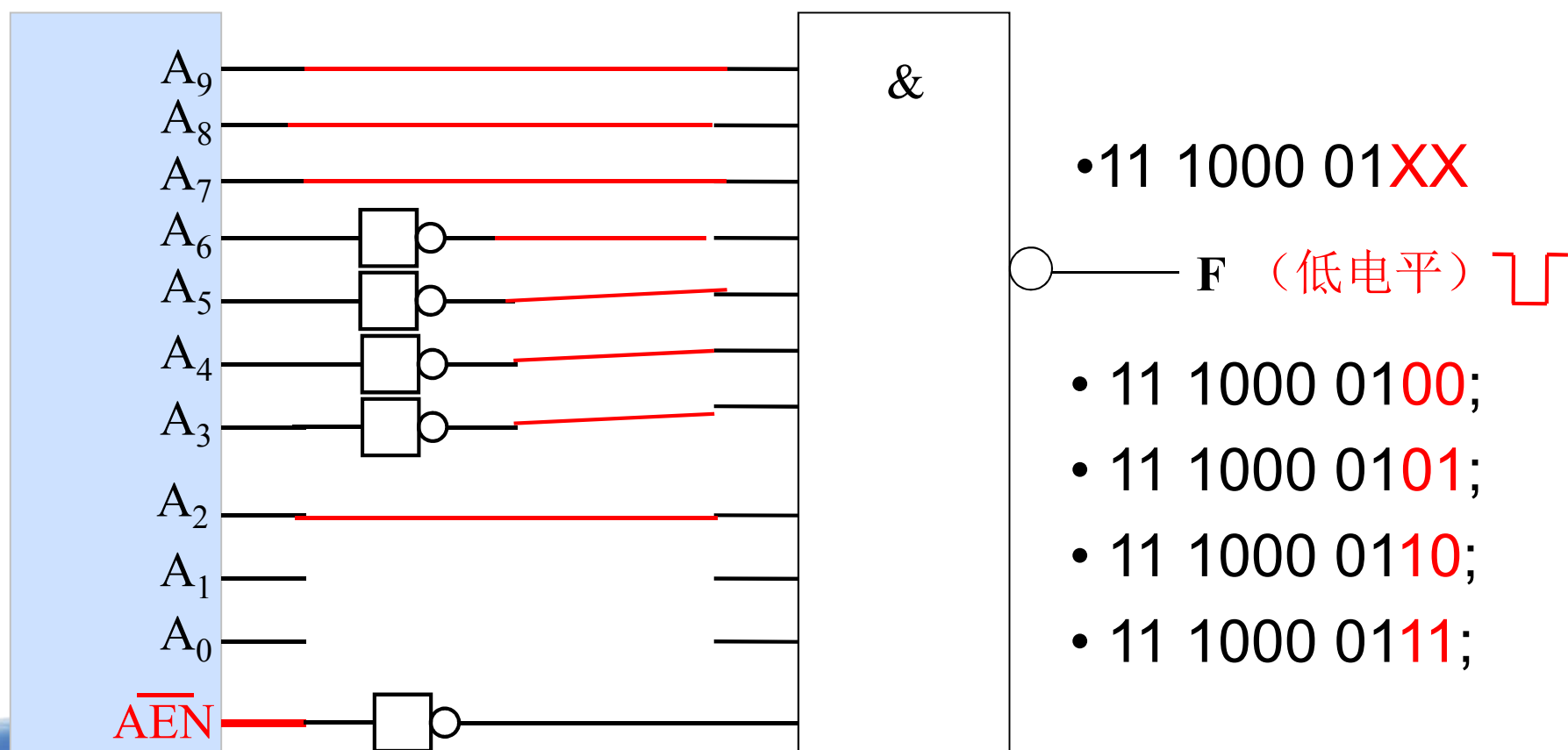
- 11 1000 0111;

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

## ● 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A1A0没有参与译码。

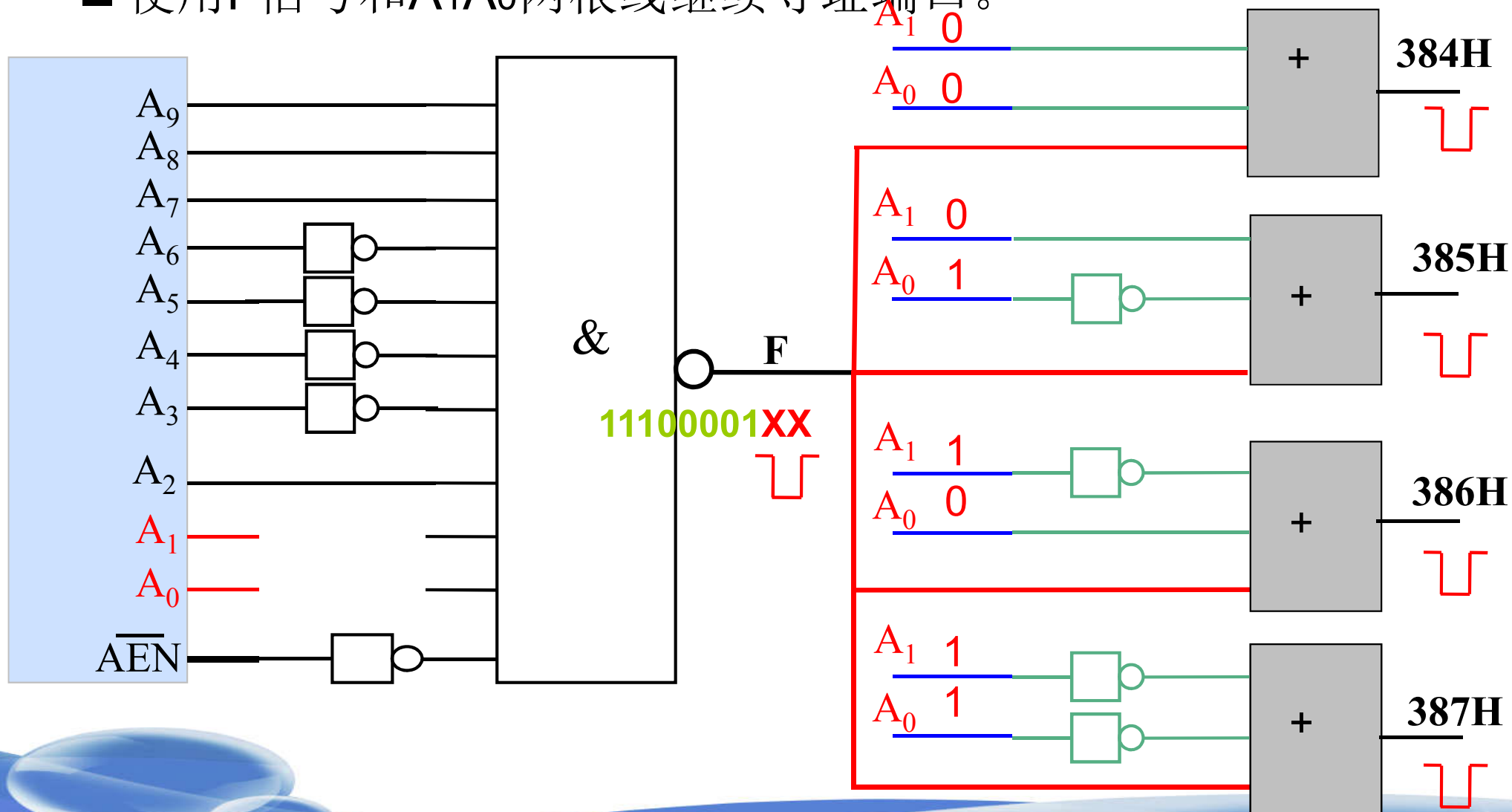




# 端口地址译码

## ● 第2步：选中特定端口

■ 使用F信号和A<sub>1</sub>A<sub>0</sub>两根线继续寻址端口。



# 端口地址译码(续)[课堂作业]

- 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

- 74LS32或门(四2)，74LS04非门(六)

- 74LS30与非门(二8)，74LS20与非门(二4)

- 输入信号：A9-A0，IOM，RD，WR

## 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。

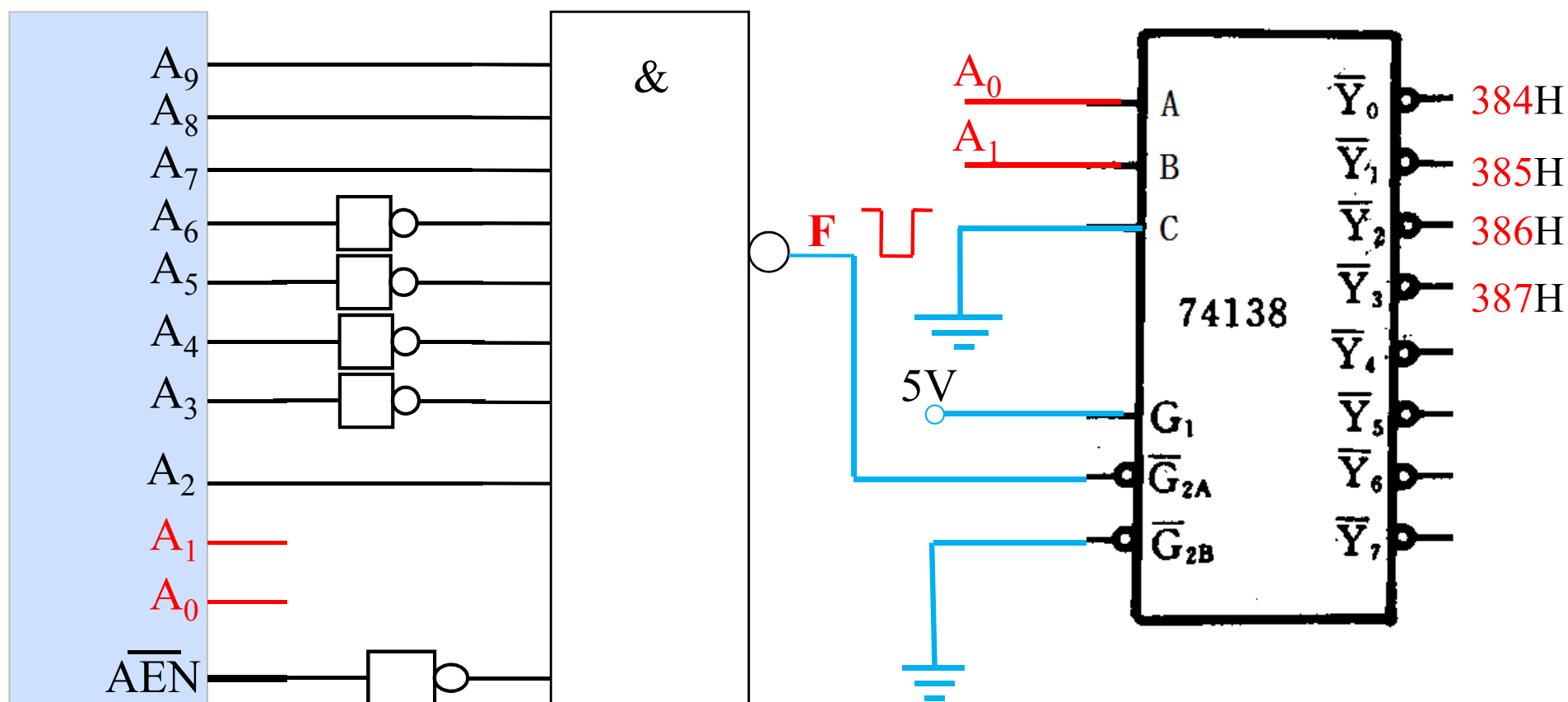
- ◆ 1) 画出地址译码电路【门电路】

- ◆ 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）

# 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。



# 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

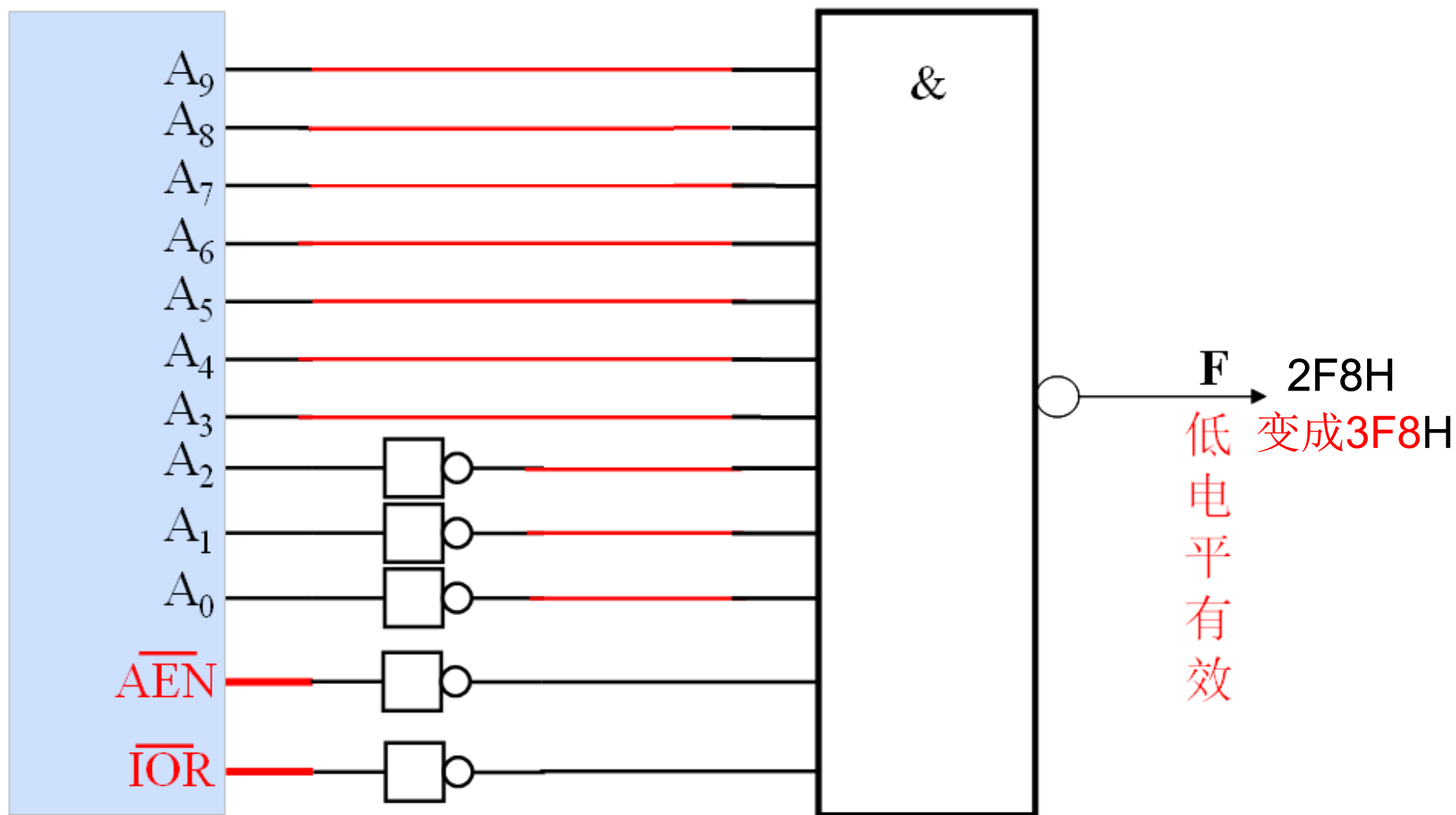
##### □ 门电路

##### □ 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

# 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

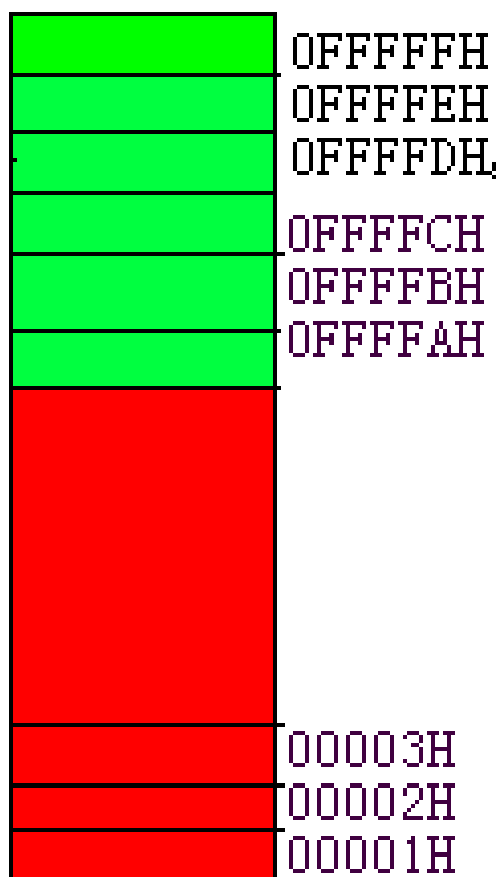
- 统一编址（存储器映射方式）

- ◆ 端口地址和存储器地址统一编址

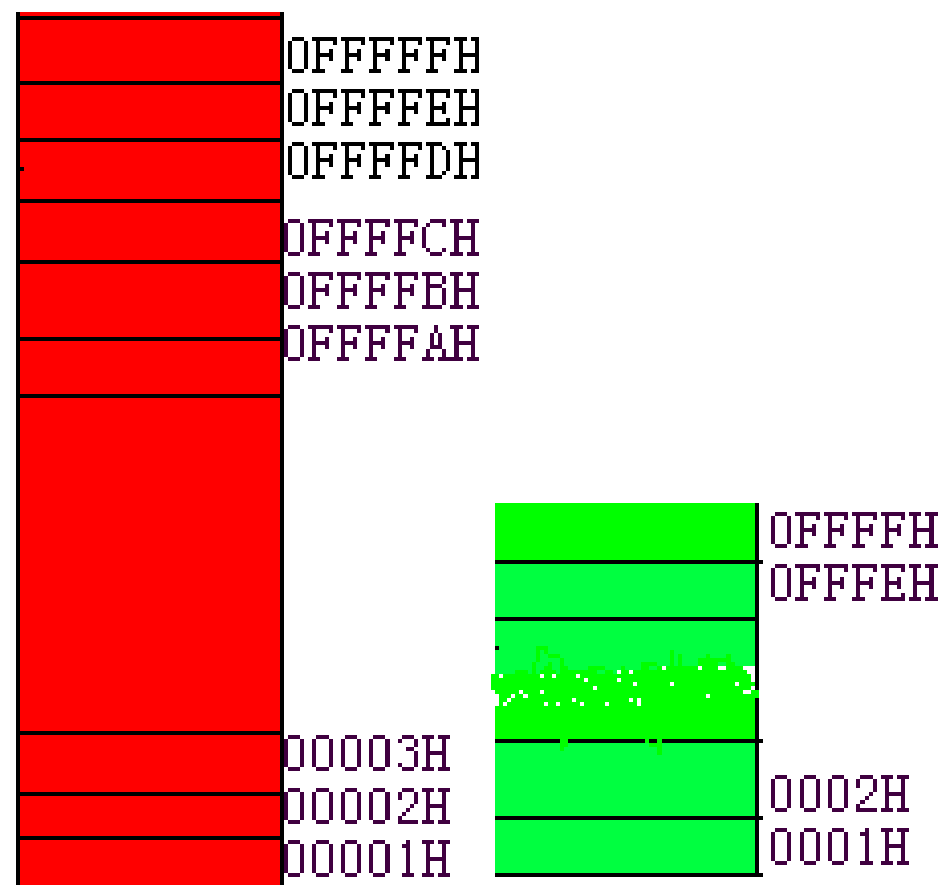


## ● 两种编址方式

IO  
端  
口  
使  
用



(a) 统一编址

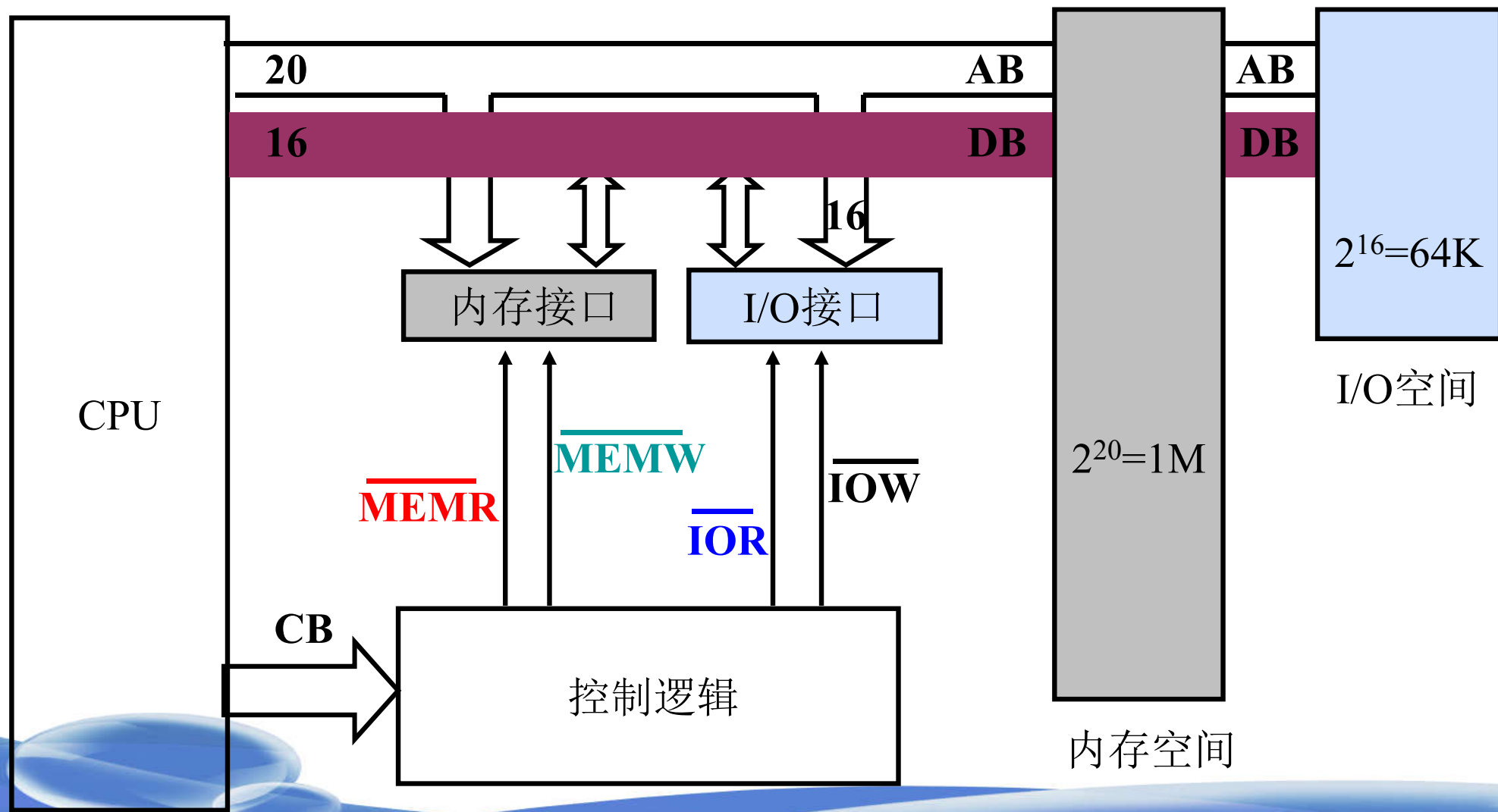


(b) 独立编址

# 端口地址编址方式（续）

## ● 独立编址的端口访问原理

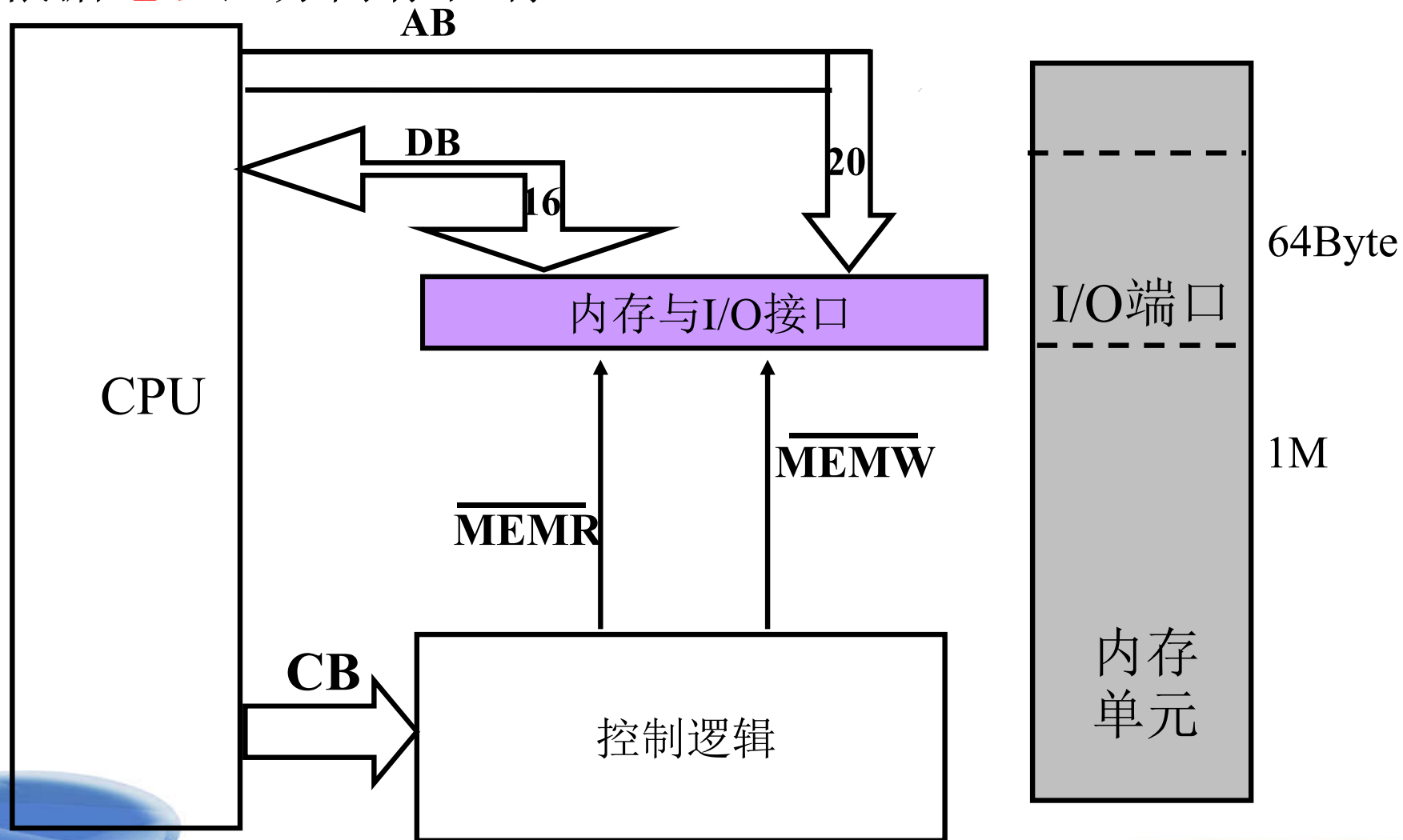
■ 根据指令（**IN/OUT** | **MOV**）区分内存和端口。



# 端口地址编址方式（续）

- 统一编址的端口访问原理

- 根据地址区分内存和端口。



# 端口地址编址方式（续）

- 独立编址系统的特点

- 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。



## 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。

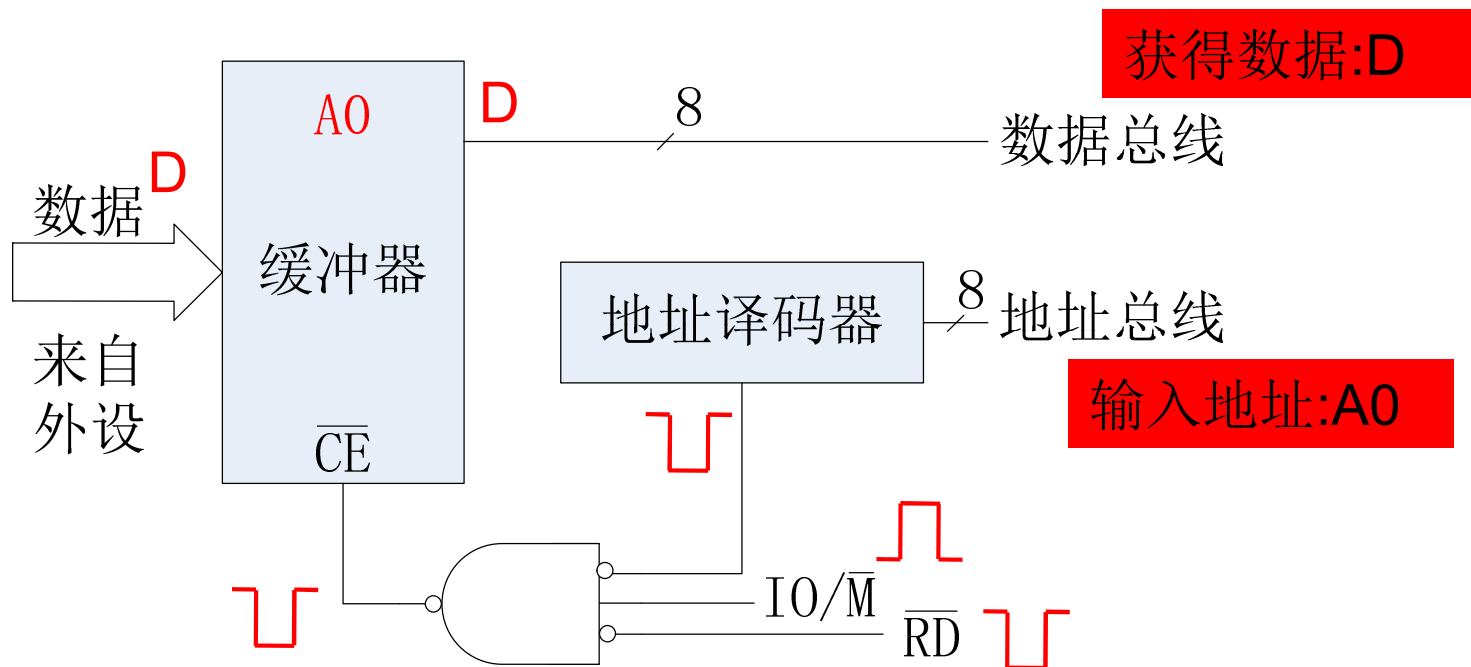
- 由于不查询外设状态，接口电路**不需要状态端口**

- 说明

- ◆通常接口在硬件上确保端口读/写操作能同步进行。

# 无条件传送——输入

- IN AL, A0H

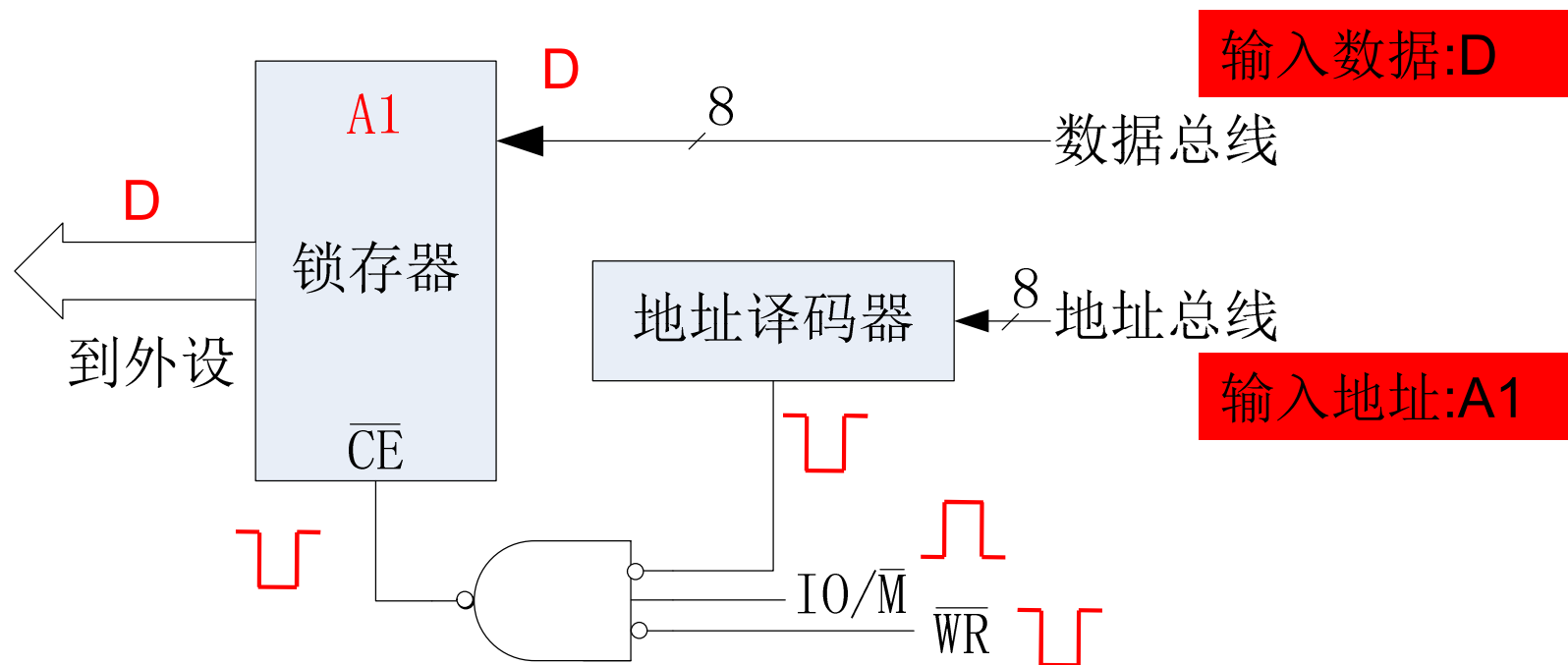


- 缓冲器：例如74LS244/245



# 无条件传送——输出

- OUT **A1H**, AL



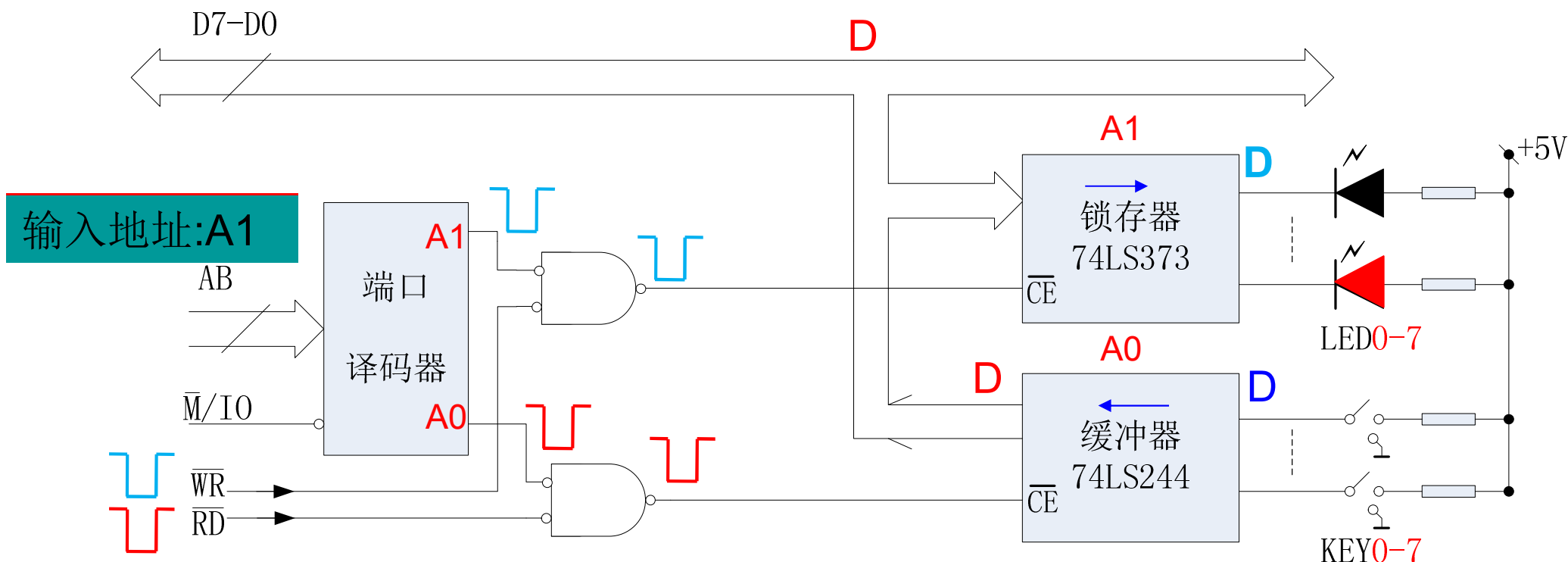
- 锁存器：例如**74LS373**

# 例：无条件传送系统

- 输入：8位开关的状态(KEY<sub>0-7</sub>)，地址A0h
- 输出：8位LED的亮/灭(LED<sub>0-7</sub>)，地址A1h
- 接口芯片：缓冲器74LS244，锁存器74LS373

IN AL, A0H

OUT A1H, AL

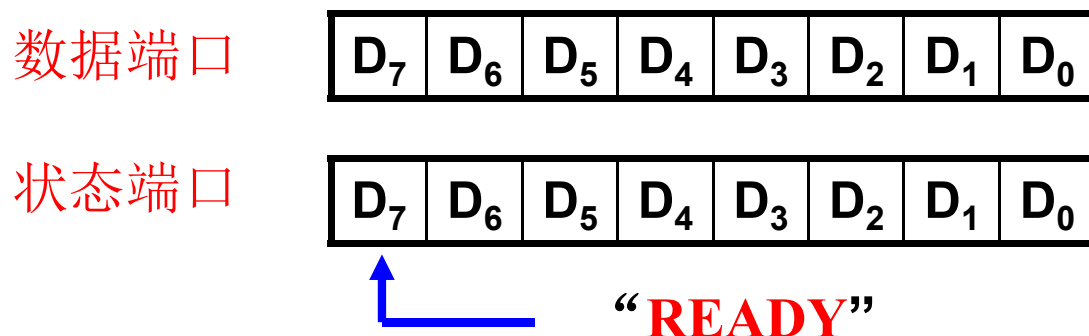


# 查询传送方式（异步传送）

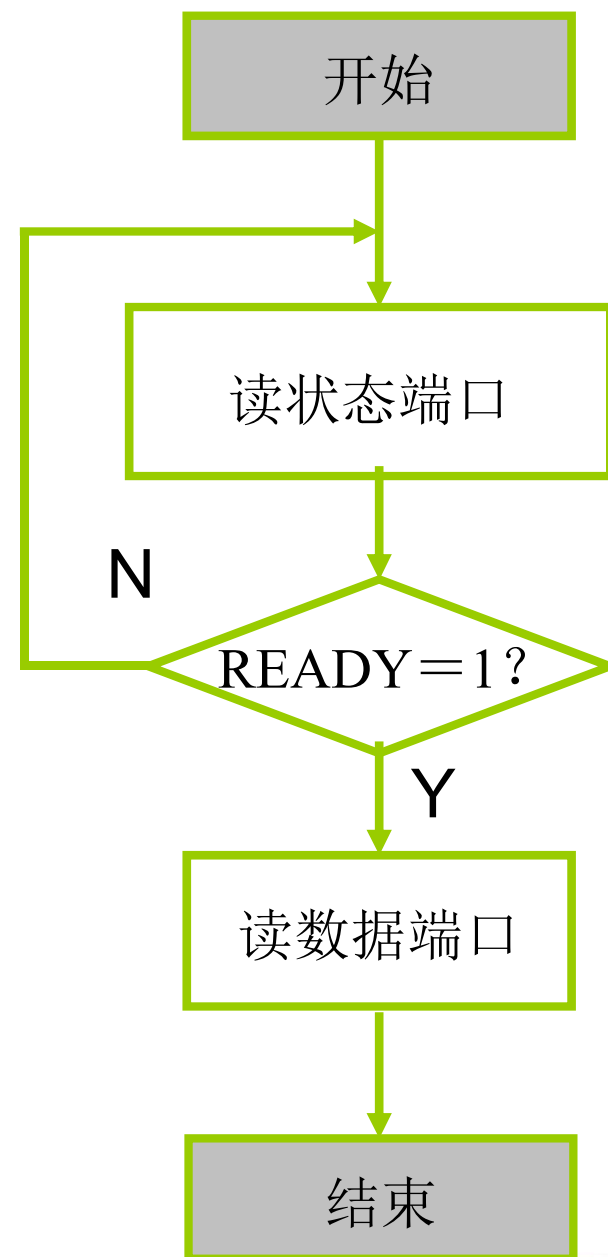
- 传送数据之前先确定外设是否准备好？！
- 传送过程
  - （1）先获取外设状态：执行IN指令读取外设状态端口
  - （2）根据外设状态判断：
    - ◆ 如果状态是“忙碌”或“未准备就绪”，则回到（1）；
    - ◆ 如果状态是“空闲”或“准备已就绪”，则continue；
  - （3）执行数据传送：对数据端口执行OUT或IN指令。
- 说明
  - 查询式接口要有数据端口和状态端口。
  - 端口一般都是8位，状态端口一般只需其中1位即可。

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

### 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



# 查询传送方式——输出

- 过程：读状态端口→读数据端口

数据端口

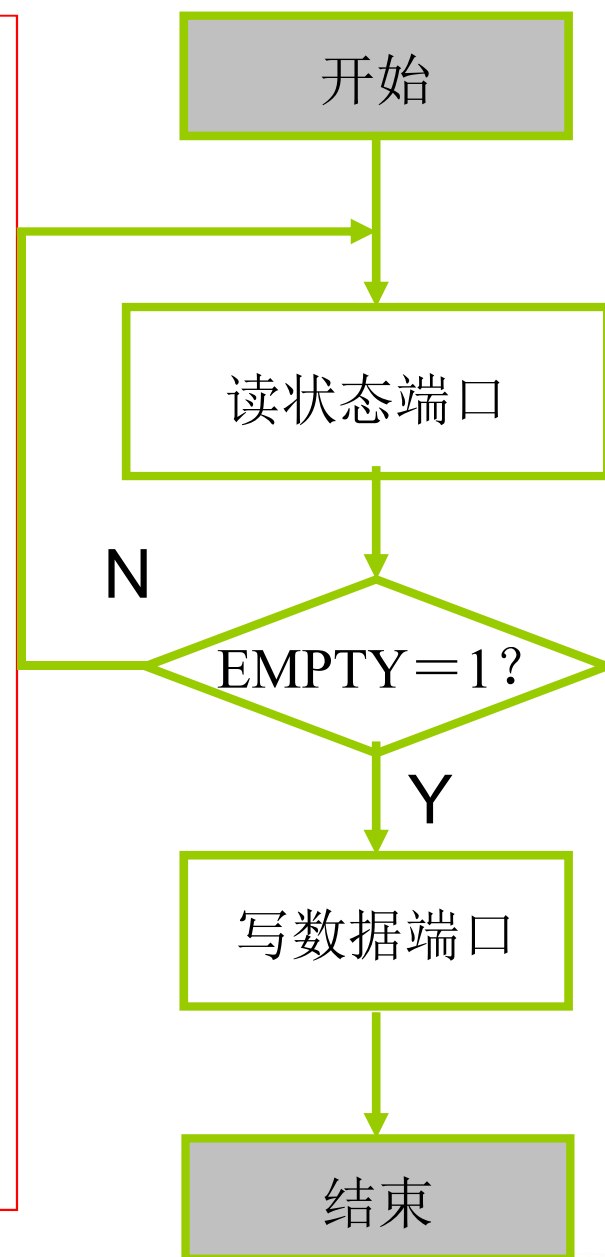
D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

状态端口

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

↑ “EMPTY”

- **EMPTY**为1时表明设备缓冲区为空（即空闲），能接收来自CPU的数据。
- 一旦OUT执行完成，设备完成了数据接收，**EMPTY**自动变0（即忙碌）



- 输出过程的典型程序

```
POLL: IN      AL, PORT_State ;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_Data, AL ;向数据寄存器中输出数据
```

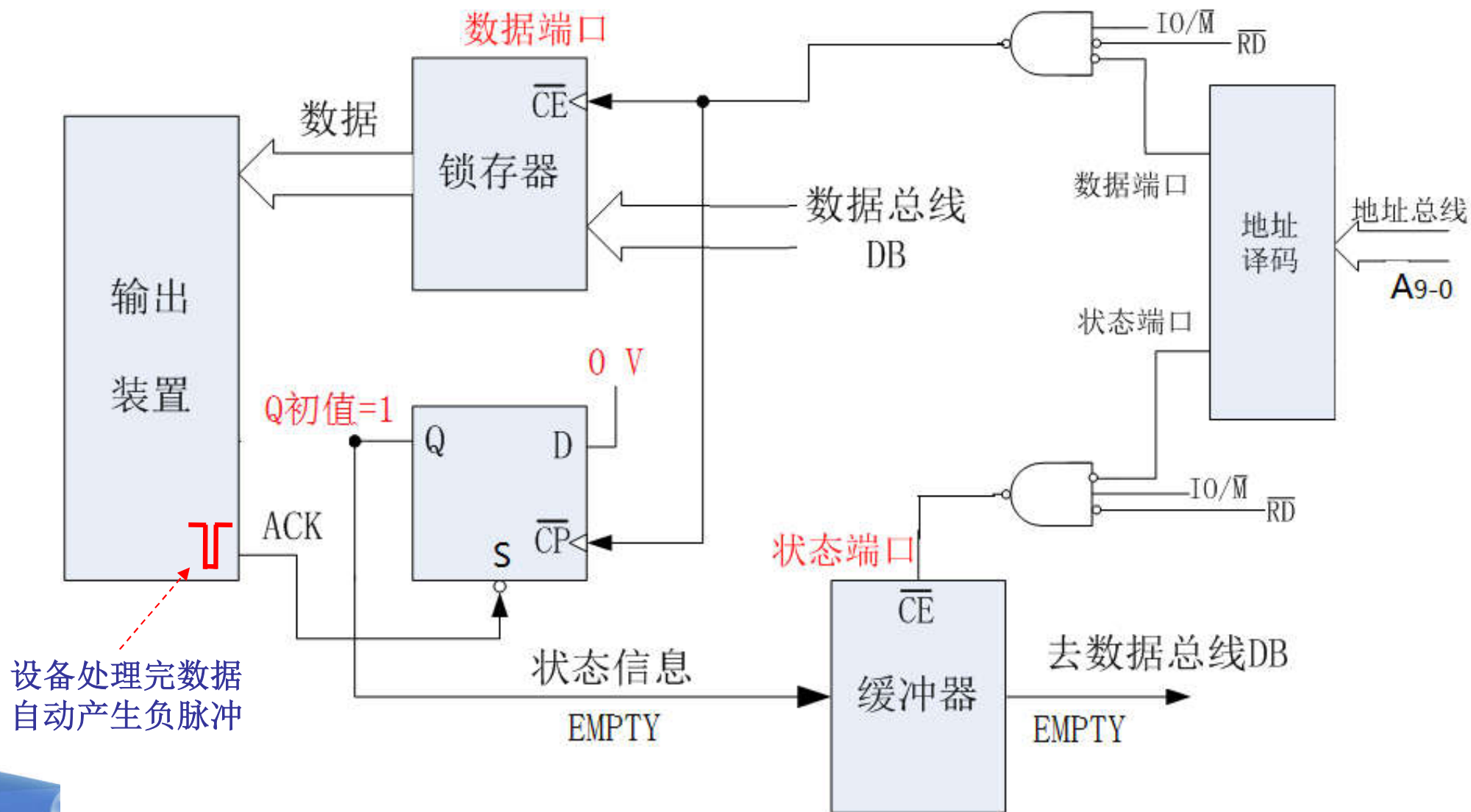
指令简介:

- (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



# 查询传送方式——输出

## ● 输出的电路原理



# 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）