

《微机原理与接口》

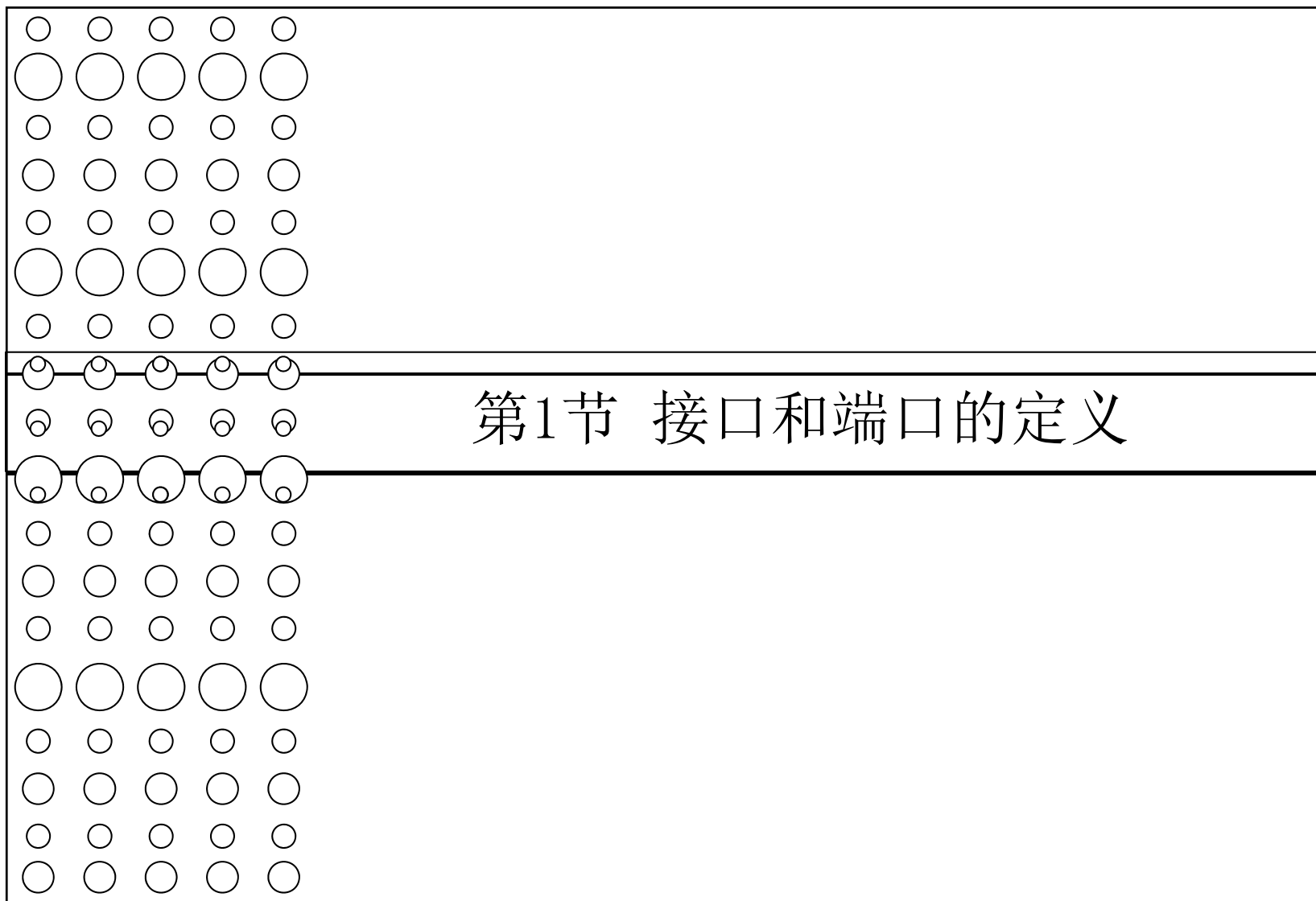
## 第3章 接口概念和原理

教师：苏曙光

华中科技大学软件学院

## ● 第三章 接口概念

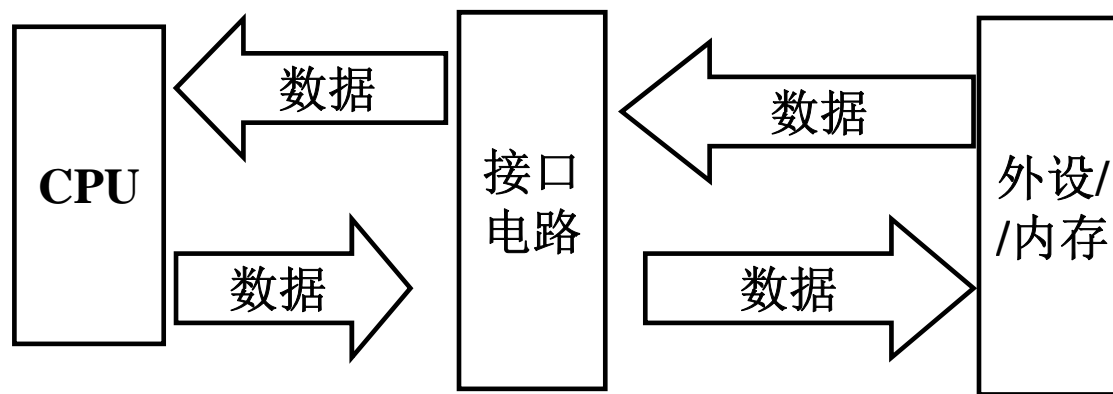
- 1. 接口和端口的定义
- 2. 端口的访问方式
- 3. 端口地址的编址
- 4. 数据传输方式
- 5. 8088输入输出综合实例
- 6. 8088的存储器



## 第1节 接口定义

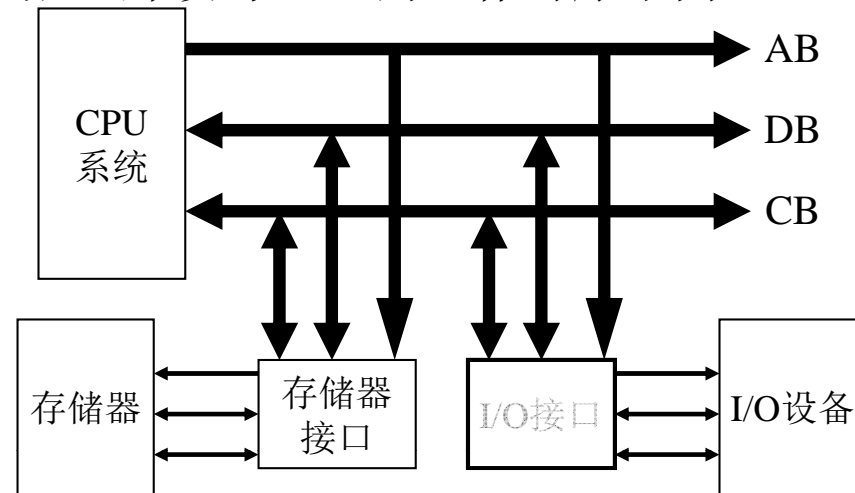
### ● 接口定义

- 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）

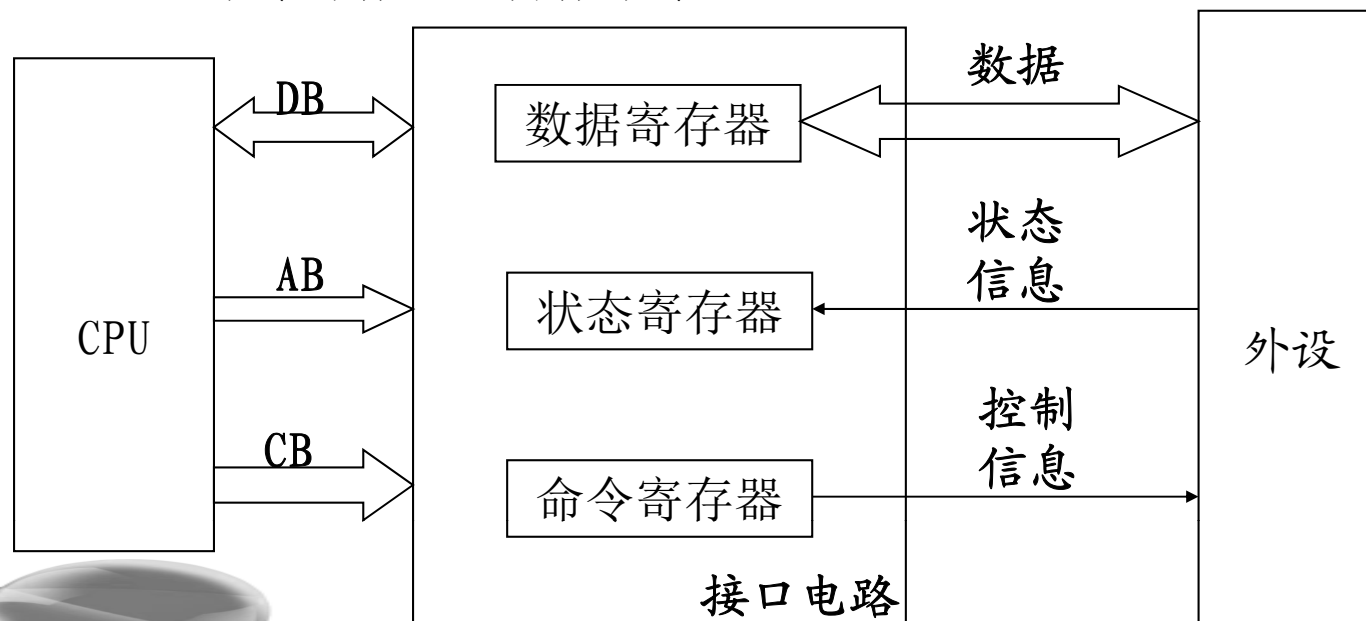
- 各种外设都必须通过接口才能和CPU（或总线）相连
  - 寻址：通过接口区分设备；
  - 缓冲：外设与CPU工作速度不一致；
  - 转换：外设与CPU信息格式、类型、幅度不一致；
  - 时序匹配：外设与CPU的工作时序不同。



## 1. 接口定义（续）

### ● 接口电路的组成

- 一组数据寄存器，暂存数据
- 一组状态寄存器，暂存状态。
- 一组命令寄存器，暂存命令。



## 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I/O芯片
  - 后768个端口：100h-3FFh, 扩展槽常规外设I/O。

# 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

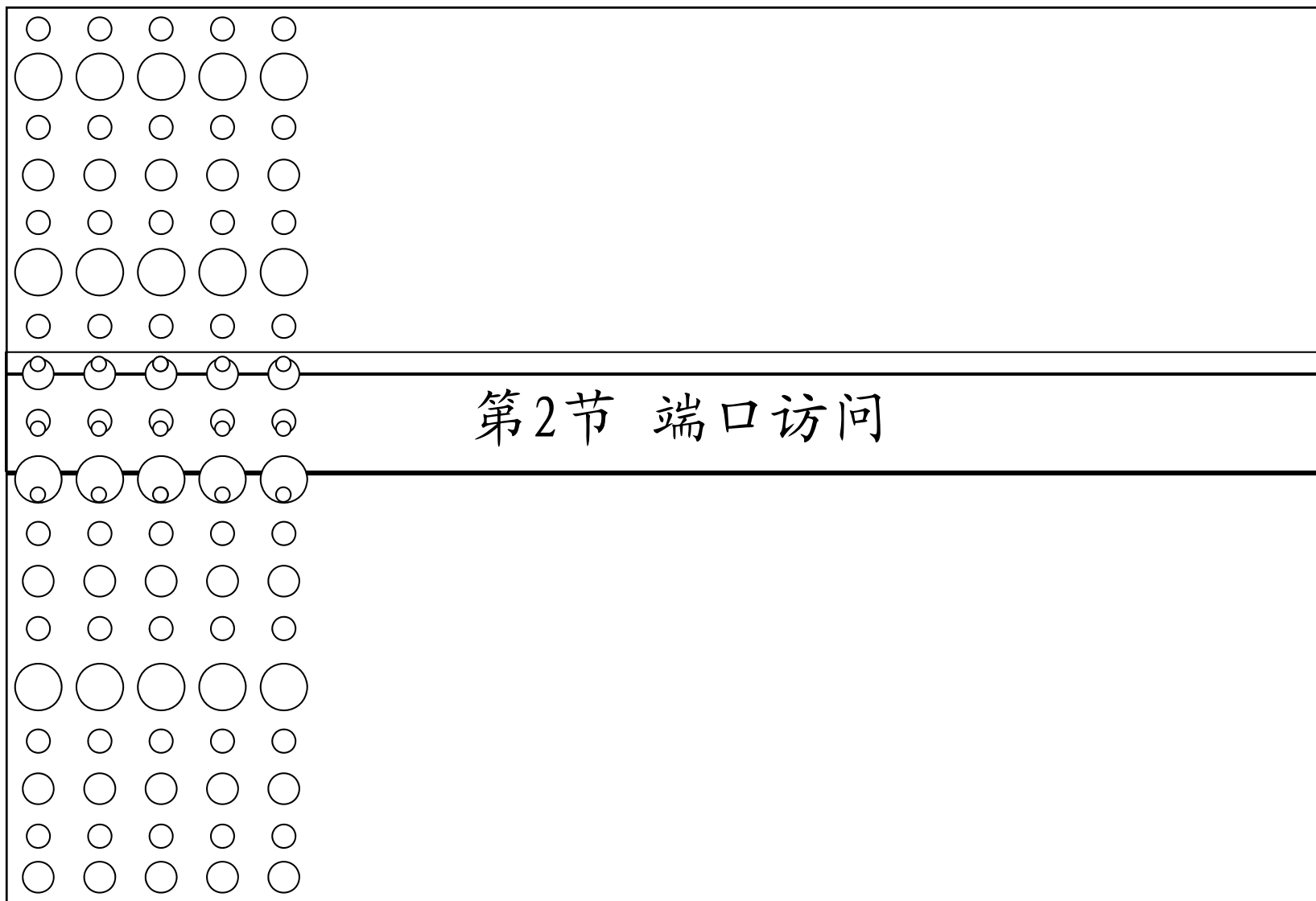
计算机管理

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



## 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), 读(IN)
- 区别设备操作

- 访问指令

- 输入指令 (读) IN
  - ◆ 从指定端口取信息送入累加器。
- 输出指令 (写) OUT
  - ◆ 把累加器的信息送往指定端口。

## 2. 端口的访问（续）

### ● 输入指令 IN

#### ■ 四种形式【PORT, DX 指端口地址】

◆ IN AL, PORT

◆ IN AX, PORT

◆ IN AL, DX

◆ IN AX, DX

#### ■ 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）

### ● 输出指令 OUT

#### ■ 四种形式【PORT, DX 指端口地址】

◆ 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

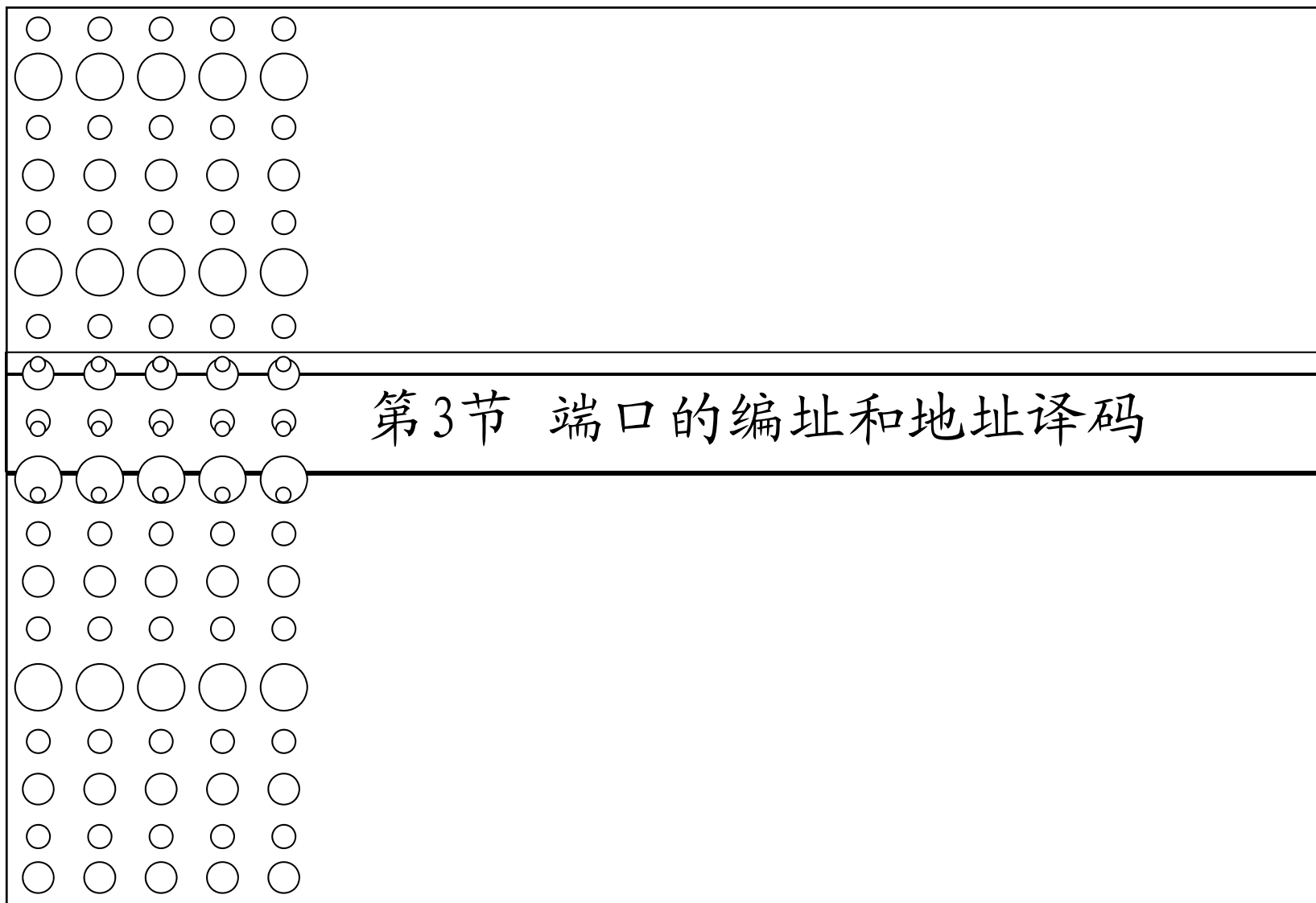
◆ OUT DX, AX ;数据是双字节; 端口地址双字节

#### ■ 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



## 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

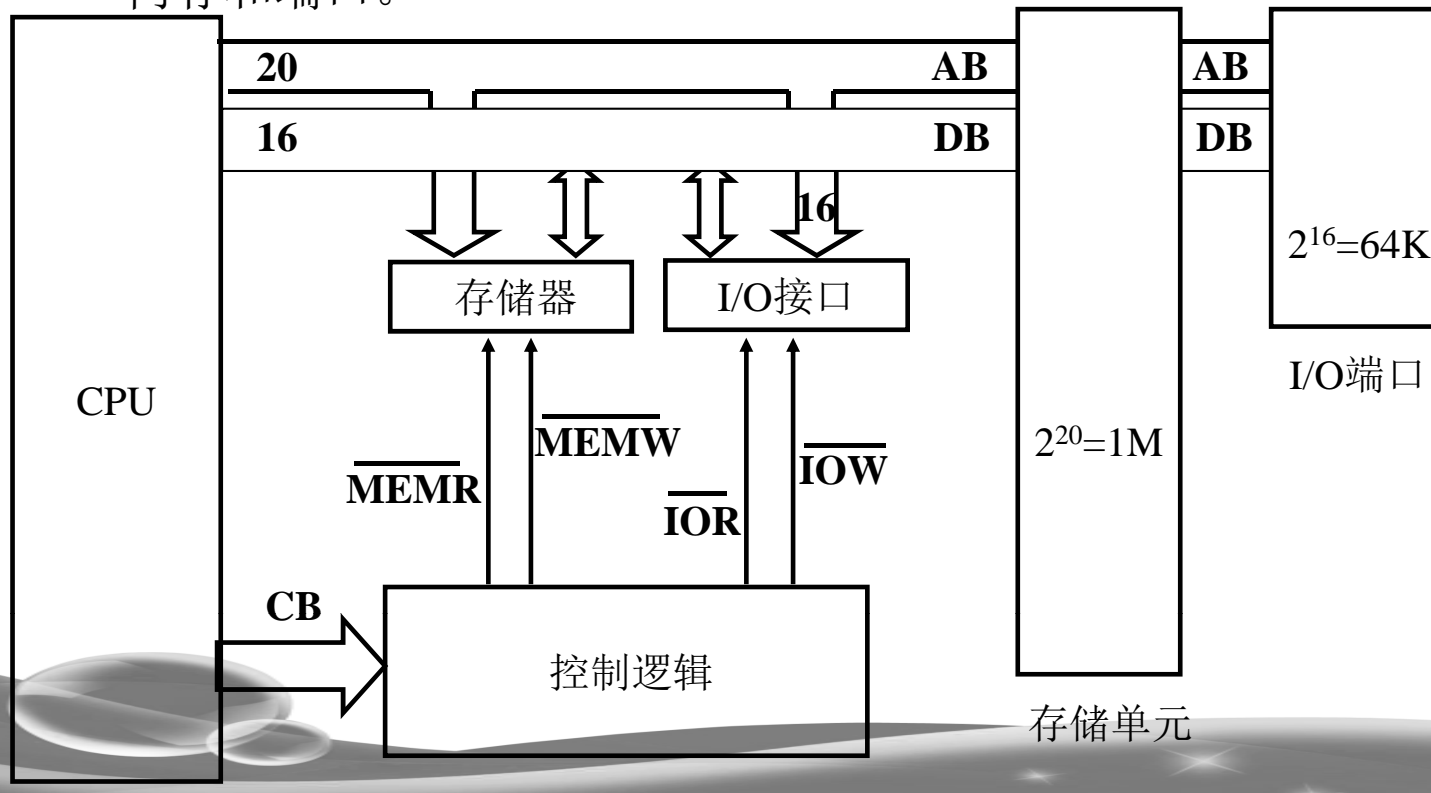
- 统一编址（存储器映射方式）

- ◆ 端口地址和存储器地址统一编址

## 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。

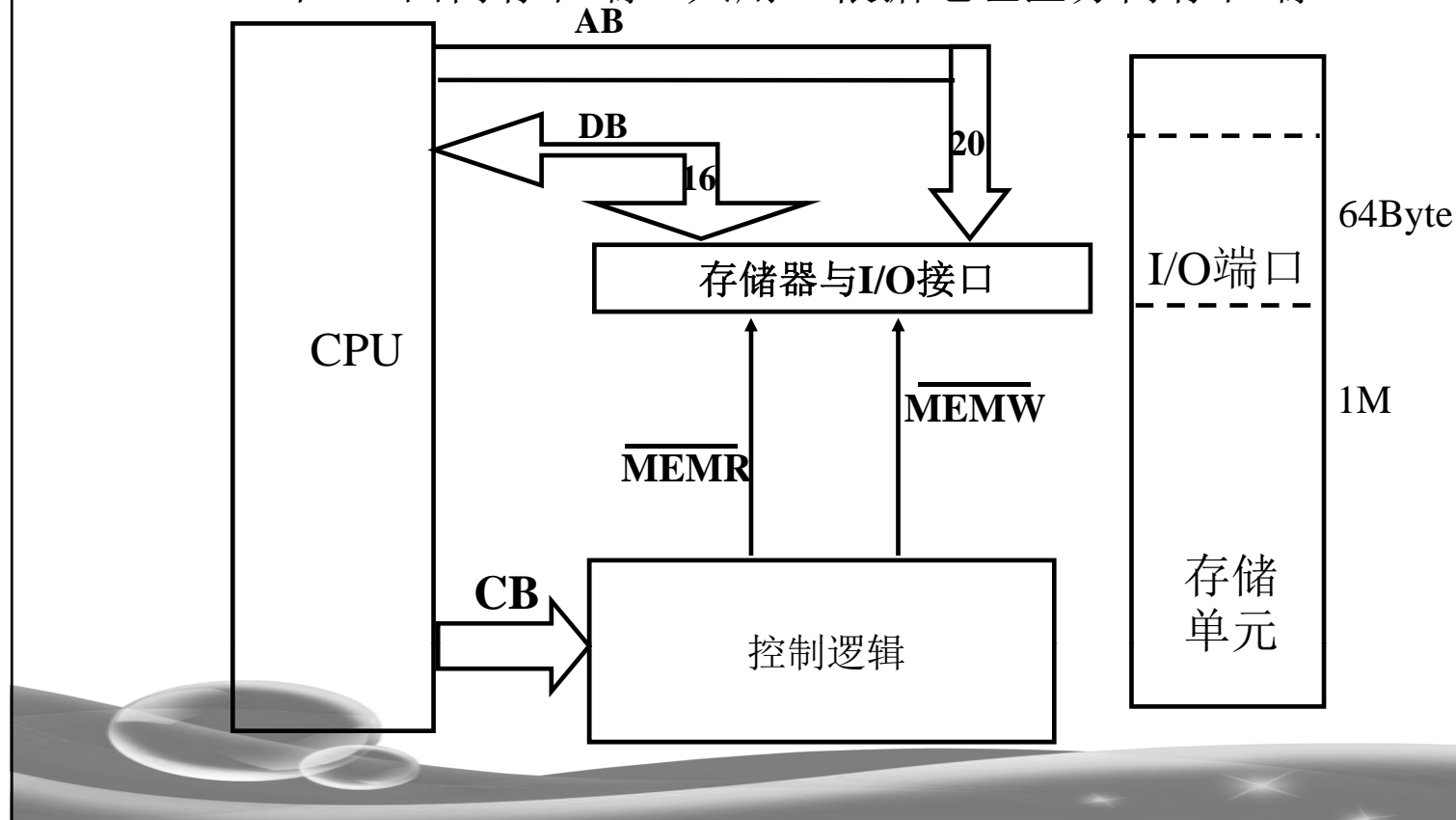




## 端口地址编址方式（续）

- 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。



## 端口地址译码

- 概念：根据给定的地址识别端口（接口/设备）。
  - 端口地址译码电路
- 三个前提
  - 有效I/O地址线10位： **$A_9 \sim 0$**
  - 考虑DMA操作：地址允许信号**AEN**
    - ◆ **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；
  - 端口读写属性（只读/只写/可读可写）

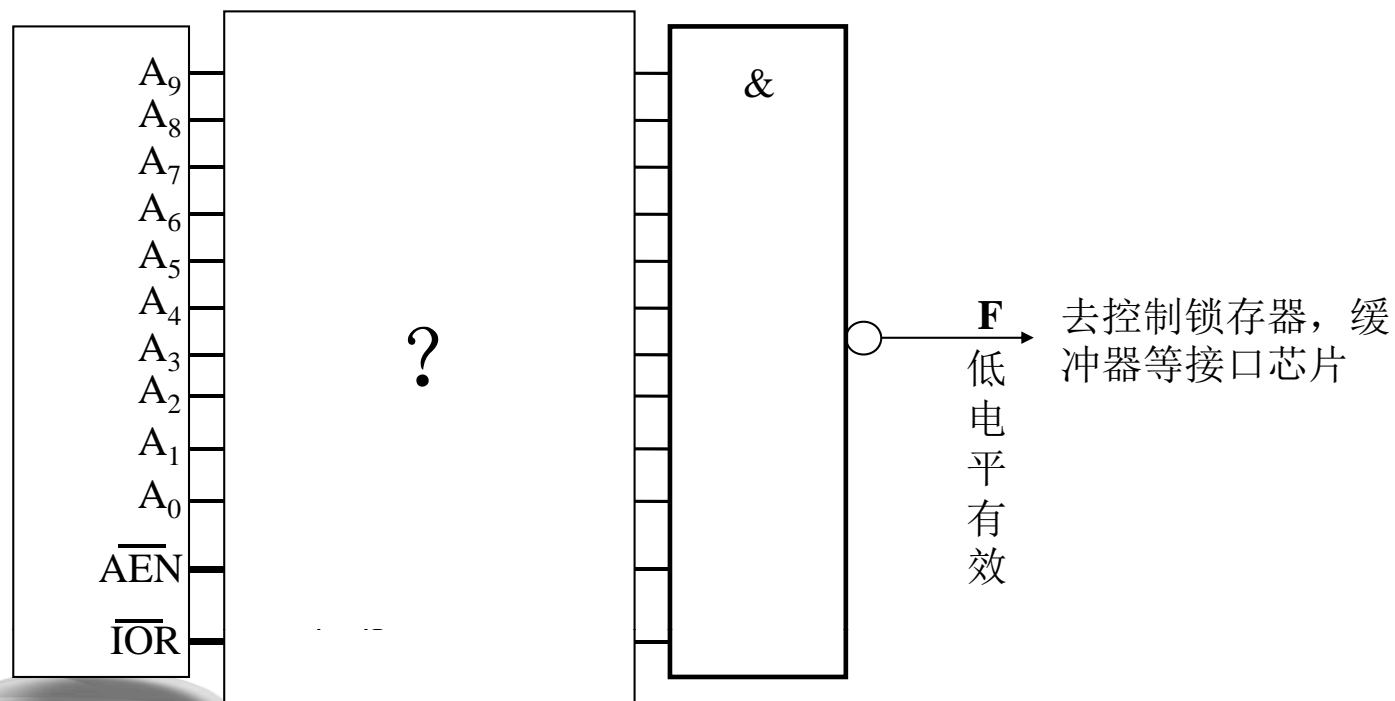
## 译码接口电路的设计例子

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



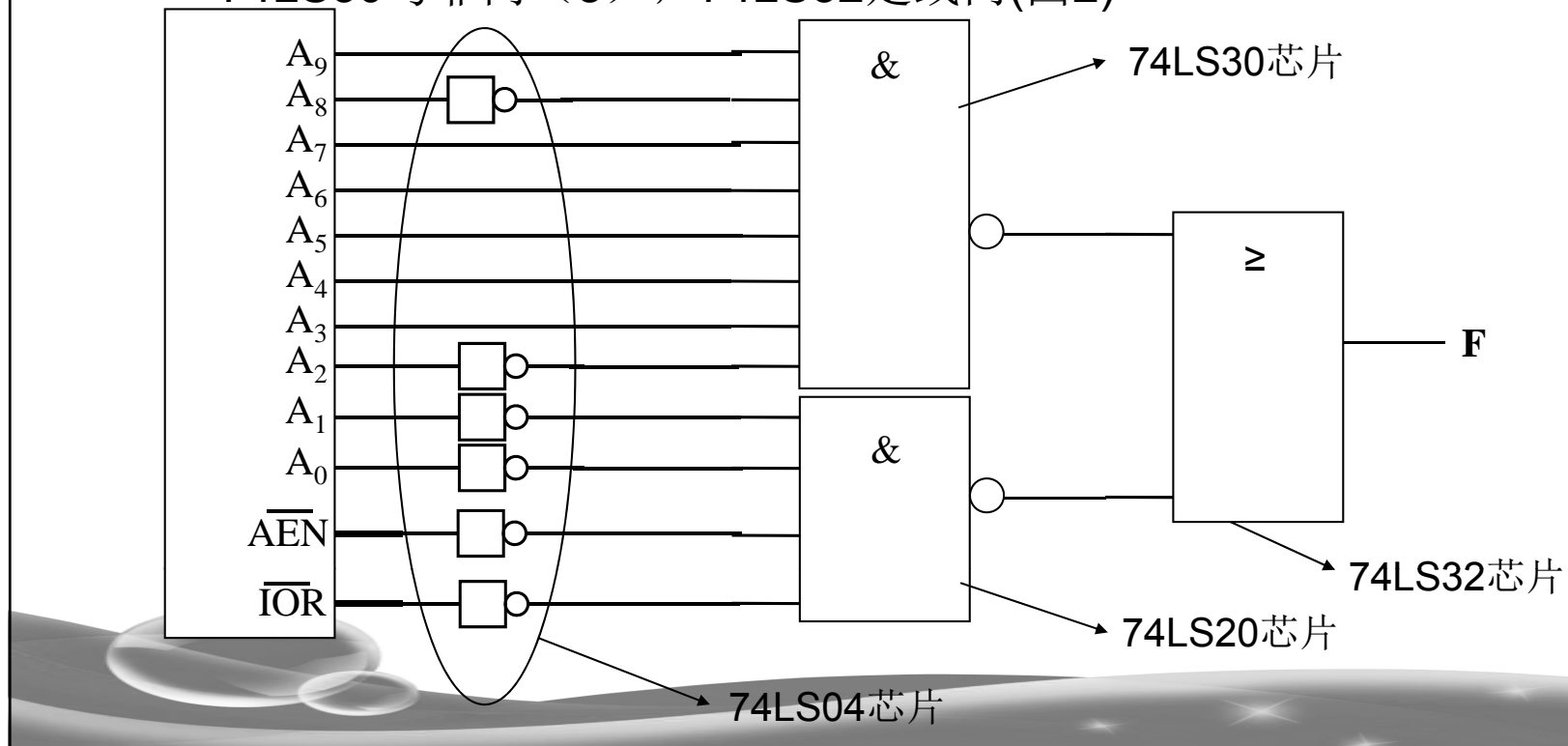
$$F = A_9 \bar{A}_8 A_7 A_6 A_5 A_4 A_3 \bar{A}_2 \bar{A}_1 \bar{A}_0 \cdot \bar{AEN} \cdot \bar{IOR}$$

## 端口地址译码

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

■ 74LS04非门（6），74LS20与非门（二4）

■ 74LS30与非门（8），74LS32是或门(四2)



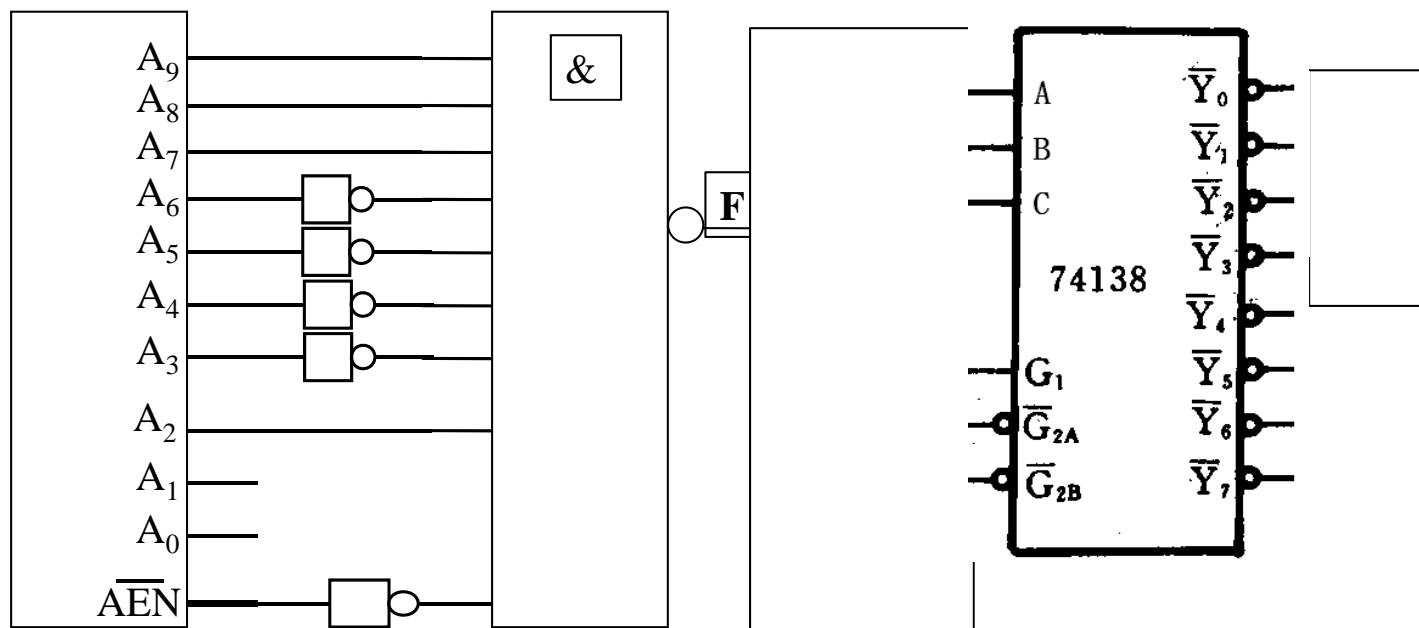
## 端口地址译码(续)

- 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) 改成74LS04/20/30/32等芯片构成
- 步骤
  - ◆ 第1步：选中接口
  - ◆ 第2步：选中接口中的某个端口

## 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A1,A0分别连到138的A, B引脚，结果会怎样？Y0,Y1,Y2, Y3的地址是多少？

## 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

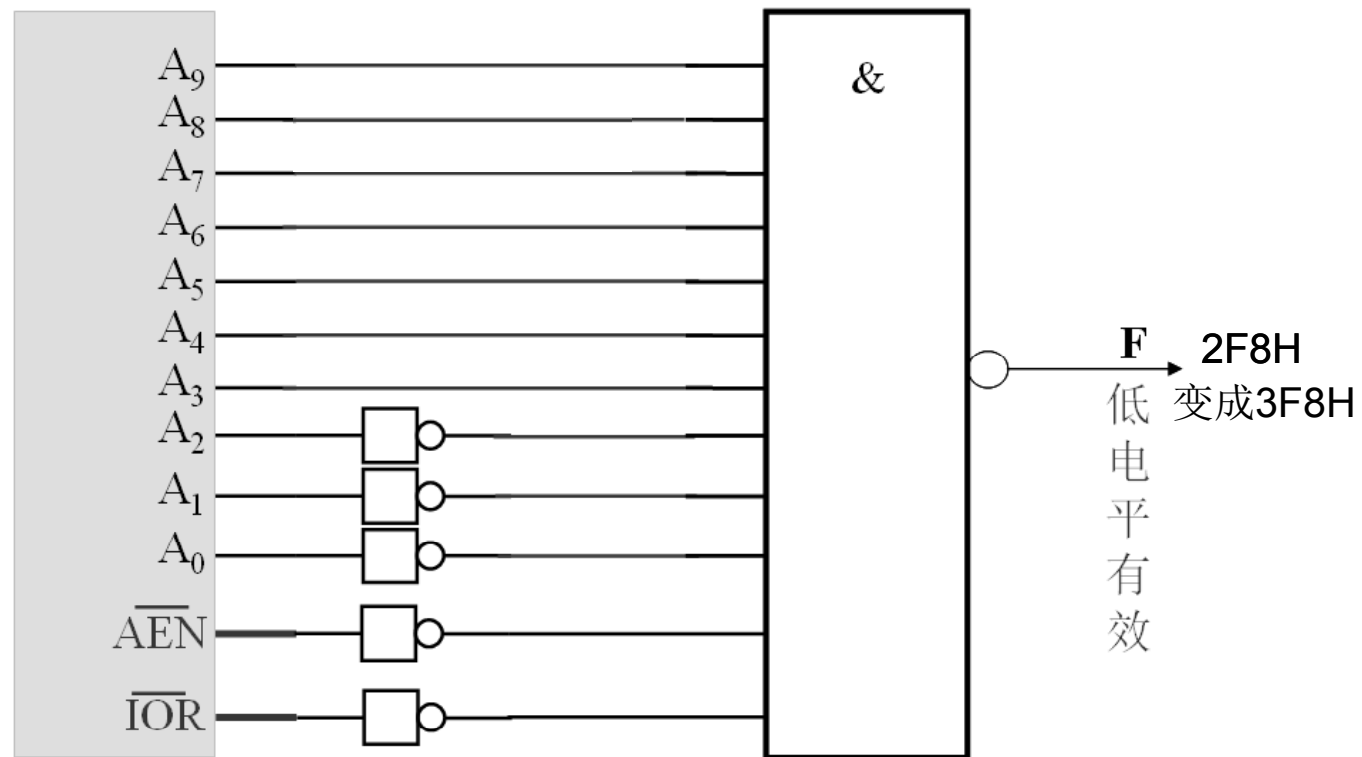
- 译码器（简洁，可靠）

- 地址可变

- ◆ 通过跳线或编程改变端口的地址

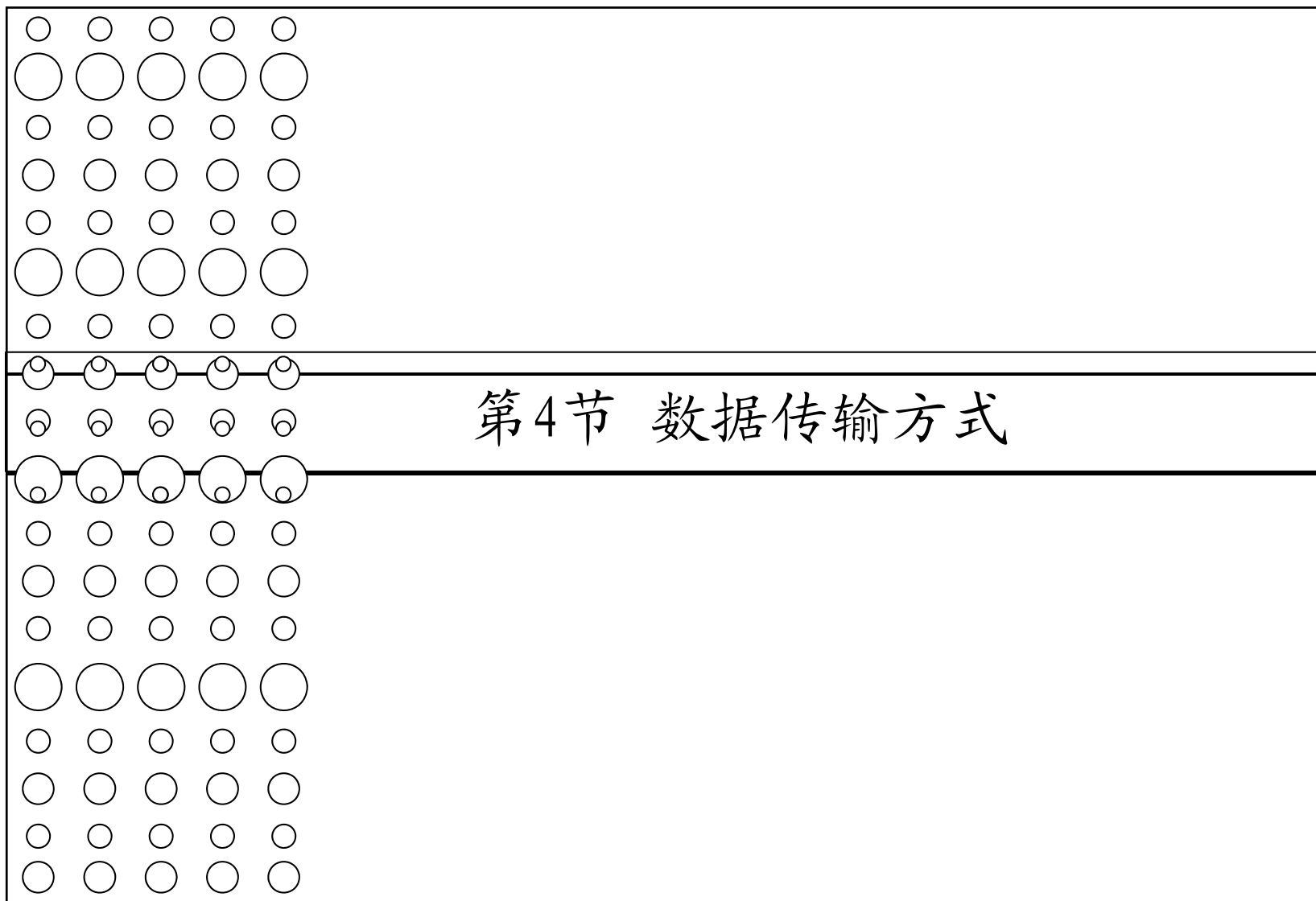


- 通过跳线或编程改变端口的地址



● 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



- 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

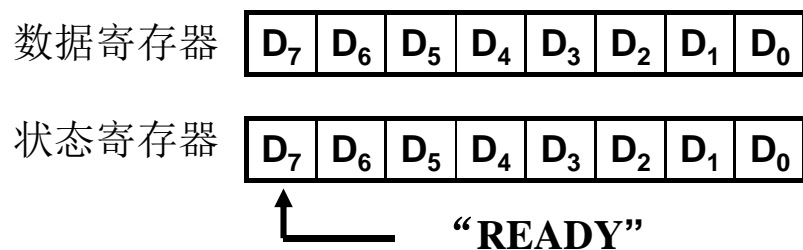
- ◆ 通常接口硬件确保端口读写操作能同步进行。

## 查询传送方式（异步传送）

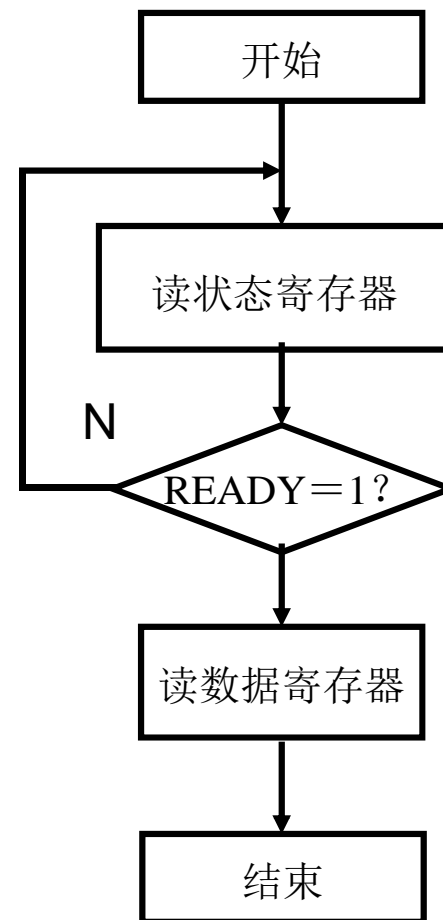
- 传送数据之前先确定外设是否准备好？！
- 传送过程
  - （1）先获取外设状态：执行IN指令读取外设状态端口
  - （2）根据外设状态判断：
    - ◆ 如果状态是“忙碌”或“未准备就绪”，则回到（1）；
    - ◆ 如果状态是“空闲”或“准备已就绪”，则continue；
  - （3）执行数据传送：对数据端口执行OUT或IN指令。
- 说明
  - 查询式接口要有数据端口和状态端口。
  - 端口一般都是8位，状态端口一般只需其中1位即可。

## 查询传送方式——输入

- 过程：读状态寄存器→读数据寄存器。



- 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    ; 读状态寄存器: 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    ; 读数据寄存器: 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
- (2) JZ: 等于0则转移



## 查询传送方式——输出

- 过程：读状态寄存器→读数据寄存器

数据寄存器 

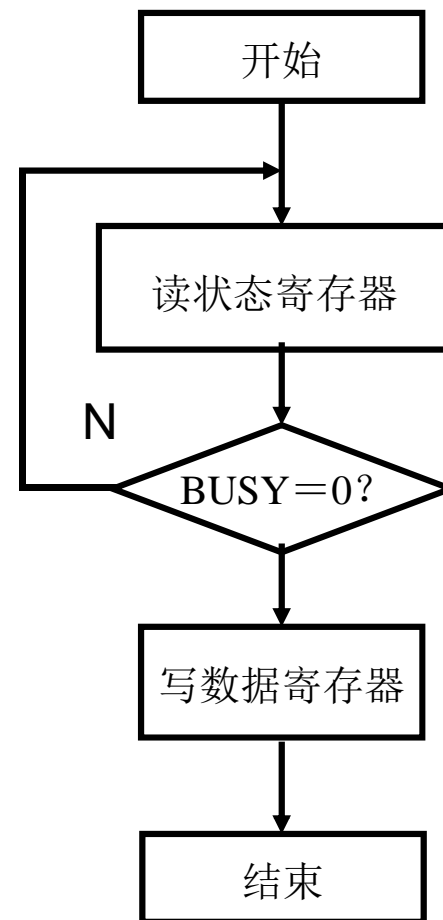
D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

状态寄存器 

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>
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

↑  
“BUSY”

- BUSY为0时表明设备空闲，能接收来自CPU输出的数据。
- 一旦OUT指令执行完成，CPU完成数据输出，BUSY自动变1。



● 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BUSY位是否为0
        JNZ  POLL        ; 外设忙碌转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:

**TEST:** AND,不影响操作数, 仅影响标志位: ZF

**JNZ:** 不等于0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）