微机原理与接口技术

第六章 基本输入输出接口

第六章 基本输入输出接口

- 教学重点

- I/O接口电路的典型结构
- 无条件传送方式
- 查询传送方式
- 中断工作过程

（1） I/O接口概述

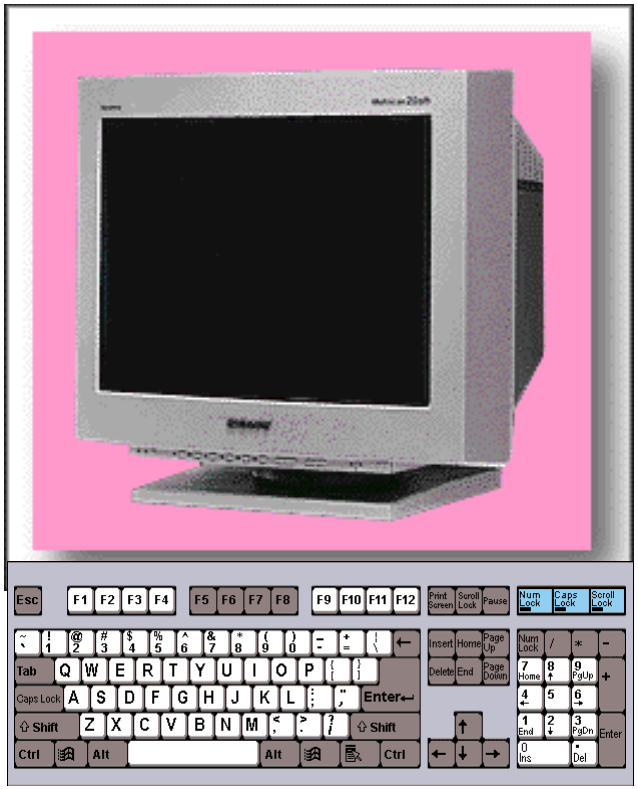
- 不同的外部设备均需要与CPU交互
- **工作原理不同**

  机械、电子、机电、电磁……
- **传送信息类型多样**

  数字量、模拟量、开关量
- **传送速度差别极大**
- **传送方式不尽相同**

  串行、并行
- **编码方式不同**

  二进制、BCD码、ASCII码……



（1） I/O接口概述

为什么需要I/O接口（电路）？

- 微机的外部设备多种多样，工作原理、驱动方式、信息格式、以及工作速度方面彼此差别很大。

- 它们不能与CPU直接相连，必须经过中间电路再与系统相连，这部分电路被称为I/O接口电路。

接口存在的必要性

- **1）信号转换：**
- 外部设备可能使用**模拟信号、数字信号**，对于数字信号，**2进制、N进制**都有可能，且**电平定义也不一致**；而计算机系统仅使用2进制数字信号，且系统总线信号的电平定义是统一的。

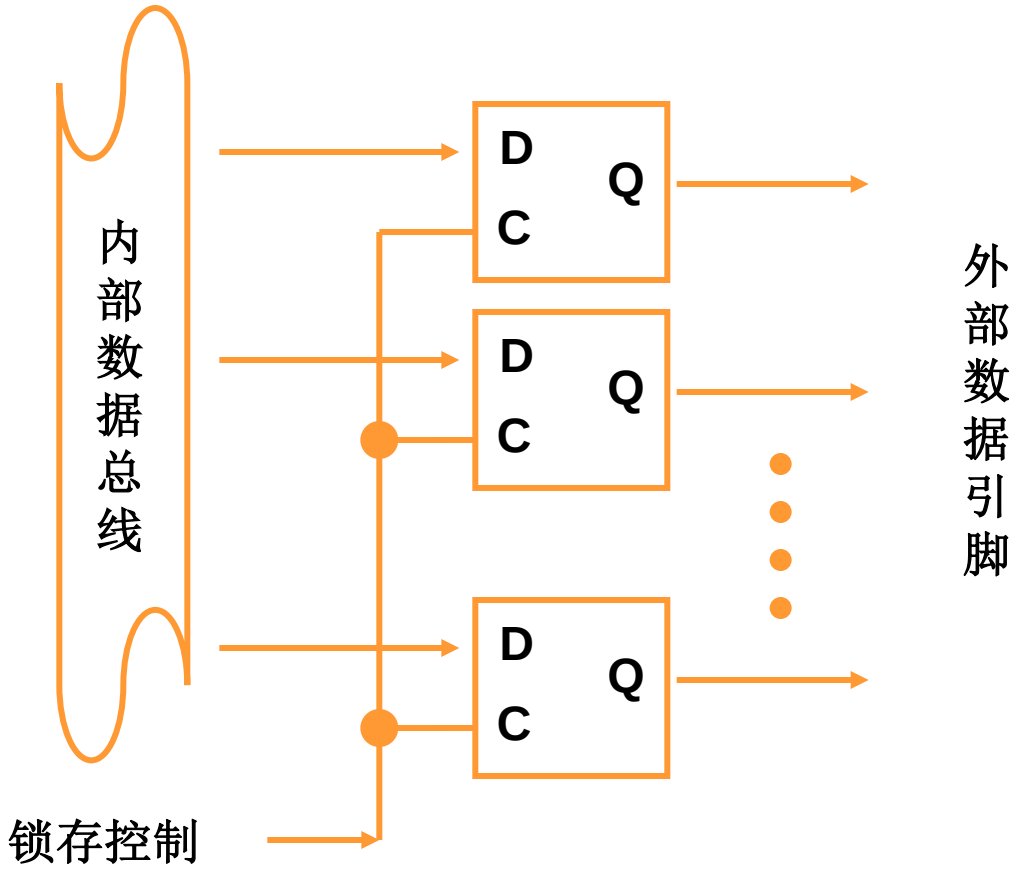
- 需要一种中间电路连接总线与外部设备，并要求这种中间电路能够完成总线与外部设备间的信号转换。

接口存在的必要性

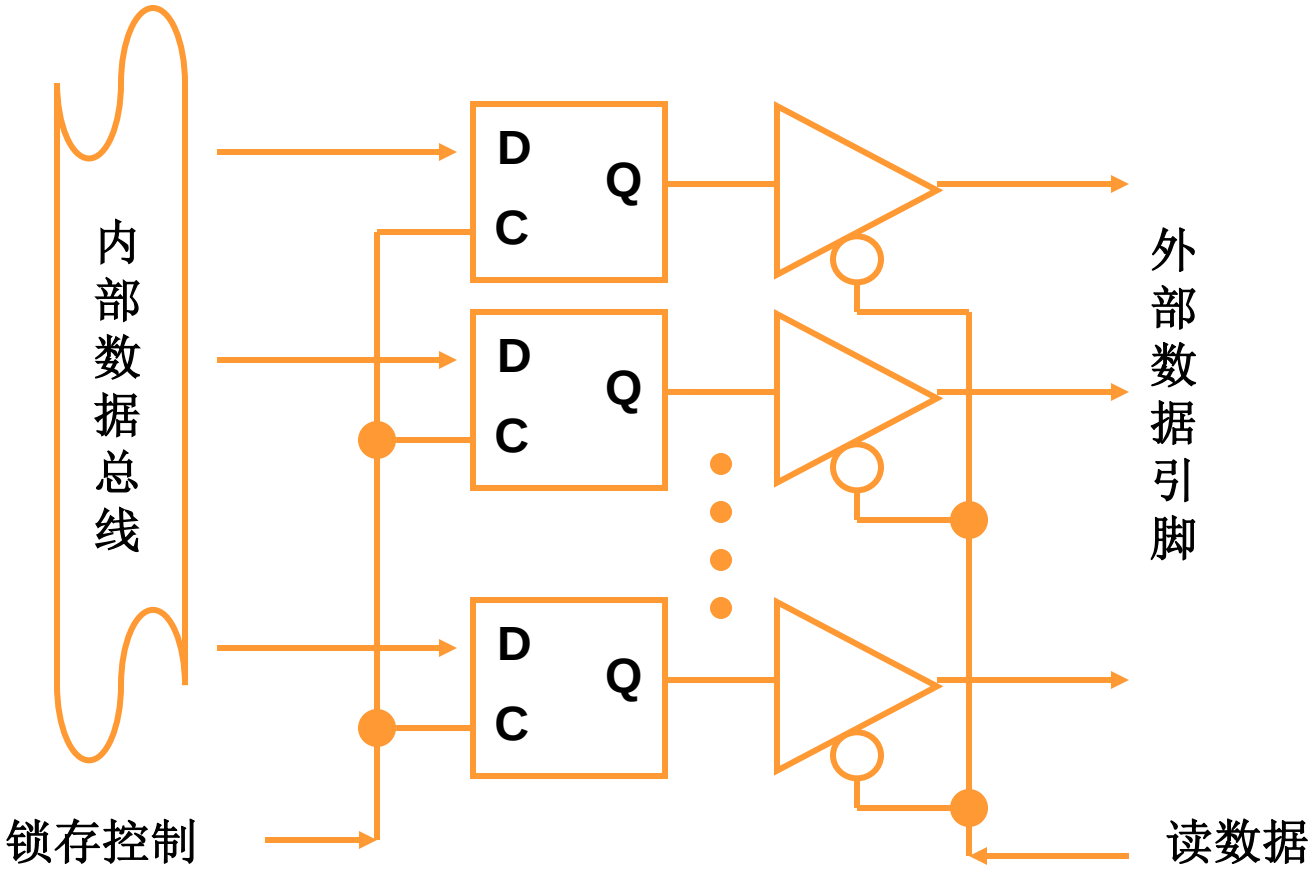
- **2）数据缓冲：**外部设备与CPU间存在速度差异，直接进行数据交换会导致数据丢失。

- 需要提供一种中间电路，为CPU与外部设备间的数据交换提供同步与缓冲，避免数据丢失。

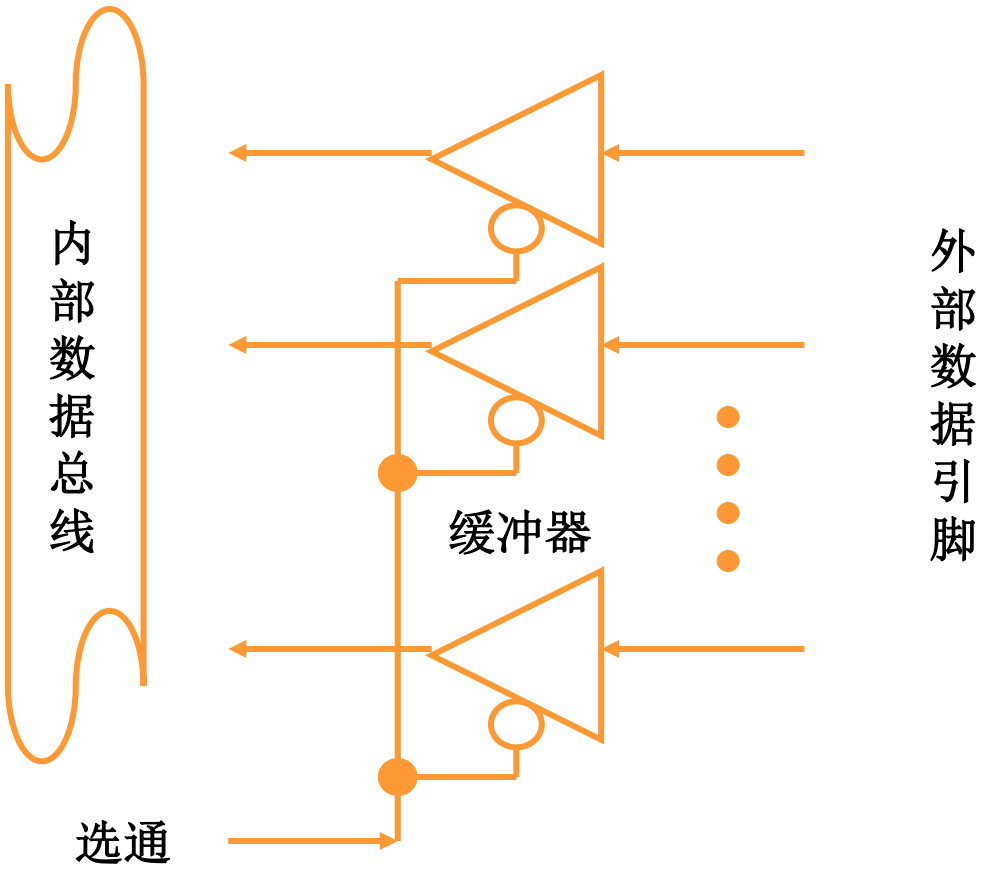
接口的输出锁存环节



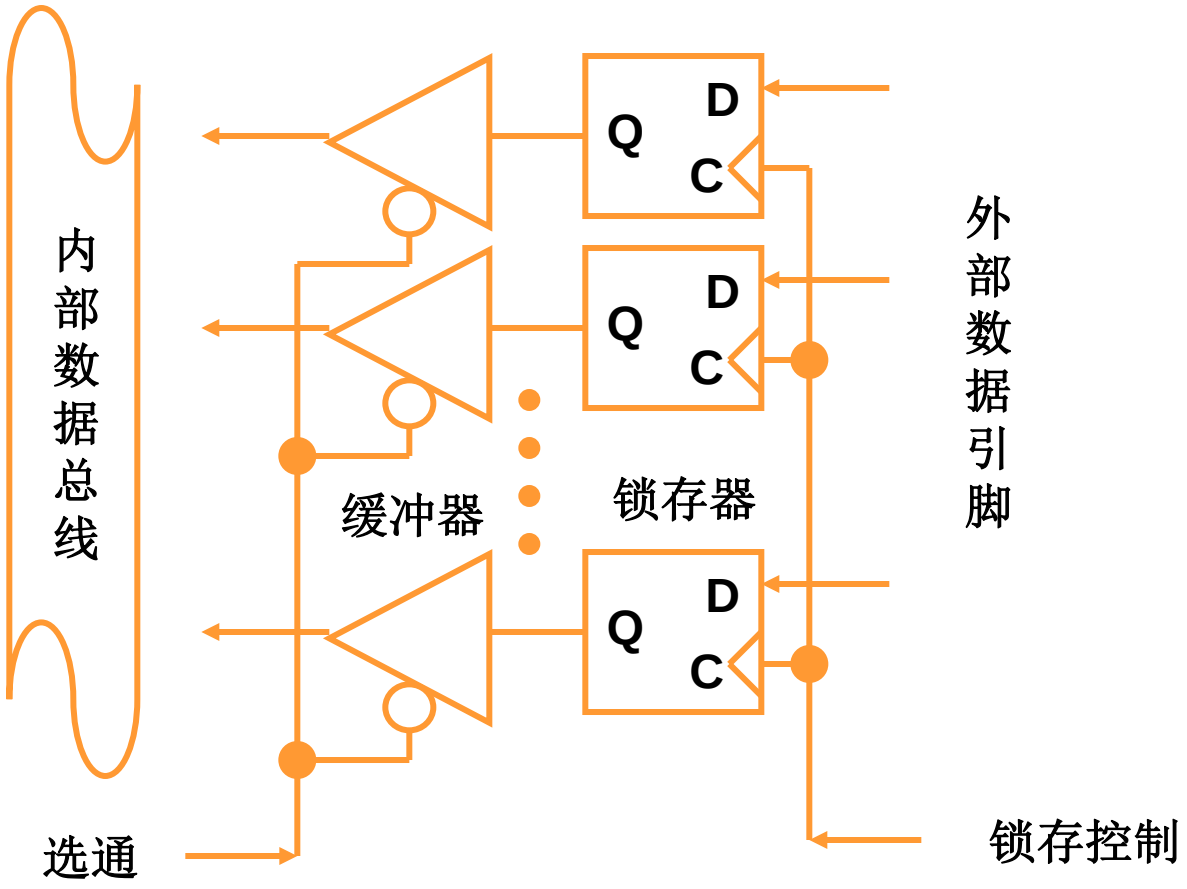
接口的输出锁存、缓冲环节



接口的输入缓冲环节

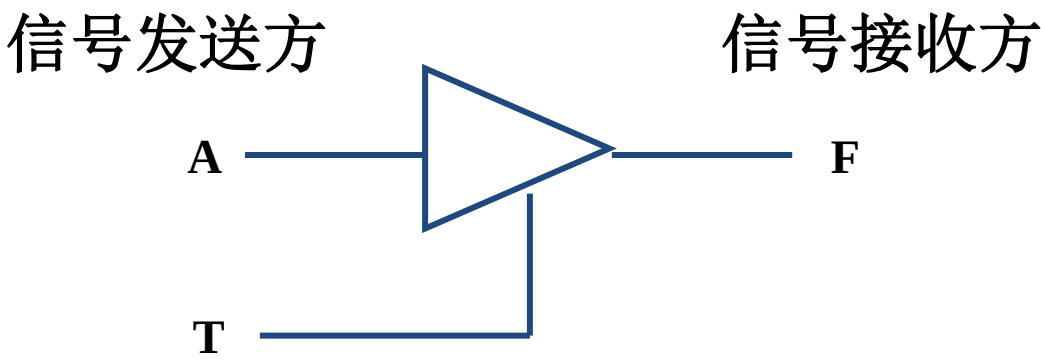


接口的输入锁存、缓冲环节



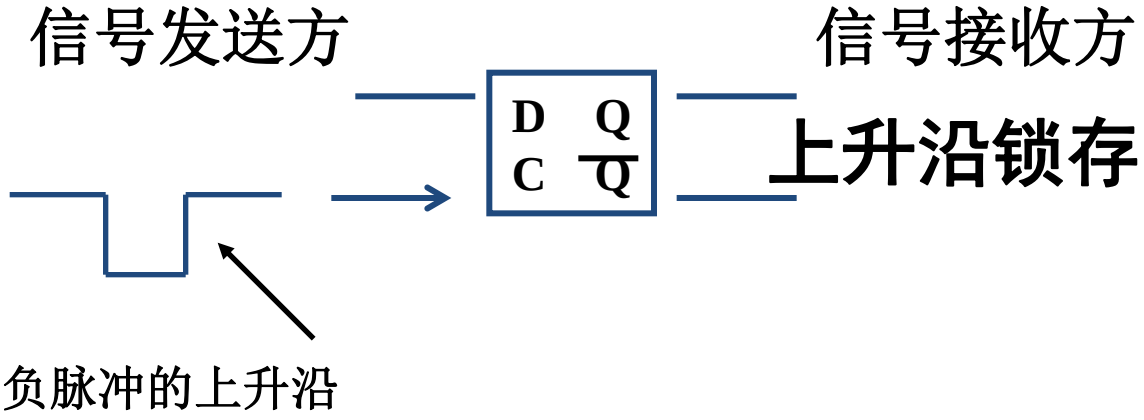
缓冲环节的功能

- 1）缓冲功能：信号发送方保持信号，直到接收方准备好接收为止，此功能为接收方提供缓冲。
- 2）隔离功能：如果有多个信号发送方，则接收方可以通过缓冲环节从各发送方依次接收信号。

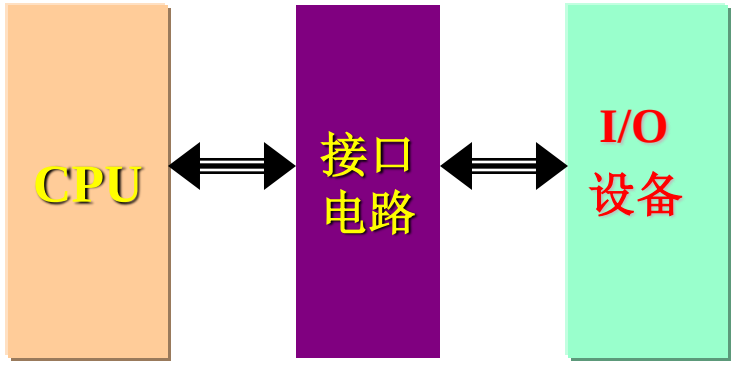


锁存环节的功能

- 暂存功能：通过锁存环节，发送方将信号暂时锁存并保持，之后发送方不再继续保持信号，此功能保证发送方的工作效率。



（1） I/O接口概述



什么是I/O接口（电路）？

- I/O接口是位于系统与外设间、用来协助完成数据传送和控制任务的逻辑电路
- PC机系统板的可编程接口芯片、I/O总线槽的电路板（适配器）都是接口电路

（1） I/O接口概述

什么是微机接口技术？

- 处理微机系统与外设间联系的技术
- 具有软硬结合的特点
- 根据应用系统的需要，使用和构造相应的接口电路，编制配套的接口程序，支持和连接有关的设备

I/O接口的主要功能

(1) 对输入输出数据进行缓冲和锁存

输出接口有**锁存环节**，输入接口有**缓冲环节**

实际的电路常用：

输出锁存缓冲环节，输入锁存缓冲环节

(2) 对信号的形式和数据的格式进行变换

微机直接处理：**数字量、开关量、脉冲量**

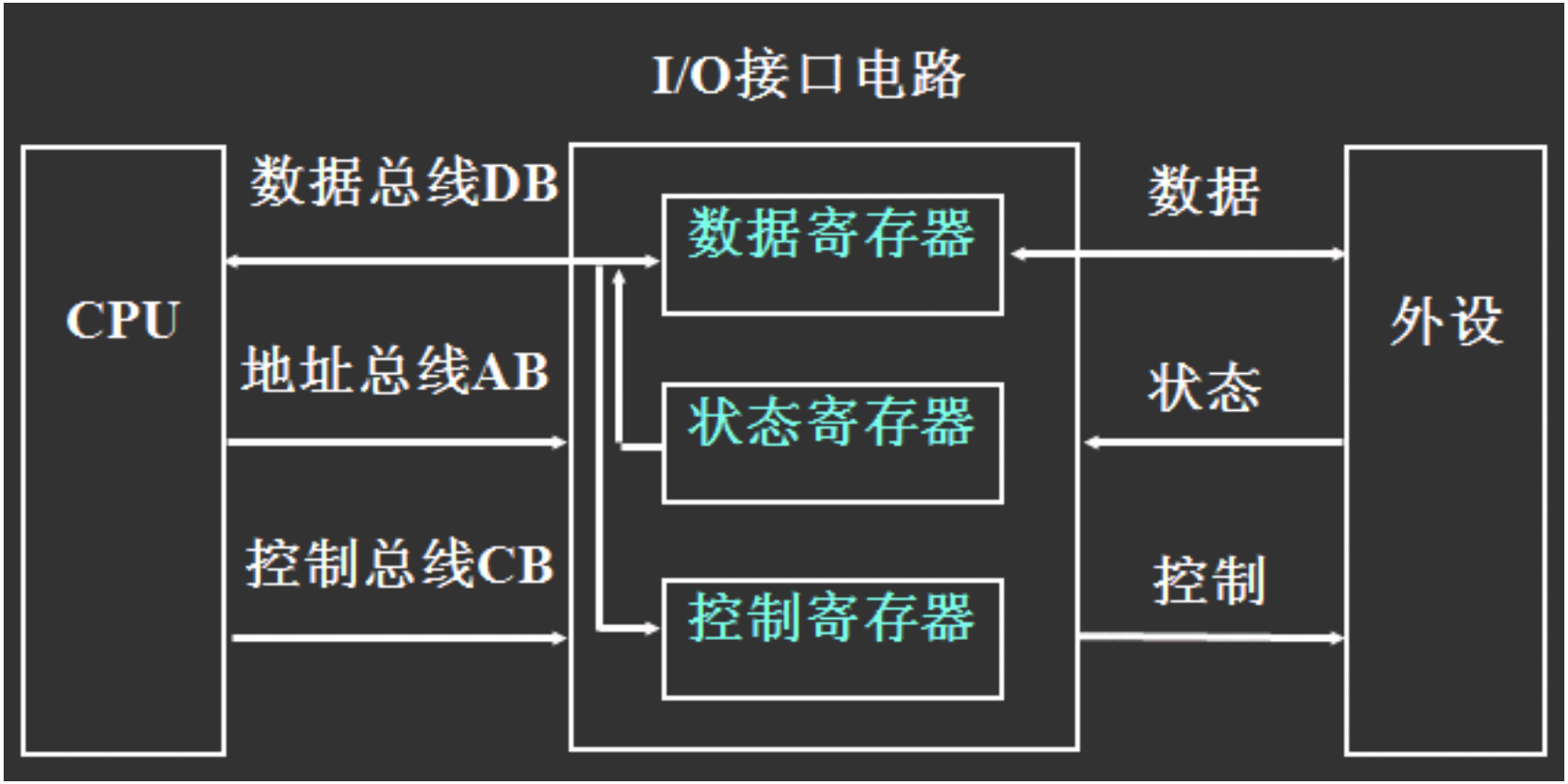
(3) 对I/O端口进行寻址

(4) 与CPU和I/O设备进行联络

（2） I/O接口的典型结构（I/O接口的共通性质）

1） 接口电路的内部结构
2） 接口电路的外部特性
3） 接口电路芯片的分类
4） 接口电路的可编程性

I/O接口的典型结构（内部、外部）



1）接口电路的内部结构

- CPU与外设主要有数据、状态和控制信息需要相互交换，于是从应用角度看内部：

数据寄存器

– 保存外设给CPU和CPU发往外设的数据

状态寄存器

– 保存外设或接口电路的状态

控制寄存器

– 保存CPU给外设或接口电路的命令

2）接口电路的外部特性

- 主要体现在引脚上，分成**两侧信号**

面向CPU一侧的信号：

- 用于与CPU连接
- 主要是数据、地址和控制信号

面向外设一侧的信号：

- 用于与外设连接
- 提供的信号种类繁多
- 功能定义、时序及有效电平等差异较大

3）接口电路芯片的分类

- 接口电路核心部分往往是一块或数块大规模集成电路芯片（接口芯片）：
- **通用接口芯片**
  - 支持通用的数据输入输出和控制的接口芯片
- **面向外设的专用接口芯片**
  - 针对某种外设设计、与该种外设接口
- **面向微机系统的专用接口芯片**
  - 与CPU和系统配套使用，以增强其总体功能

4）接口电路的可编程性

- 许多接口电路具有**多种功能和工作方式**，可以通过编程的方法**选定其中一种**
- 接口需要进行物理连接，还需要编写接口软件
- 接口软件有两类：
  - **初始化程序段**——设定芯片工作方式等
  - **数据交换程序段**——管理、控制、驱动外设，负责外设和系统间信息交换

（3）I/O端口的编址

- 端口的概念：
- **端口**泛指I/O地址，通常对应寄存器
- 一个接口电路可以具有多个I/O端口，每个端口用来保存和交换不同的信息

（3）I/O端口的编址

- 数据寄存器、状态寄存器和控制寄存器占有的I/O地址常依次被称为**数据端口**、**状态端口**和**控制端口**，用于保存数据、状态和控制信息

- 输入、输出端口可以是同一个I/O地址：
- 1）1个物理端口可读可写
- 2）1个物理端口只读，另1个物理端口只写，可公用同一地址

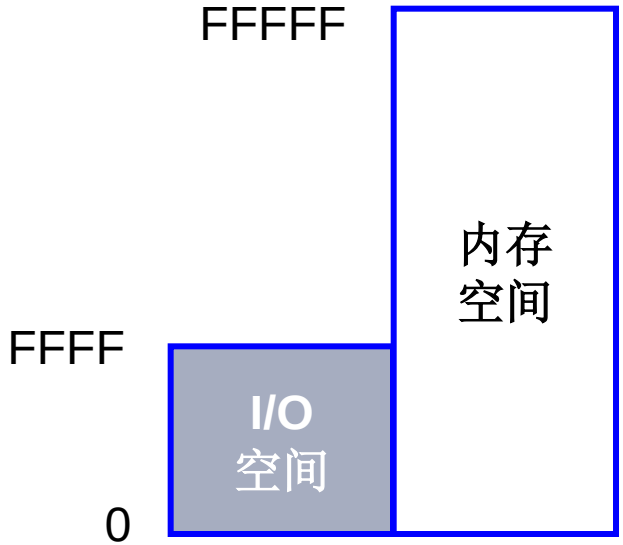
（3）I/O端口的编址

接口电路占用的I/O端口有两类编排形式

- I/O端口**独立编址**
  - I/O地址空间独立于存储地址空间
  - 如8086/8088
- I/O端口与存储器**统一编址**
  - 它们共享一个地址空间
  - 如M6800

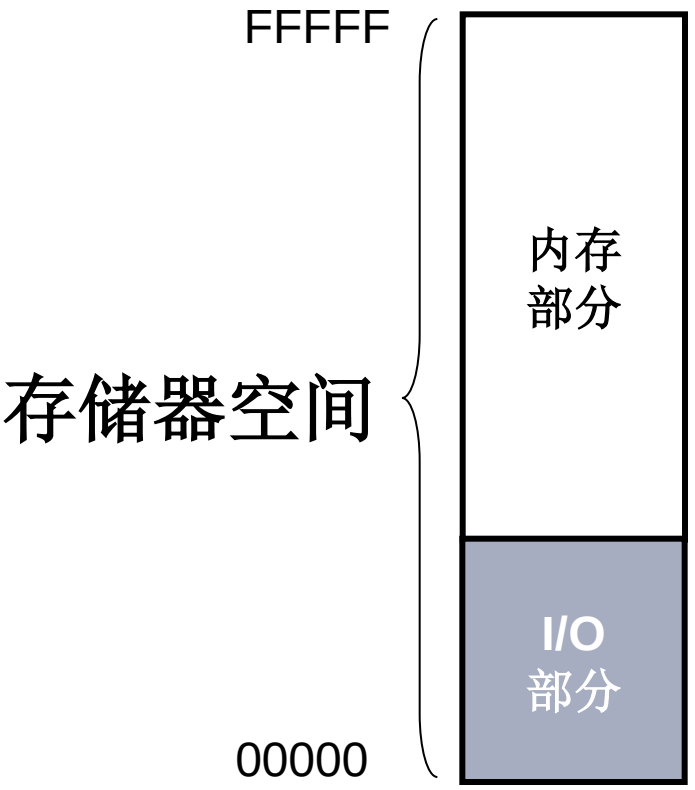
1）I/O端口单独编址

- 优点：
  - I/O端口的地址空间独立
  - 控制和地址译码电路相对简单
  - 专门的I/O指令使程序清晰易读
- 缺点：
  - I/O指令没有存储器指令丰富
- **80x86采用I/O端口独立编址**



2）I/O端口与存储器统一编址

- 优点：
  - 不需要专门的I/O指令
  - I/O数据存取与存储器数据存取一样灵活
- 缺点：
  - I/O端口要占去部分存储器地址空间
  - 程序不易阅读（不易分清访存和访问外设）



（4）8088/8086的输入输出指令

- 输入指令

  IN AL,i8　　;字节输入，直接寻址

  IN AL,DX　　;字节输入，间接寻址

  IN AX,i8　　;字输入，直接寻址

  IN AX,DX　　;字输入，间接寻址

- 输出指令

  OUT i8,AL　　;字节输出，直接寻址

  OUT DX,AL　　;字节输出，间接寻址

  OUT i8,AX　　;字输出，直接寻址

  OUT DX,AX　　;字输出，间接寻址

8088/8086的I/O端口

- 8088只能通过输入输出指令与外设进行数据交换；呈现给程序员的外设是**端口（Port）**，即I/O地址

- 8086用于寻址外设端口的地址线为16条，端口最多为2^{16}＝65536（64K）个，**端口号（端口地址）为0000H ~ FFFFH**

- 每个端口地址对应一个字节空间

I/O寻址方式

- 8088/8086的端口有64K个，无需分段，设计有两种寻址方式

  - **直接寻址**：只用于寻址00H ~ FFH前256个端口，操作数i8表示端口号

  - **间接寻址**：可用于寻址全部64K个端口，DX寄存器的值就是端口号，对端口号大于FFH的端口只能采用间接寻址方式

数据交换方式

- 如果输入输出一个字节，使用AL寄存器
- 如果输入输出一个字，使用AX寄存器

数据交换方式

- **输入一个字**，实际上是从连续两个端口输入两个字节，分别送AL（对应低地址端口）和AH（对应高地址端口）

- **输出一个字**，实际上是将AL（对应低地址端口）和AH（对应高地址端口）两个字节的内容输出给连续两个端口

IN指令（从20H端口输入一个字）

```
; 方法1：字量输入，直接寻址
  in ax,20h

; 方法2：字量输入，间接寻址
  mov dx,20h
  in ax,dx
```

IN指令（从20H端口输入一个字）

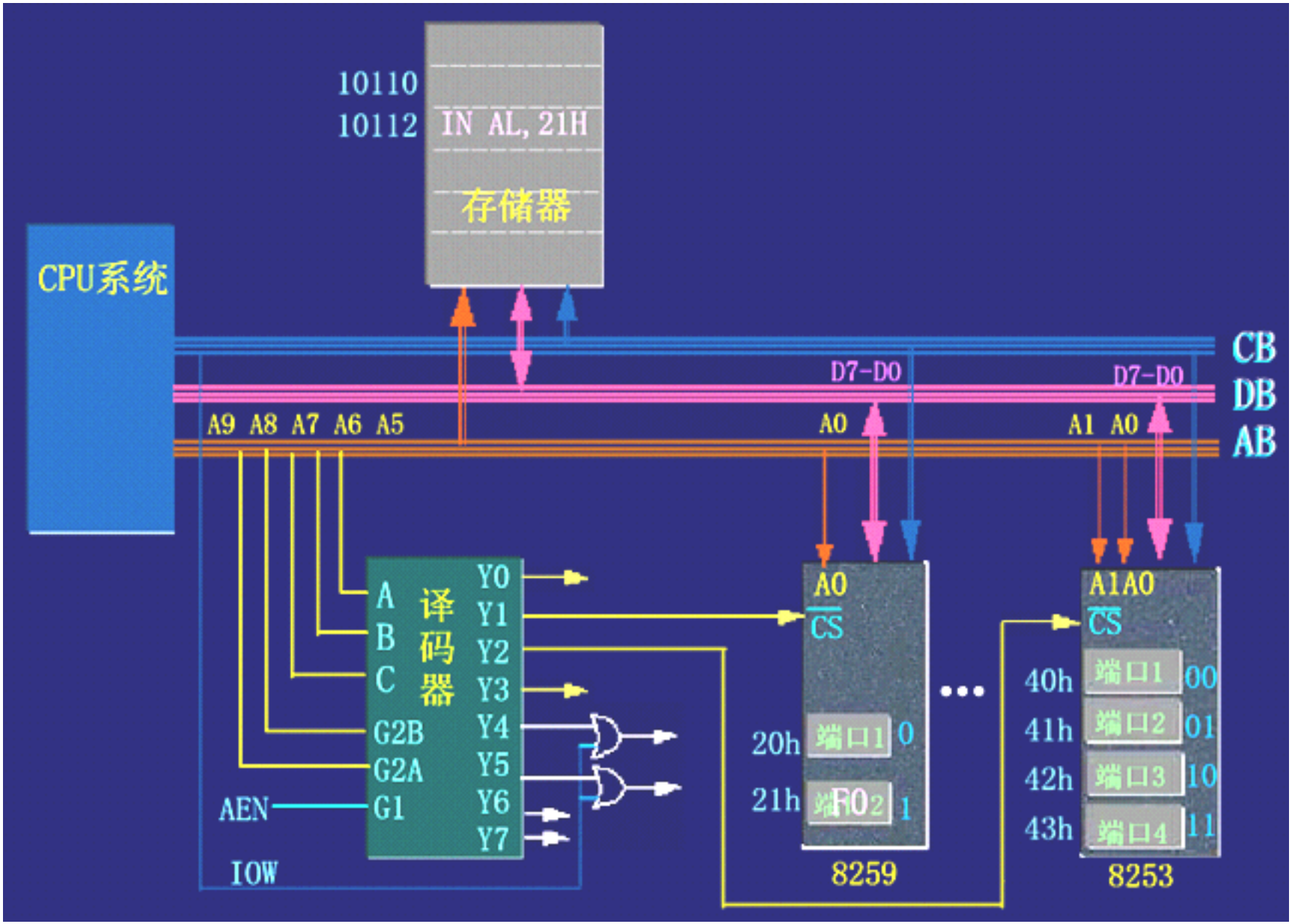
```
；方法3：字节输入，直接寻址
  in al,21h
  mov ah,al
  in al,20h

；方法4：字节输入，间接寻址
  mov dx,21h
  in al,dx
  mov ah,al
  dec dx
  in al,dx
```

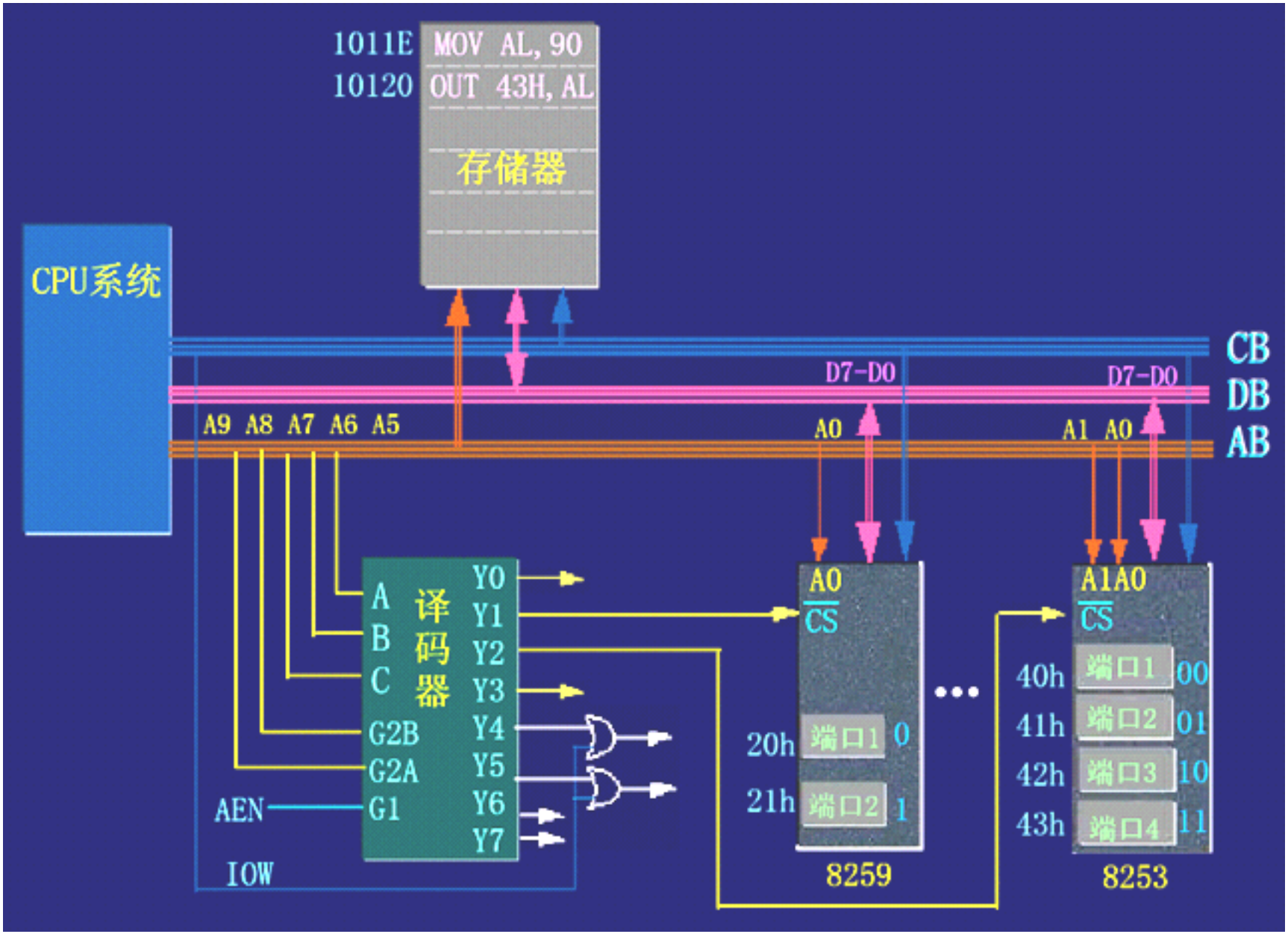
OUT指令（向300H端口输出一个字节）

```
；唯一的方法：间接寻址，字节量输出
  mov al,bvar ；bvar是字节变量
  mov dx,300h
  out dx,al
```

IN AL, 21H


10110
10112
IN AL, 21H
存储器
CPU系统
CB
DB
AB
D7-D0
D7-D0
A9 A8 A7 A6 A5
A0
A1 A0
A
B
C
G2B
G2A
AEN
G1
译码器
Y0
Y1
Y2
Y3
Y4
Y5
Y6
Y7
IOW
A0
CS
20h
端口1
0
21h
端口2
1
8259
A1A0
CS
40h
端口1
00
41h
端口2
01
42h
端口3
10
43h
端口4
11
8253

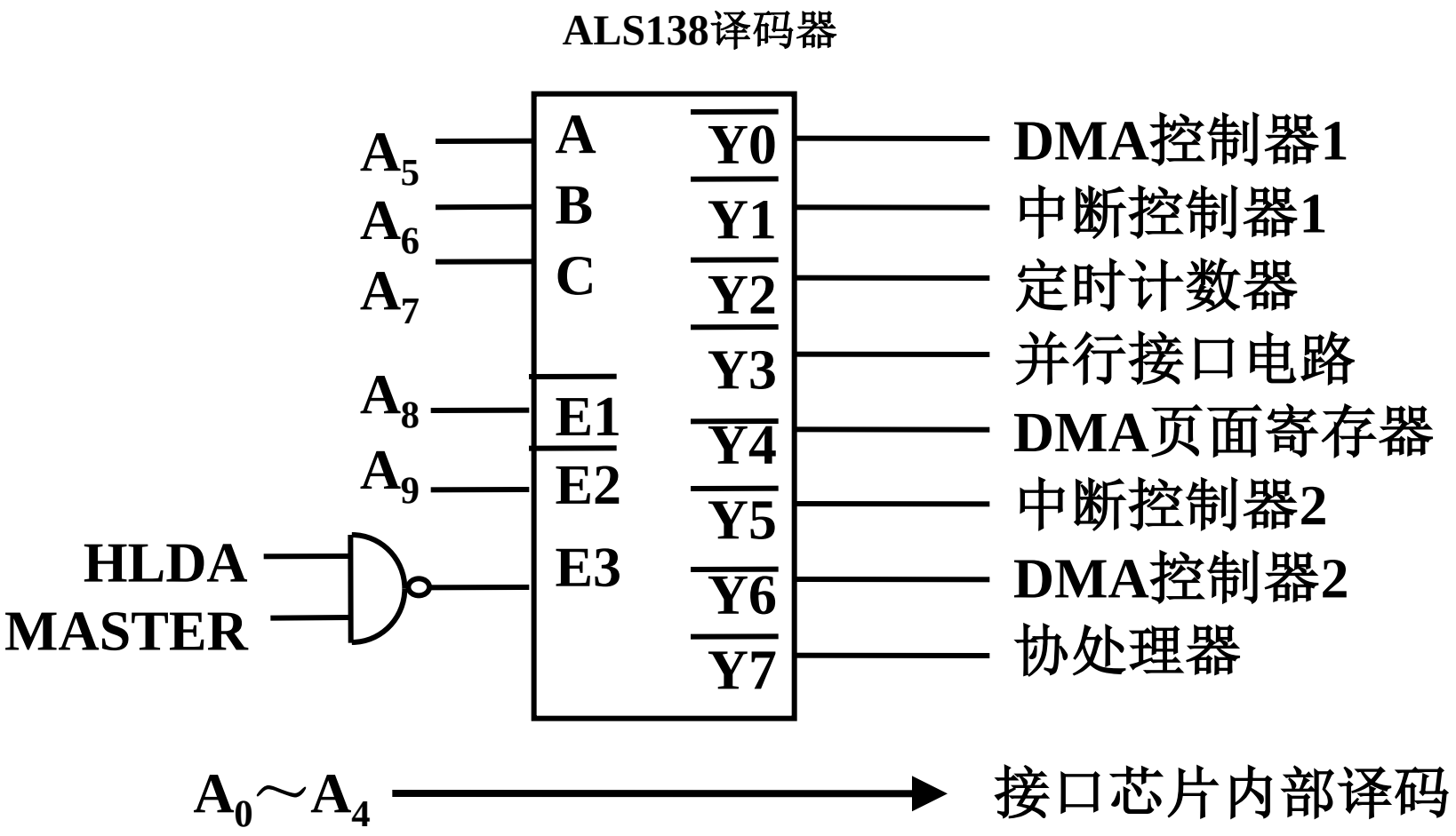
OUT 43H, AL


1011E MOV AL, 90
10120 OUT 43H, AL
存储器
CPU系统
CB
DB
AB
D7-D0
D7-D0
A9 A8 A7 A6 A5
A0
A1 A0
A
B
C
G2B
G2A
G1
AEN
IOW
译码器
Y0
Y1
Y2
Y3
Y4
Y5
Y6
Y7
A0
CS
20h 端口1 0
21h 端口2 1
8259
A1A0
CS
40h 端口1 00
41h 端口2 01
42h 端口3 10
43h 端口4 11
8253

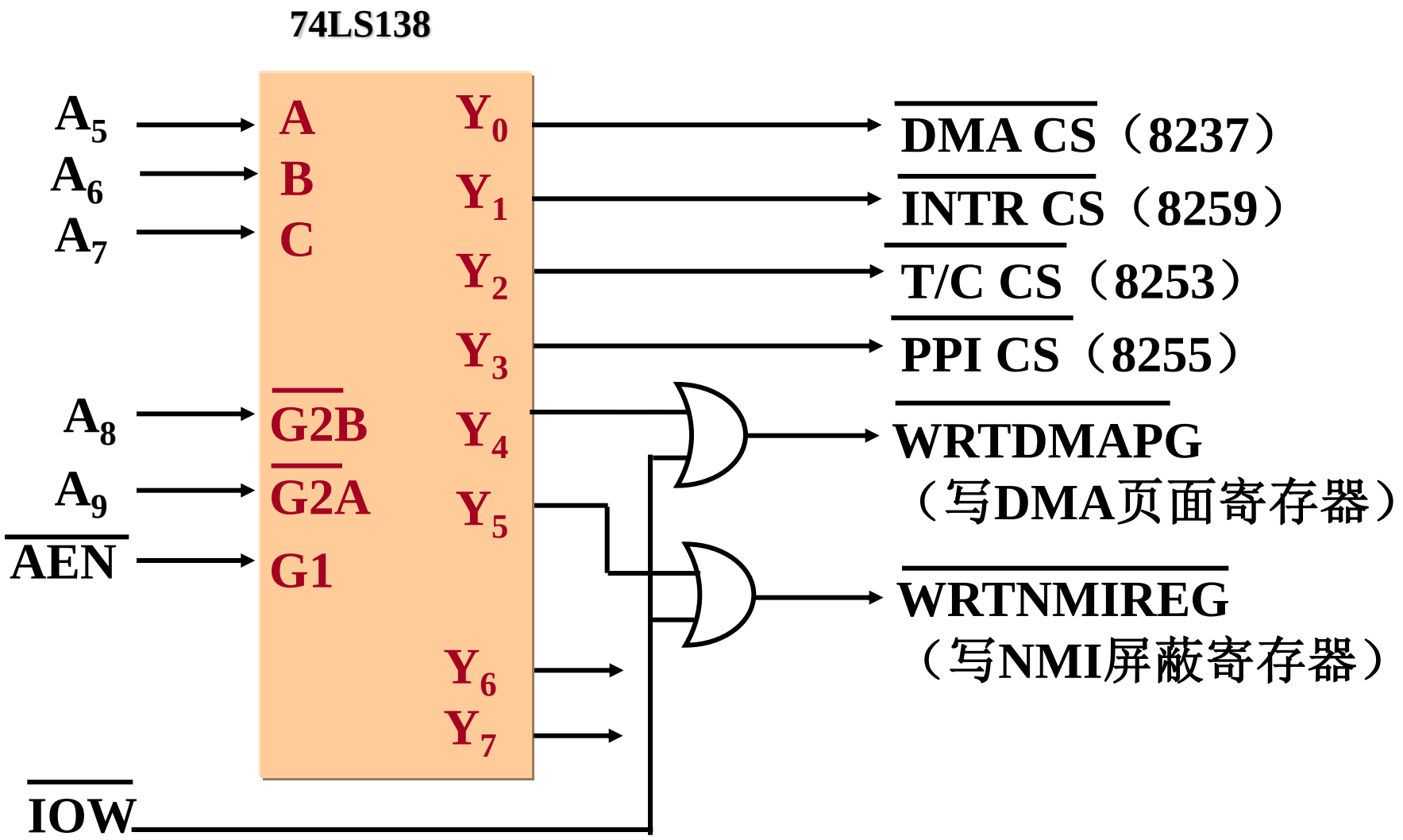
（5）I/O地址的译码

- I/O地址的译码方法与存储器地址的译码方法一样，但有它的特点：
  - 部分译码时，通常是中间地址线不连接
  - 部分译码也有最低地址线不连接的情况
  - 每个接口电路通常只占用几个I/O地址，这时可以利用基本逻辑门电路进行地址译码
  - 除采用译码器、门电路进行译码外，I/O地址译码还经常采用可编程逻辑器件PLD
  - 为了给系统一定的选择余地，有些接口电路利用比较器、开关或跨接器等进行多组I/O地址的译码

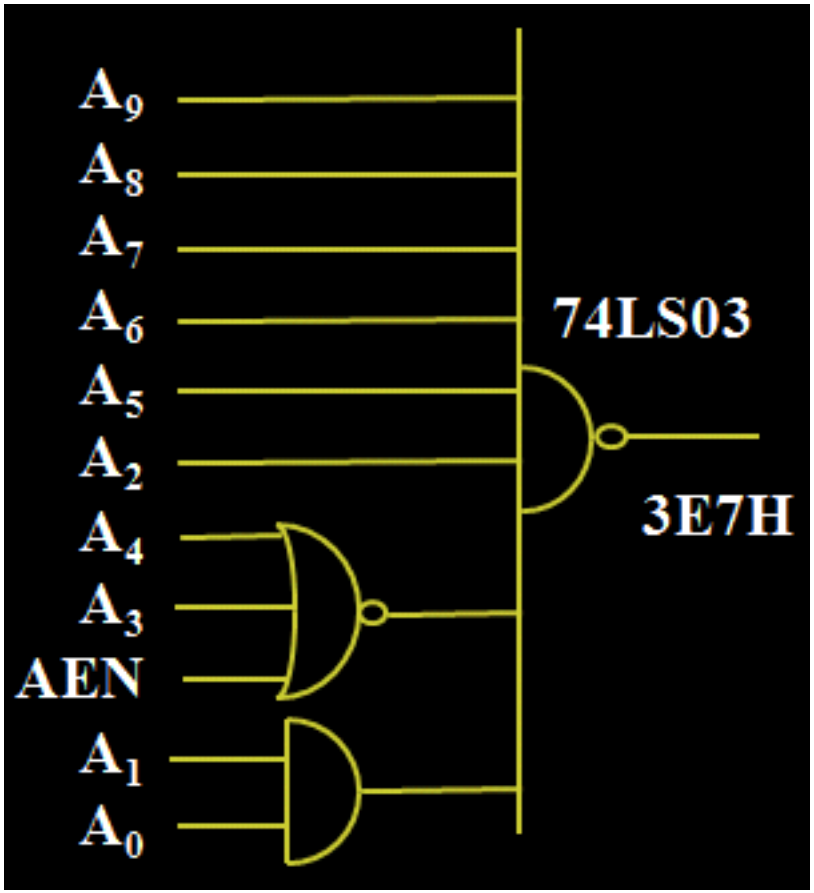
IBM PC/AT主机板的I/O译码电路



IBM PC/XT主机板的I/O译码电路



逻辑门电路进行I/O地址译码



此类地址译码器通常针对单一地址进行译码，位于板级接口电路内部，属于分布式译码结构。

（6）数据传送方式

- **程序控制下的数据传送**——通过CPU执行程序中的I/O指令来完成传送，又分为：**无条件传送、查询传送、中断传送**

（6）数据传送方式

- **直接存储器存取（DMA）**——传送请求由外设向DMA控制器（DMAC）提出，后者向CPU申请总线，最后DMAC利用系统总线来完成外设和存储器间的数据传送

- **I/O处理机**——CPU委托专门的I/O处理机来管理外设，完成传送和相应的数据处理

1）无条件传送方式及其接口

- 在CPU与慢速变化的设备（或简单设备）交换数据时，可以认为它们总是处于“就绪”状态，随时可以进行数据传送，这就是无条件传送，或称立即传送、同步传送

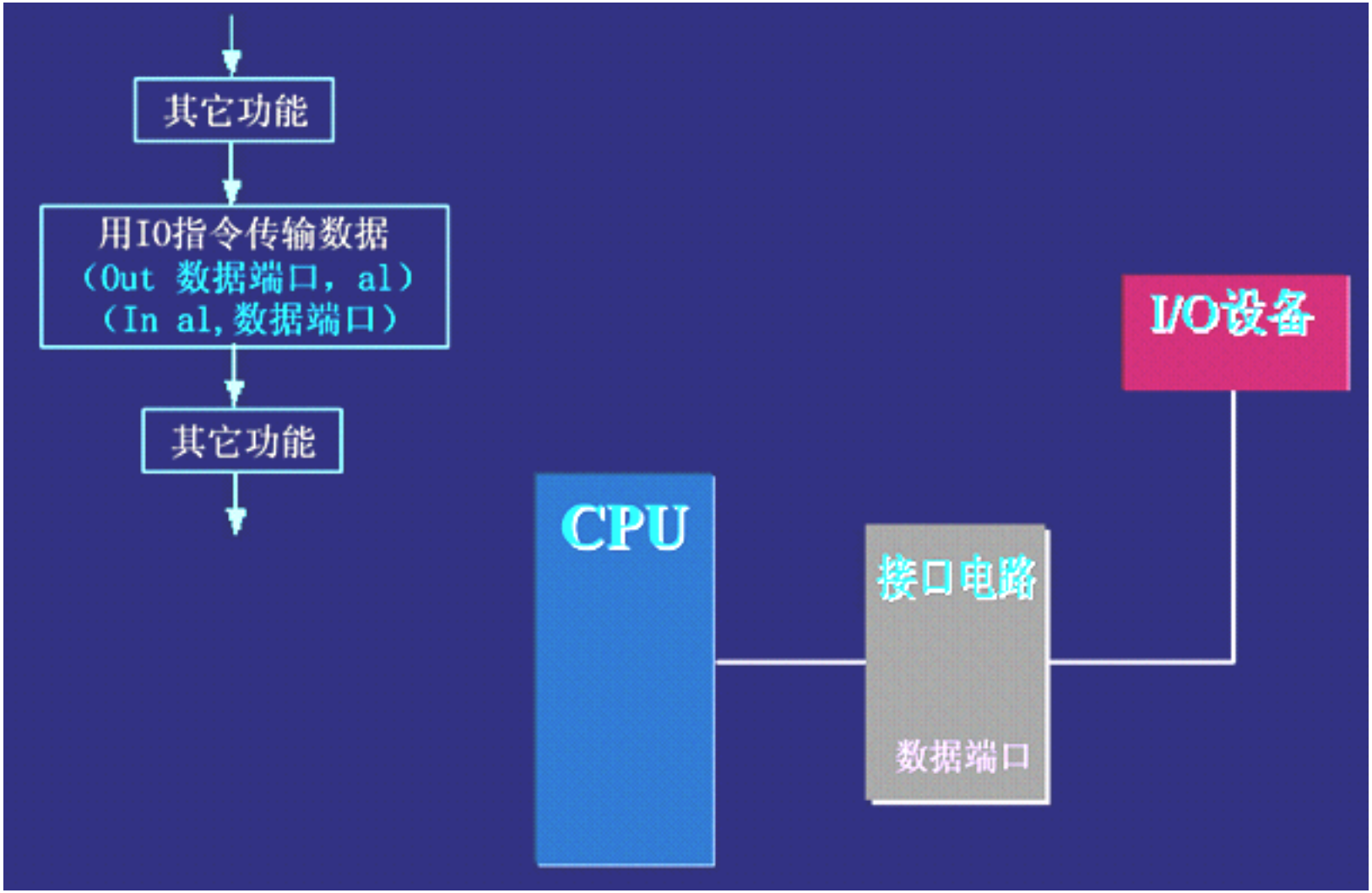
- 适合于简单设备，如LED数码管、开关等

1）无条件传送方式及其接口

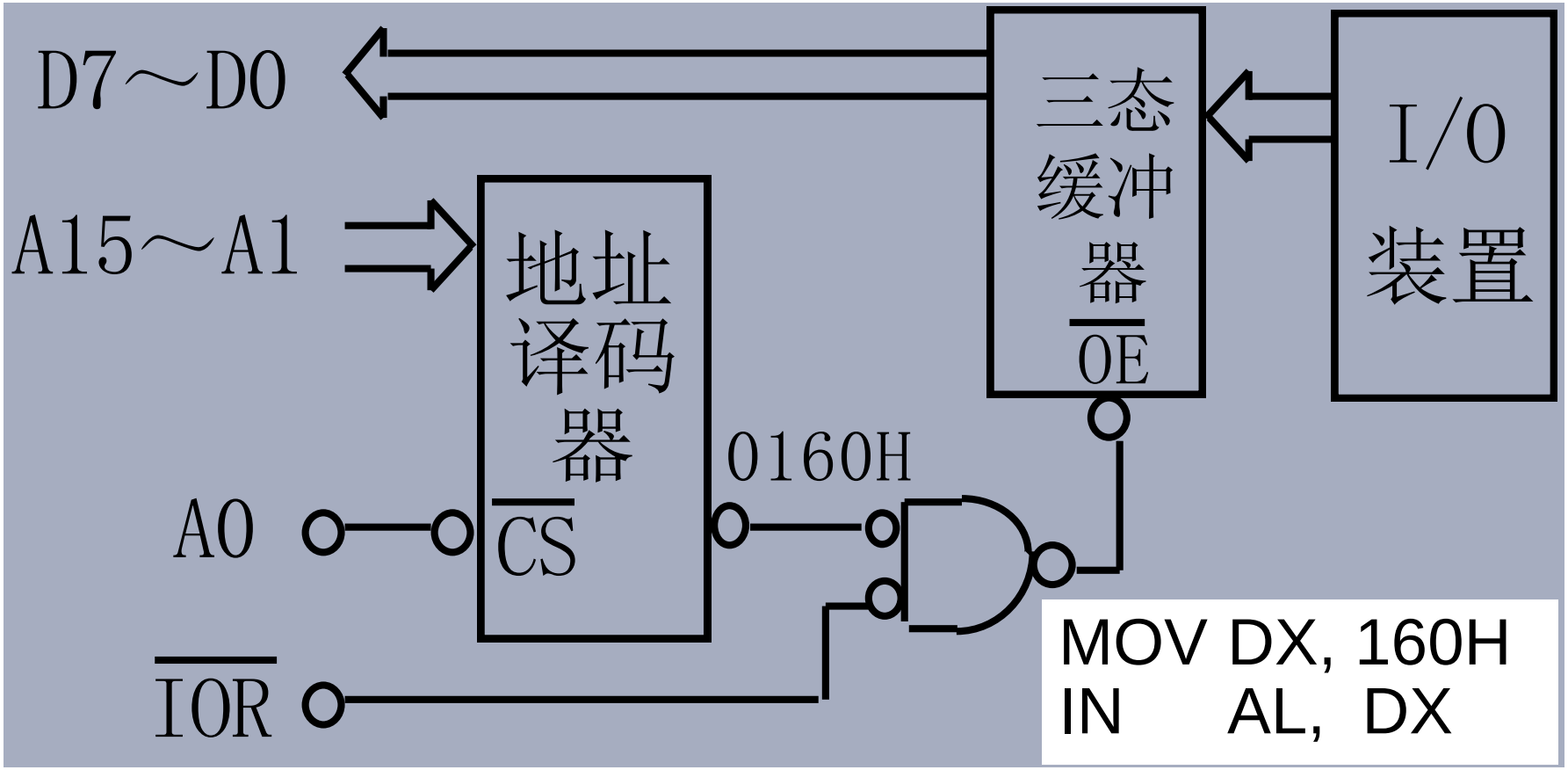
- 无条件传送的接口和操作均十分简单

- 这种传送有**前提：外设必须随时就绪**

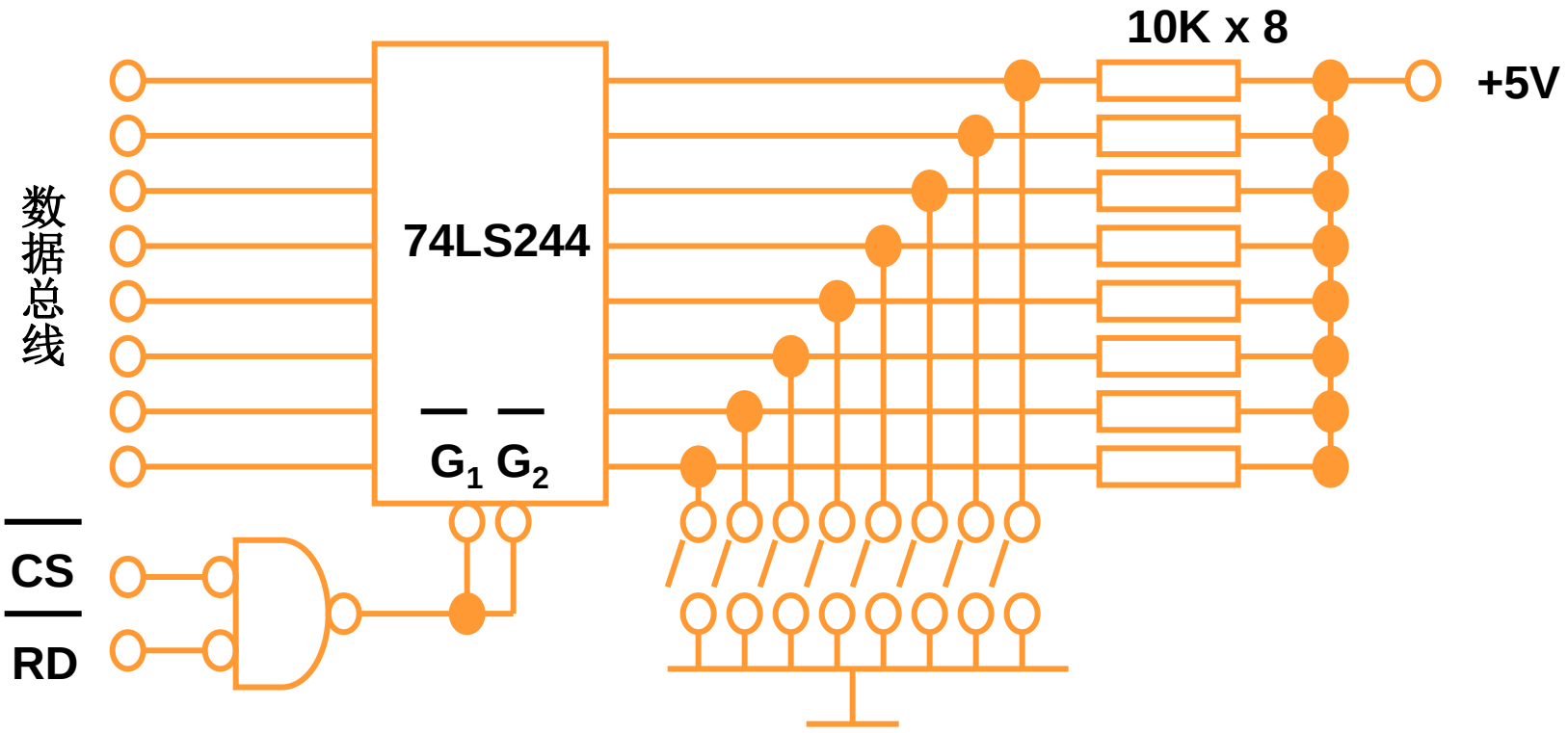
无条件传送流程



无条件传送：输入示例

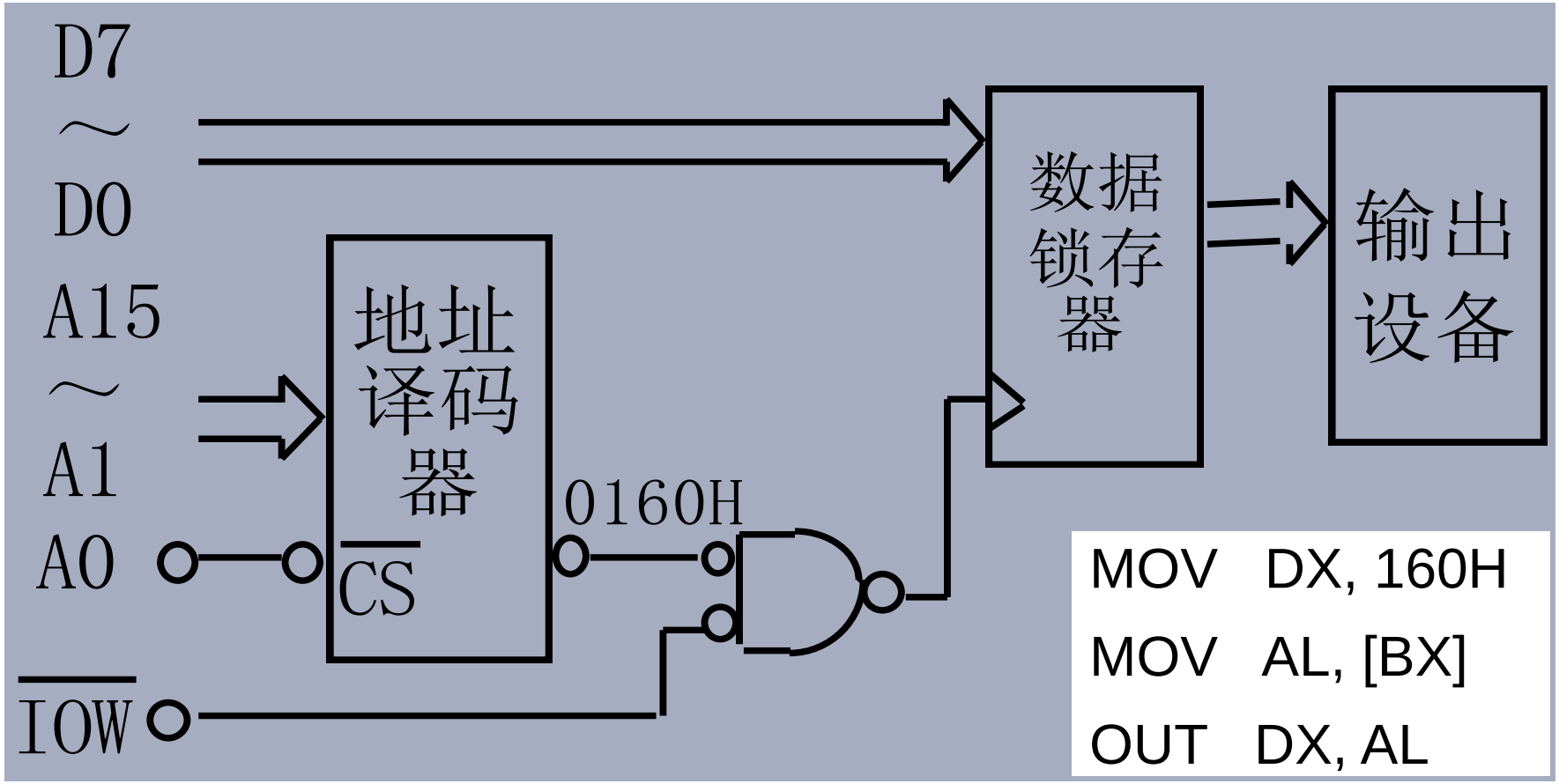

D7～D0
A15～A1
地址
译码
器
CS
A0
IOR
0160H
三态
缓冲
器
OE
I/O
装置
MOV DX, 160H
IN AL, DX

无条件传送：输入示例

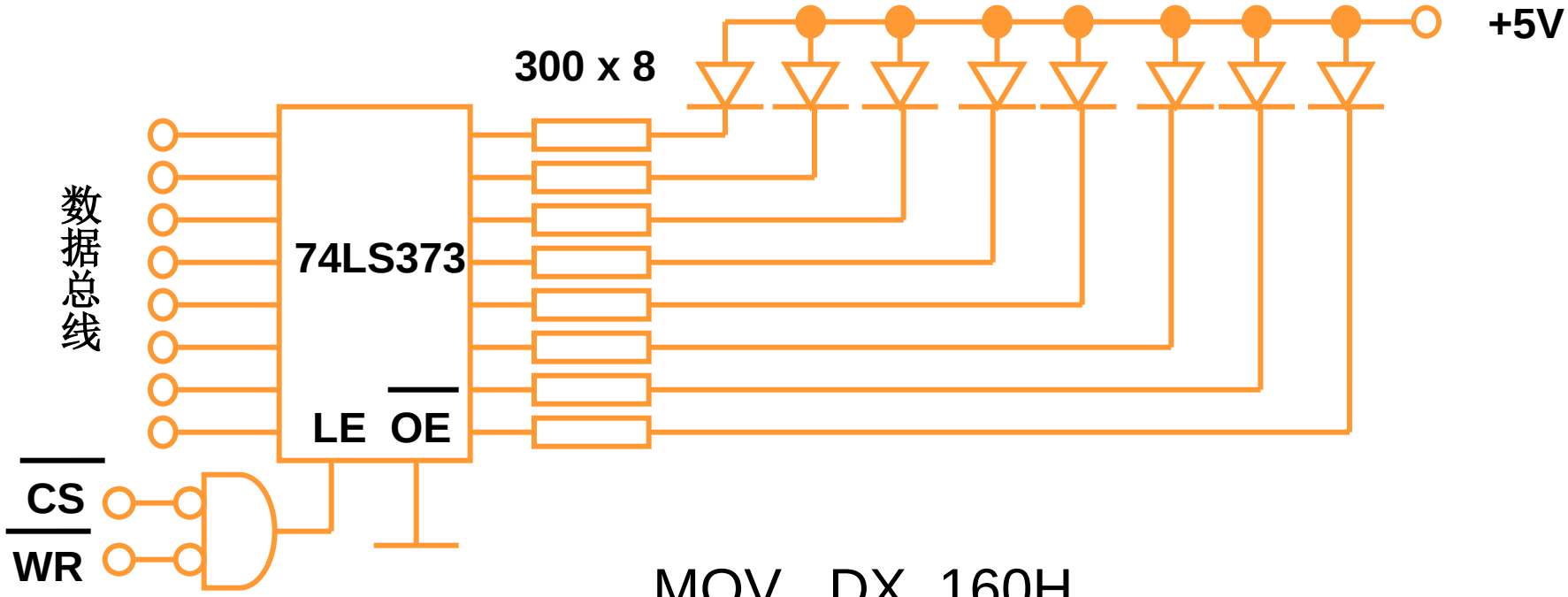



```
MOV DX, 160H
IN    AL,  DX
```

无条件传送：输出示例

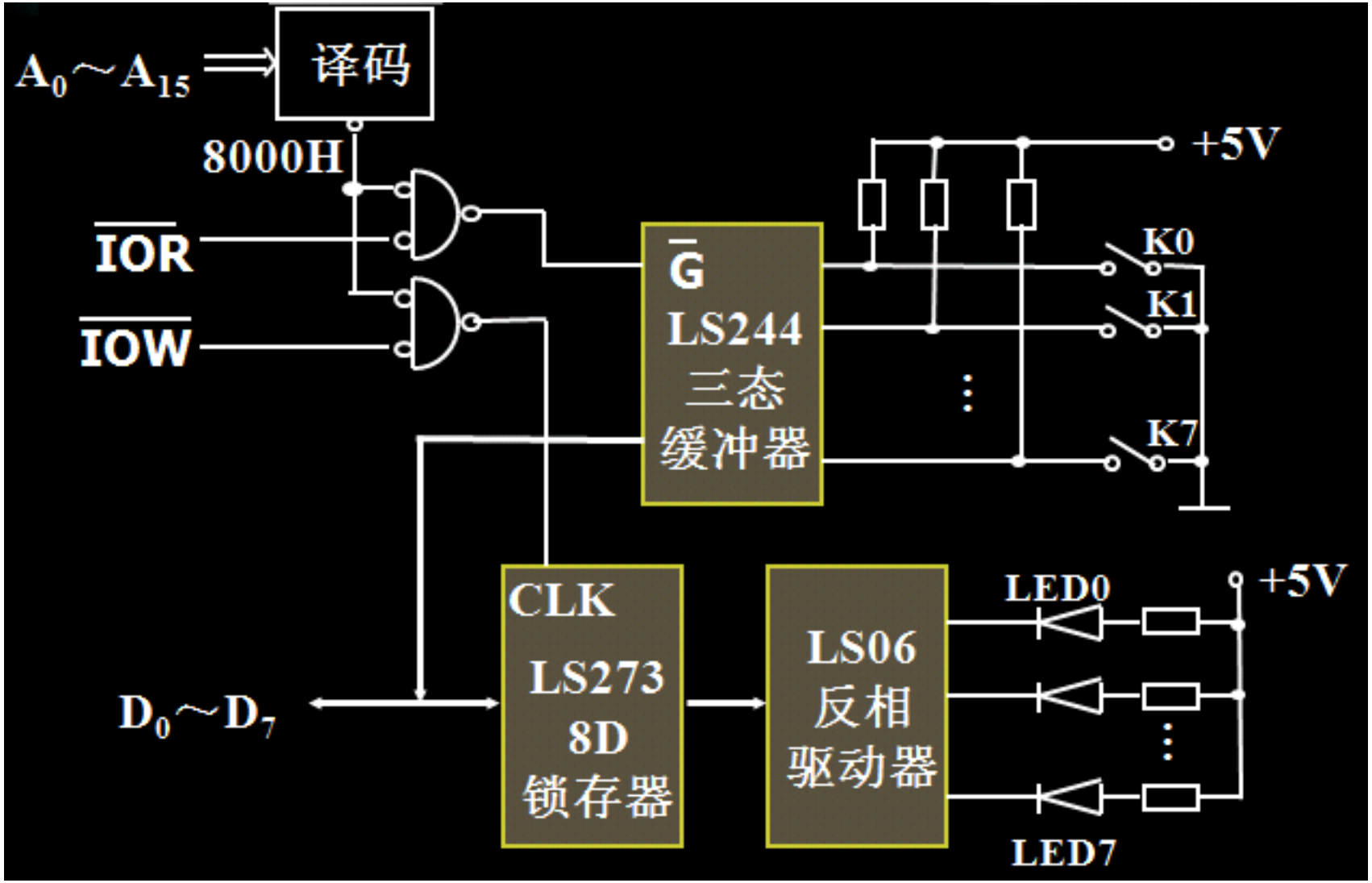

D7
～
D0
A15
～
A1
A0
IOW
地址译码器
CS
0160H
数据锁存器
输出设备
MOV DX, 160H
MOV AL, [BX]
OUT DX, AL

无条件传送：输出示例




```
MOV   DX, 160H
MOV   AL, [BX]
OUT   DX, AL
```

无条件传送：输入输出接口



无条件传送：输入输出接口

```
next:  mov dx,8000h ;DX指向数据端口
       in al,dx     ;从输入端口读开关状态
       not al       ;反相
       out dx,al    ;送输出端口显示
       call delay   ;调子程序延时
       jmp next     ;重复
```

2）查询传送方式及其接口

- CPU需要先了解（**查询**）外设的工作状态，然后在外设可以交换信息的情况下（**就绪**）实现数据输入或输出

- 对多个外设的情况，则CPU按一定顺序依次查询（轮询）。先查询的外设将优先进行数据交换

- 查询传送的特点是：工作可靠，适用面宽，但传送效率低

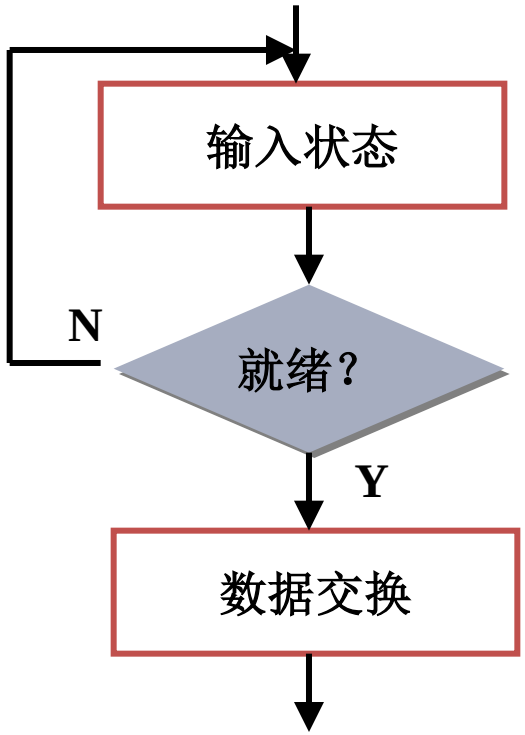
“就绪”状态

- 在**输入场合**
  - “就绪”说明输入接口已准备好送往CPU的数据，正等着CPU来读取
  - 该状态也可用接口中数据缓冲器已“满”来描述
- 在**输出场合**
  - “就绪”说明输出接口已做好准备，等待接收CPU要输出的数据
  - 该状态也可用接口数据缓冲器已“空”、或者用接口（外设）“闲”或不“忙（Busy）”来描述

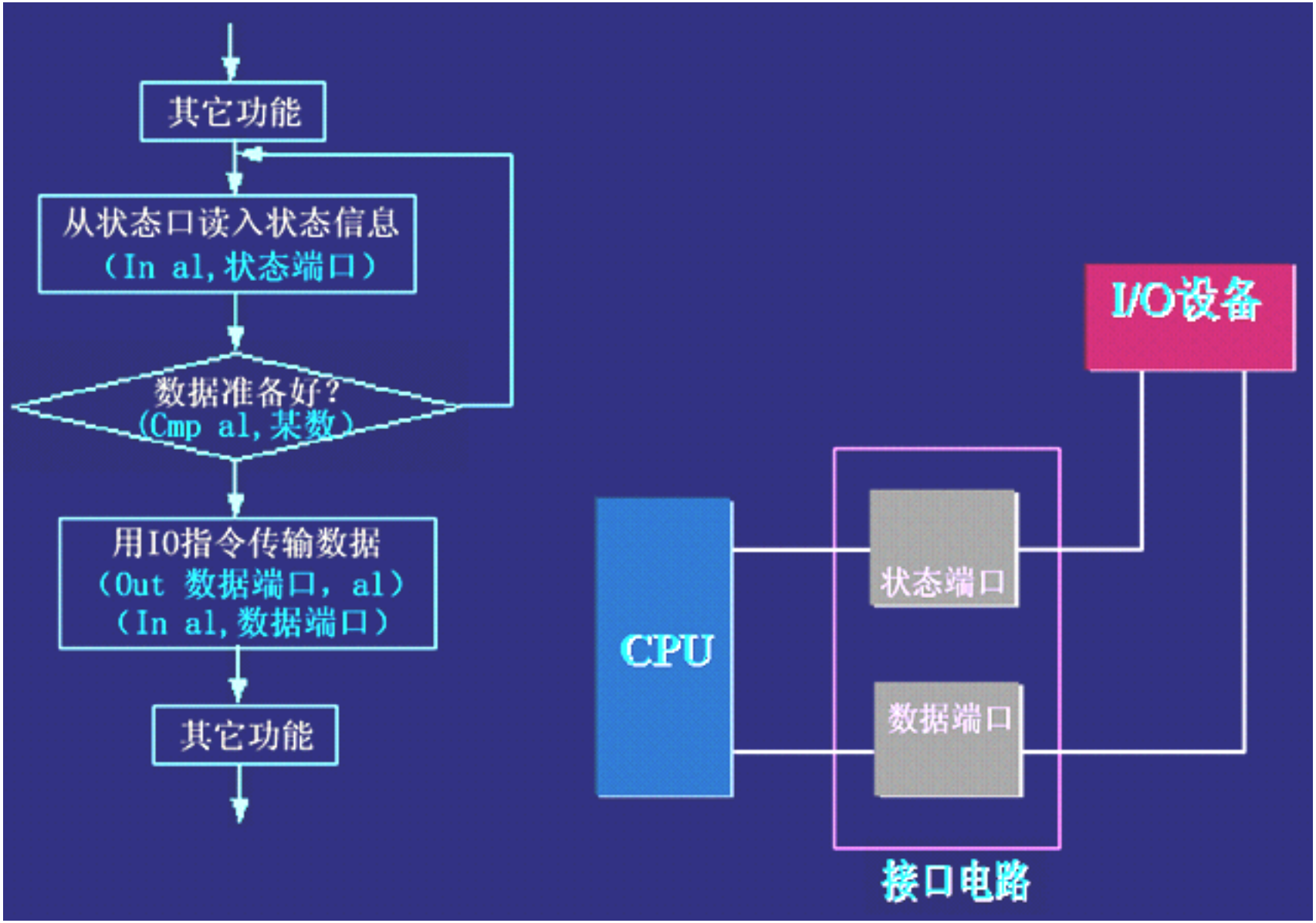
查询传送的两个环节

1. **查询环节**
- 寻址状态口
- 读取状态寄存器的标志位
- 若不就绪就继续查询，直至就绪

2. **传送环节**
- 寻址数据口
- 是输入，通过输入指令从数据端口读入数据
- 是输出，通过输出指令向数据端口输出数据



查询传送流程


其它功能
从状态口读入状态信息
（In al,状态端口）
数据准备好?
(Cmp al,某数)
用IO指令传输数据
（Out 数据端口，al）
（In al,数据端口）
其它功能
I/O设备
CPU
状态端口
数据端口
接口电路

查询输入接口程序示例

```
        mov dx,8000h      ;DX指向状态端口
status: in al,dx          ;读状态端口
        test al,01h       ;测试标志位D0
        jz status         ;D0＝0，未就绪，继续查询
        inc dx            ;D0＝1，就绪，DX指向数据端口
        in al,dx          ;从数据端口输入数据
```

查询输出接口程序示例

```
        mov dx,8000h      ;DX指向状态端口
status: in al,dx          ;读取状态端口的状态数据
        test al,80h       ;测试标志位D7
        jnz status        ;D7＝1，未就绪，继续查询
        inc dx            ;D7＝0，就绪，DX指向数据端口
        mov al,buf        ;变量buf送AL
        out dx,al         ;将数据输出给数据端口
```

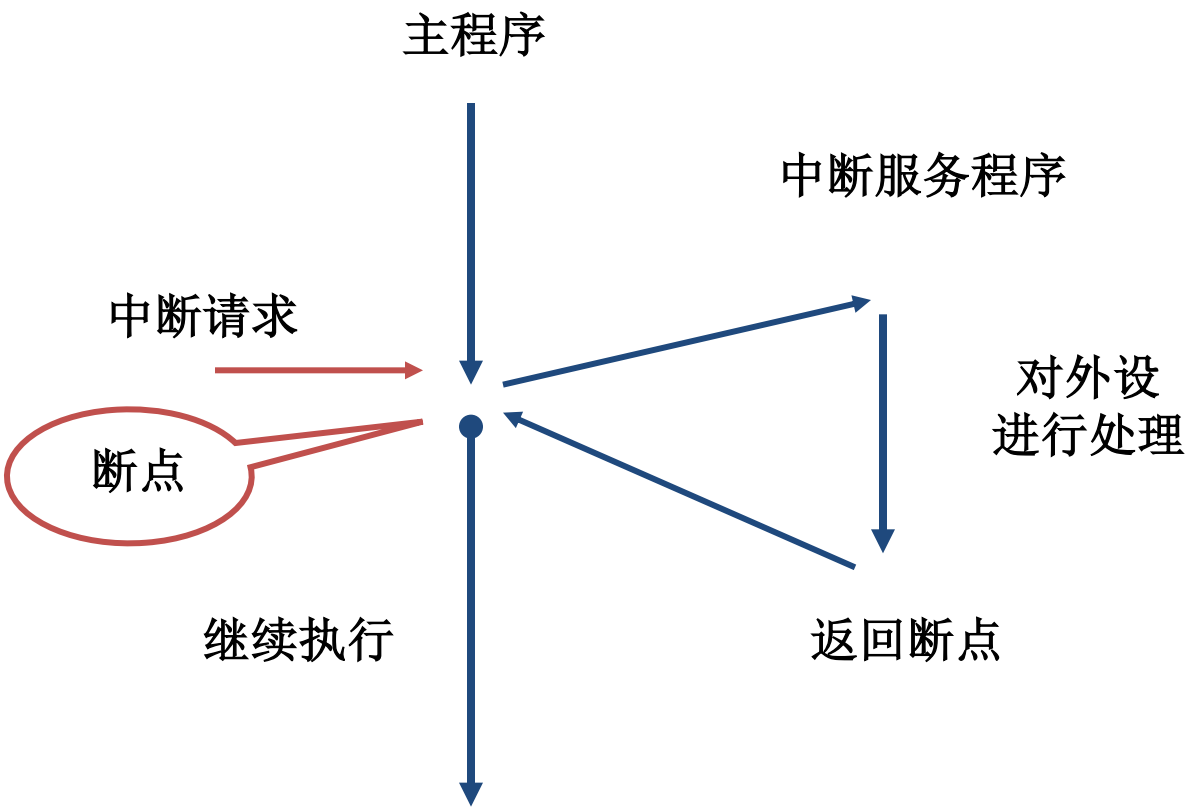
查询方式的EEPROM编程

```
next:   mov al,55h      ;写入内容＝55H
        mov [bx],al     ;写入存储单元
        nop             ;空操作指令，起延时作用
        nop
next1:  in al,dx        ;查询状态口
        test al,01h     ;测试D0
        jz next1        ;D0＝0，芯片还在写入
        inc bx          ;D0＝1，写毕，指针移动
        loop next       ;循环至全部字节写完
```

3）中断传送方式

- CPU在执行程序中，被内部或外部的事件所打断，转去执行一段预先安排好的中断服务程序；服务结束后，又返回原来的断点，继续执行原来的程序

3）中断传送方式



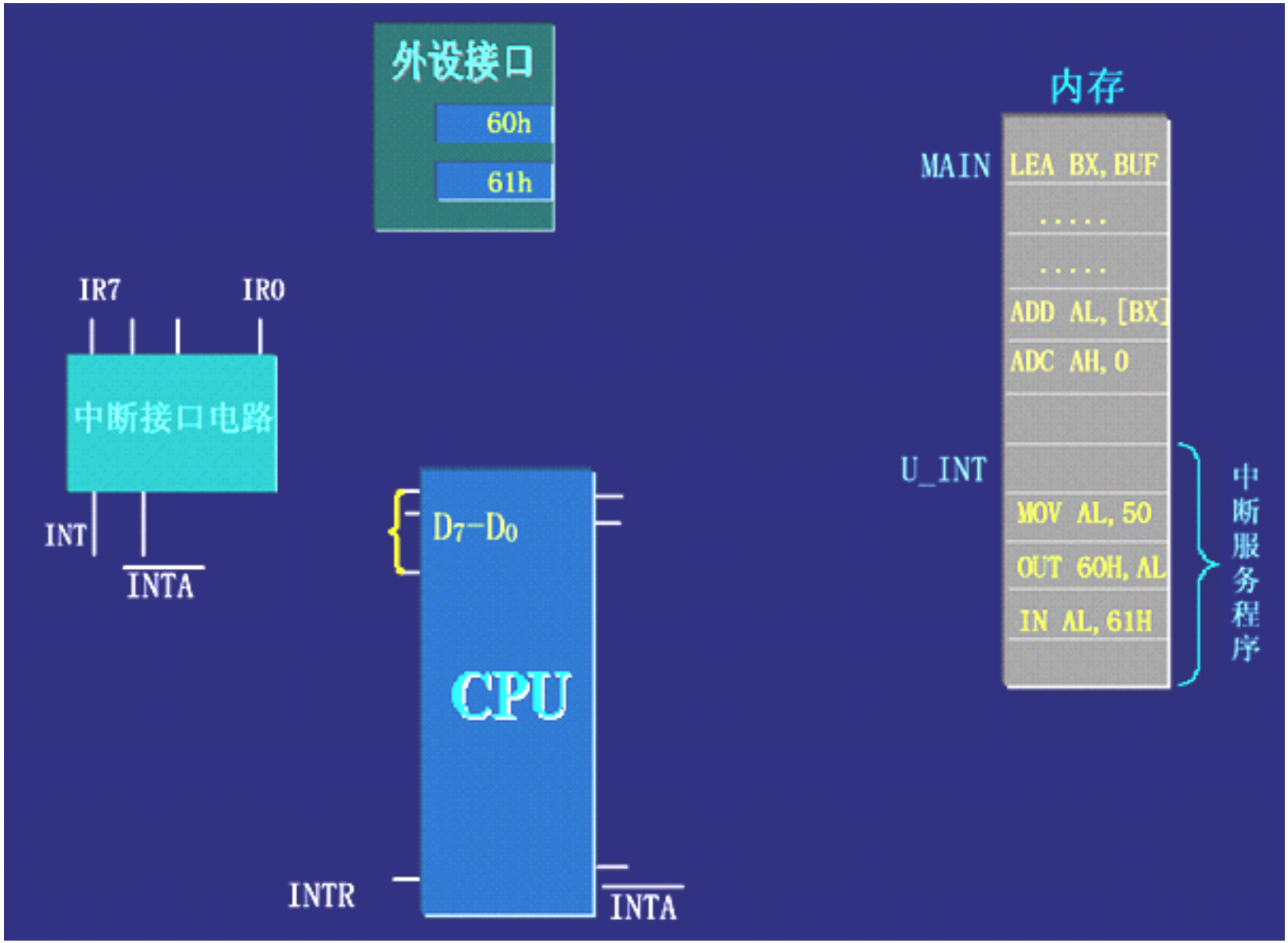
中断工作过程

- 中断服务是进行数据交换的实质性环节

- 中断请求 =>中断响应=>关中断=>断点保护=>中断识别=>现场保护=>中断服务=>恢复现场=>开中断=>中断返回

- 上图中硬件、软件完成的步骤分别使用不同颜色表示。

中断传送流程


外设接口
60h
61h
内存
MAIN
LEA BX, BUF
.....
.....
ADD AL, [BX]
ADC AH, 0
U_INT
MOV AL, 50
OUT 60H, AL
IN AL, 61H
中断服务程序
IR7
IR0
中断接口电路
INT
INTA
D7-D0
CPU
INTR
INTA

3）中断传送方式

- 中断传送是一种效率更高的程序传送方式
- 进行传送的中断服务程序是预先设计好的，常驻内存的程序
- 中断请求是外设随机向CPU提出的
- CPU对请求的检测是有规律的：一般是在每条指令的最后一个时钟周期采样中断请求输入引脚

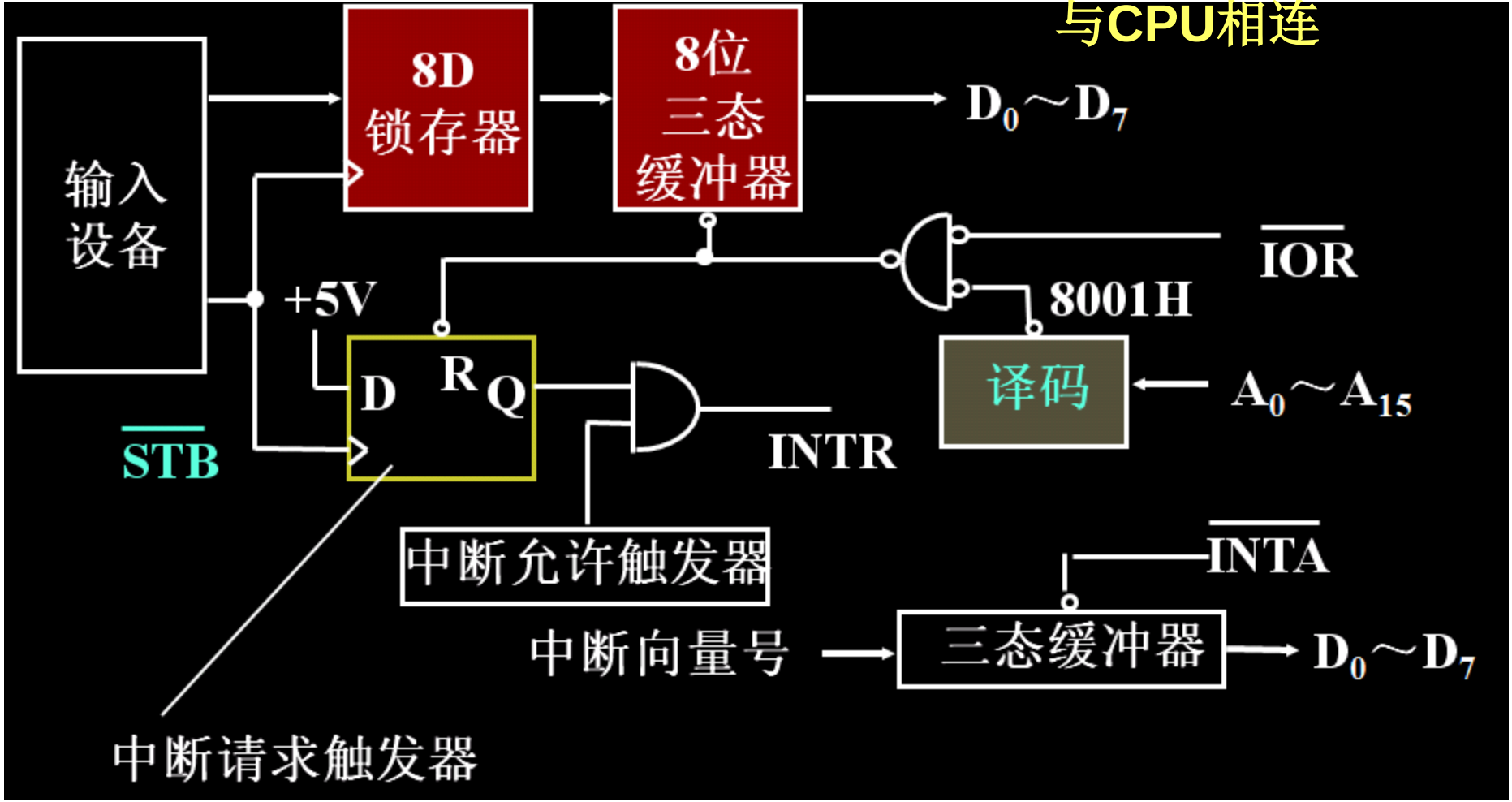
3）中断传送方式

- 本课程主要论述中断在输入和输出方面的应用

- 中断还有着非常广泛的应用：例如操作系统使用定时中断剥夺应用程序对CPU的控制权。

中断输入接口

不通过8259，直接与CPU相连



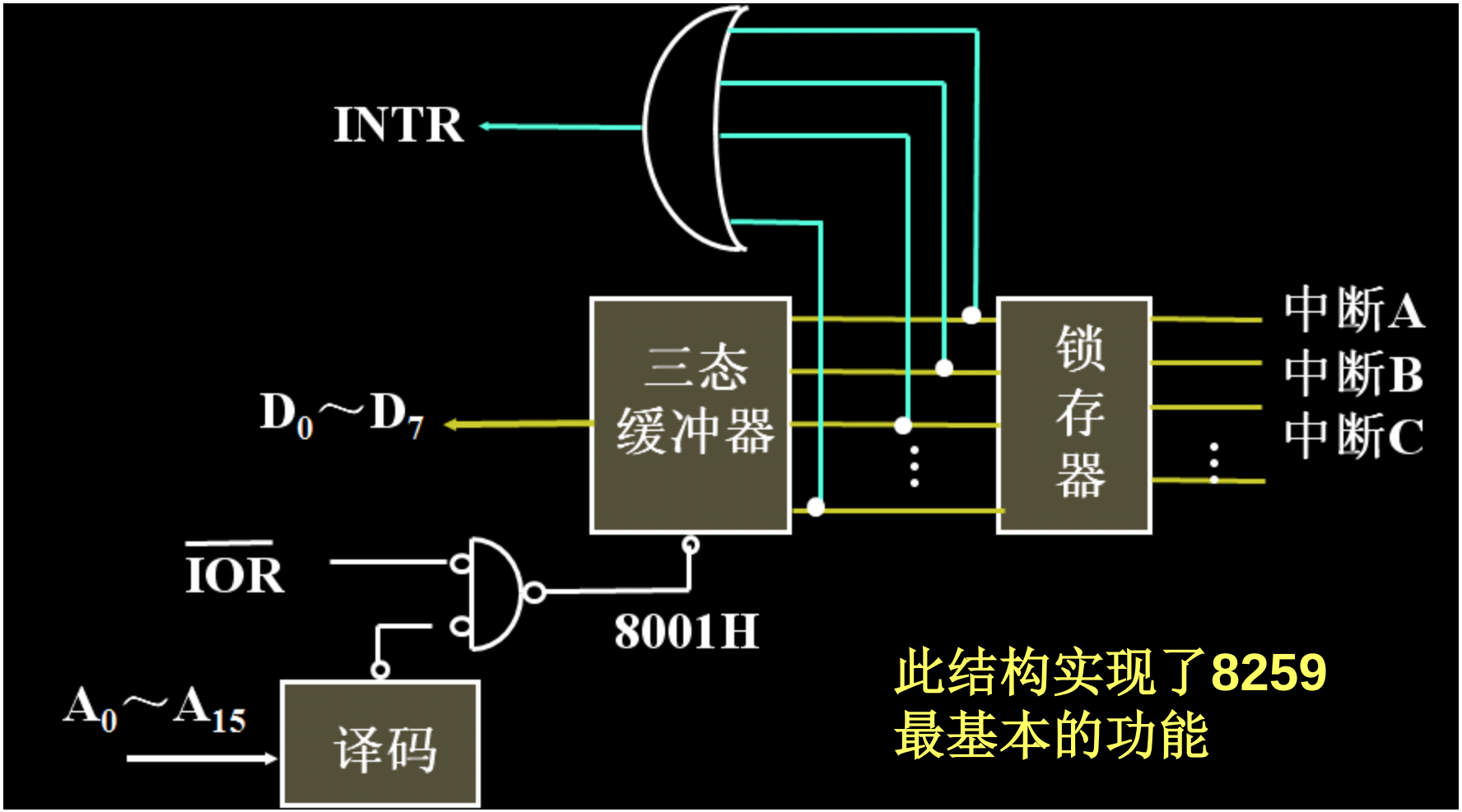
中断源的识别

问题1：系统有多个中断请求，CPU如何识别中断源？

解答1：向量中断

解答2：中断查询

中断（源）查询接口

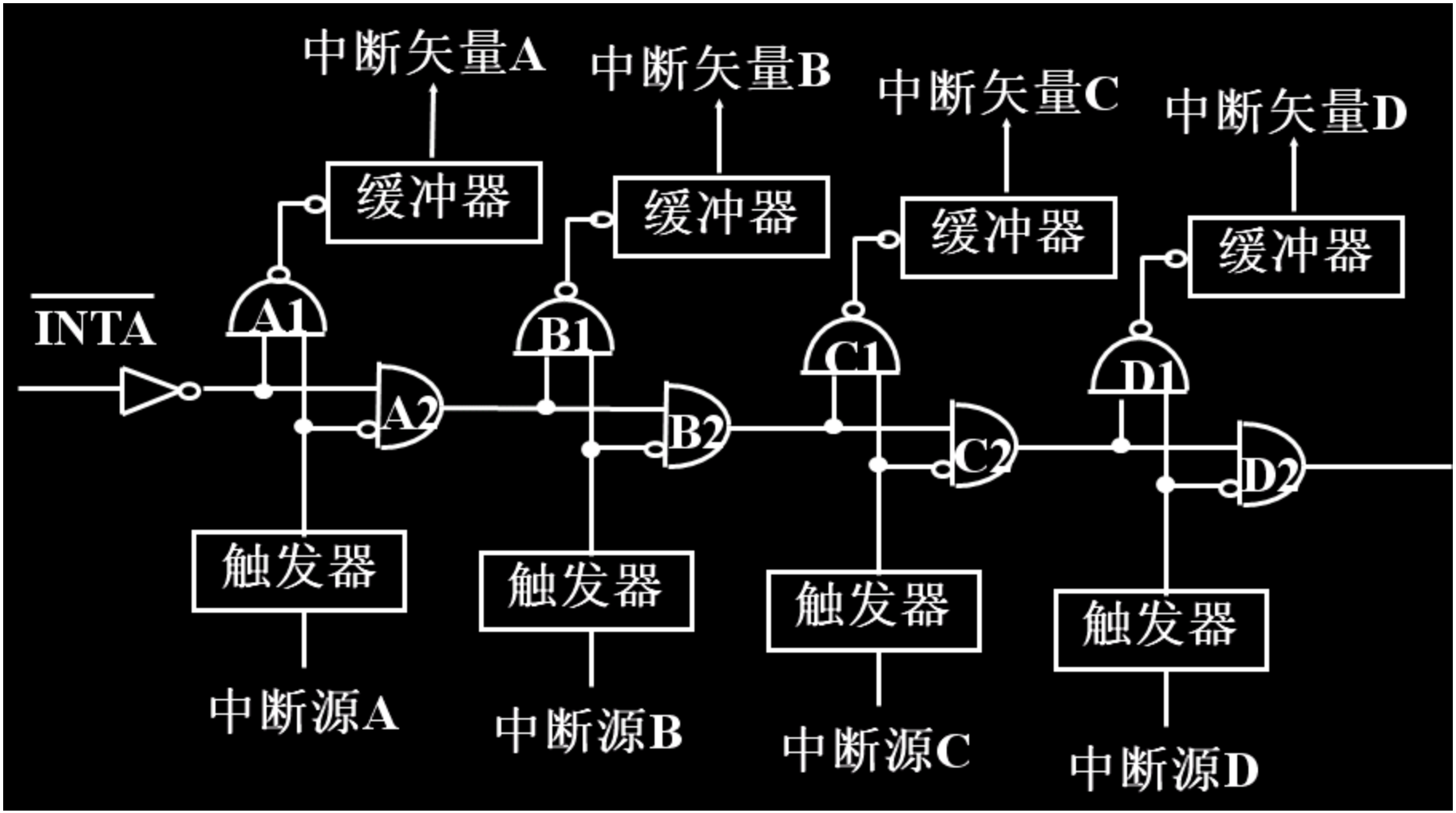

INTR
$D_0 \sim D_7$
三态
缓冲器
锁
存
器
中断A
中断B
中断C
$\overline{IOR}$
8001H
$A_0 \sim A_{15}$
译码
此结构实现了8259
最基本的功能

中断优先权排队

问题2：有多个中断同时请求，CPU如何应对？

解答：链式优先权排队电路

链式中断优先权排队电路



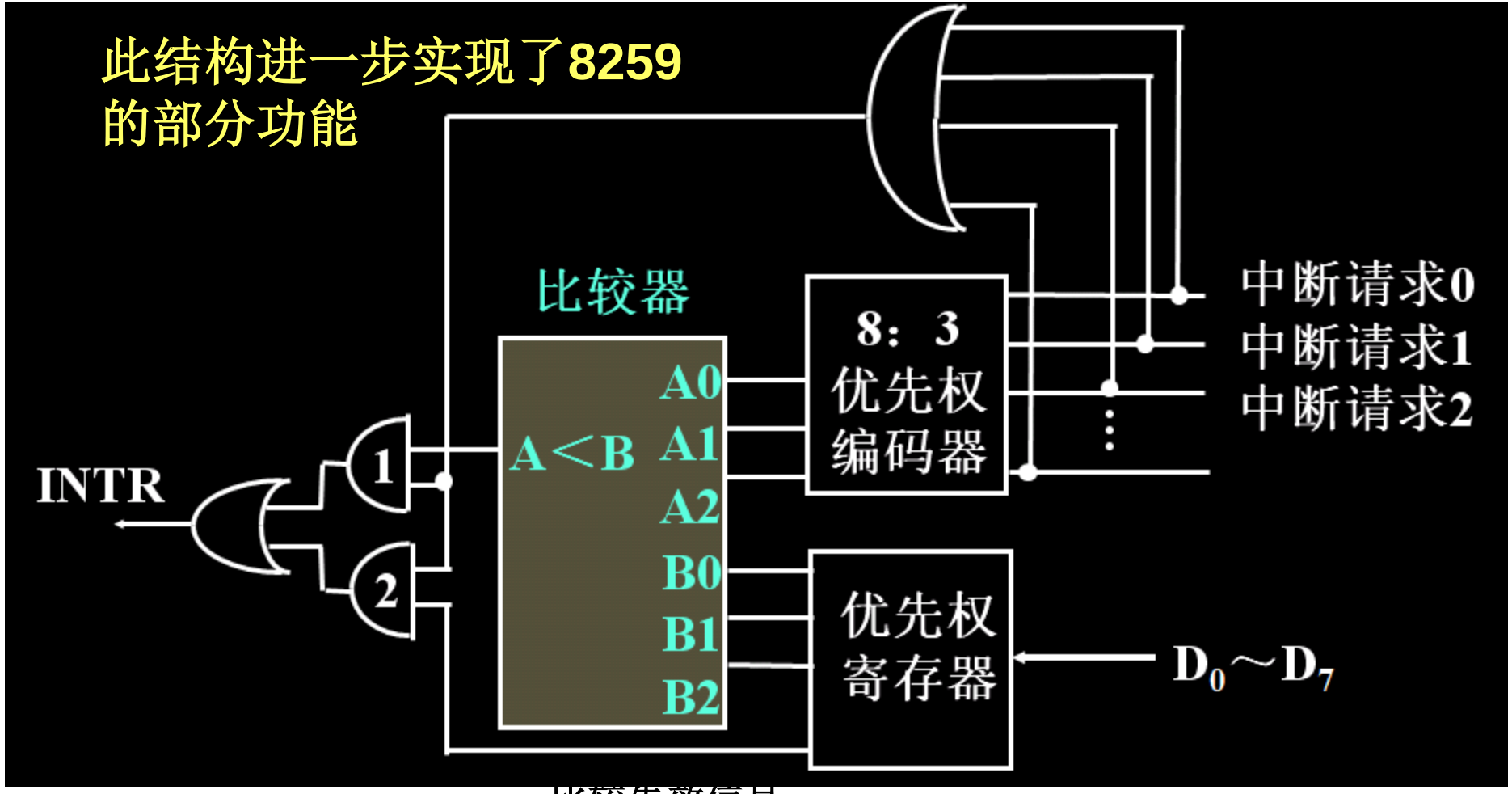
中断嵌套

问题3：中断处理过程中，
又有中断提出请求，应如何处理？

解答：优先权编码电路

- **除了硬件上能够识别更高级中断请求外，软件上在中断服务程序中需要开放中断，才能实现中断嵌套**

中断优先权编码电路


此结构进一步实现了8259
的部分功能
比较器
A<B
A0
A1
A2
B0
B1
B2
8：3
优先权
编码器
中断请求0
中断请求1
中断请求2
INTR
1
2
优先权
寄存器
D0～D7
比较失效信号

4） DMA传送方式

- 希望克服程序控制传送的不足：

  外设→CPU→存储器

  外设←CPU←存储器

- 直接存储器存取DMA：

  外设→存储器

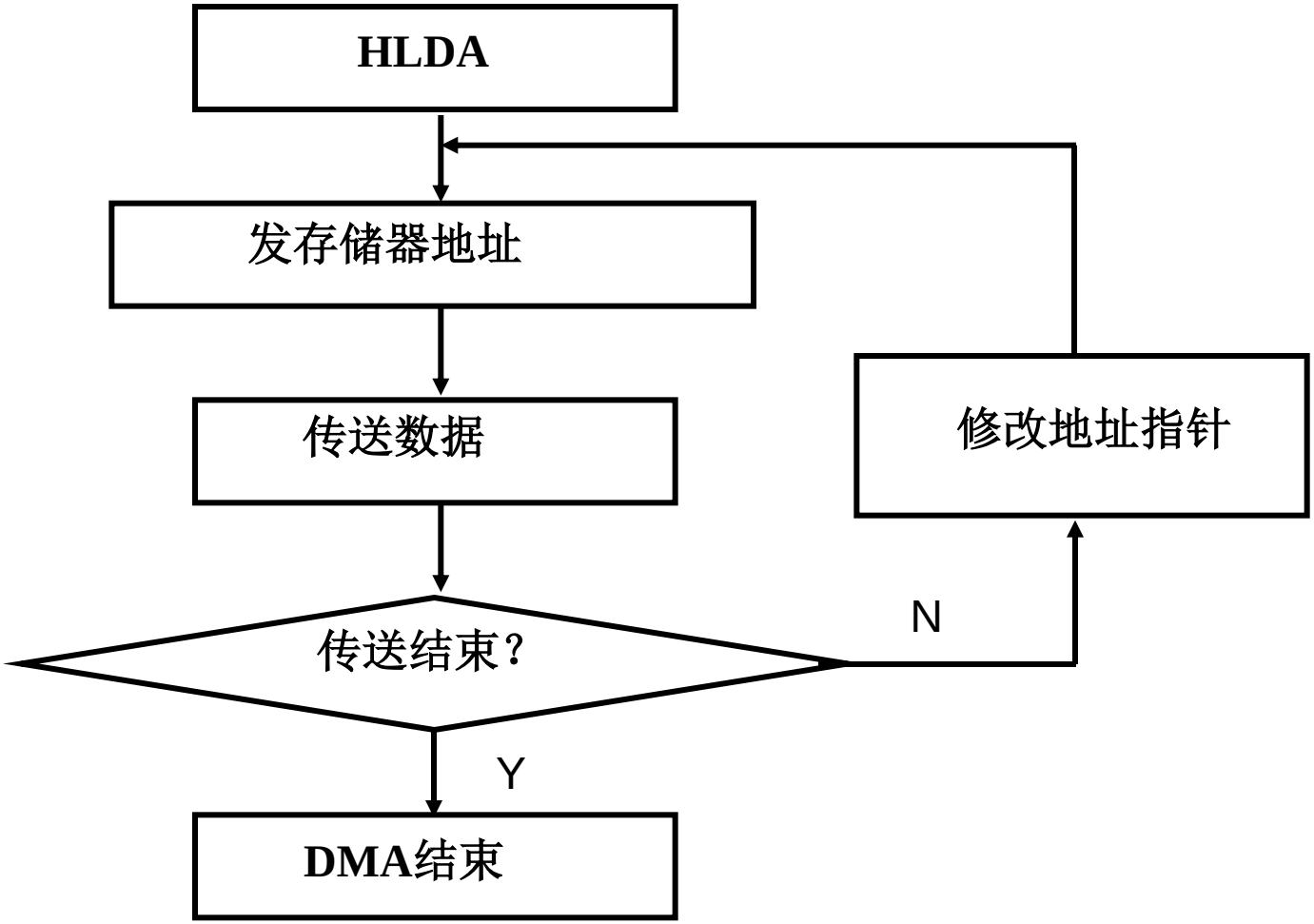
  外设←存储器

- CPU释放总线，由DMA控制器管理

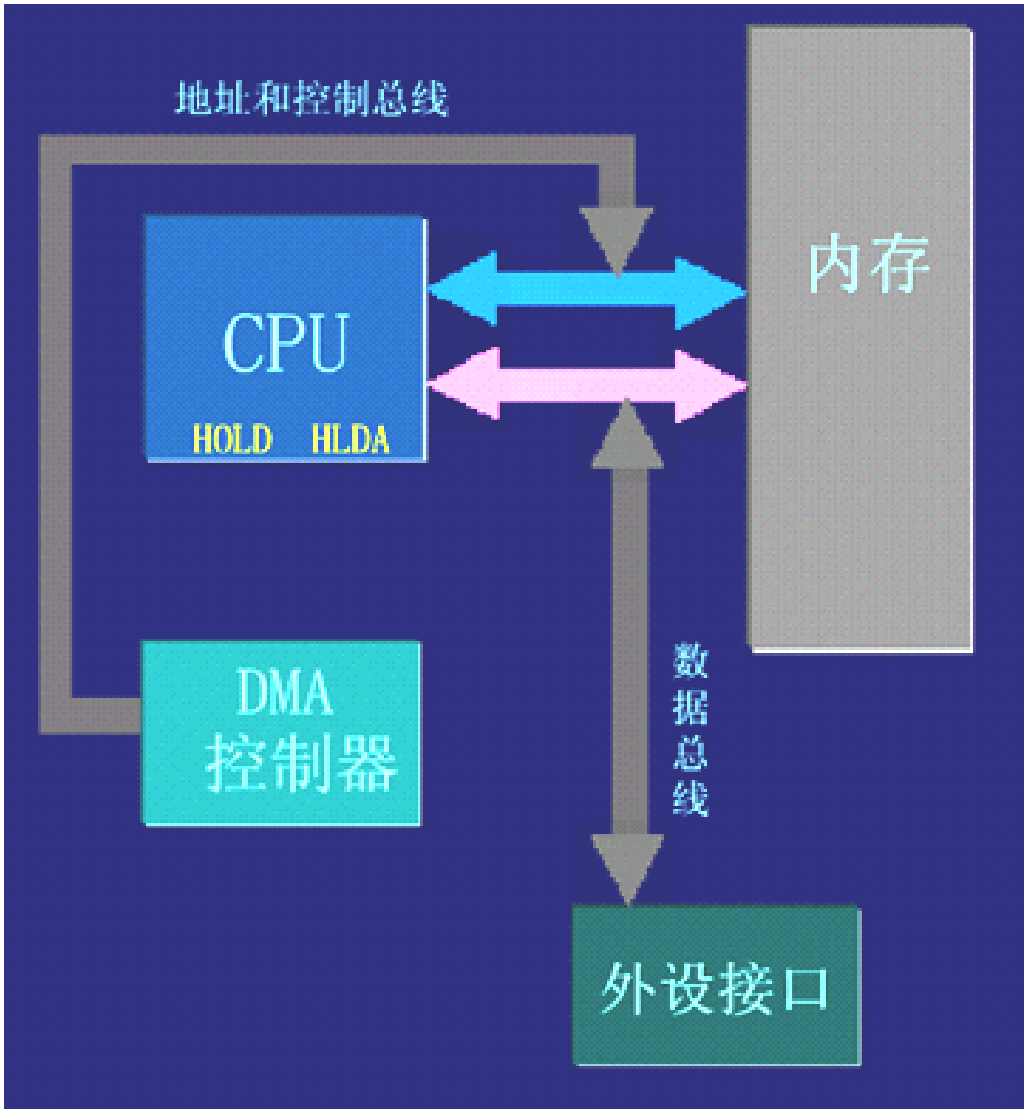
DMA传送的工作过程

1. CPU对DMA控制器进行初始化设置
2. 外设、DMAC和CPU三者通过应答信号建立联系：CPU将总线交给DMAC控制
3. DMA传送
   - DMA读存储器：存储器 → 外设
   - DMA写存储器：存储器 ← 外设
4. 自动增减地址和计数，判断传送完成否

DMA传送流程



DMA传送流程



传送方式的比较

- 无条件传送：慢速外设需与CPU保持同步
- 查询传送： 简单实用，效率较低
- 中断传送：外设主动，可与CPU并行工作，但每次传送需要大量额外时间开销
- DMA传送：DMAC控制，外设直接和存储器进行数据传送，适合大量、快速数据传送

第6章教学要求

- 1. 了解I/O接口电路的主要功能、内部和外部特点、端口编址方法、I/O地址译码特点
- 2. 掌握输入输出指令
- 3. 掌握无条件、查询传送方式

第6章教学要求

- 4. 理解中断、中断源、中断工作过程、中断源识别、优先权排队和中断嵌套

- 5. 理解DMA传送的工作过程