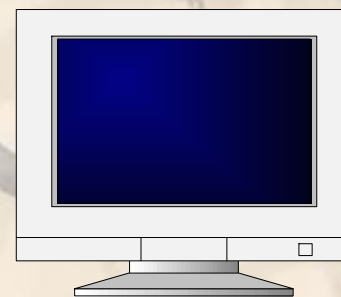


# 第6章

# 输入输出及中断技术





# 主要内容

---

- 基本概念
  - I/O接口和端口
  - 端口的编址方式
- 简单接口芯片及其应用
- 基本输入输出方法
- 中断的基本概念及工作过程



# 6.1 基本概念

---



# 了解和掌握：

---

- I/O接口的基本功能
- 端口的概念及编址方式
- I/O地址译码



## 6.1.1 I/O系统的特点

---

- 复杂性
- 异步性
- 实时性
- 与设备无关性

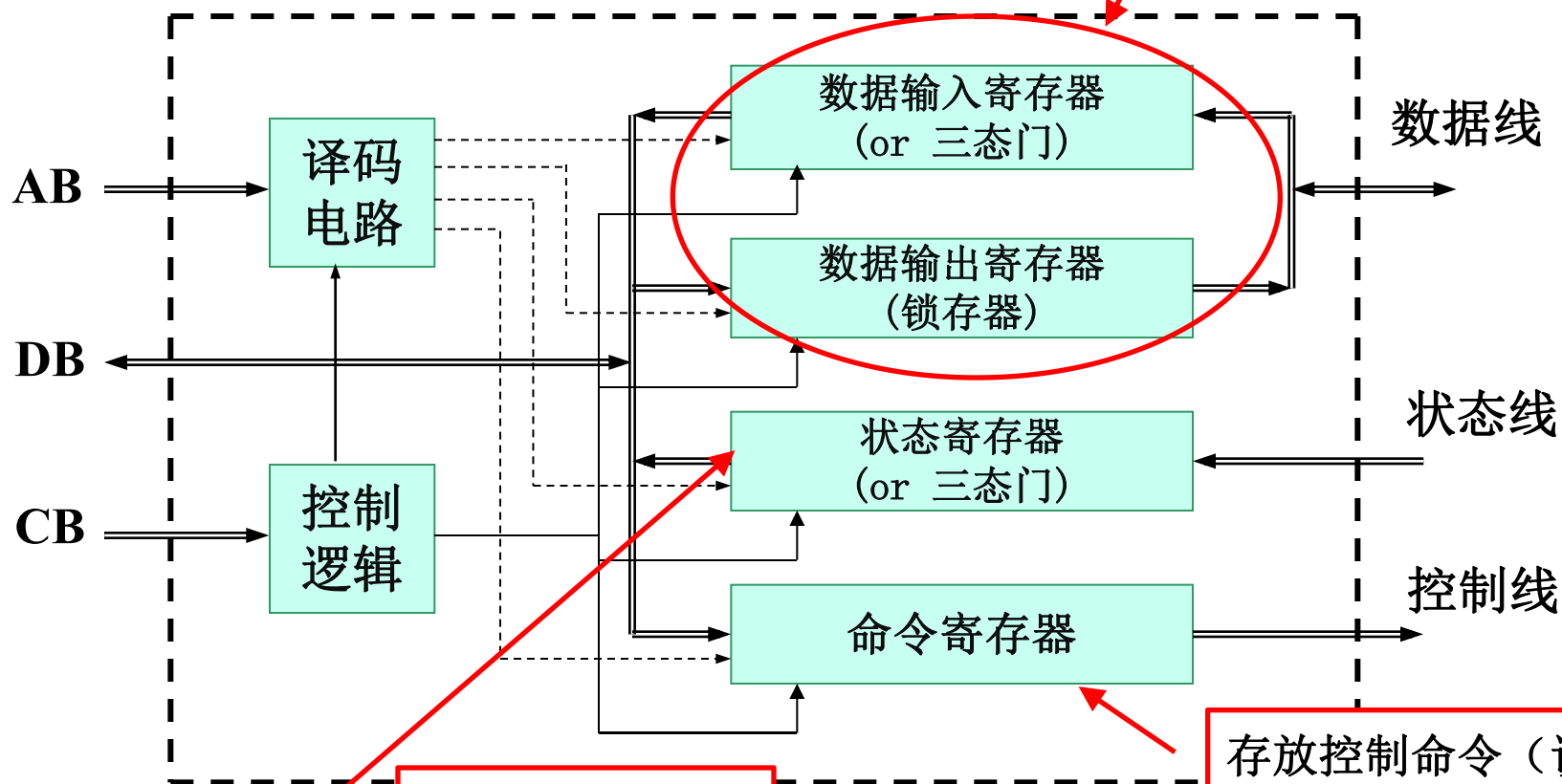


## 6.1.2 I/O接口的基本功能

---

- 1 接口要解决的问题
  - 速度匹配 —————→ 数据的缓冲与暂存
  - 信号的驱动能力 —————→ 信号驱动
  - 信号形式和电平的匹配 —————→ 信号类型转换
  - 信息格式 —————→ 信号格式转换
  - 时序匹配（定时关系）
  - 总线隔离（三态门）

## 2. 接口的基本构成与功能



暂存输入  
输出数据

保存外设当前状态，供CPU读取

存放控制命令（设定接口功能、参数和工作方式）



# I/O端口

---

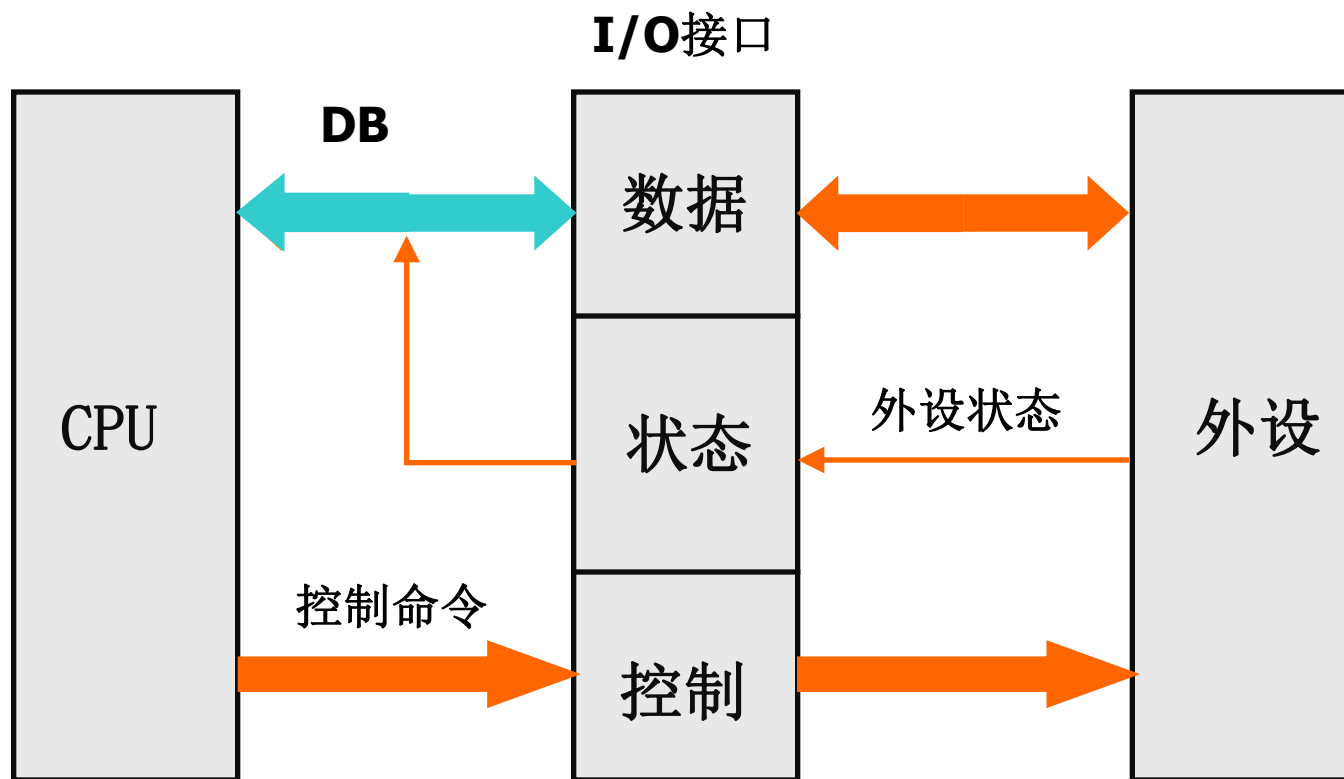
- 端口：

- 接口电路中用于缓存数据及控制信息的部件

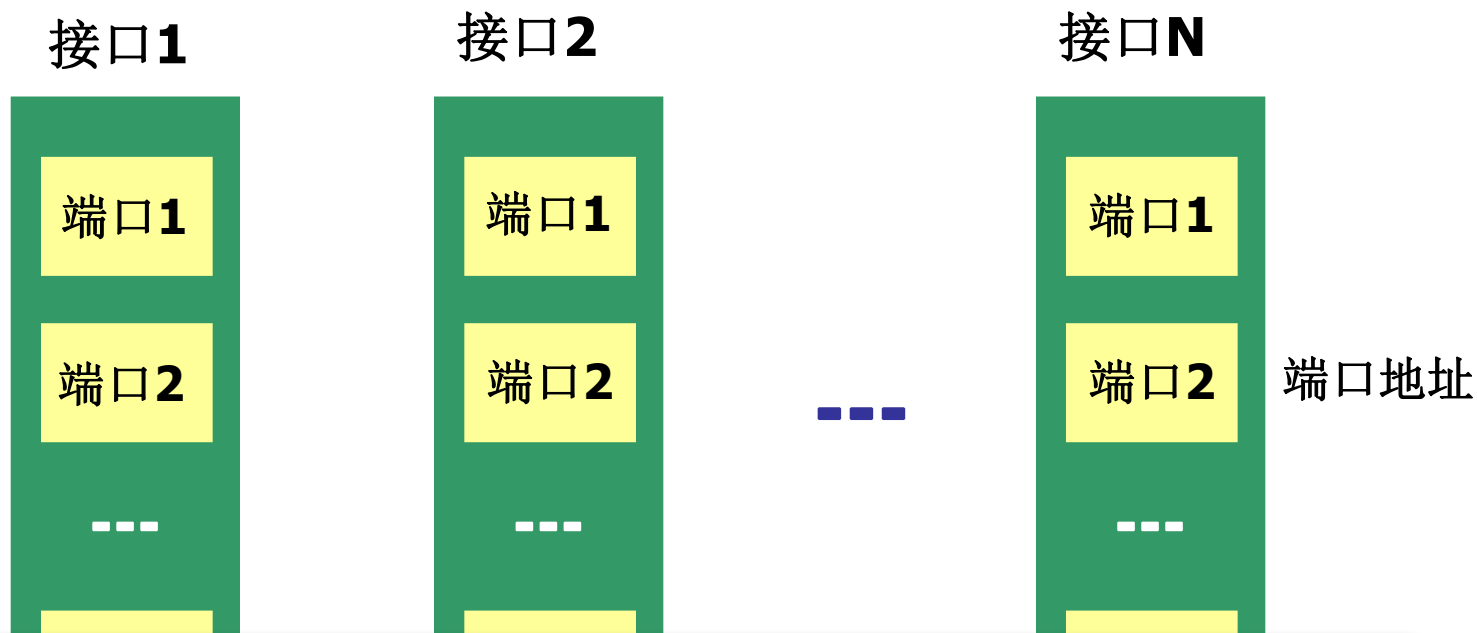
端口 { 数据端口  
状态端口  
控制端口



# I/O端口



# I/O端口编址



每个端口地址=片选地址（高位地址）+片内地址



## 6.1.3 I/O端口的编址方式

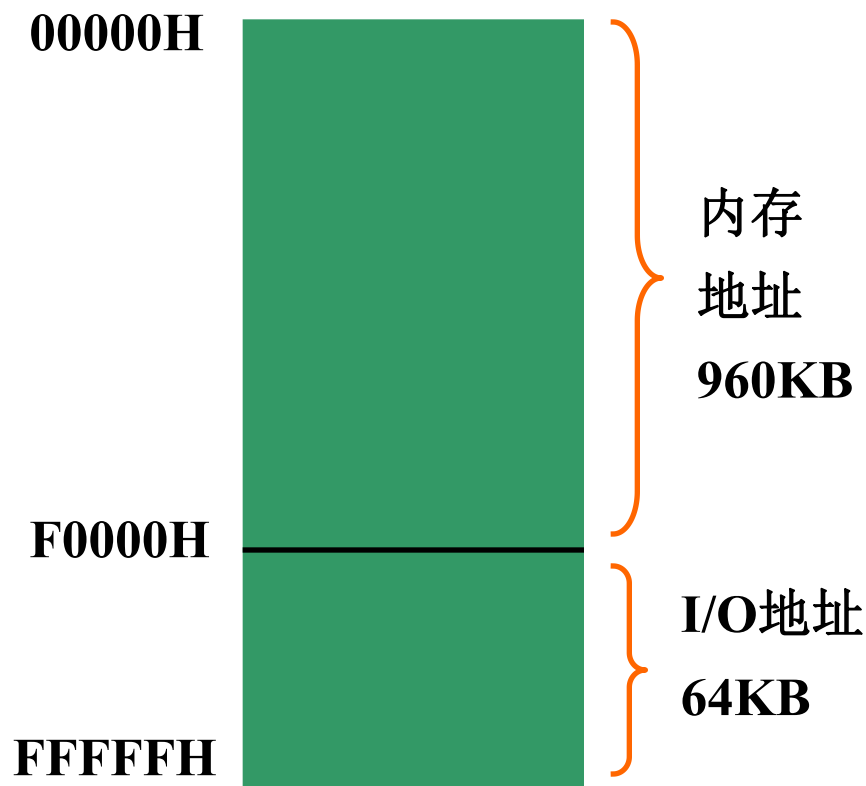
---

- 8086/8088寻址端口的能力：
  - 64K个端口
- 端口的编址方式：
  - 与内存统一编址
  - 独立编址

# 端口与内存的统一编址

## 特点:

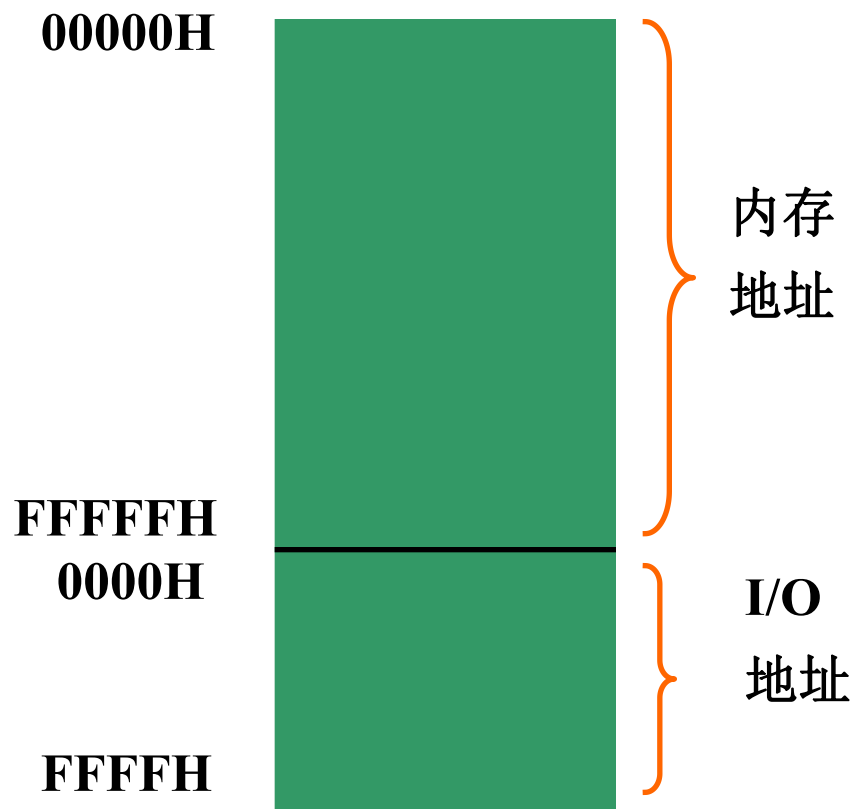
- 指令及控制信号统一;
- 内存地址资源减少



# 端口的独立编址

特点:

- 内存地址资源充分利用
- 能够应用于端口的指令较少





# 8088/8086的I/O端口编址

---

- 采用I/O独立编址方式(但地址线与存储器共用)
- 地址线上的地址信号用IO/#M来区分
- I/O操作只使用20根地址线中的**16根**:  $A_{15} \sim A_0$
- 可寻址的I/O端口数为64K (65536) 个
- I/O地址范围为0~FFFFH
- IBM PC只使用了1024个I/O地址 (0~3FFH)



## 4. I/O地址的译码

---

- 目的：
  - 确定端口的地址
- 参加译码的信号：
  - **#IOR**, **#IOW**, 高位地址信号
- 产生条件
  - **IO/#M=1**
  - **#RD=0** → **#IOR=0**
  - **#WR=0** → **#IOW=0**
- OUT指令将使总线的IOW信号有效
- IN指令将使总线的IOR信号有效



# I/O译码的地址信号

---

- 当接口只有一个端口时：
  - 无片内地址，全部地址信号均为高位地址（可全部参与译码），译码输出直接选择该端口；
- 当接口具有多个端口时：
  - 则16位地址线的高位参与译码（决定接口的基地址），而低位则用于确定要访问哪一个端口。

由于端口资源丰富，端口地址译码常采用部分地址译码





# I/O地址译码例

---

- 某外设接口有4个端口，地址为2F0H——2F3H，由A<sub>15</sub>~A<sub>2</sub>译码得到，而A<sub>1</sub>、A<sub>0</sub>用来区分接口中的4个端口。试画该接口与系统的连接图。

# I/O地址译码例

## ■ 地址范围:

**A11**

■	×	×	×	×	0	0	1	0	1	1	1	1	0	0	0	0
■	×	×	×	×	0	0	1	0	1	1	1	1	0	0	1	1

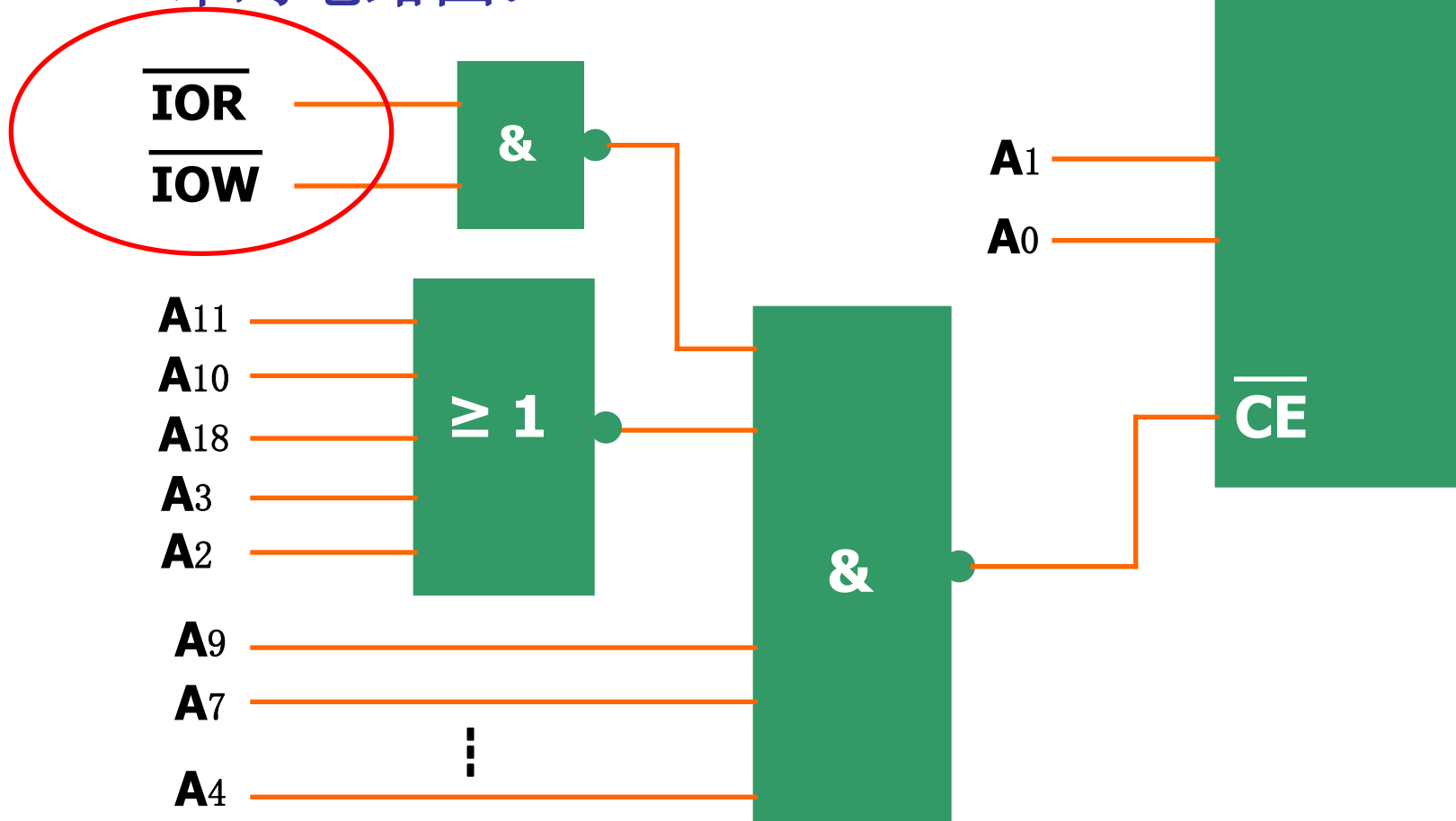
任意状态

片内地址

图中不接入

# I/O地址译码例

## 译码电路图：





以下哪些属于存储器扩展方法? (多选)

- ☒ A 位扩展
- ☒ B 字扩展
- ☒ C 字位扩展
- ☐ D 总线扩展

提交



## 6.2 简单接口电路

---



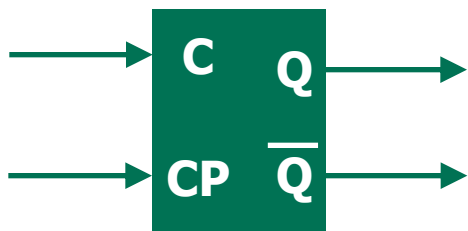
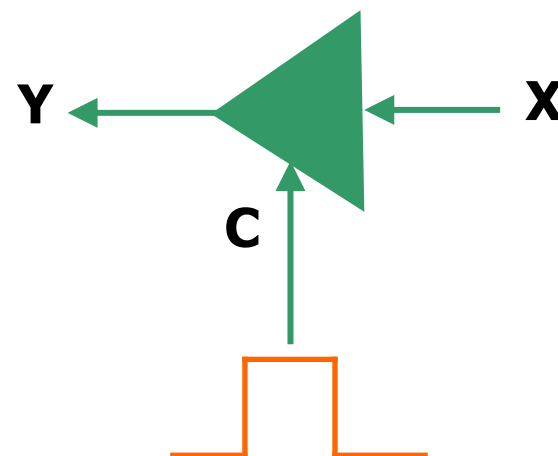
# 1. 接口的类型及特点

---

- 按传输信息的方向分类：
  - 输入接口
  - 输出接口
- 按传输信息的类型分类：
  - 数字接口
  - 模拟接口
- 按传输信息的方式分类：
  - 并行接口
  - 串行接口

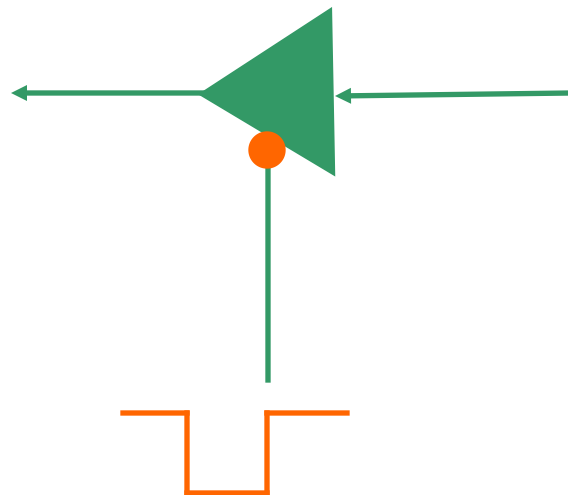
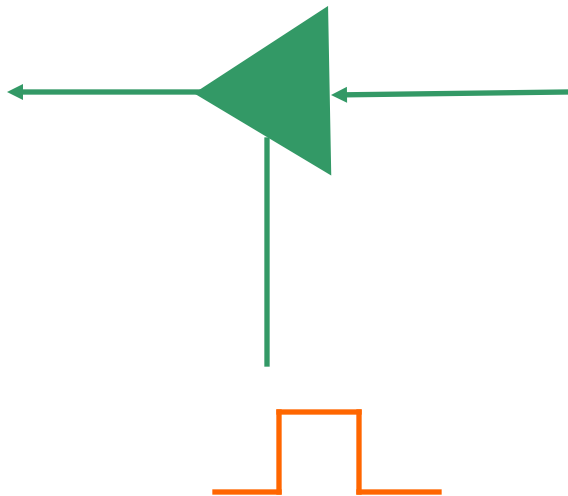
# 接口特点

- 输入接口：
  - 要求对数据具有控制能力
  - 常用三态门实现
- 输出接口：
  - 要求对数据具有锁存能力
  - 常用锁存器实现



# 三态门接口

- 高电平、低电平、高阻态







# 74LS244

---

- 含8个三态门的集成电路芯片
- 在外设具有数据保持能力时用来输入接口
- 74LS244应用例
  - 教材p245

P244图



## 3. 锁存器接口

---

- 通常由D触发器构成;
- 特点:
  - 具有对数据的锁存能力
  - 不具备对数据的控制能力



# 常用锁存器芯片

---

- 74LS273

- 8D触发器，不具备数据的控制能力



P246图

- 74LS373

- 含三态的8D触发器，具有对数据的控制能力。
  - 既可以做输入接口，也可以做输出接口。



# I/O接口综合应用例

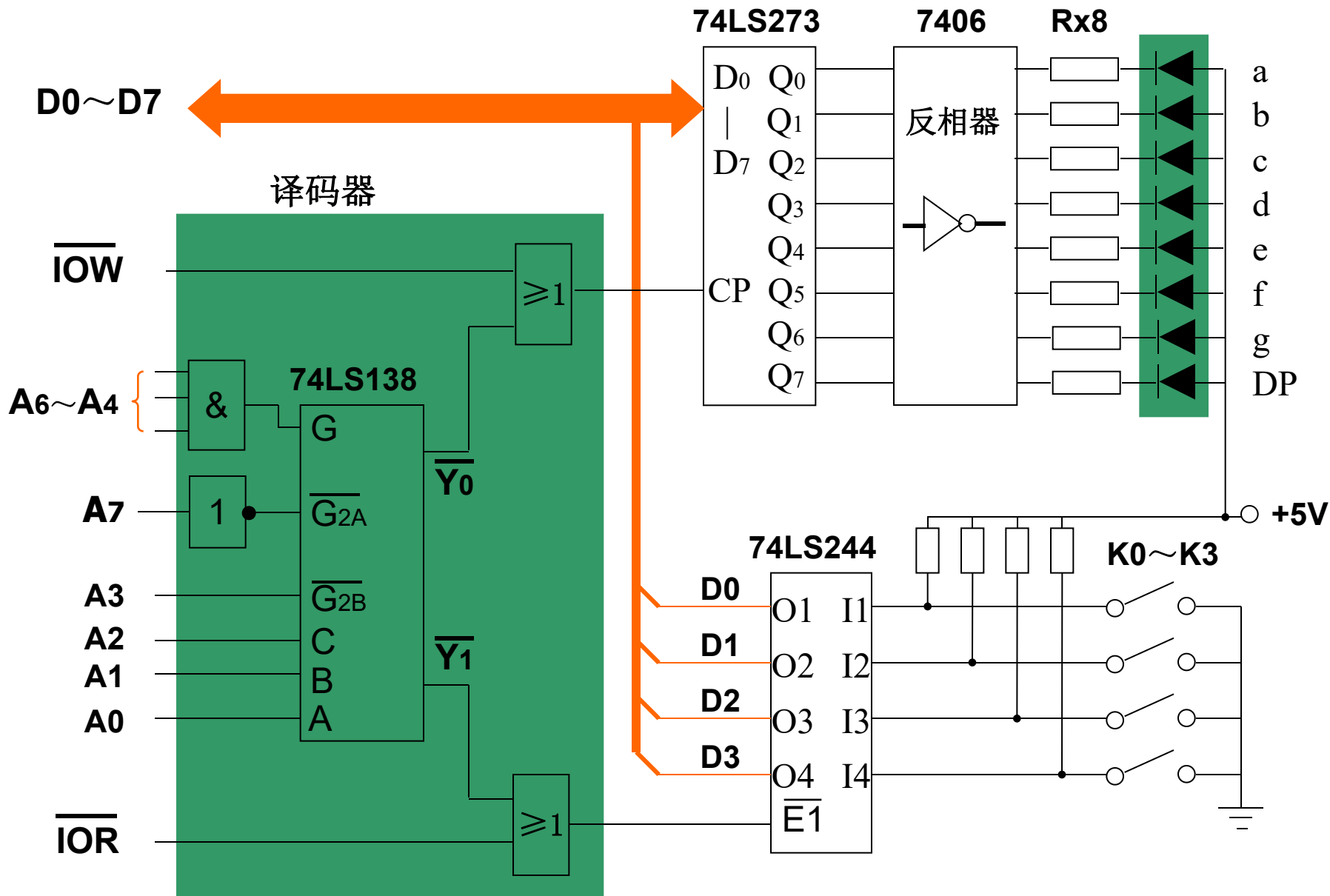
---














- 根据开关状态在7段数码管上显示数字或符号
- 设输出接口的地址为F0H
- 设输入接口地址为F1H
- 当开关的状态分别为0000~1111时，在7段数码管上对应显示'0'~'F'

7段数码管图见教材p248

**F1H = 1111 0001**

# A0



符号	形状	7段码 .gfedcba	符号	形状	7段码 .gfedcba
'0'		00111111	'8'		01111111
'1'		00000110	'9'		01100111
'2'		01011011	'A'		01110111
'3'		01001111	'B'		01111100
'4'		01100110	'C'		00111001
'5'		01101101	'D'		01011110
'6'		01111101	'E'		01111001
'7'		00000111	'F'		01110001

# I/O接口综合应用例 —— 程序段

.....

```
Seg7 DB 3FH,06H,5BH,4FH,66H,6DH,7DH,07H,  
       7FH,67H,77H,7CH,39H,5EH,79H,71H
```

.....

```
LEA  BX, Seg7  
MOV  AH, 0  
GO: IN  AL, 0F1H  
     AND  AL, 0FH  
     MOV  SI, AX  
     MOV  AL, [BX+SI]  
     OUT  0F0H, AL  
     JMP  GO
```



## 6.3 基本输入/输出方法

---





# 基本输入/输出方法

---

无条件传送

查询式传送

中断方式传送

直接存储器存取 (DMA)

} 程序控制方式



# 1. 无条件传送

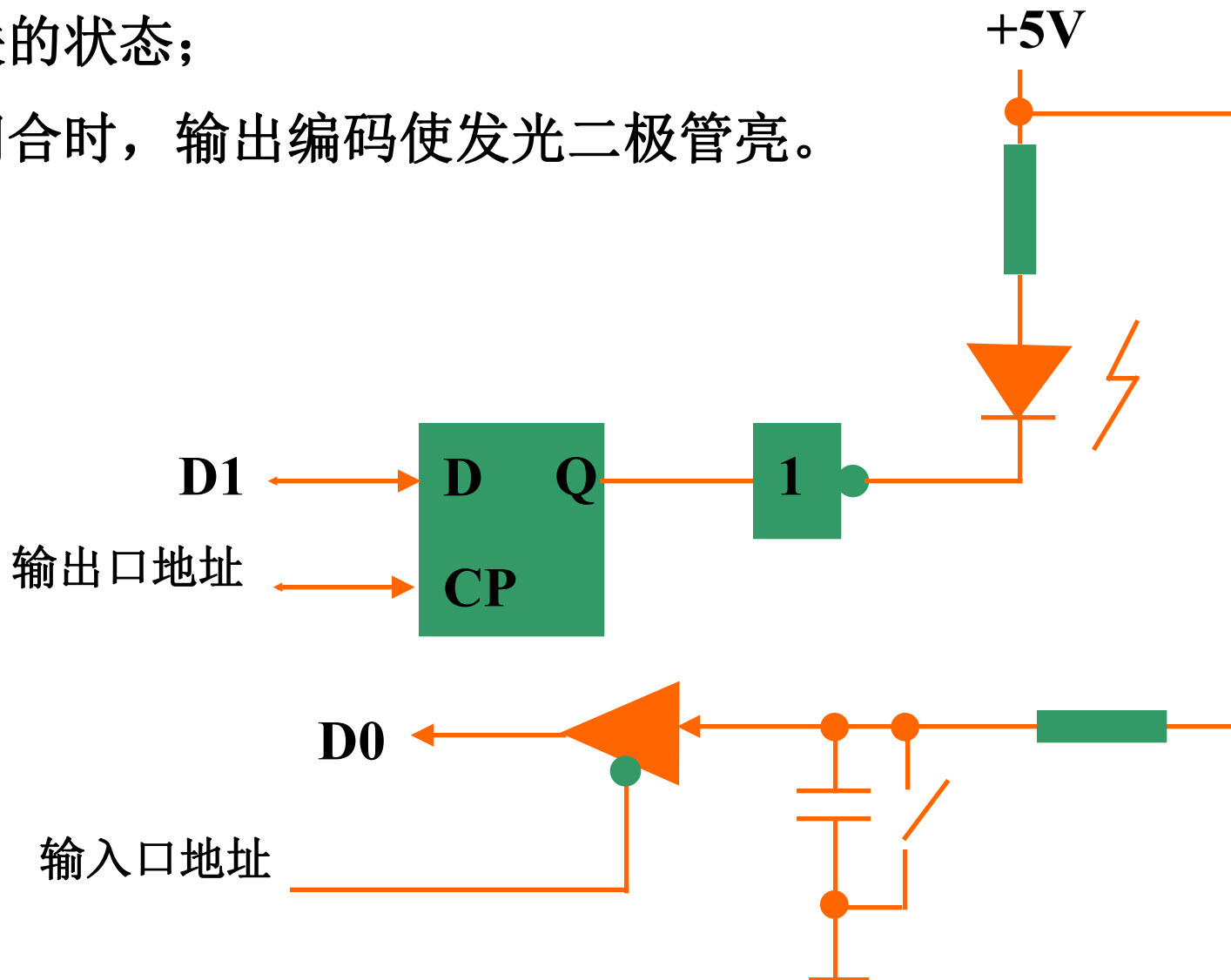
---

- 要求外设总是处于准备好状态
- 优点：
  - 软件及接口硬件简单
- 缺点：
  - 只适用于简单外设，适应范围较窄

# 无条件传送例

读取开关的状态；

当开关闭合时，输出编码使发光二极管亮。



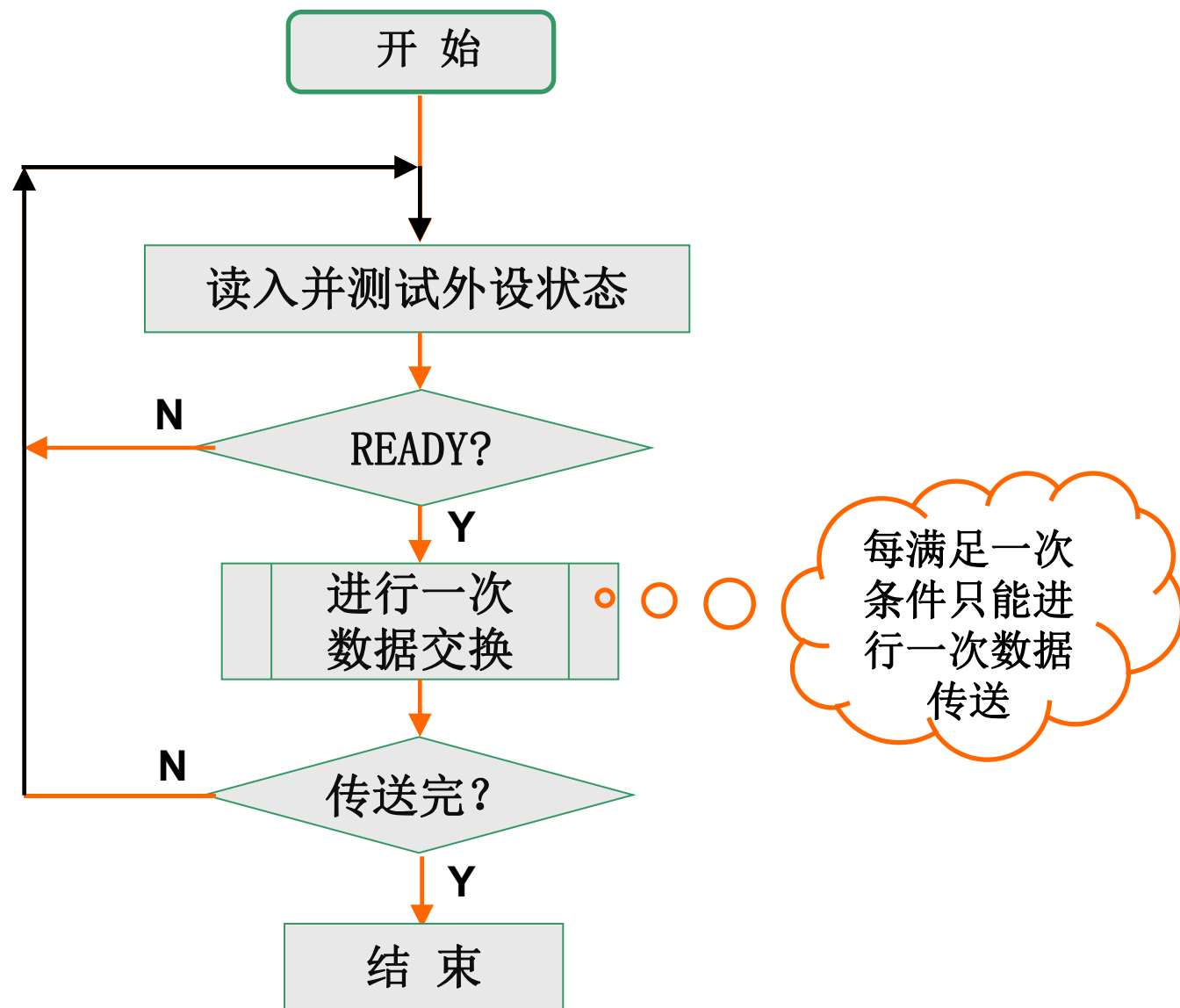


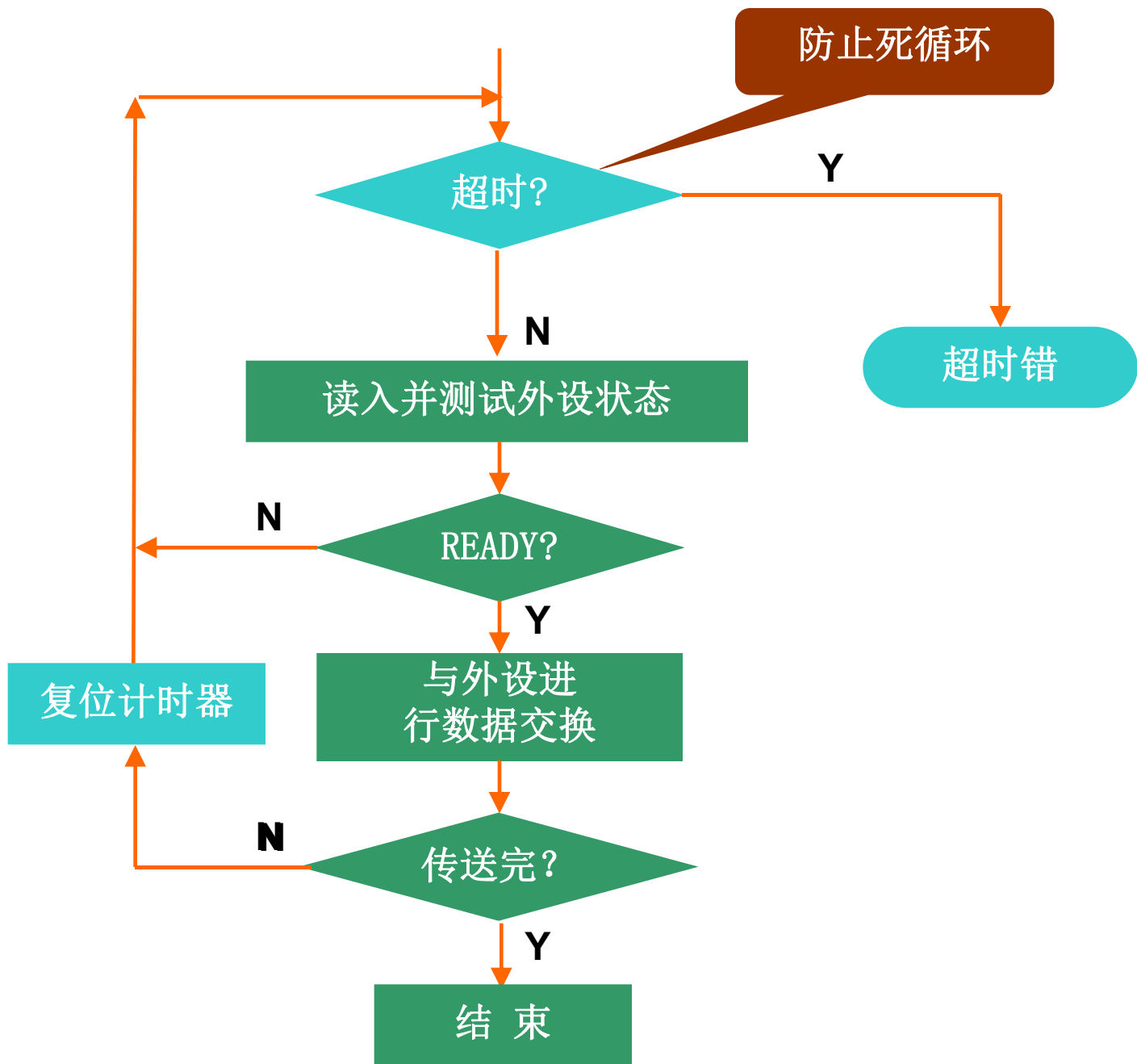
## 2. 查询工作方式

---

- 仅当条件满足时才能进行数据传送；
- 每满足一次条件只能进行一次数据传送。
- 适用场合：
  - 外设并不总是准备好
  - 对传送速率和效率要求不高
- 工作条件：
  - 外设应提供设备状态信息
  - 接口应具备状态端口

# 查询工作方式流程图







# 查询工作方式例

---

- 外设状态端口地址为03FBH，第5位(bit5)为状态标志（=1忙，=0准备好）
- 外设数据端口地址为03F8H，写入数据会使状态标志置1；外设把数据读走后又把它置0。
- 试画出其电路图，并将DATA下100B数据输出。



# 查询工作方式例

---

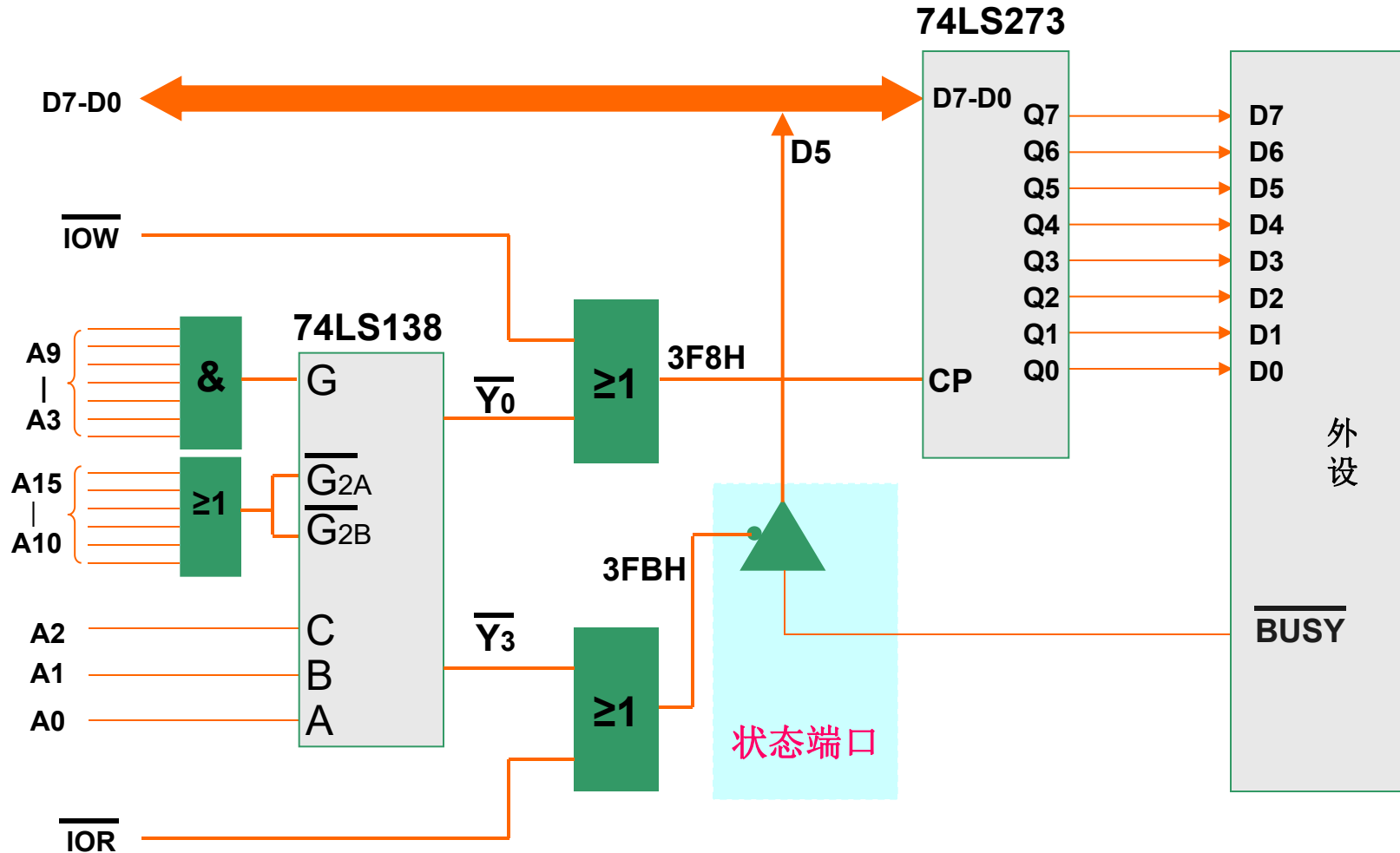
## ■ 题目分析：

- 外设有1位状态位，需要通过输入接口将状态信息输入系统；
  - 可选择**一个三态门或74LS244**接口
- 数据需由系统输出，需要通过输出接口
  - 可选择**74LS273**接口
- 输入接口地址= 03FBH，bit5=1表示“忙”；
- 输出接口地址=03F8H
- 待输出数据在内存中的首地址=DATA；
- 待输出数据块大小=100B



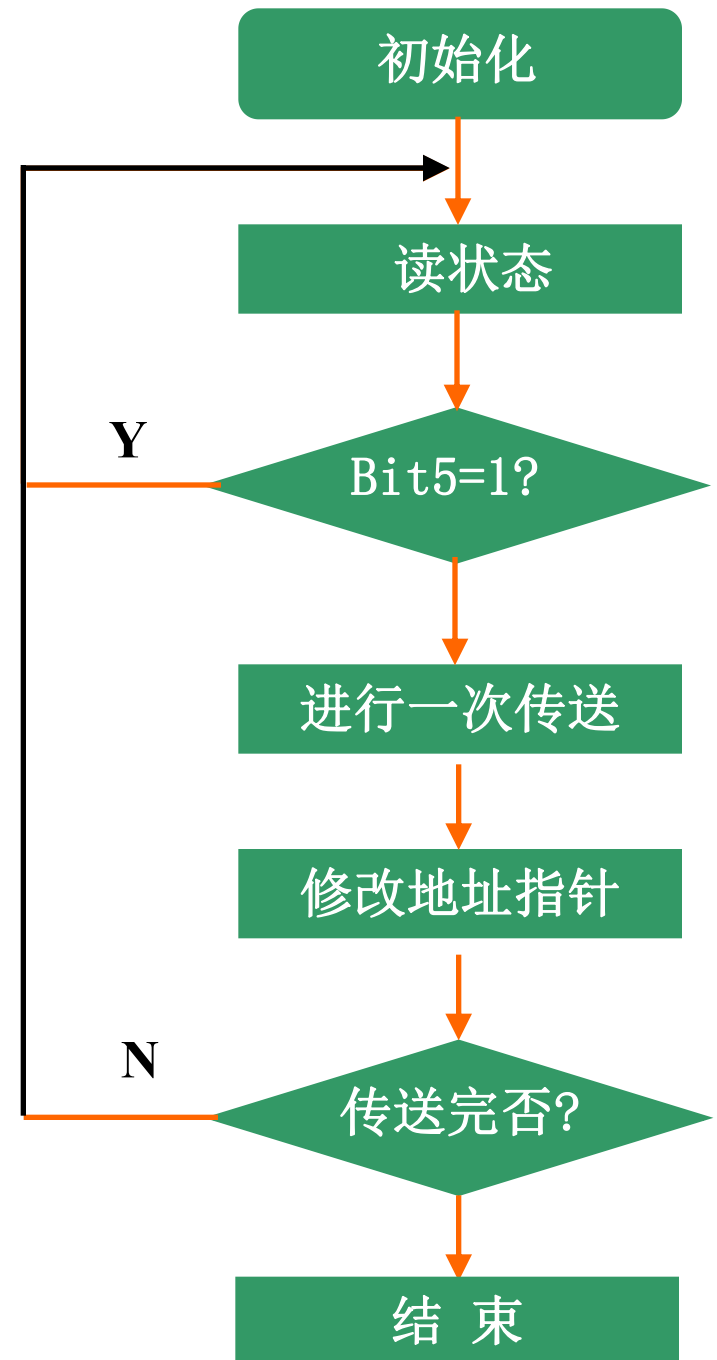
状态端口地址: **0000 0011 1111 1011** 外设状态端口地址为**03FBH**

数据端口地址: **0000 0011 1111 1000** 外设数据端口地址为**03F8H**



# 控制程序

```
LEA SI,DATA
MOV CX,100
AGAIN: MOV DX,03FBH
WAITT: IN AL,DX
      TEST AL,20H
      JNZ WAITT
      MOV DX,03F8H
      MOV AL, [SI]
      OUT DX, AL
      INC SI
      LOOP AGAIN
      HLT
```



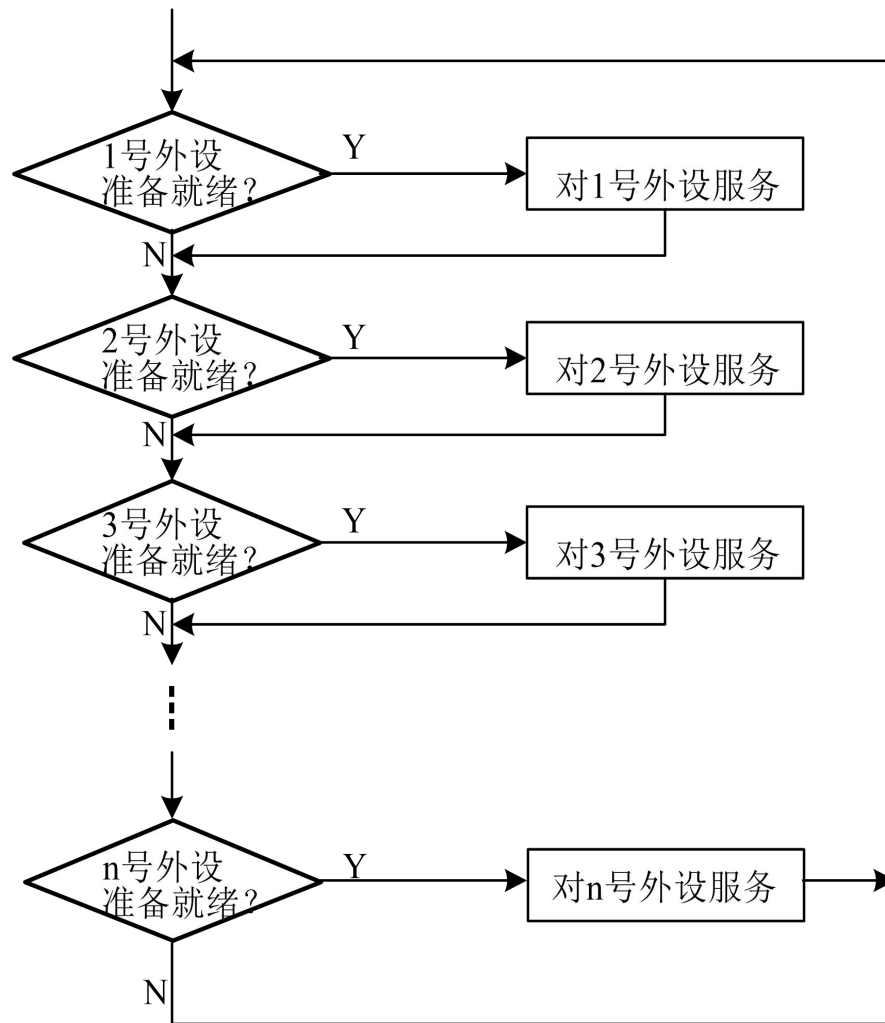
# 查询工作方式

## ■ 优点:

- 软硬件比较简单

## ■ 缺点:

- CPU效率低，数据传送的实时性差，速度较慢





## 3. 中断控制方式

---

- 特点:

- 外设需要在需要时向CPU提出请求，CPU再去为它服务。服务结束后或在外设不需要时，CPU可执行自己的程序。

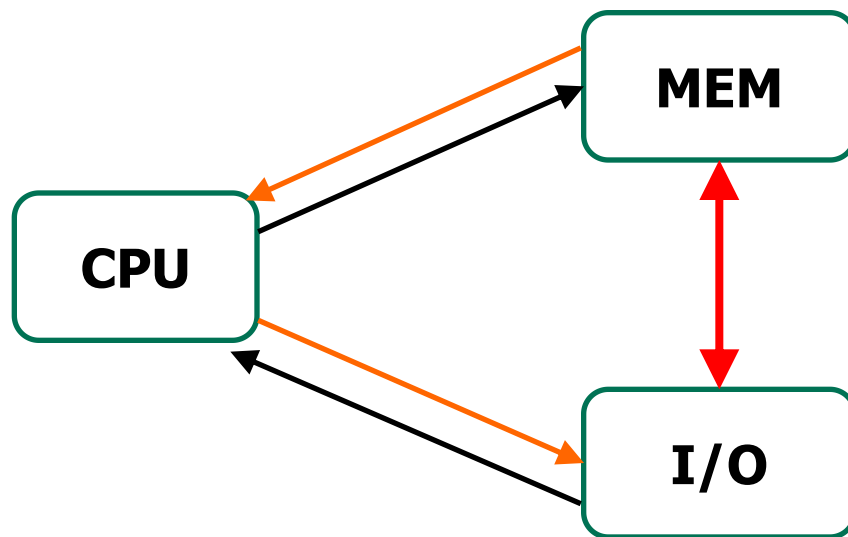
- 优点:

- CPU效率高，实时性好，速度快。

- 缺点:

- 程序编制相对较为复杂。

# 以上三种I/O方式的共性





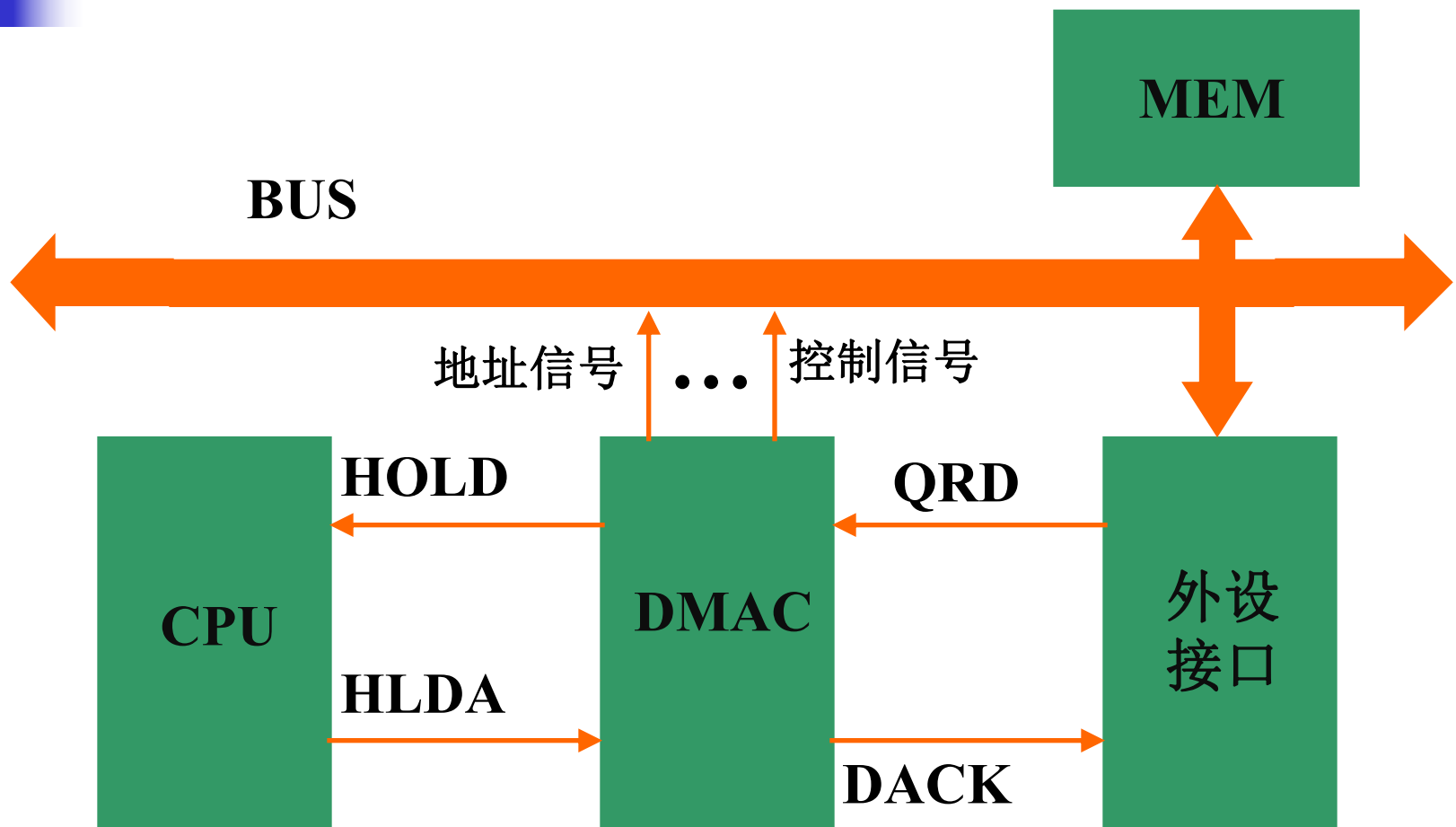
## 4. DMA控制方式

---

### ■ 特点:

- 外设直接与存储器进行数据交换，CPU不再担当数据传输的中介者；
- 总线由DMA控制器（DMAC）进行控制（CPU要放弃总线控制权），内存/外设的地址和读写控制信号均由DMAC提供。

# DMA控制方式





# DMA控制方式的工作过程

---

- 外设向DMA控制器发出“DMA传送请求”信号DRQ;
- DMA控制器收到请求后, 向CPU发出“总线请求”信号HOLD;
- CPU在完成当前总线周期后会立即发出HLDA信号, 对HOLD信号进行响应;
- DMA控制器收到HLDA信号后, 就开始控制总线, 并向外设发出DMA响应信号DACK。



# DMA工作方式

## ■ 周期窃取:

- 每个DMA周期只传送一个字节或一个字就立即释放总线。

## ■ 数据块传送:

要求接口要有较大缓存

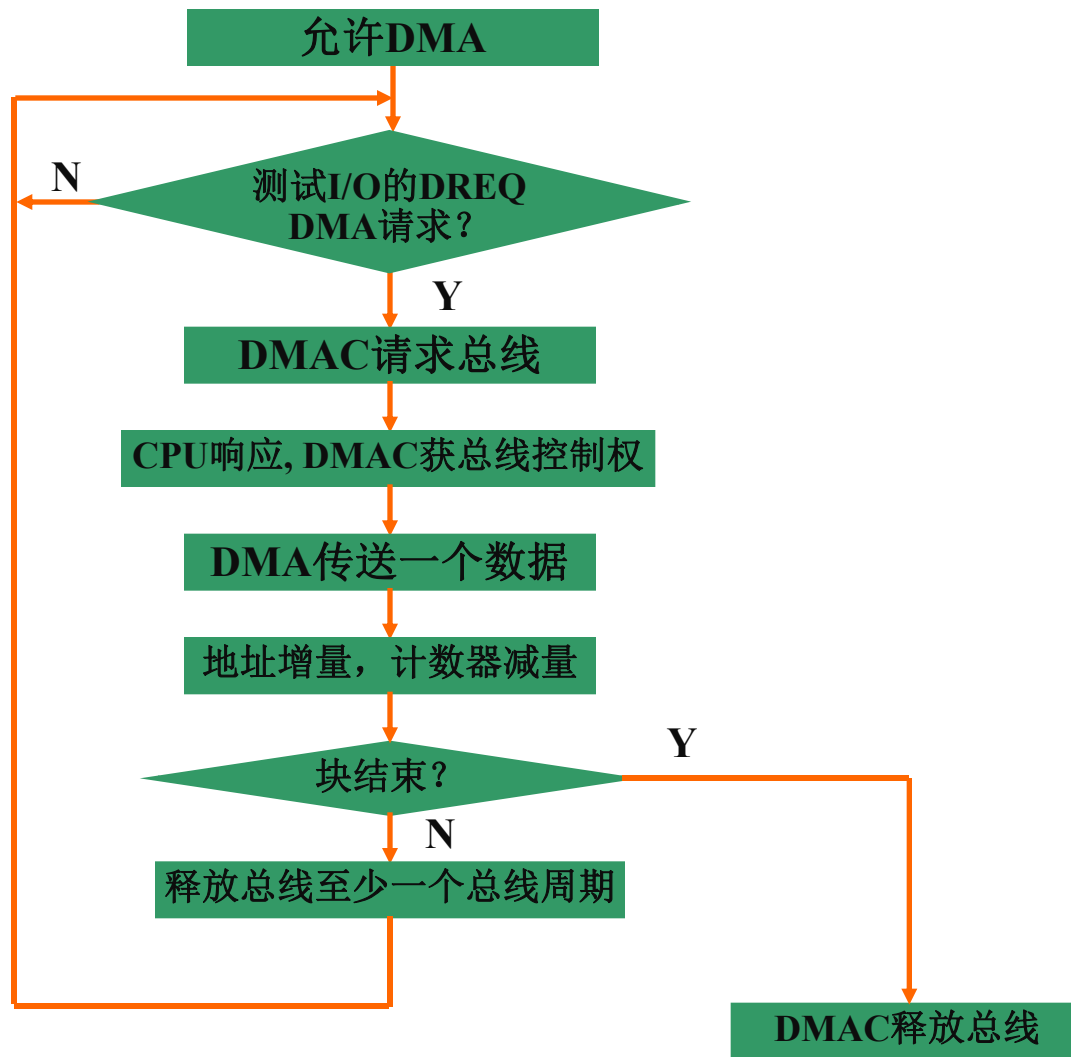
- DMAC在申请到总线后，将一块数据传送完后才释放总线，而不管中间DREQ是否有效。

## ■ 直接存取方式:

对硬件有较高要求

- DMA的数据传送请求直接发到主存储器，在得到响应后，整个工作过程在DMA控制器中由硬件完成。

# 周期窃取的DMA方式:

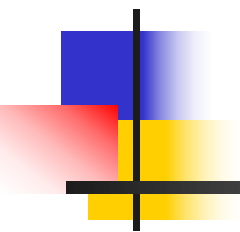




# DMA控制方式

---

- 数据传输由DMA硬件来控制，数据直接在内存和外设之间交换，可以达到很高的传输速率。
- 控制复杂，硬件成本相对较高。



# 6.4 中断技术

---



# 掌握：

---

- 中断的基本概念
- 中断响应的一般过程
- 中断向量表及其初始化
- 8088/8086中断系统



## 6.4.1 中断的基本概念

### ■ 中断：

- CPU执行程序时，由于发生了某种随机的事件（外部或内部），引起CPU暂时中断正在运行的程序，转去执行一段特殊的服务程序，以处理该事件，该事件处理完后又返回被中断的程序继续执行，这一过程称为中断。



中断服务（处理）子程序



中断源



# 引入中断的原因

---

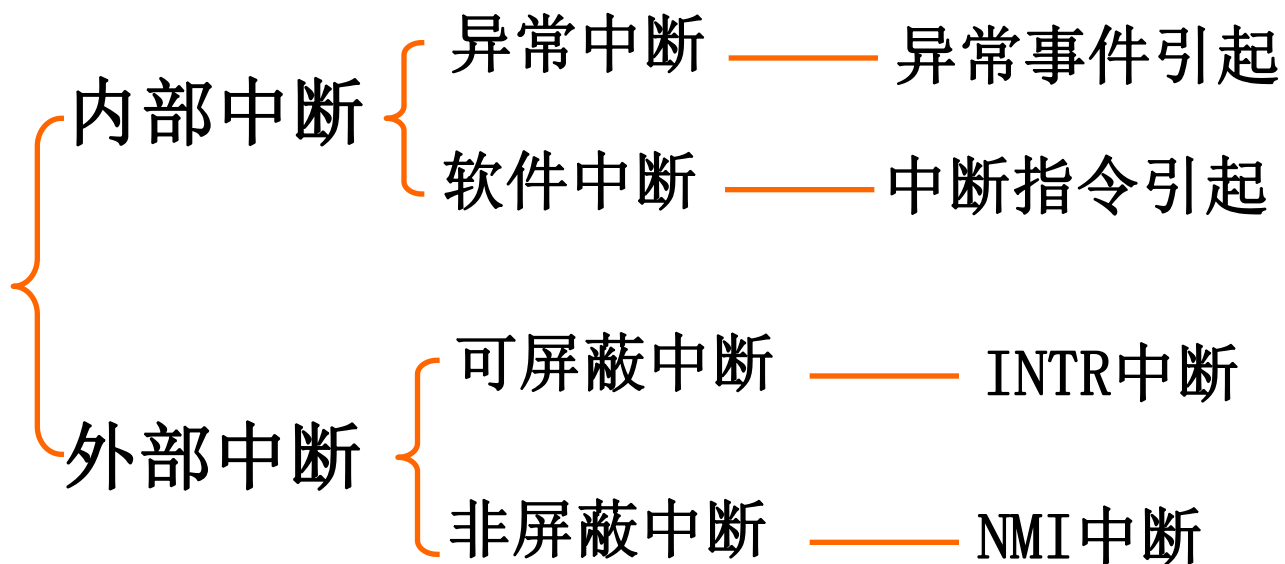
- 提高对外设请求的响应实时性。
- 提高了CPU的利用率
  - 避免了CPU不断检测外设状态的过程



# 中断类型

---

- 根据中断请求的来源分为：







## 6.4.2 外部中断响应的一般过程

---

- 中断请求
- 中断源识别及中断判优
- 中断响应
- 中断处理（服务）
- 中断返回



# 中断请求

---

**NMI**  
**INTR**

- 中断请求信号应保持到中断被处理为止；
- CPU响应中断后，中断请求信号应及时撤销。



# 中断源识别

- 软件查询法
- 中断矢量法
  - 由中断源提供中断类型号，CPU根据类型确定中断源。
- 当有多个中断源同时提出请求时，需要确定首先响应哪一个中断源 → 中断判优
  - 优先级法则
    - 低优先级的中断程序允许被高优先级的中断源所中断

中断源识别及判优由硬件系统完成

# 中断判优

## ■ 软件判优

- 顺序查询中断请求，先查询的先服务
  - 即先查询的优先级别高

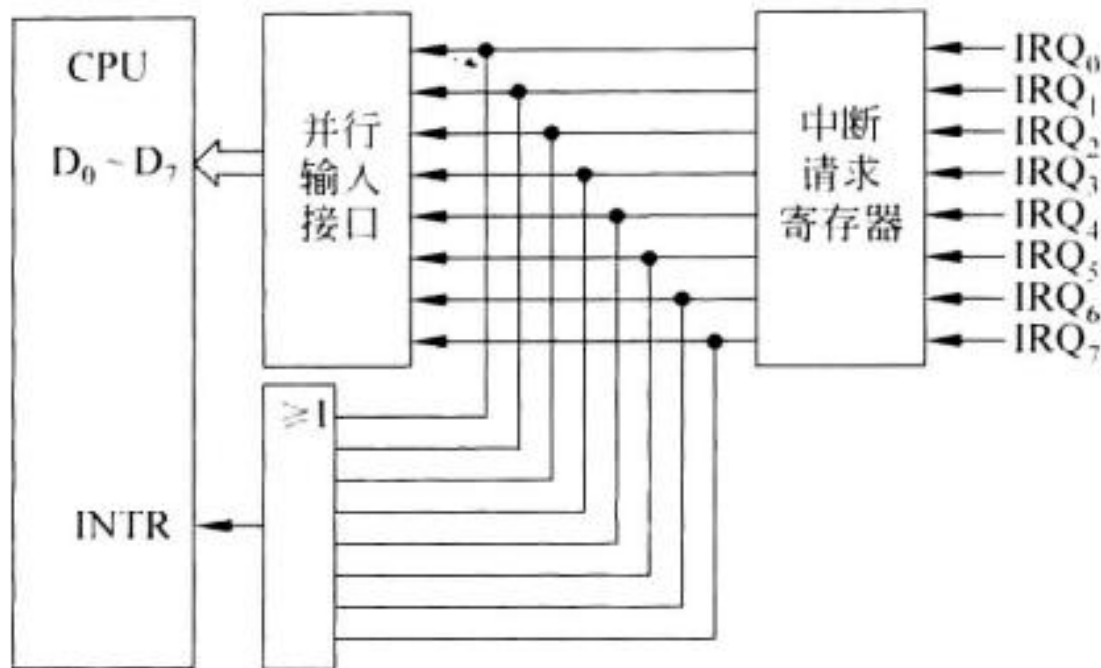


图 6-18 软件判优的电路原理图

## 硬件判优

**(1)**中断控制器判优（常用方法）

**(2)**链式判优（现在已经很少采用）



# 中断响应

---

- 向中断源发出 $\overline{\text{INTA}}$ 中断响应信号;
- 关中断
- 保护硬件现场
  - 将FLAGS压入堆栈
- 保护断点
  - 将CS、IP压入堆栈
- 获得中断服务程序入口地址

由  
硬  
件  
系  
统  
完  
成



# 中断处理

---

- 执行中断服务子程序
- 中断服务子程序的特点：
  - 为“远过程”
  - 用IRET指令返回



# 中断服务子程序完成的工作

---

- 关中断，保护现场，保护断点，找入口地址
- 保护软件现场（参数）
- 开中断（STI）
- 中断处理
- 关中断（CLI）
- 恢复现场
- 中断返回





# 中断返回

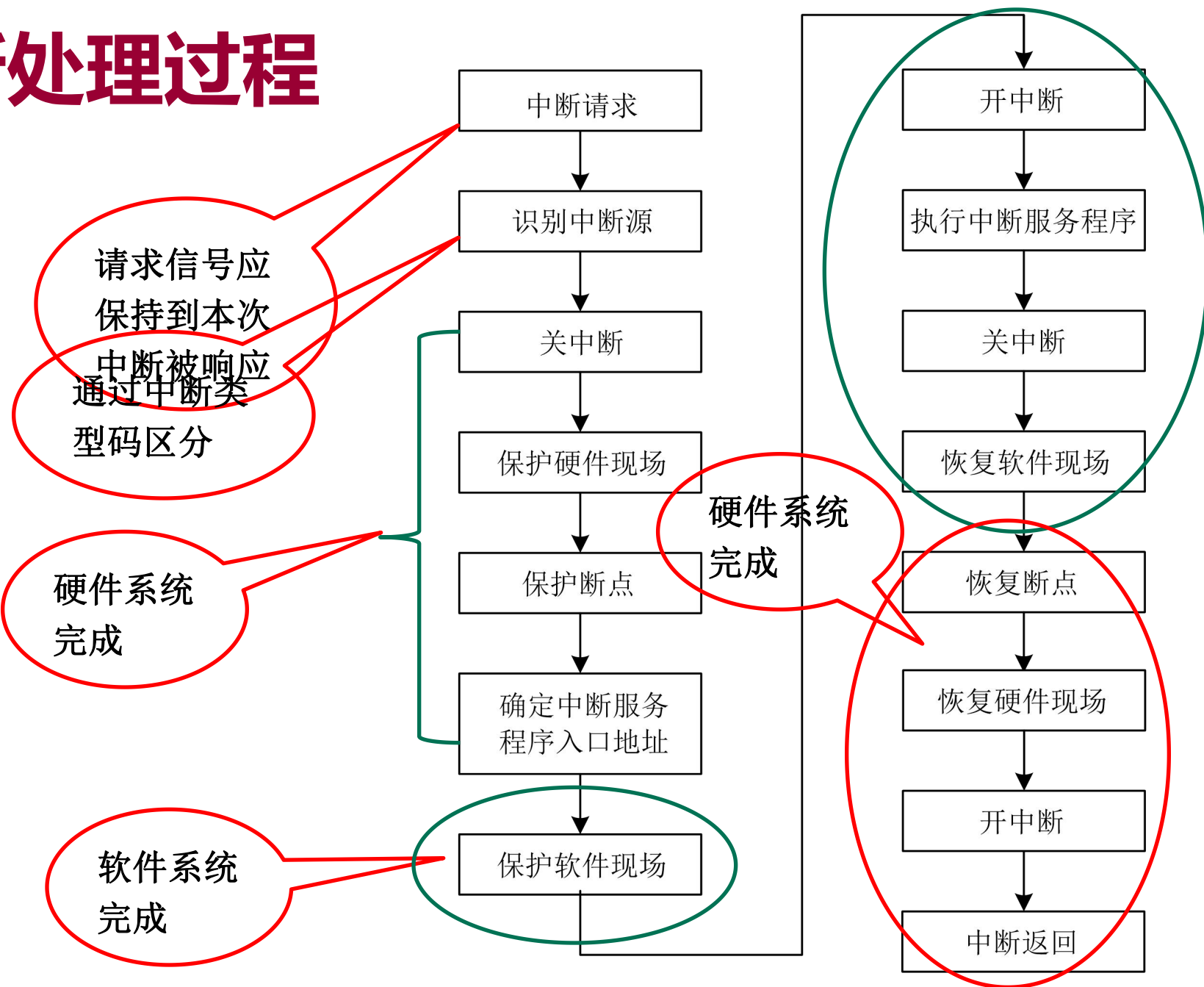
---

- 执行IRET指令，使IP、CS和FLAGS从堆栈弹出

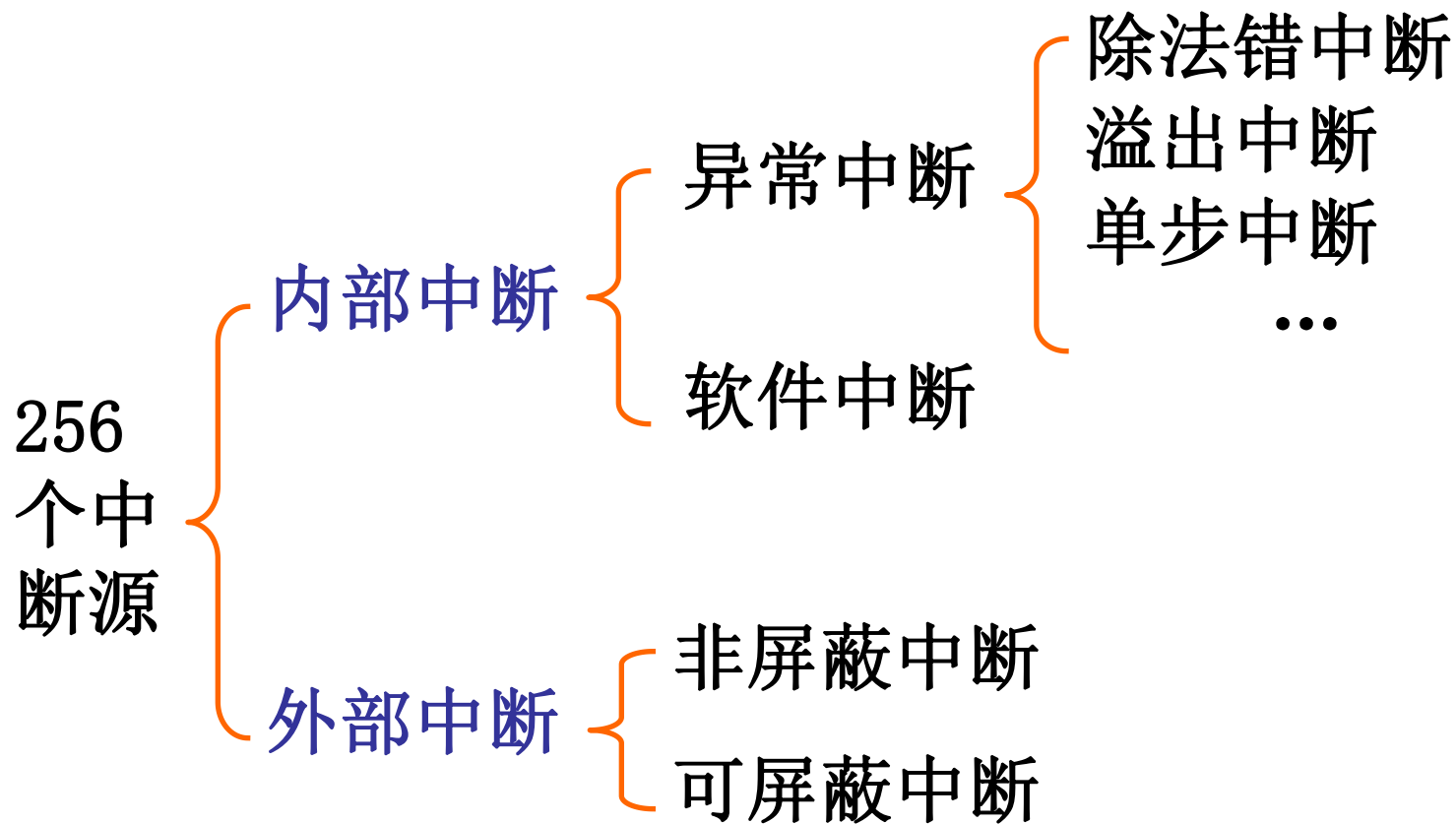


恢复断点和硬件现场

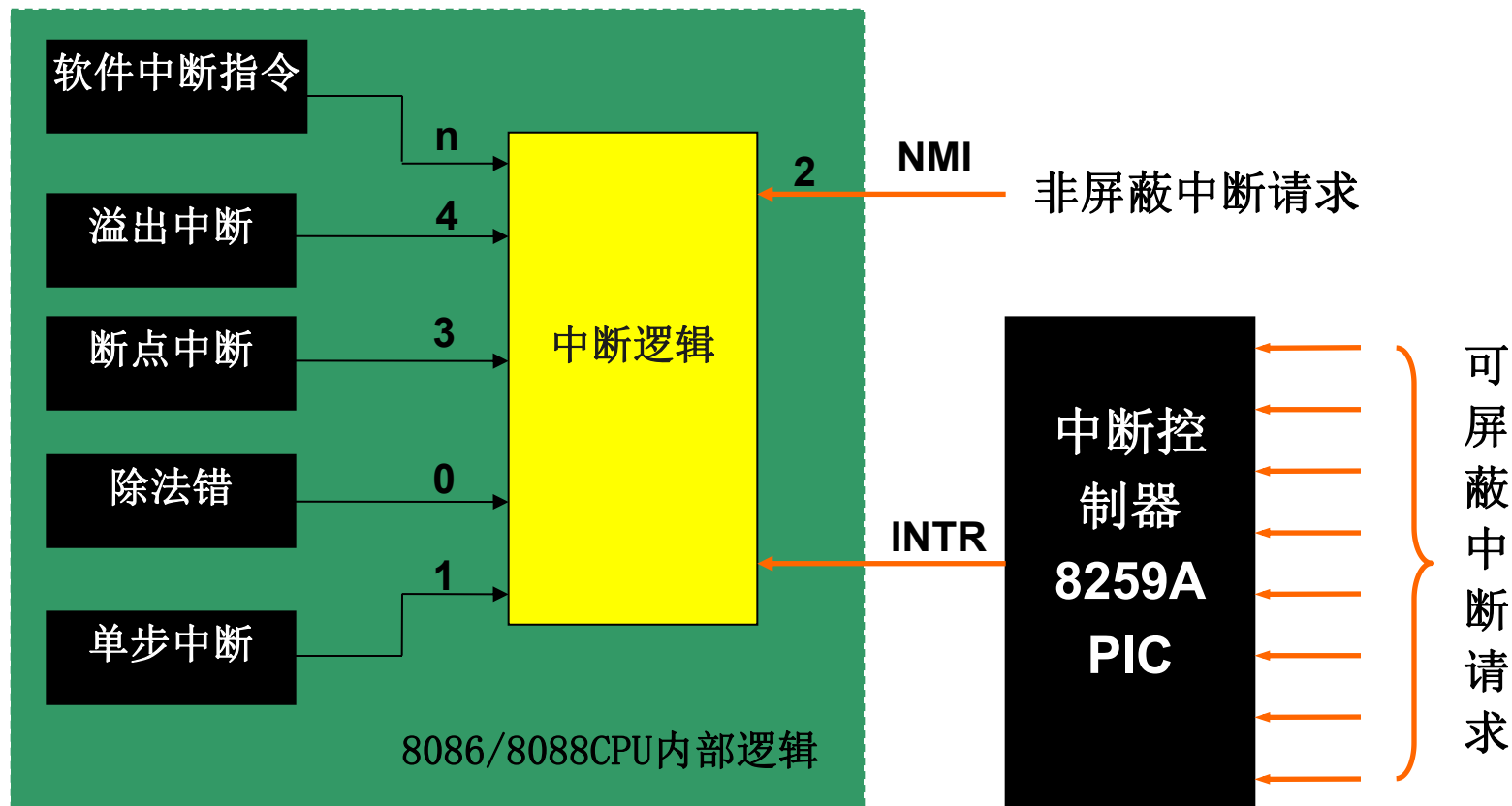
# 中断处理过程



## 6.4.3 8088/8086中断系统



# 8086/8088中断源类型:





---

**1. 内部中断**

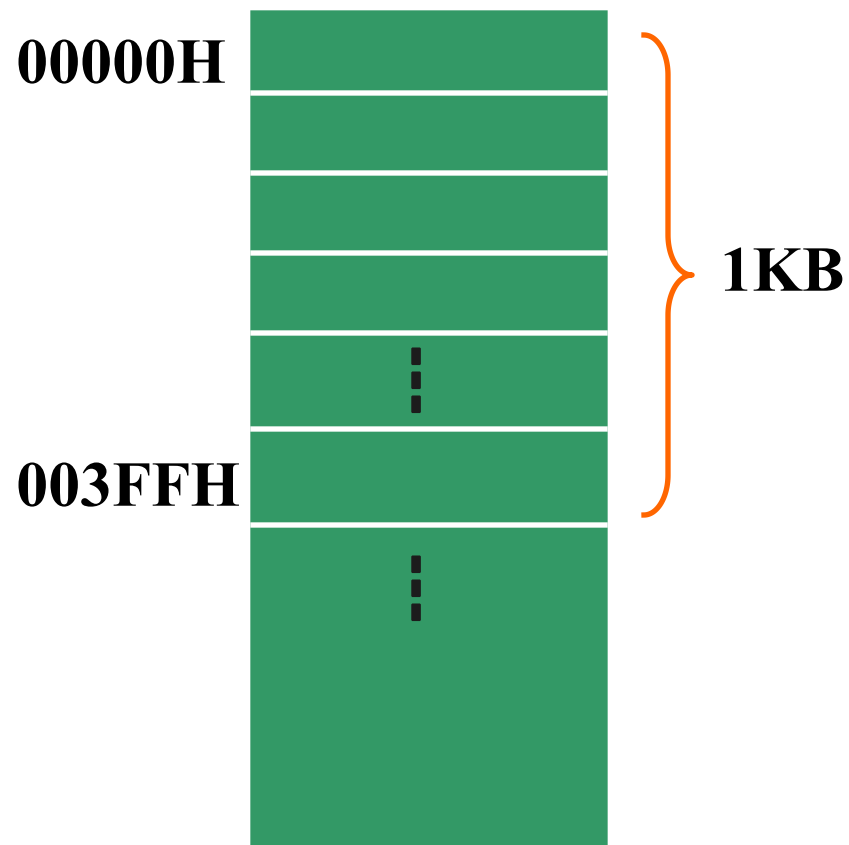
**2. 外部中断**

**见教材P261**



# 中断向量表

---





# 中断向量表

---

- 存放各类中断的中断服务程序的入口地址；
- 每个入口占用4 Bytes，其中：
  - 低地址字单元中存放入口的偏移地址，高地址字单元中存放入口的段基址；
- 表的地址位于内存的00000H~003FFH，大小为1KB，共256个入口。



# 中断向量表的初始化

---

- 将用户自定义的中断服务程序入口地址放入向量表
- 注意点：
  - 向量表所在的段地址=0
  - 存放子程序入口的单元的偏移地址= $n \times 4$
- 例：
  - 将中断向量码为48H的服务程序入口地址放入向量表





# 中断向量表的初始化

---

- MOV AX, 0000H
- MOV DS, AX
- MOV SI, 0120H
- MOV BX, OFFSET TIMER
- MOV [SI], BX
- MOV BX, SEG TIMER
- MOV [SI+2], BX



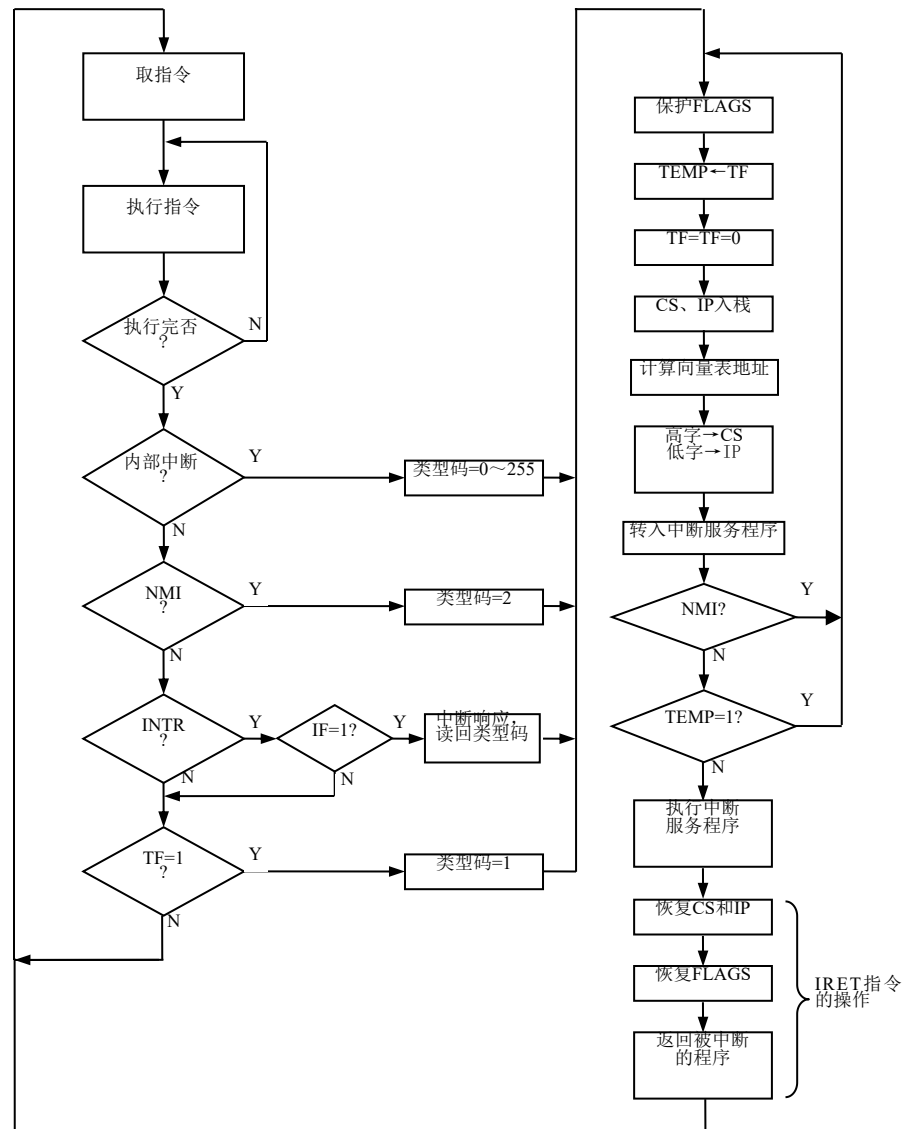
## 5. 8088内部中断与NMI中断

---

特点:

- 无 $\overline{\text{INTA}}$ 周期
- 中断类型码固定或由指令给出

# 8088/8086 中断响应和处理流程



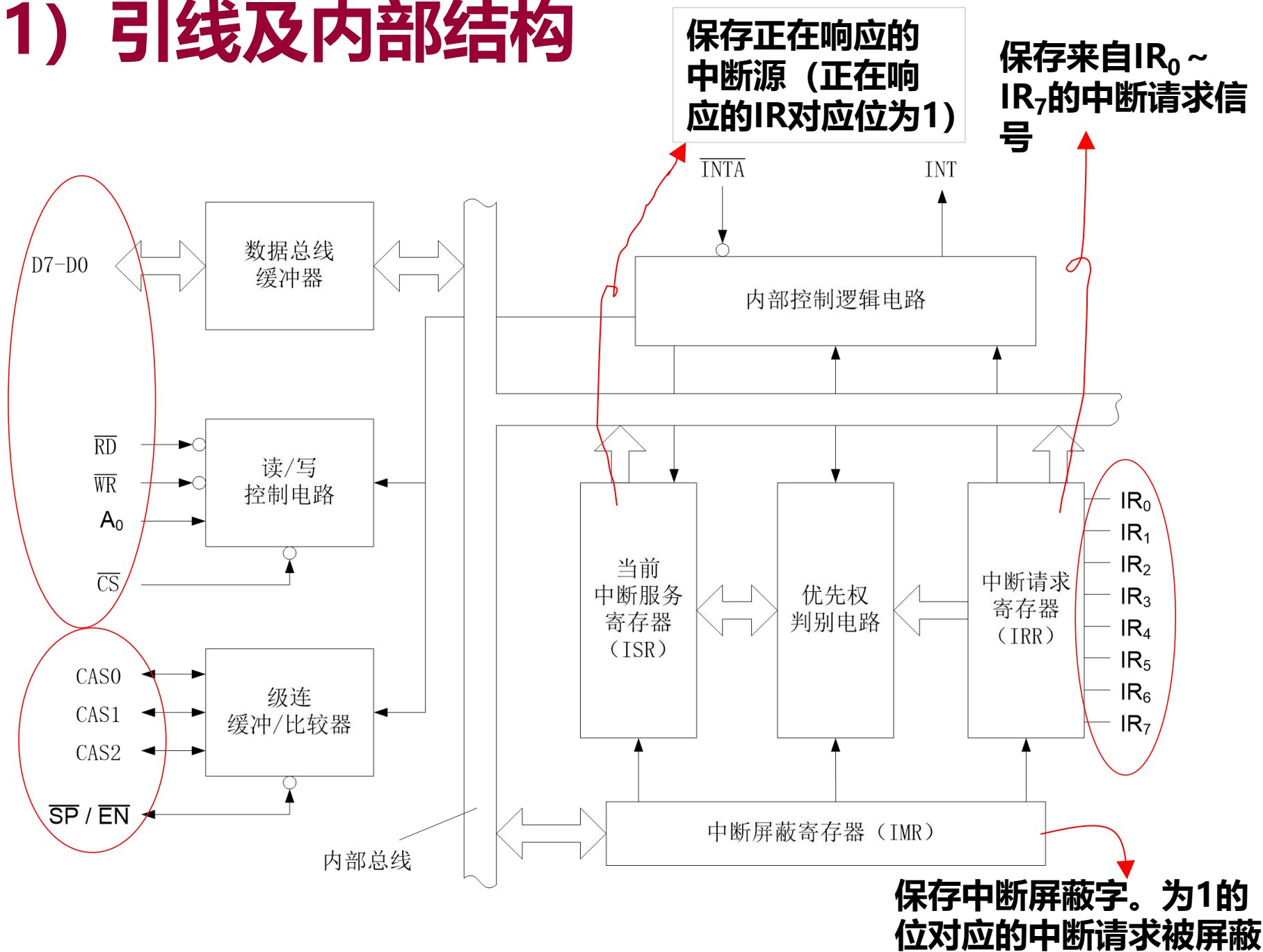


## 6. 中断控制器8259A

---

- 可编程中断控制器，用于可屏蔽中断管理。
- 单片8259A可管理8个外边中断源，通过级联，最多可以管理64个外部中断源。
- 了解：
  - 主要引线及内部结构
  - 工作过程
  - 工作方式

# 1) 引线及内部结构





# 1) 引线及内部结构

---

## ■ 中断请求寄存器IRR

- 保存8条外界中断请求信号 $IR_0 \sim IR_7$ 的请求状态
- $D_i$ 位为1表示 $IR_i$ 引脚有中断请求；为0表示无请求
- 中断处理后，及时撤销。

## ■ 中断服务寄存器ISR

- 保存正在被8259A服务着的中断状态
- $IS_i$ 为1表示 $IR_i$ 中断正在服务中；为0表示没有被服务
- 收到EOI命令相应位被清除

## ■ 中断屏蔽寄存器IMR

- 保存对中断请求信号IR的屏蔽状态
- $D_i$ 位为1表示 $IR_i$ 中断被屏蔽（禁止）；为0表示允许

## 2) 8259A的工作过程

(1) 初始化8259A

(2) 判断有无中断请求，如果有，则：

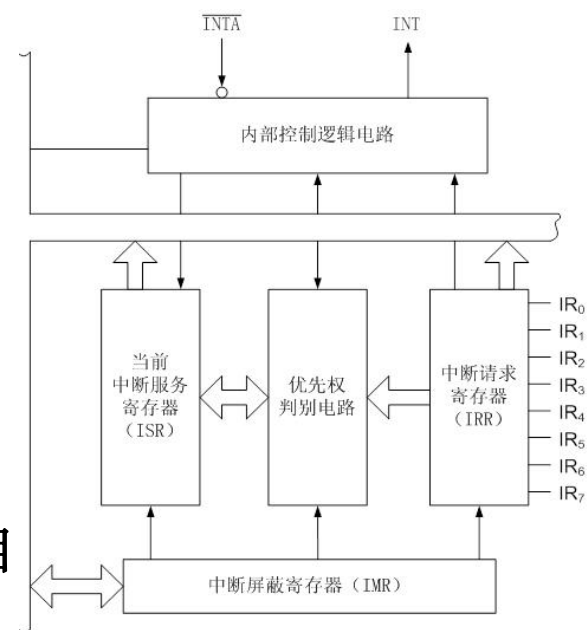
① 将有请求的 $IR_i$ 所对应的**中断请求寄存器**的相应位置1。

① 若相应中断未被屏蔽，则由INT引脚向CPU发出中断请求。

② 若CPU处于开中断状态，则当前指令执行完后，CPU发出#INTA响应信号。

③ 第一个#INTA脉冲使优先级最高的**中断服务寄存器ISR**的对应位置1，并使相应的**IRR**位复位。

④ 第二个#INTA周期用于获取选定的中断源所对应的中断类型码。CPU读取该中断类型码并乘以4，就可以从中断向量表中取出中断服务程序的入口地址并转去执行。





### 3) 8259A的工作方式

---

- ① 中断优先级
- ② 中断嵌套方式
- ③ 中断结束处理
- ④ 中断源屏蔽方式
- ⑤ 中断触发方式
- ⑥ 级联工作方式





# (1) 8259A的中断优先级设定

---

- 可以通过软件命令的方式设定各中断源的中断优先级
- 固定优先级
  - 默认IR<sub>0</sub>优先级最高，IR<sub>7</sub>优先级最低
  - 可通过程序改变。如：
    - **IR3→IR4→IR5→IR6→IR7→IR0→IR1→IR2**
- 循环优先级
  - 当某中断源被响应后，其优先级自动降为最低，原来比它低一级的中断则为最高级，依次排列。

## (2) 8259A的中断嵌套

### ■ 中断嵌套:

- 更高优先级的中断可以打断当前的中断处理过程

普通全嵌套方式



**响应某中断后，仅会响应更高优先级的中断请求。对同级或优先级更低的中断请求将被屏蔽**

特殊全嵌套方式



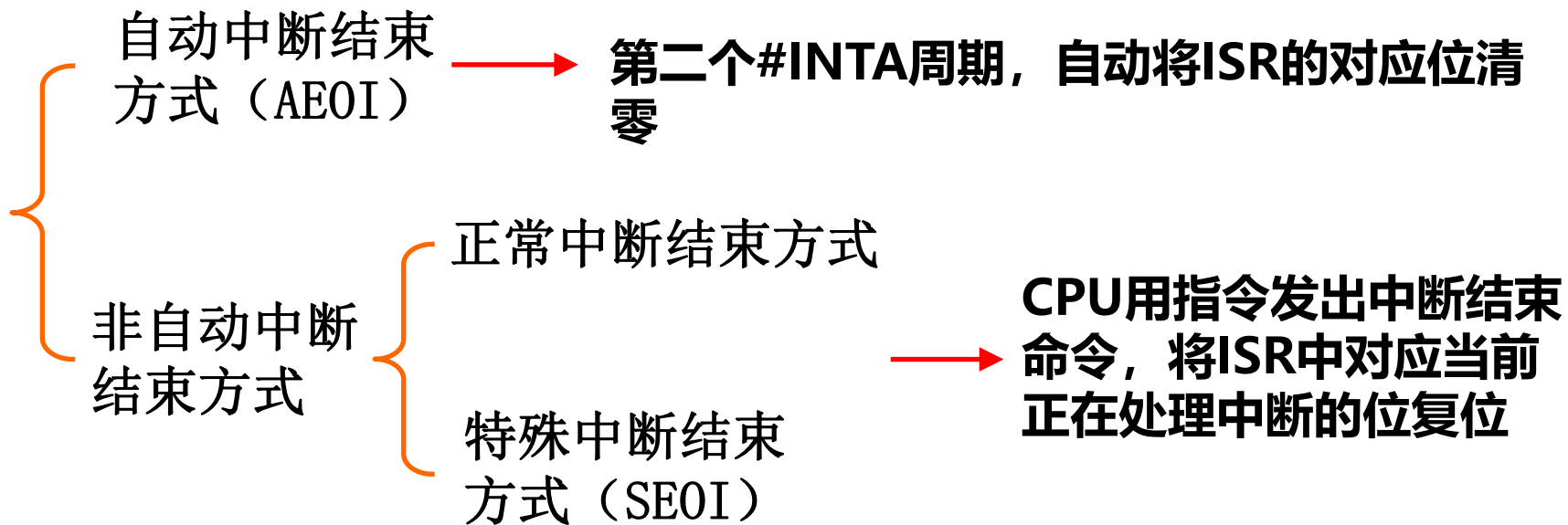
**响应某中断后，会响应与它同级或更高优先级的中断请求**

**一般用于级联情况下的主片的中断优先级设置**

## (3) 中断结束处理

### ■ 中断结束处理:

- 当中断服务程序结束时，将中断服务寄存器ISR中相应位（ $IS_i$ ）清零。





## (3) 中断结束处理

一次中断服务结束后，CPU可用中断结束命令EOI通知8259A，使ISR中的相应位复位(清零)。

### ➤ 自动EOI方式

在第二个中断响应周期  $\overline{\text{INTA}}$  信号的后沿将ISR中相应位清0，而不需要CPU发送EOI指令。

◆ 非自动EOI方式：要求在中断服务程序返回前，向8259发送一个EOI命令，才能将ISR中相应位清0。

➤ 正常EOI方式：8259收到EOI命令后，将ISR级别最高的置1位清0。

➤ 特殊EOI方式：中断服务程序向8259发送一个EOI命令的同时，将当前结束的中断级别也传送给8259，8259收到后，将ISR中指定级别相应位清0。

中断结束EOI与中断返回IRET的性质不同：

➤ EOI用于清除8259A的ISR中的中断服务标志，是中断的处理

➤ EOI命令发出后，紧接着就执行IRET指令，用于返回主程序的断点，是中断服务程序的最后一句



## (4) 中断源屏蔽方式

- 通过编程使得中断屏蔽寄存器IMR相应位置0或置1，从而允许或禁止该位所对应的中断

普通屏蔽方式



若将中断屏蔽寄存器某位置1，  
则屏蔽该位所对应的中断请求

特殊屏蔽方式（SMM）

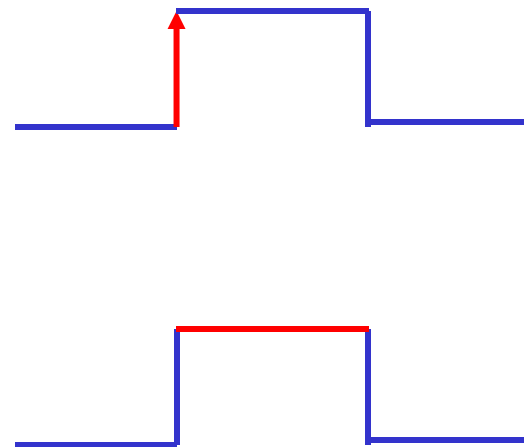


允许较低优先级的中断请求  
中断更高优先级的中断

## (5) 中断触发方式

- IR引脚的中断触发方式：
  - 边沿触发
  - 电平触发

中断请求信号IR都应维持到第一个#INTA 信号结束之前





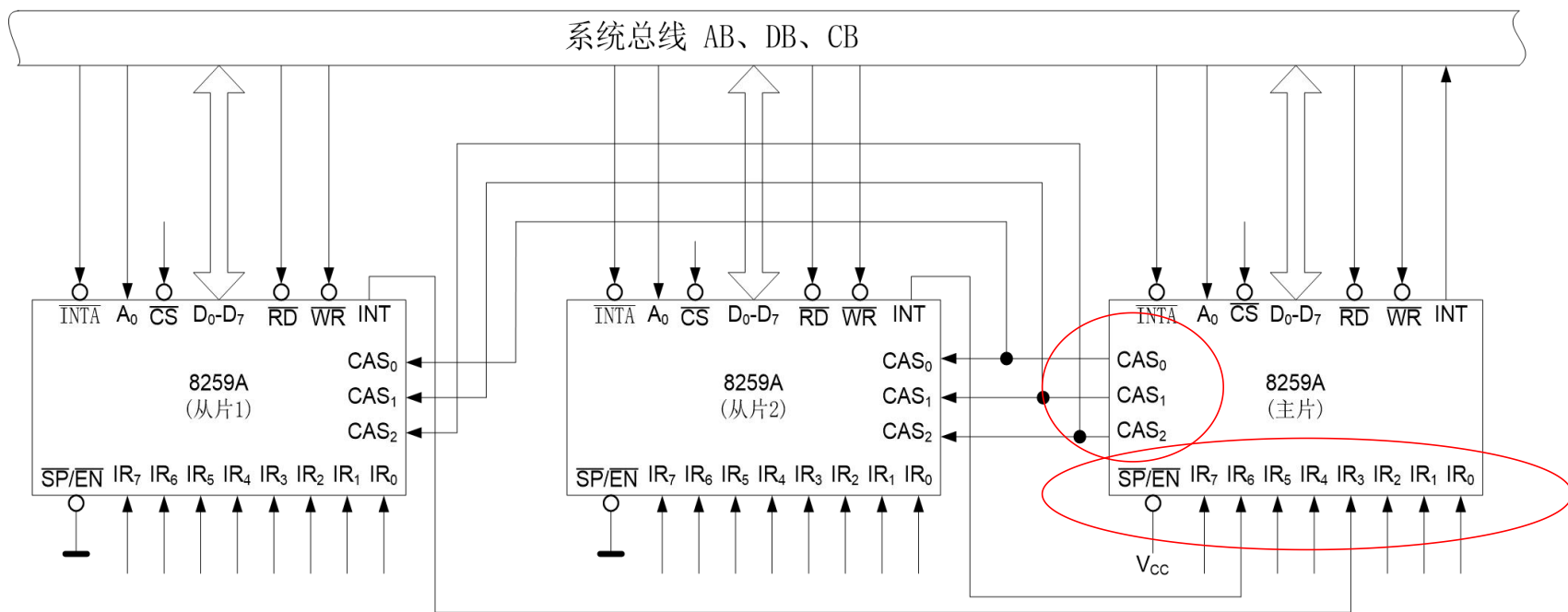
## (6) 8259A的级联

---

- 一个系统中，8259A可以级联，有一个主8259A，若干个（最多8个）从8259A
- 级联时，主8259A的三条级连线CAS0~CAS2作为输出线，连至每个从8259A的CAS0~CAS2
- 每个从8259A的中断请求信号INT，连至主8259A的一个中断请求输入端IR
- 主8259A的INT线连至CPU的中断请求输入端
- SP\*/EN\*在非缓冲方式下，规定该8259A是主片（SP\*=1）还是从片（SP\*=0）

## (6) 8259A的级联

- 设置一片为主片，将主片的IR端接入到从片的INT段。实现最多64个中断源的中断控制。







## (6) 数据线连接方式

### ■ 缓冲方式

- 在多片8259A级连的大系统中，8259A通过外部总线驱动器和数据总线相连，这就是缓冲方式。
- 8259A把SP\*/EN\*引脚作为输出端，输出允许信号，用以锁存或开启缓冲器

### ■ 非缓冲方式

- 当系统中只有一片或几片8259A芯片时，可以将数据总线直接与系统数据总线相连
- SP\*/EN\*引脚为输入端
- 若8259A级联，由其确定是主片或从片



中断型码为21H的中断，中断向量存放的地址\_\_\_\_\_。

- ☐ A 0000:0021H
- ☐ B 0000:0042H
- ☒ C 0000:0084H
- ☐ D 0000:0108H

提交



## 3) 8259A的初始化

---

- 8259A的初始化：
  - 通过软件向其写入控制命令的方法控制其工作状态。
- 包括：
  - 写入初始化命令字ICW
    - 向8259A送2~4个字节的初始化命令字ICW，写入各ICW寄存器，使其处于准备就绪状态。
  - 写入操作命令字OCW
    - 向8259A送3个字节的操作命令字OCW，以规定8259A的操作方式（中断控制方式、屏蔽某些中断源、读出8259A工作状态信息）。



# 中断程序设计的一般过程

---

## ① 确定中断类型号

- 应用程序可使用的中断类型号为60H ~ 66H和68H ~ 6FH

## ② 保存原中断向量

- 在将自己的中断程序入口地址置入中断向量表之前，应先保存该地址中原来的内容。

## ③ 将自己的中断向量放入向量表；

## ④ 设置中断屏蔽字（可选）。若编写的是硬件中断程序，应将所使用的硬件中断对应的8259A的中断屏蔽位开放；

## ⑤ CPU开中断；

## ⑥ 恢复原中断向量。

## 8259A的初始化编程



### □ 8259A的两种命令字

- ◆ 初始化命令字: ICW1~ICW4
- ◆ 操作命令字: OCW1~OCW3

□ 8259A的启动：必须先通过编程写入初始化命令字，使它处于工作起始状态；

□ 初始化过程：按照固定的顺序进行。**ICW1、ICW2必须写**，**ICW3、ICW4**视具体情况而定。



## 初始化流程



用 $ICW_1$ 设置，是否级联，  
后面是否用 $ICW_4$

用 $ICW_2$ 设置中断类型码

是否为级连方  
式？

否

是

本片为主片吗？

是

设 $ICW_3$ 的各位对应  
 $IR_0 \sim IR_7$ 的连接情况

否

设置 $ICW_3$ 的低三位为从控标志码

需要用 $ICW_4$   
吗？

否

是

用 $ICW_4$ 设置，是否为特殊完全嵌套方式；缓冲  
方式；自动结束中断方式；8086/8088系统

OCWn

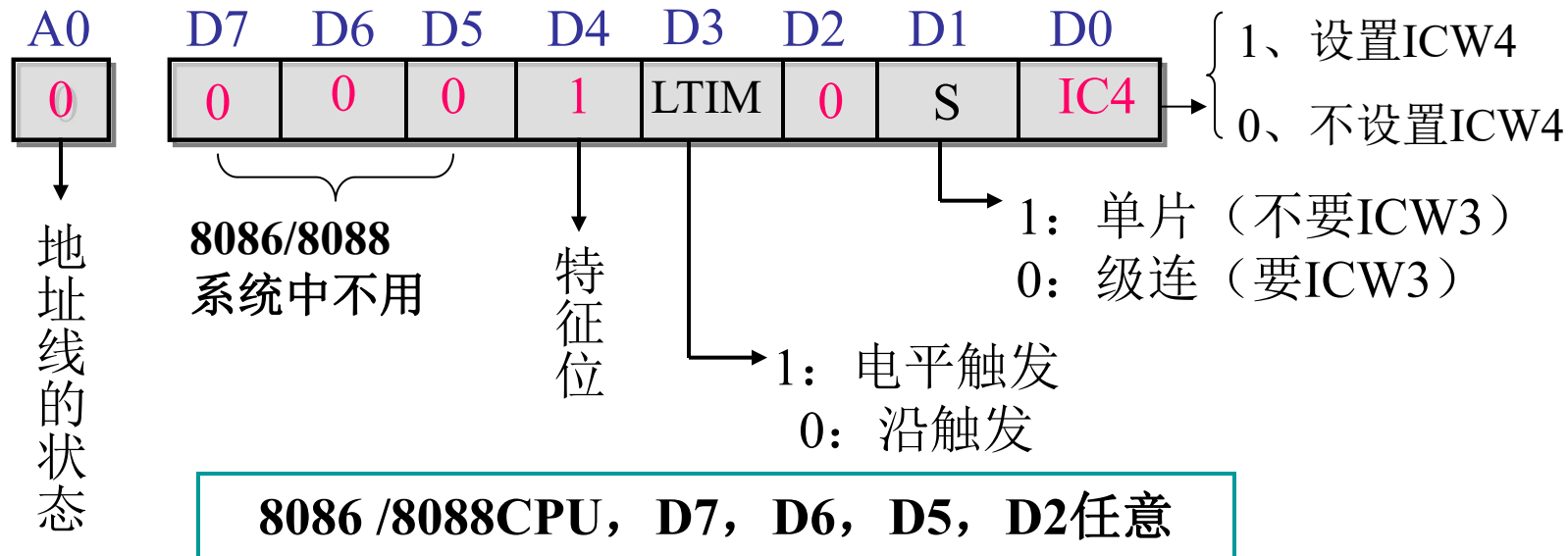
Back

Next



# 1、预置命令字

## (1) ICW1



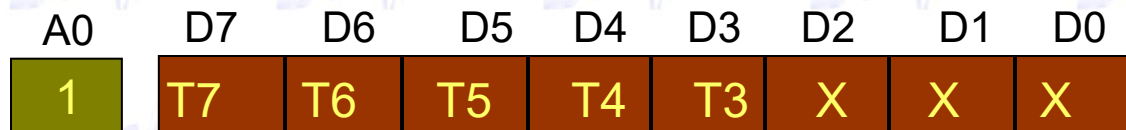
当CPU向8259A送入一条使A0=0、D4=1的命令时，该命令被译码为ICW1，它启动8259A的初始化过程，自动完成下列操作：

- 1) 清除ISR和IMR;
- 2) 设置以IR0优先级最高，IR7优先级最低的完成嵌套方式，固定优先排序;
- 3) 设定普通屏蔽方式; 4) 非自动EOI中断结束方式;
- 5) 设置读IRR方式

## 初始化命令字 ICW2

8259A提供给CPU的中断类型号是一个8位代码，是通过初始化命令ICW2提供的。但由于ICW2的低3位被8位机占用，因此只有高5位是在初始化编程时，通过命令字ICW2写入的，它的低3位是由中断请求线IRi的二进制编码（如IR4的编码为100）决定，并且是在第一个INTA到来时，将这个编码写入低3位的。ICW2的格式如下：

ICW2



中断类型码高5位

自动填入



## 初始化命令字 ICW2



- ◇ 例如，在PC系列中断系统中，硬盘中断类型号的高5位是**08H**，它的中断请求线连到8259A的IR5上，在向ICW2写入中断类型号时，只写中断类型号的高5位（**08H**），低3位可以取0：
- ◇ `MOV AL, 08H ; ICW2的内容（中断类型号高5位）`  
`OUT 21H, AL; 写入ICW2的端口（A0=1）`
- ◇ 当CPU响应硬盘中断请求时，8259A把**IR5**的编码**101**作为低3位构成一个完整的8位中断类型号**0DH**，经数据总线发送给CPU。
- ◇ 可见，外部硬中断中断源的中断号（8位代码）是由两部分构成的，即**高5位**（ICW2）和**低3位**（IRi的编码）。



## ICW3: 设置主8259和从8259的联结关系

主8259的ICW3: 指出主8259的哪个引脚上联有从8259

D7	D6	D5	D4	D3	D2	D1	D0
IR7	IR6	IR5	IR4	IR3	IR2	IR1	IR0

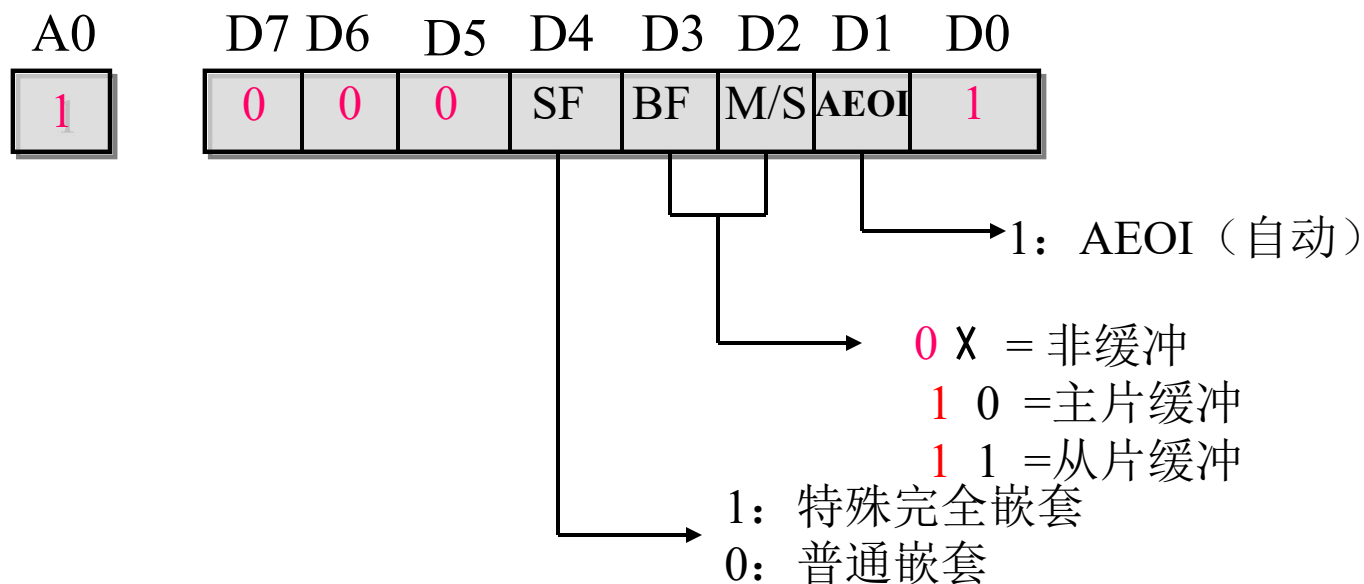
如: 主8259的IR2上联从8259, 则主8259的ICW3的D2=1

从8259的ICW3:

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	ID2	ID1	ID0

ID2~ID0编码值, 指出从8259的INT联至主8259的哪个引脚  
如联至IR5, 则ID2~ID0=101

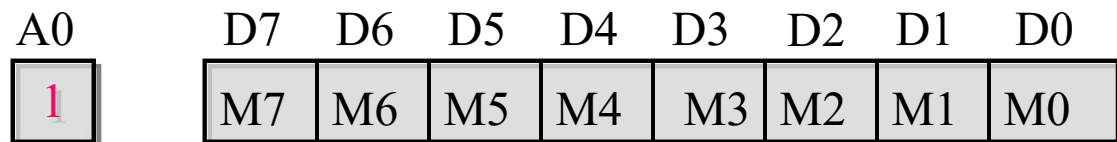
#### 4、ICW4（对8086/8088系统是必须设置的预置命令字）



- 注意：**
- (1) ICW1 ~ ICW4必需顺序写入，即使改变一个参数也需全部重新写入ICW
  - (2) 单片只写ICW1，ICW2，ICW4，不要ICW3  
级连ICW1 ~ ICW4全要，但主、从片的ICW3不同
  - (3) ICW1的A0 = 0，其它ICW的A0 = 1

## 2、操作命令字

### (1) OCW1 (实现中断屏蔽功能)



$M_i = 1$  禁止中断（屏蔽）， $M_i = 0$  允许中断

**例** 已知 IBM PC/XT 系统中  
8259A 的奇地址端口地址为 21H

#### ● 新增允许 IR2 的中断申请

IN AL, 21H ;读入原IMR的内容

AND AL, 1111 1011B ;D2=0,允许IR2的中断申请

OUT 21H, AL ;写入IMR

#### ● 禁止 IR4 的中断申请

IN AL, 21H ;读入原IMR的内容

OR AL, 0001 0000B ;D4=1,禁止IR4的中断申请

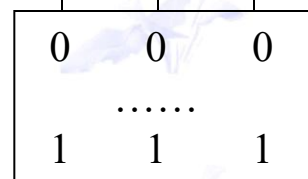
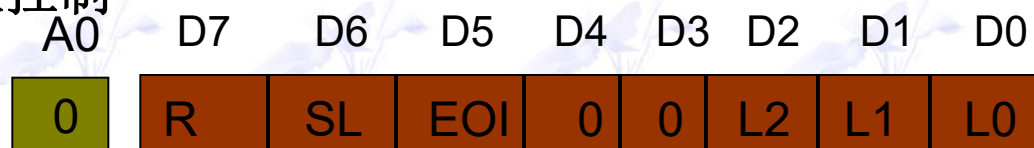
OUT 21H, AL ;写入IMR

# OCW<sub>2</sub>中断结束和优先级循环控制

◇ 中断结束方式和中断排队方式都用OCW2命令来实现，因此它的某些位功能重叠相互交叉显得比较复杂，但从使用的角度来看，它只有两个作用：

- ◆ 中断结束方式控制
- ◆ 中断优先权排队控制

OCW2



对应IRi  
的编码

R=1: 循环优先级

R=0: 固定优先级

SL=1: 由L2 L1 L0指定

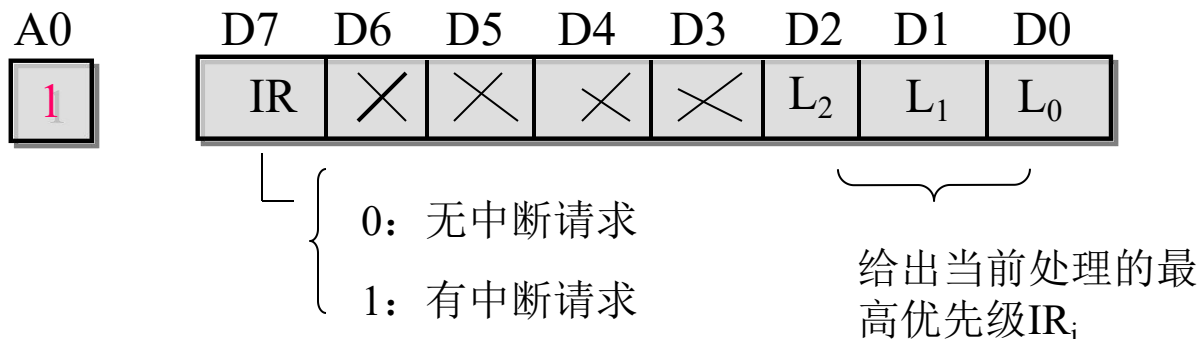
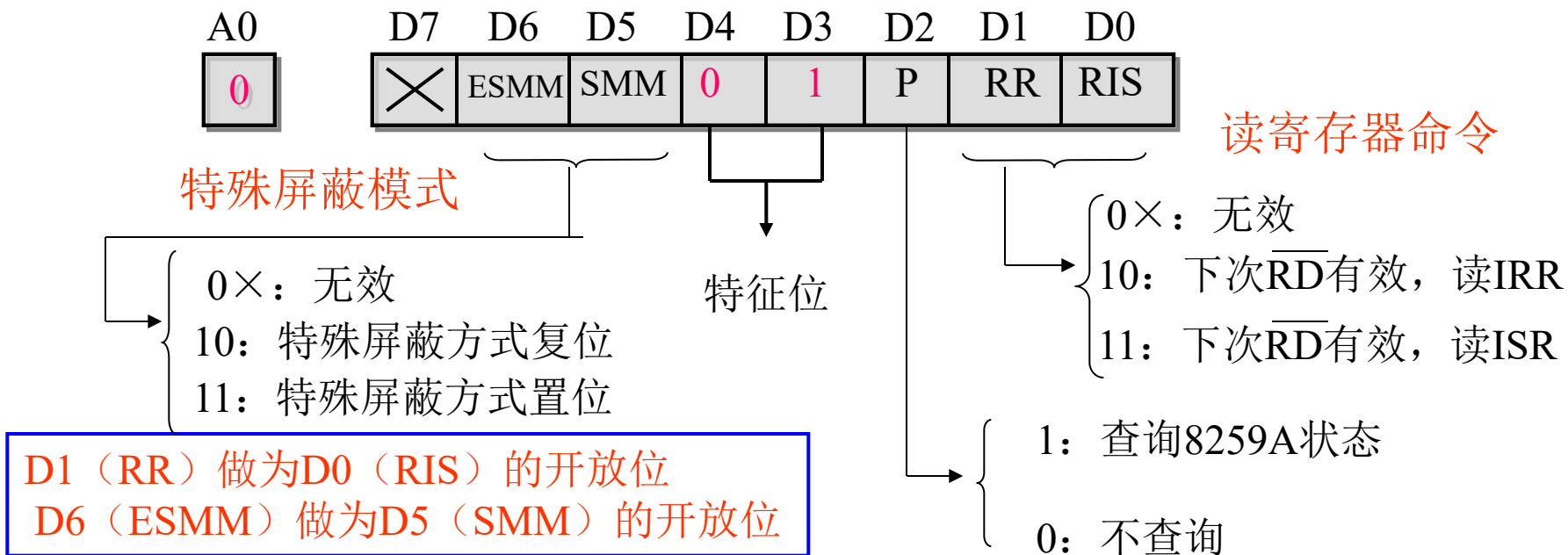
IR位

SL=0: L2 L1 L0无效

EOI=1: 中断结束

0	0	1	一般EOI,全嵌套方式
0	1	1	特殊EOI,全嵌套方式,由L2~L0指定结束IRi
1	0	1	一般EOI,优先级自动循环
1	0	0	自动EOI,优先级自动循环
0	0	0	自动EOI,取消自动优先级循环(固定优先级)
1	1	1	一般EOI,特殊优先级循环,L2~L0指定结束
1	1	0	特殊优先级循环,由L2~L0设定最低优先级
0	1	0	无操作

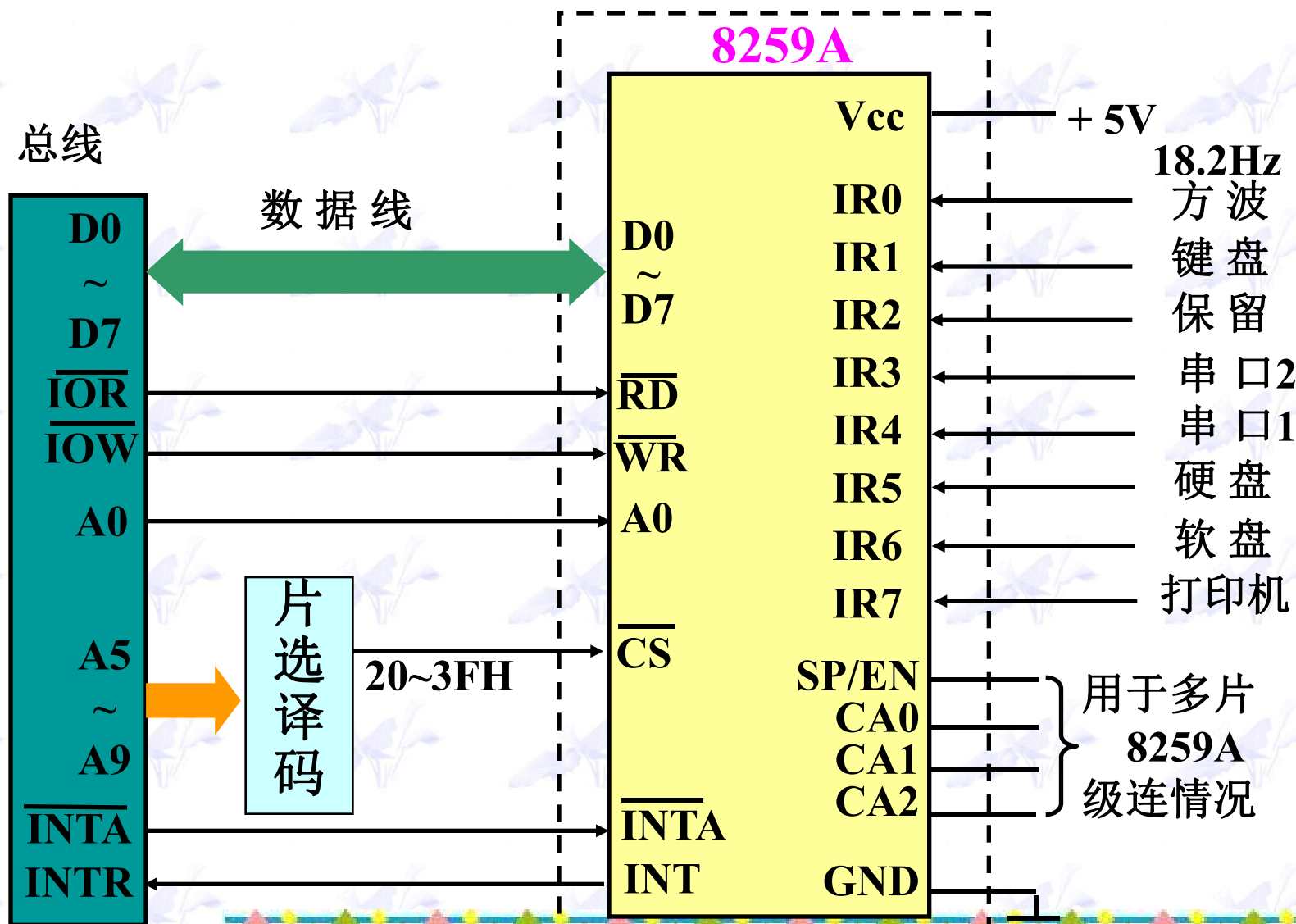
### (3) OCW3 (特殊屏蔽设置, 中断查询, 读IRR, ISR)





# ① 8259A的片选地址范围为 20 ~ 3F H

偶、奇端口地址通常取用20H、21H

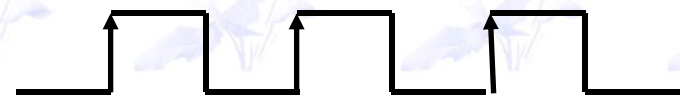


## ② 8259A的8个中断申请与外设的连接

◇ **IR0** 接一个频率为**18.2Hz**的方波；

◇ 若**CPU**对该申请给予响应，

◇ 则每秒执行**18.2**次该申请对应的中断子程。



**18.2Hz**时钟信号

◇ **IR1**接键盘接口发出的中断申请；

◇ 每按下一键，若**CPU**对该申请给予响应，

◇ 则执行一次该申请对应的中断子程。

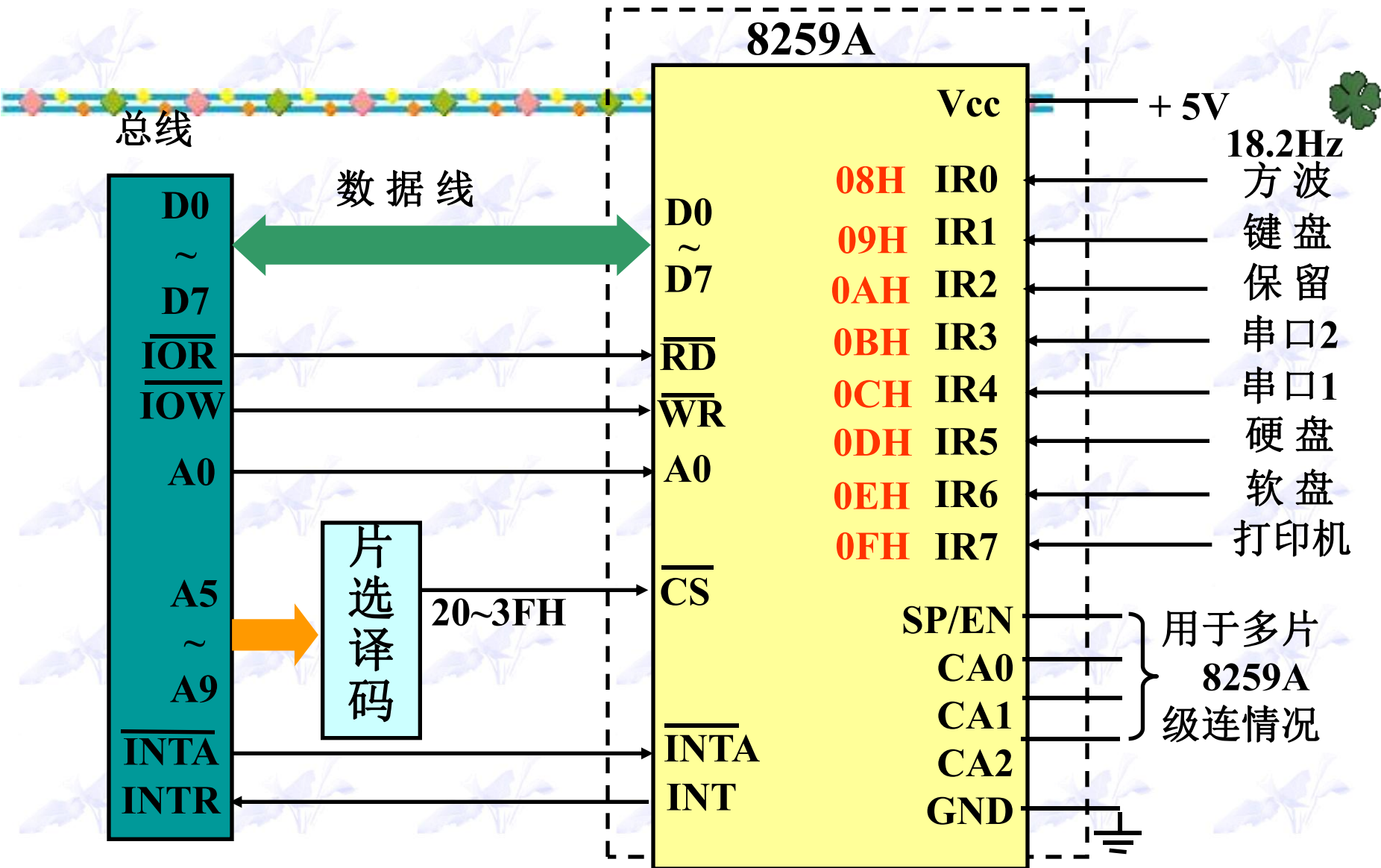
◇ **IR2~IR7**通过系统总线引出，

◇ 对应引脚信号为**IRQ2~IRQ7**。











### 例:单片使用8259A的初始化

(1)要求:IBM PC/XT 8259A ,单片,边缘触发,全嵌套方式,设定0级中断类型码为08H。

端口地址 :20H,21H。

(2)硬件连接:CAS2~CAS0不用,SP/EN接+5V。

(3)初始化命令字:

ICW1=000**1** **00****1**1B ; 边缘触发,单片,需ICW4

ICW2=0000 1000B ;设置类型码的高5位

ICW4=0000 **00****0****1**B ;全嵌套,非缓冲,非自动EOI,16位机



## 8259A初始化编程:



....

MOV AL, 13H

;ICW1:边沿触发.单片.要ICW4

OUT 20H,AL

MOV AL, 08H

;ICW2:IRQ0中断类型为08H

OUT 21H,AL

MOV AL, 01H

;ICW4全嵌套.8086系统

OUT 21H,AL

;普通EOI方式



## (2) 设计

### ◇ ① 8259A初始化编程

；主片8259A的初始化

```
MOV AL, 11H
OUT 20H, AL
JMP INTR1
INTR1: MOV AL, 08H
OUT 21H, AL
JMP INTR2
INTR2: MOV AL, 04H
OUT 21H, AL
JMP INTR3
INTR3: MOV AL, 11H
OUT 21H, AL
      ⋮
```

；从片8259A的初始化

```
MOV AL, 11H
OUT 0A0H, AL
JMP INTR5
INTR5: MOV AL, 70H
      为70H
OUT 0A1H, AL
JMP INTR6
INTR6: MOV AL, 02H
OUT 0A1H, AL
JMP INTR7
INTR7: MOV AL, 01H
OUT 0A1H, AL
      ⋮
```

；写入ICW1，设定边沿触发，级联方式

；延时，等待8259A操作结束，下同  
；写入ICW2，设定IRQ0的中断类型号为08H

；写入ICW3，设定主片IRQ2级联方式

；写入ICW4，设定特殊全嵌套方式，一般EOI方式

；写入ICW1，设定边沿触发，级联方式

；写入ICW2，设定从片IR0，即IRQ8的中断类型号

；写入ICW3，设定从片级联到主片的IRQ2

；写入ICW4，设定普通全嵌套方式，一般EOI方式





## ② 级联工作编程



；读ISR的内容

```
MOV AL, 0FH
OUT 0A0H, AL
NOP
IN AL, 0A0H
⋮
```

；写入OCW3，读ISR命令

；延时，等待8259A操作结束  
；读出ISR

；向从片发EOI命令

```
MOV AL, 20H
OUT 0A0H, AL
⋮
```

；写从片EOI命令(OCW2)

；向主片发EOI命令

```
MOV AL, 20H
OUT 20H, AL
⋮
```

；写主片EOI命令(OCW2)





- ◇ 保护现场能力，以防中断程序出故障。
- ◇ 在设置中断向量时，首先关中断，在初始化后再开中断。
- ◇ 在中断程序的入口处要立即开中断，并在程序结束之前予以恢复。
- ◇ 硬件中断程序执行IRET指令前，应向8259A发出中断结束命令EOI。
- ◇ 硬件中断服务程序不要使用DOS系统功能调用即INT 21H，服务程序若要控制I/O设备，最好调用ROM-BIOS功能或对I/O直接编程。







# 8259A芯片的初始化流程

