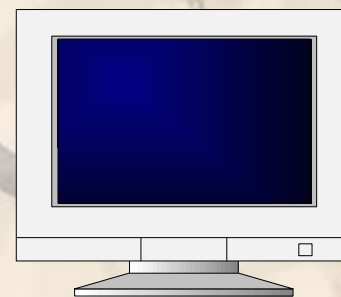


第6章

输入输出及中断技术





主要内容

- 基本概念
 - 输入输出系统
 - **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
- 中断控制器**8259A**介绍
- **DMA**控制器**8237**介绍



§ 6.1 输入输出系统



了解和掌握：

- **I/O**系统组成及主要特点
- 接口的基本功能
- 端口的概念
- 端口的编址方式
- **I/O**地址译码



一、输入输出系统

- 输入输出系统:

- 计算机系统中除CPU和内存储器之外的部分

I/O系统 { 输入输出设备
输入输出接口
输入输出软件



输入输出系统特点

■ 复杂性

- 输入输出设备、处理器、操作系统的复杂性

■ 异步性

- 工作速度和时序不一致

■ 实时性

- 控制的时效性。**I/O**系统保证处理器对不同设备的请求提供及时服务。

■ 与设备无关性

- 接口的标准化，由操作系统屏蔽了设备的差异。



二、I/O接口系统及接口

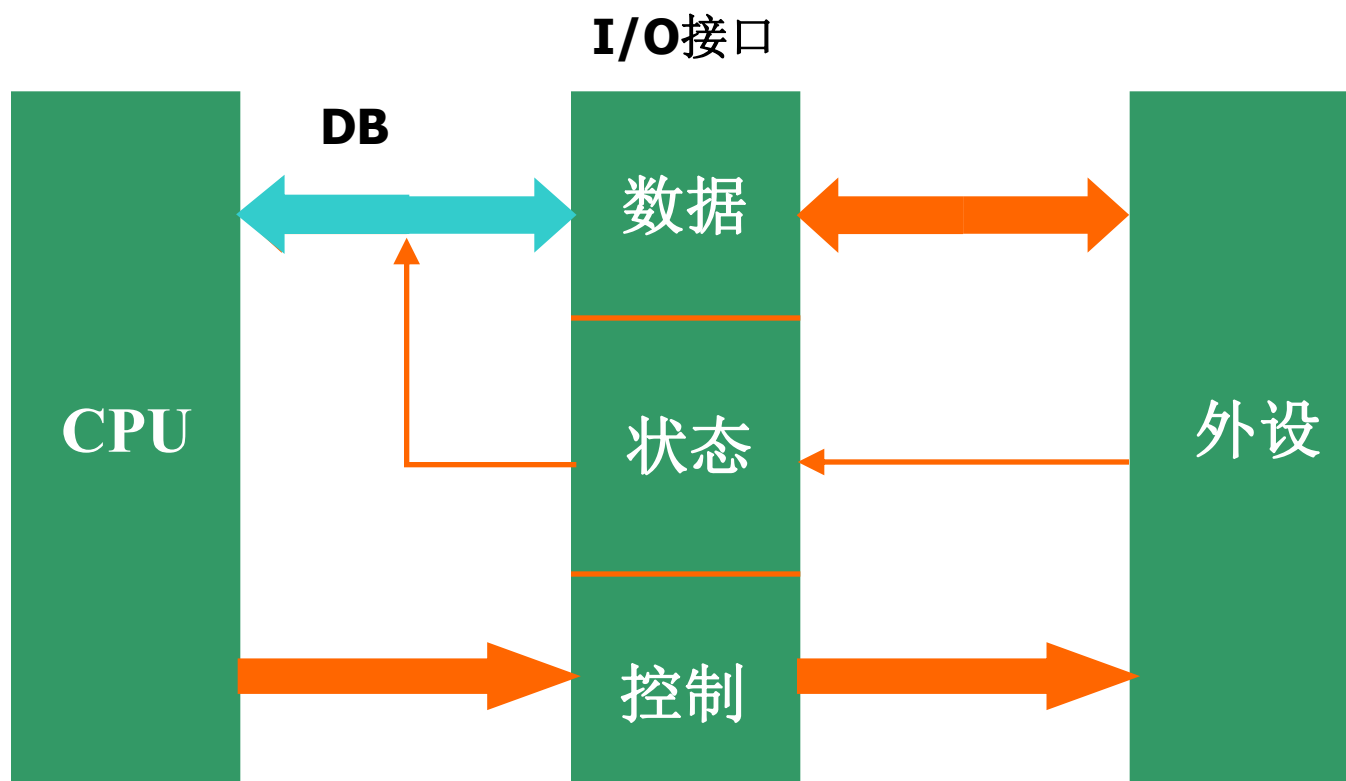
- 有关**I/O**系统的概念、特点，以及**I/O**接口的功能等，请参阅教材描述，自行学习。
- 总体上，**I/O**接口应具备以下功能：
 - 数据的缓冲与暂存
 - 信号电平与类型的转换
 - 增加信号的驱动能力
 - 对外设进行监测、控制与管理，中断处理



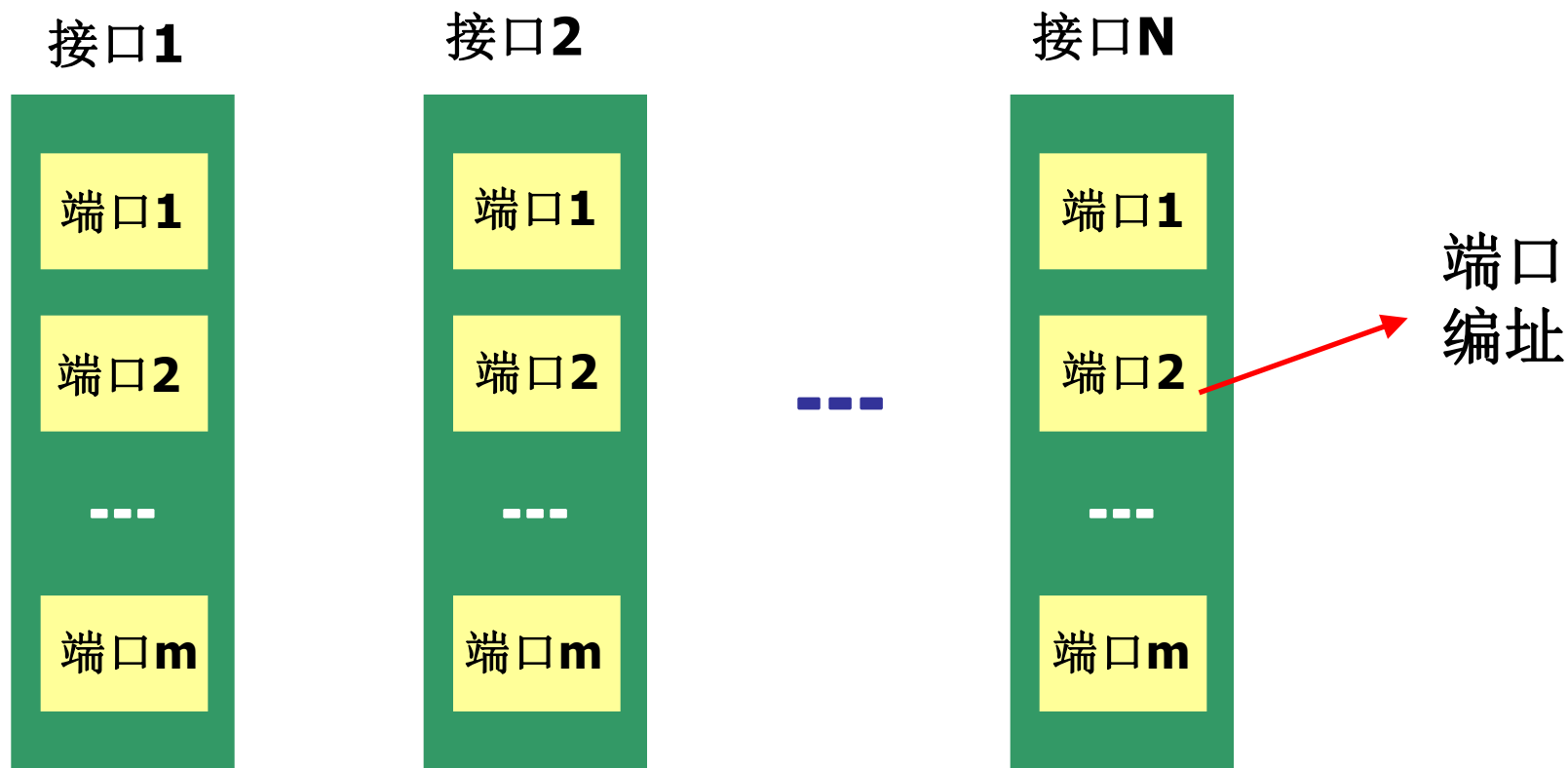
三、I/O端口

- 端口
 - 接口中的寄存器
- 端口的主要作用
 - 信息的缓存
- 端口类型
 - 数据端口
 - 缓存输入和输出的数据
 - 状态端口
 - 缓存需要输入的外设工作状态
 - 控制端口
 - 缓存由系统输出的各种控制信息

I/O端口



I/O系统中的接口和端口



端口地址 = 芯片地址（高位地址） + 片内地址



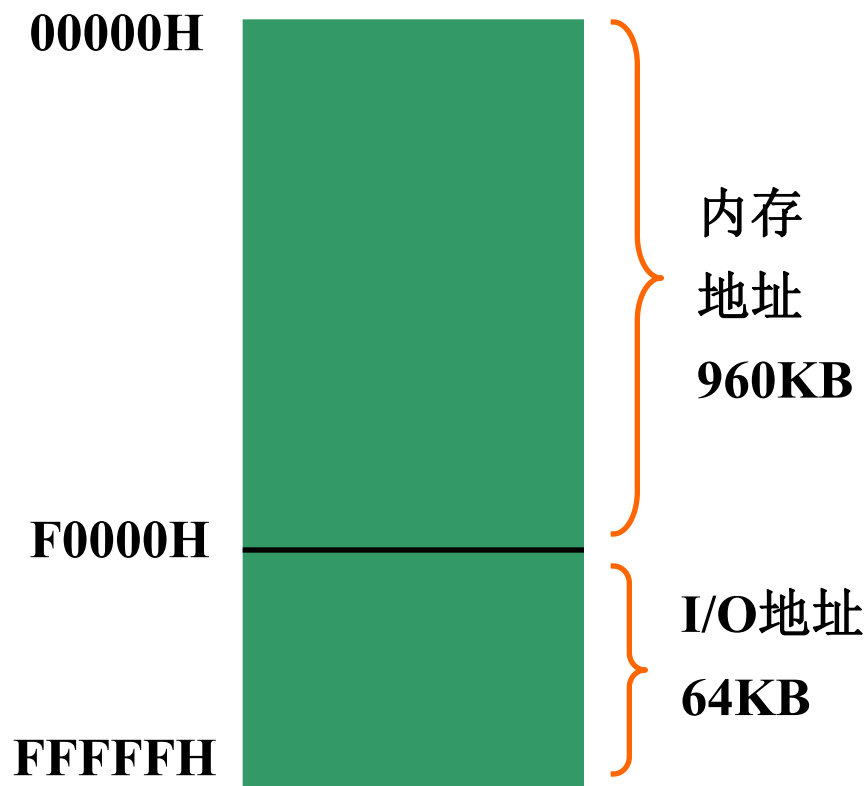
四、I/O端口的编址方式

- 编址方式:
 - 与内存统一编址
 - 独立编址

1. 端口与内存的统一编址

特点:

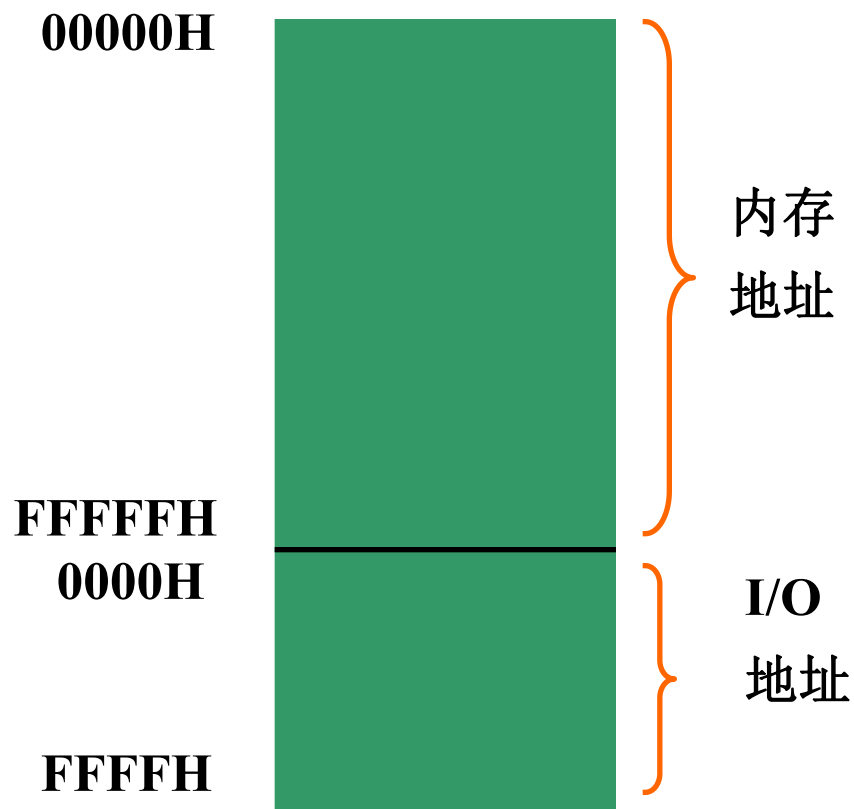
- 指令及控制信号统一
- 内存地址资源减少



2. 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/ \overline{M}** 来区分
- **I/O**操作只使用**20**根地址线中的**16根**: **A₁₅~A₀**
- 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**)



五、I/O地址译码

- 目的
 - 确定端口的地址
- 寻址端口的信号
 - **IOR**、**IOW**
 - **A15 ~ A0**
- 参加译码的信号：
 - **IOR**, **IOW**, 高位地址信号

IN指令将使总线中的该信号有效

OUT指令将使总线中的该信号有效



I/O地址译码

- 一个接口电路中可以有多个端口，也可以只有一个端口。
- 对全地址译码：
 - 当接口只有一个端口时， **16**位地址线应全部参与译码，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），低位用于寻址接口中要访问的端口。
- 对部分地址译码：
 - 仅用部分地址信号参与译码
 - 对含多个端口的接口，最低的几位直接连到接口。



I/O地址译码

I/O系统中，因地址资源丰富，多采用部分地址译码。

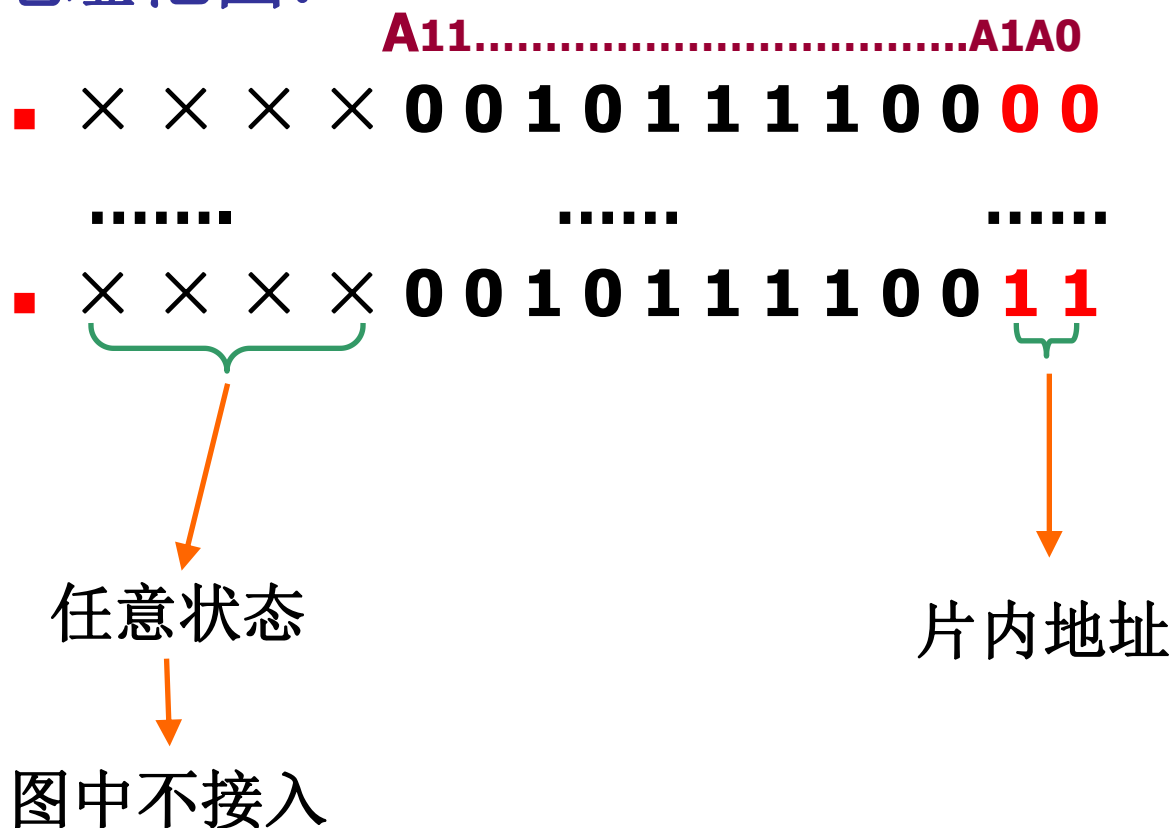


I/O地址译码例

- 例：某外设接口有**4**个端口，地址为**2F0H—2F3H**，由**A₁₅~A₂**译码得到，而**A₁、A₀**用来区分接口中的**4**个端口。试画出该接口与系统的连接图。
- 题目分析：
 - 寻址端口的地址信号最多为**16bit**，题中仅用**12bit**就能表示其地址——故采用部分地址译码
 - 该接口电路中含有**4**个端口，片内端口寻址需**2**位地址信号，其余**10**位为片选地址信号。

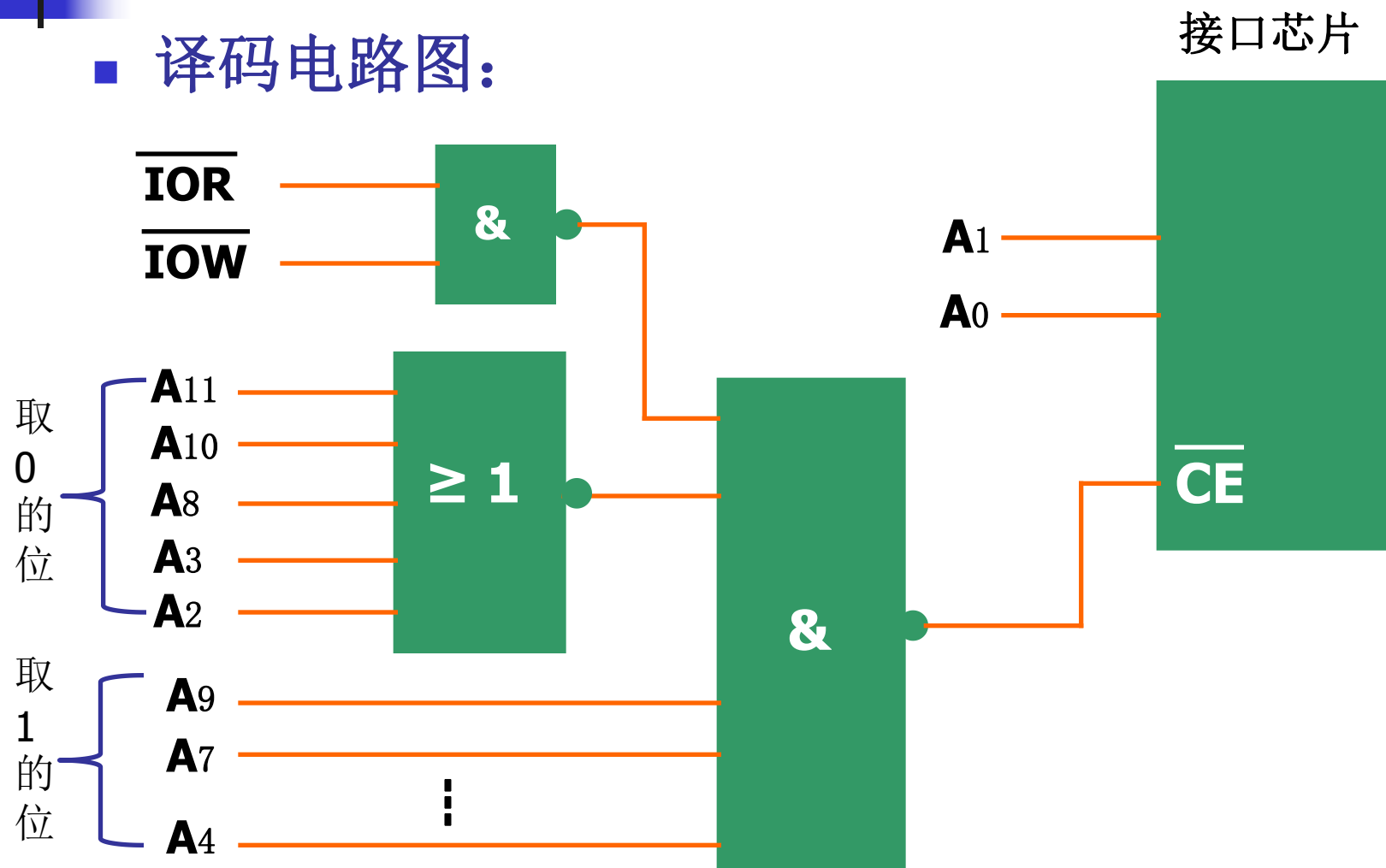
I/O地址译码例

■ 地址范围:

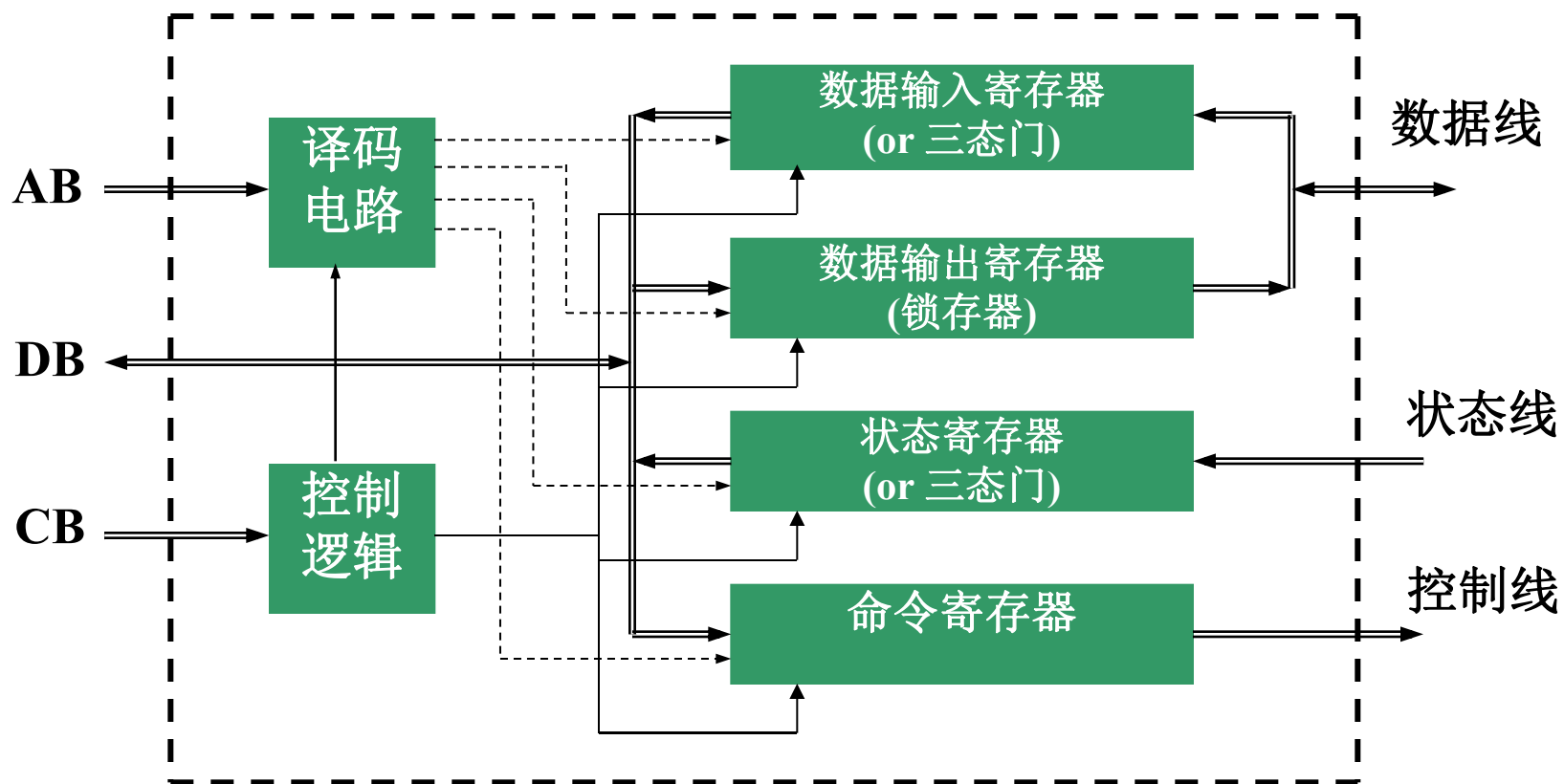


I/O地址译码例

译码电路图：



六、接口的基本构成





接口的基本构成

- 数据输入/输出寄存器
 - 暂存输入/输出的数据
- 命令寄存器
 - 存放控制命令
 - 设定接口功能、工作参数和工作方式。
- 状态寄存器
 - 保存外设当前状态，以供**CPU**读取。



七、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（允许数据送到数据线）
- 常用三态门实现

■ 输出接口：

- 要求对数据具有锁存能力（接收后保持数据不变）
- 常用锁存器实现



§ 6.2 简单接口电路



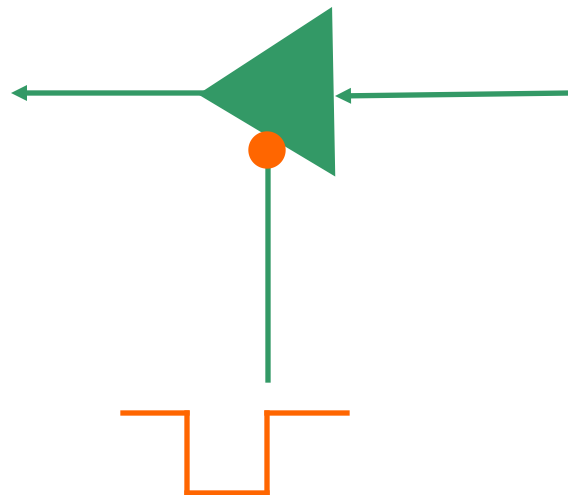
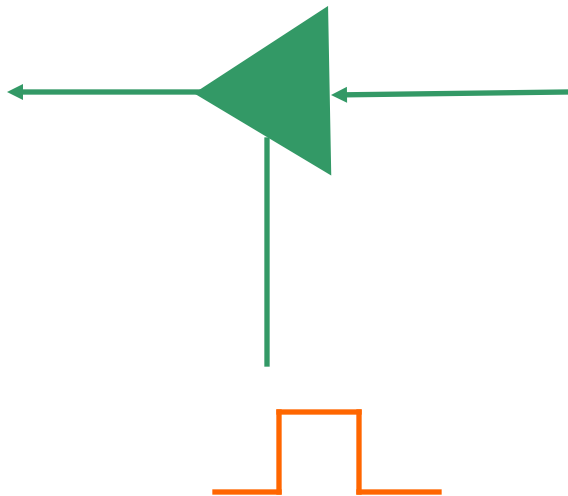
掌握：

- 两类简单接口芯片的应用



一、三态门接口

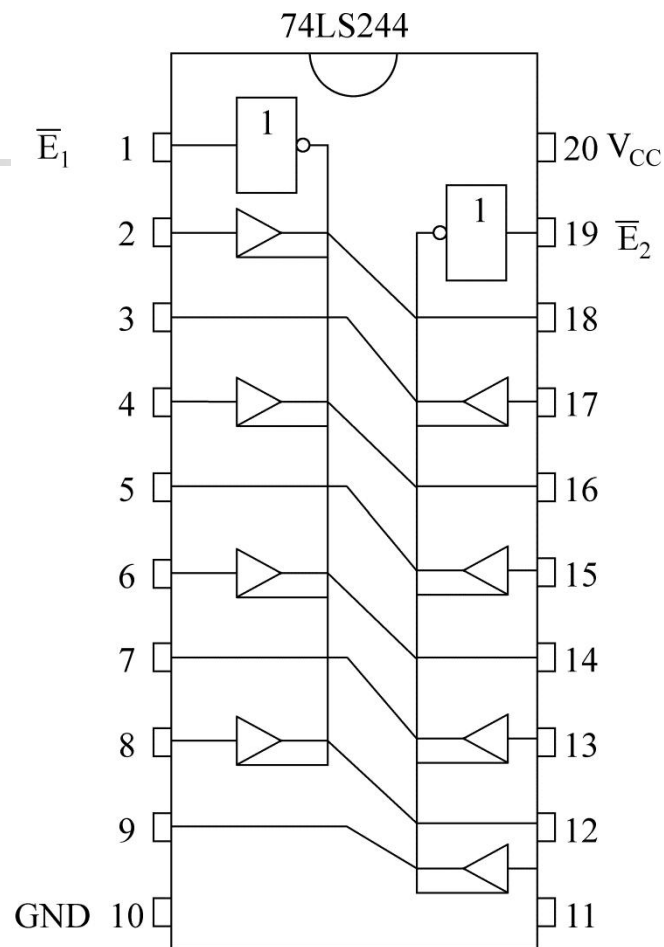
- 高电平、低电平、高阻态



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数据
- **74LS244**应用例
 - 教材**p239**

P239图





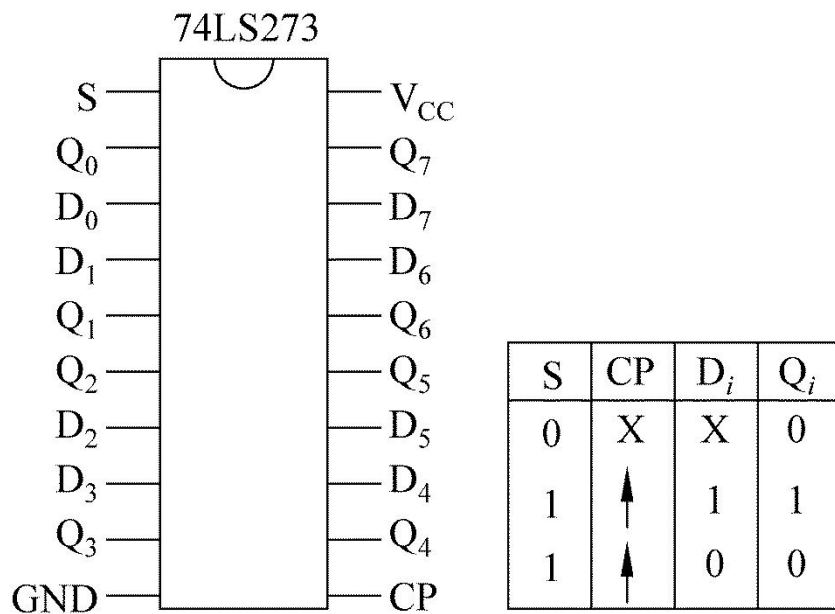
二、锁存器接口

- 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

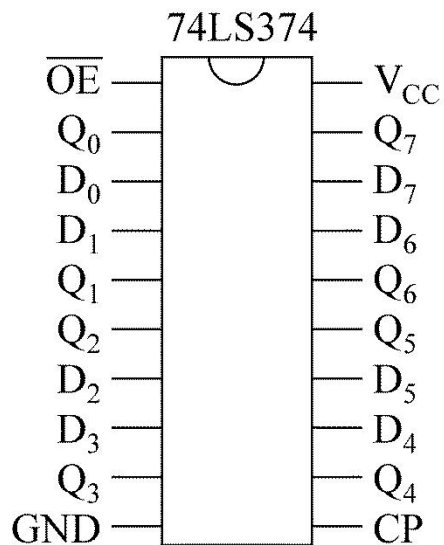
■ 74LS273

- 8D触发器
- 不具备数据的控制能力

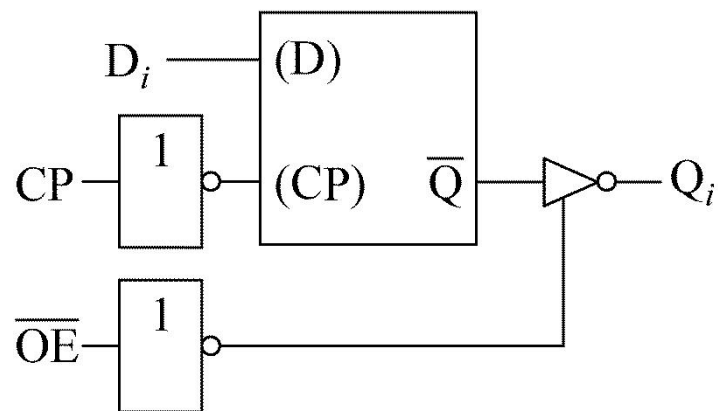


■ 74LS373和74LS374

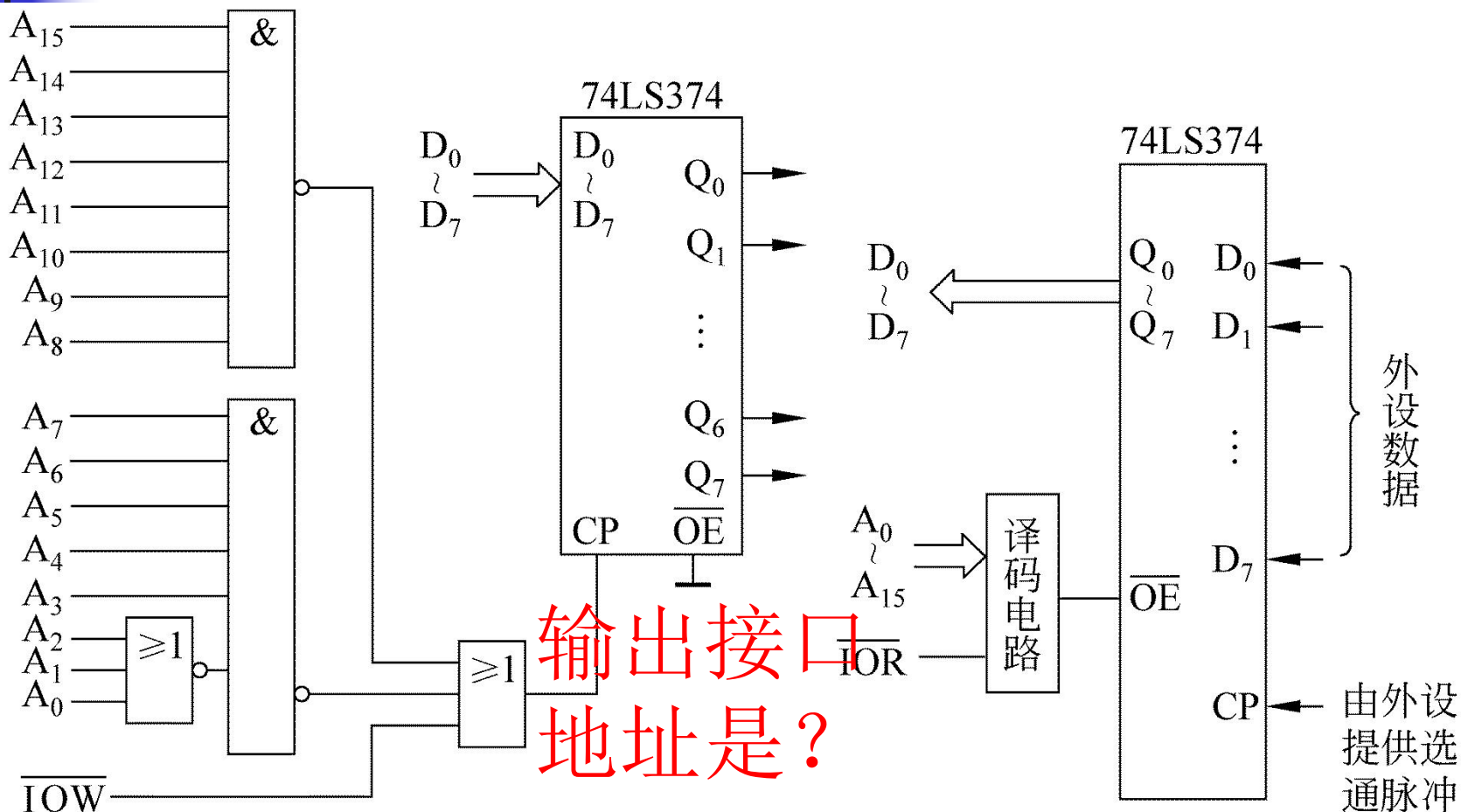
- 含三态的**8D**触发器，并具有对数据的控制能力。
- 既可以做输入接口，也可以做输出接口。



D_i	CP	\overline{OE}	Q_i
1	\uparrow	0	1
0	\uparrow	0	0
X	X	1	高阻



74LS374作输入输出接口示例



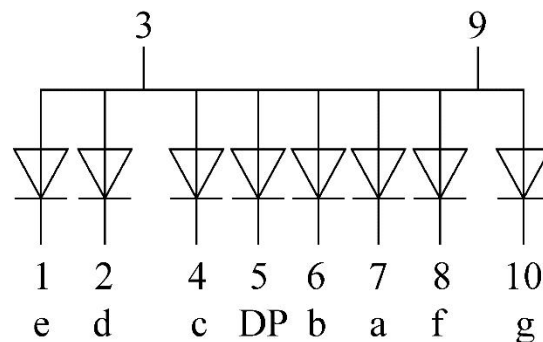
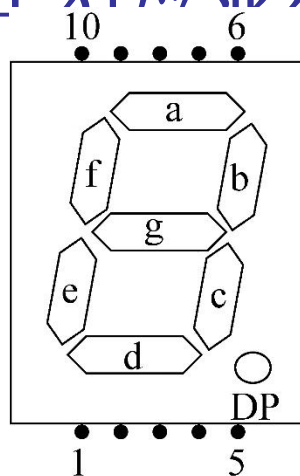
(a) 74LS374 用作输出接口

(b) 74LS374 用作输入接口

I/O接口综合应用例

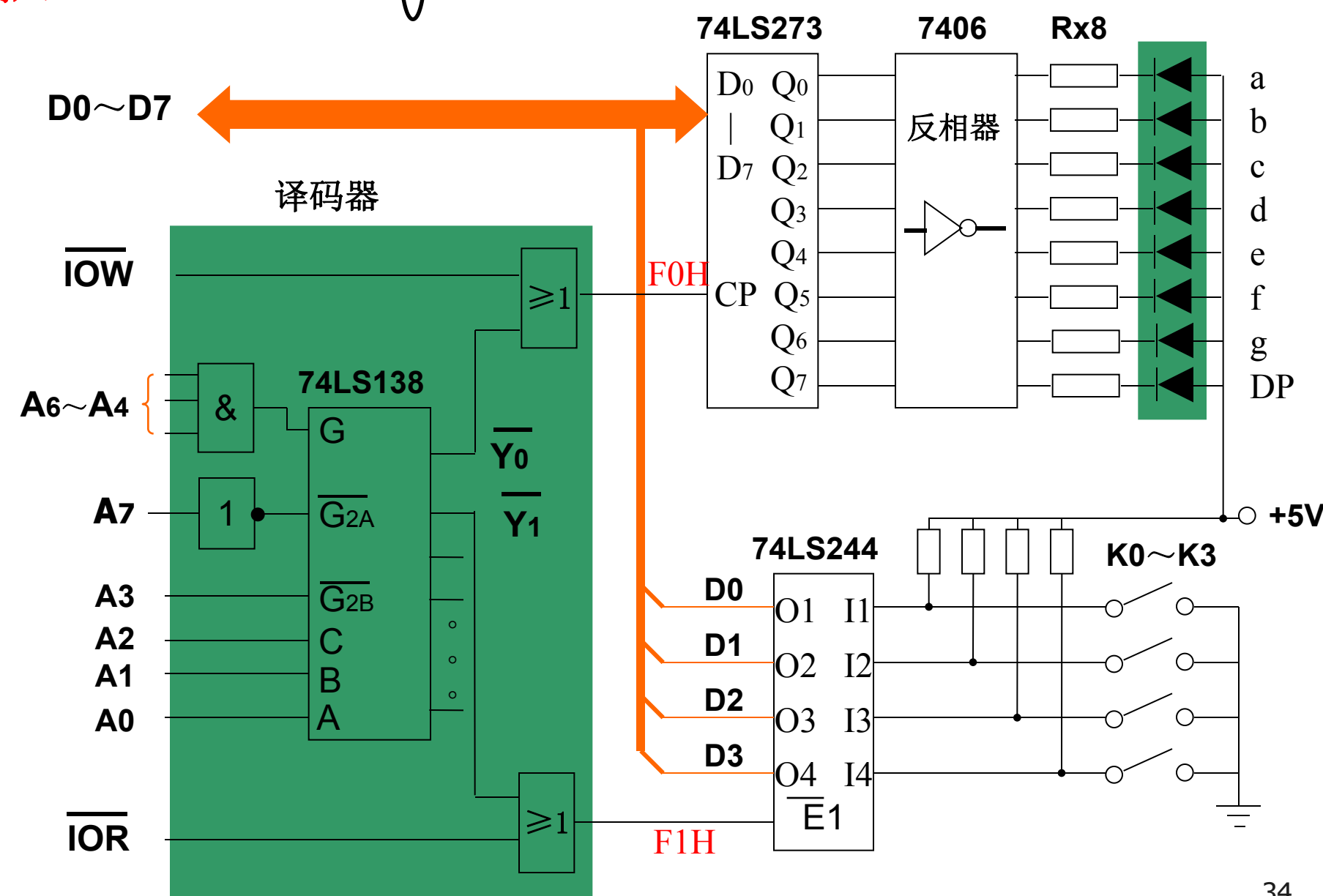
- 根据开关状态在**7**段数码管上显示数字或符号
- 设输出接口的地址为**F0H**
- 设输入接口地址为**F1H**
- 当**4**个开关的状态分别为**0000~1111**时，在**7**段数码管上对应显示'**0**'~'**F**'

共阳极数码管结构示意图



















输出口: F0H = 1111 000 0
 输入口: F1H = 1111 000 1

A0的两个取值分别用于显示输出和开关输入的控制



显示符号与输出数据对应表

符号	形状	7段码 Dp gfedcba D7 -- D0	符号	形状	7段码 Dp gfedcba D7 -- D0
'0'		00111111,3FH	'8'		01111111,7FH
'1'		00000110,06H	'9'		01100111,67H
'2'		01011011,5BH	'A'		01110111,77H
'3'		01001111,4FH	'B'		01111100,7CH
'4'		01100110,66H	'C'		00111001,39H
'5'		01101101,6DH	'D'		01011110,5EH
'6'		01111101,7DH	'E'		01111001,79H
'7'		00000111,07H	'F'		01110001,71H



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

} 程序控制方式

中断方式传送

直接存储器存取(**DMA**)



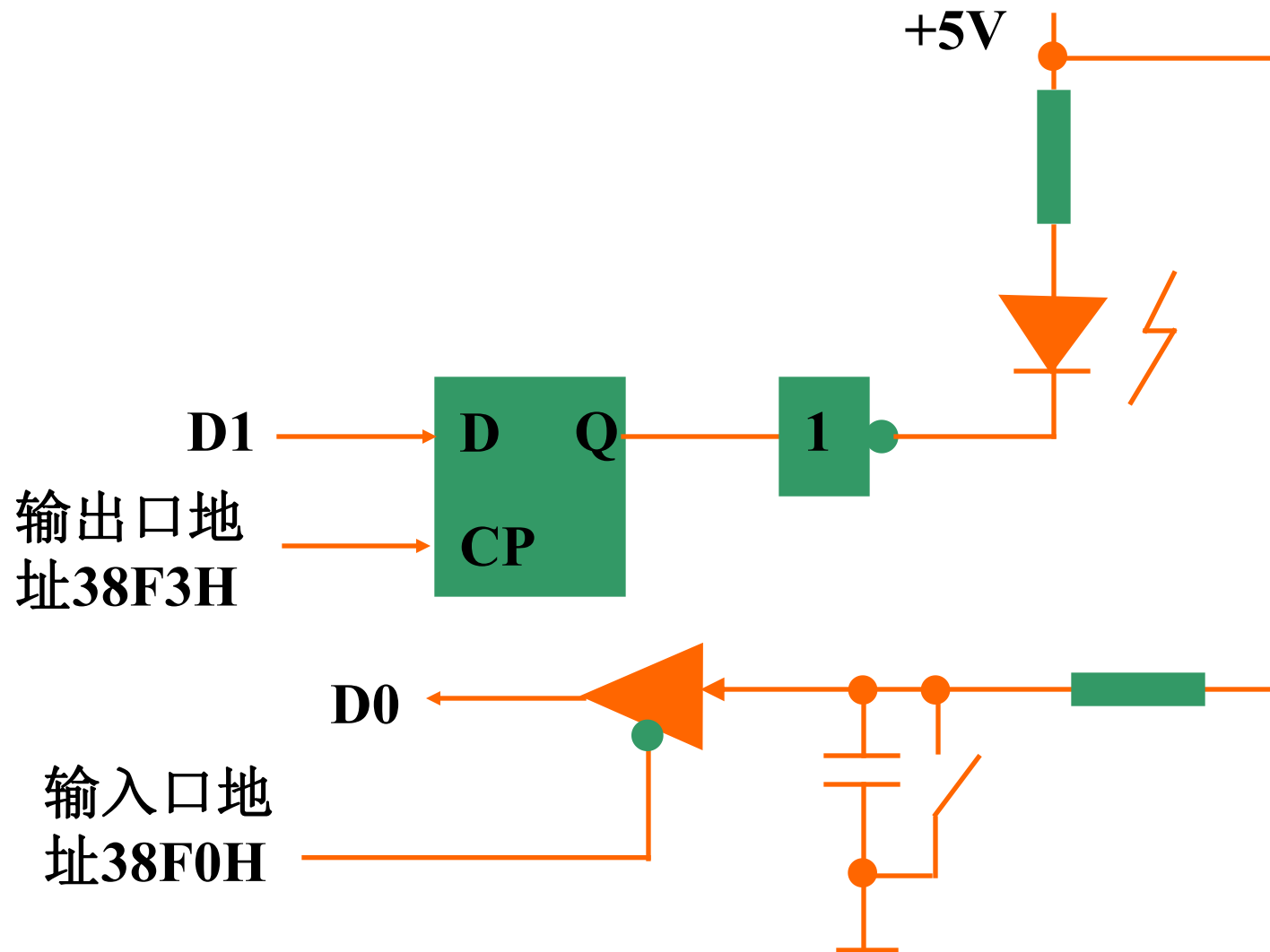
一、无条件传送

- 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。

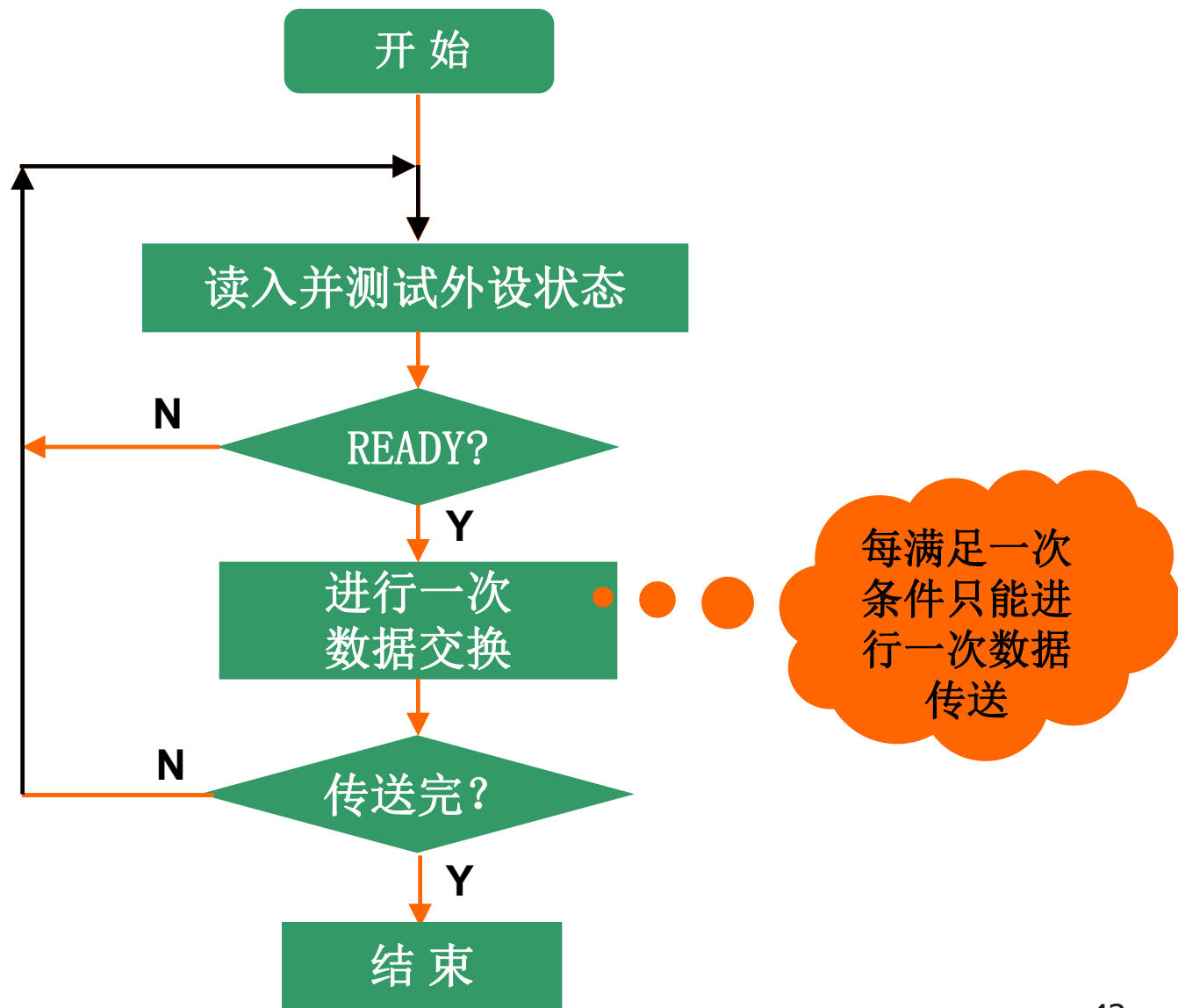


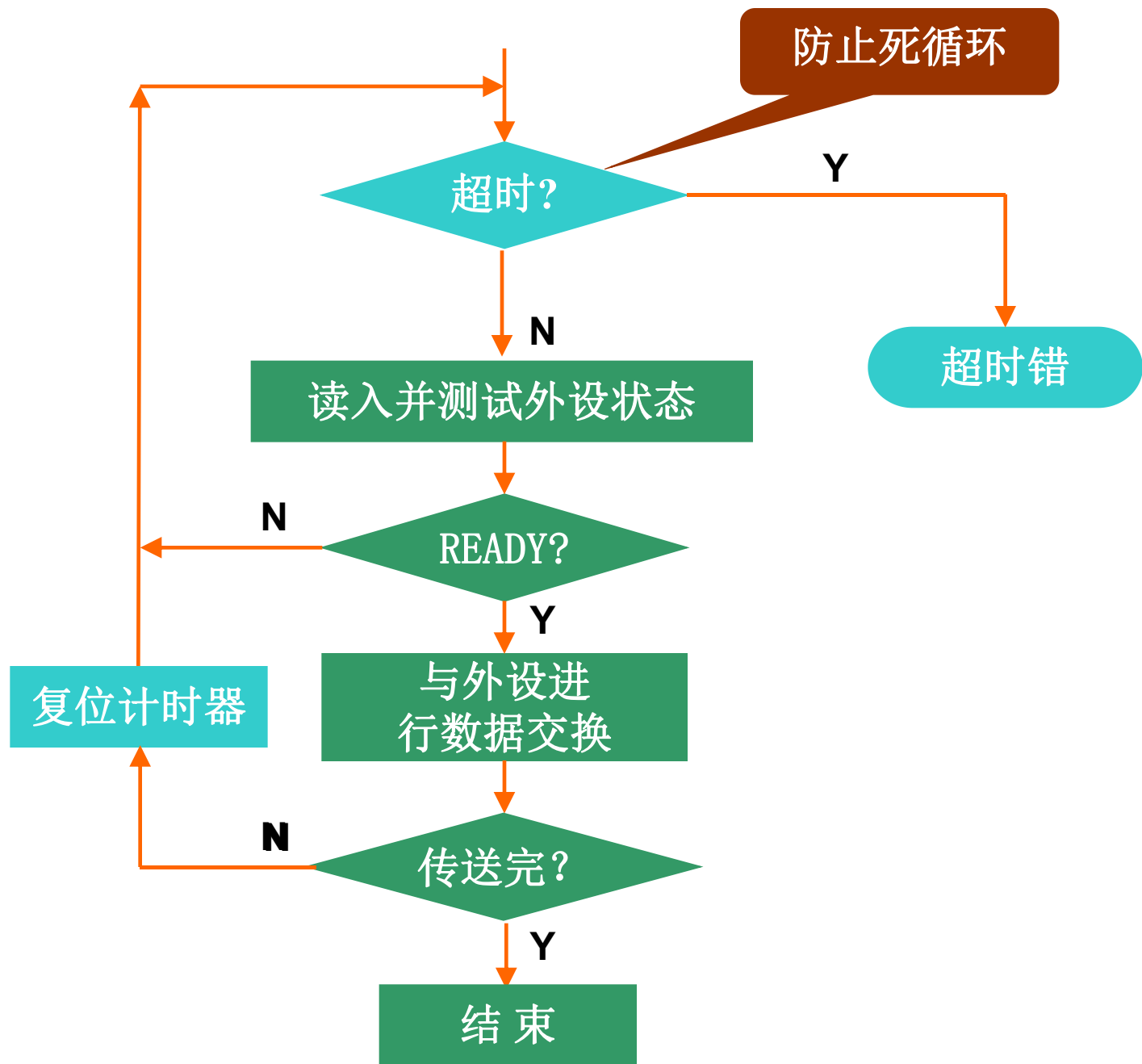


二、查询工作方式

- 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



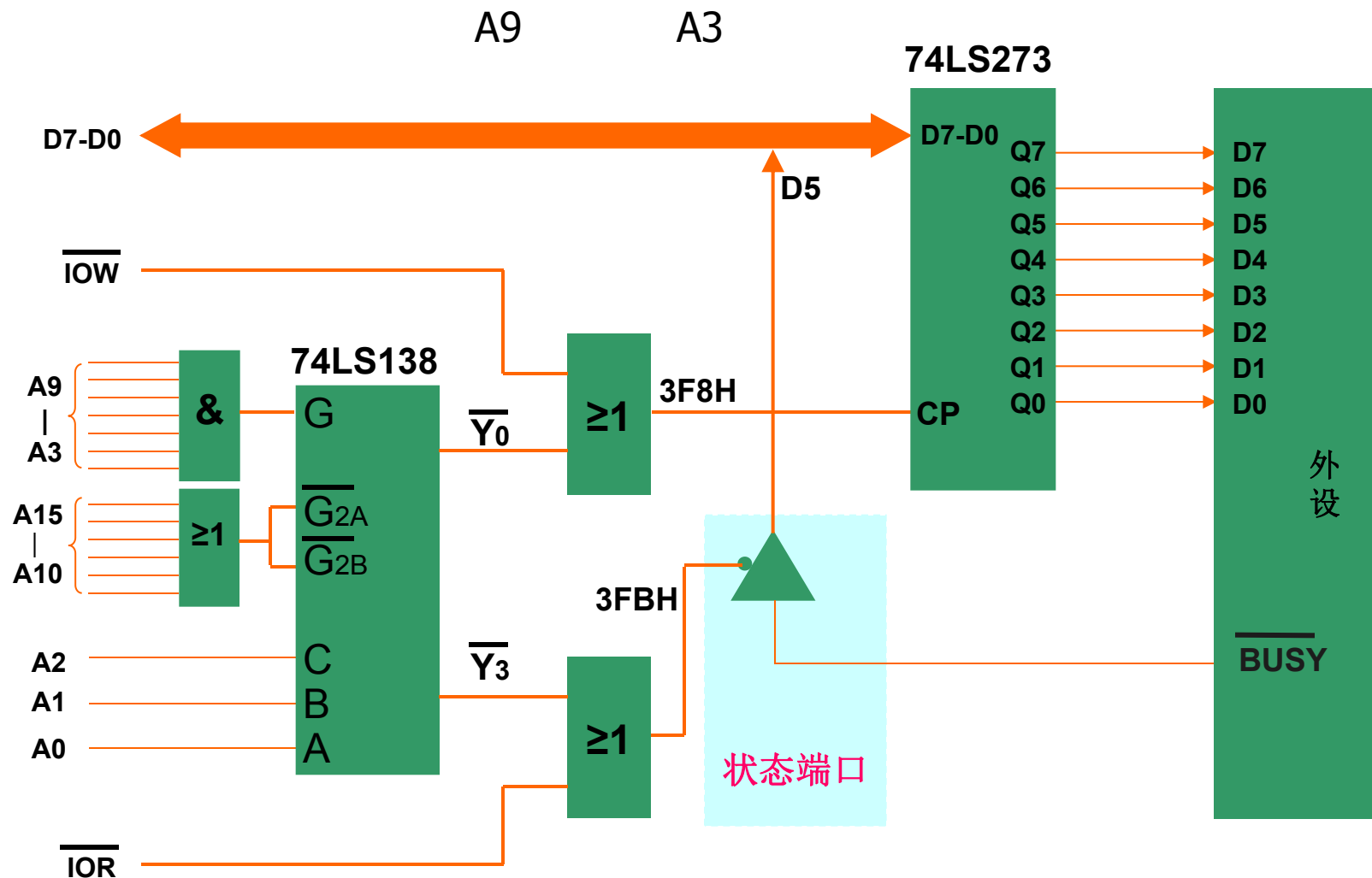




查询工作方式的例子

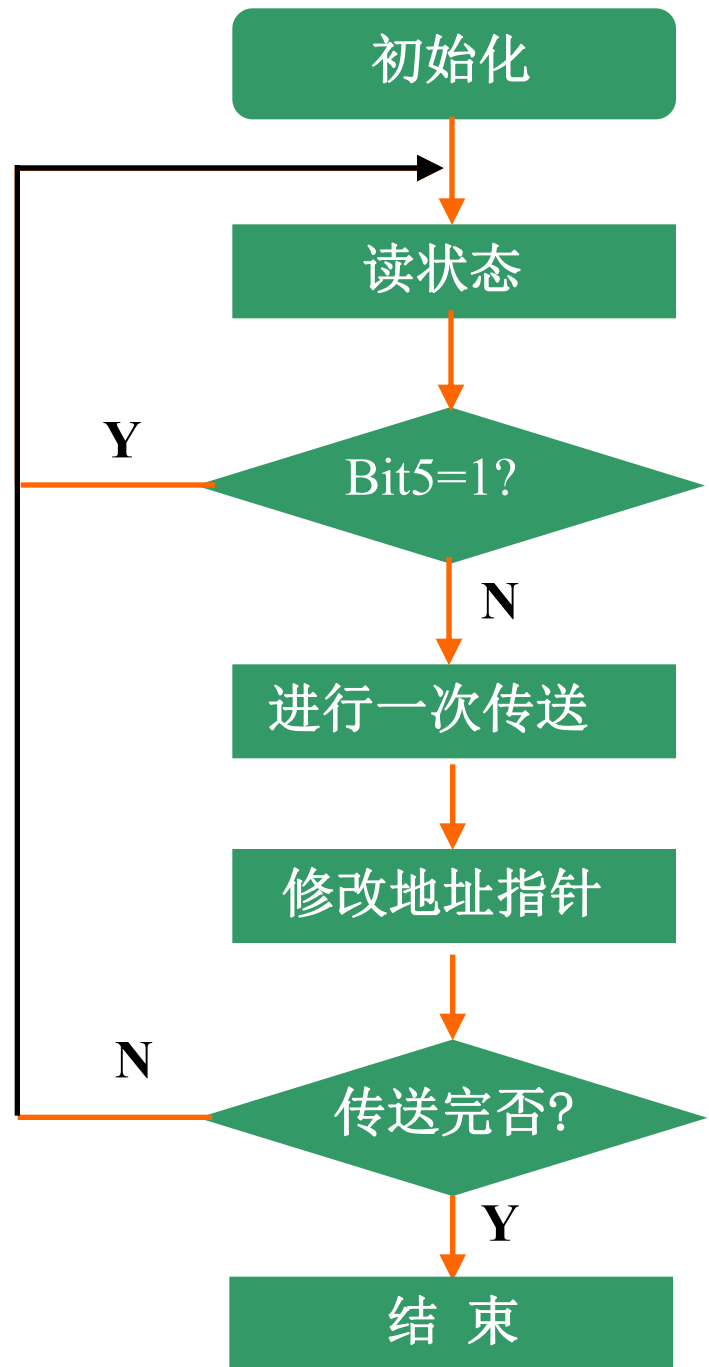
- 外设状态端口地址为03FBH，第5位(bit5)为状态标志（=1忙，=0准备好）
- 外设数据端口地址为03F8H，写入数据会使状态标志置1；外设把数据读走后又把它置0。
- 试画出其电路图，并将BUF中的100个字节数据输出。

数据端口地址: 0000 0011 1111 1000 3F8H



控制程序

```
LEA SI,BUF
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

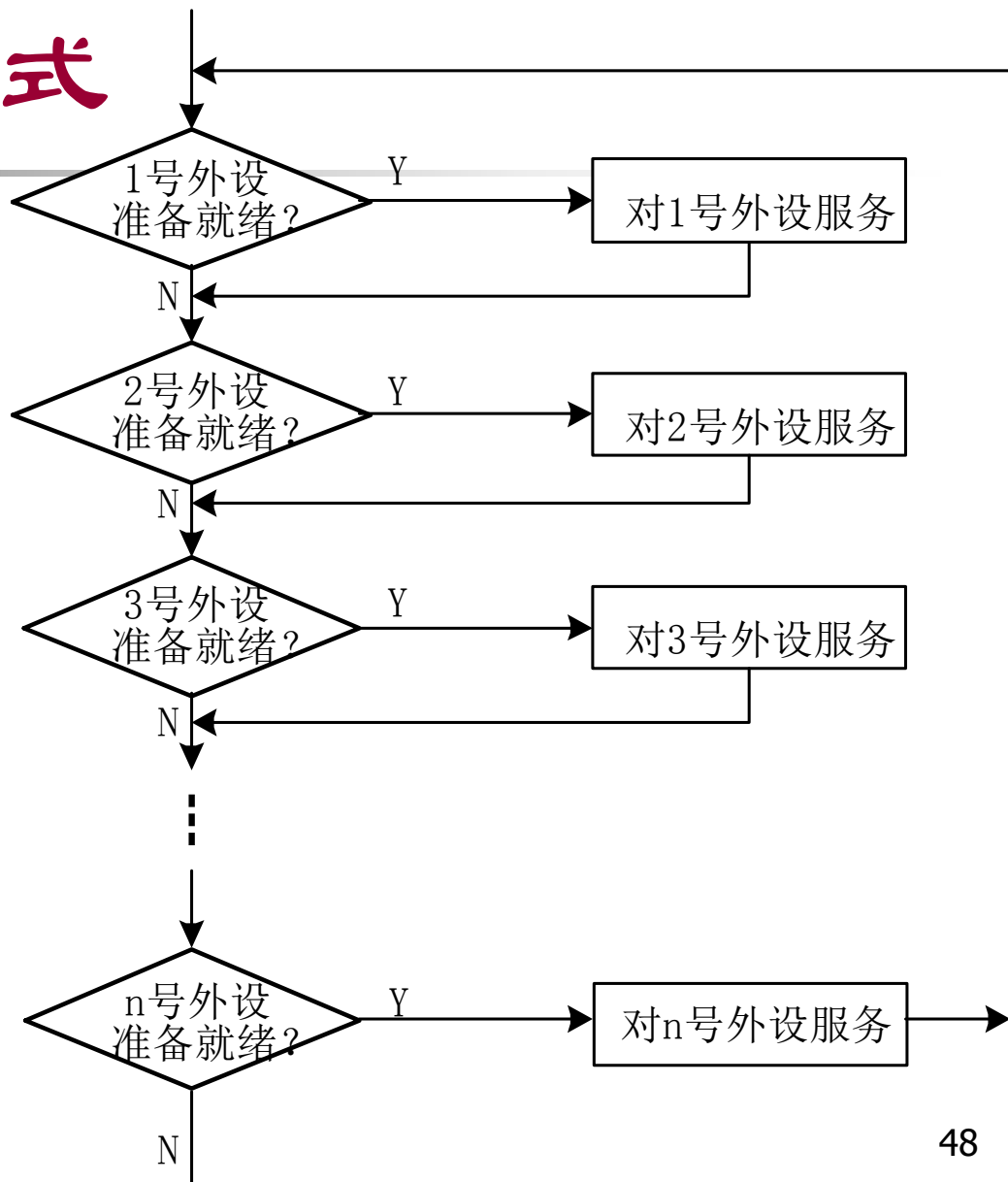
■ 优点:

- 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

一般工作流程





三、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:

- 程序编制相对较复杂。



以上三种I/O方式的共性

- 信息的传送均需通过**CPU**
- 软件：
 - 外设与内存之间的数据传送是通过**CPU**执行程序来完成的（**PIO**方式）；
- 硬件：
 - **I/O**接口和存储器的读写控制信号、地址信号都是由**CPU**发出的。
- 缺点：
 - 程序的执行速度限定了传送的最大速度

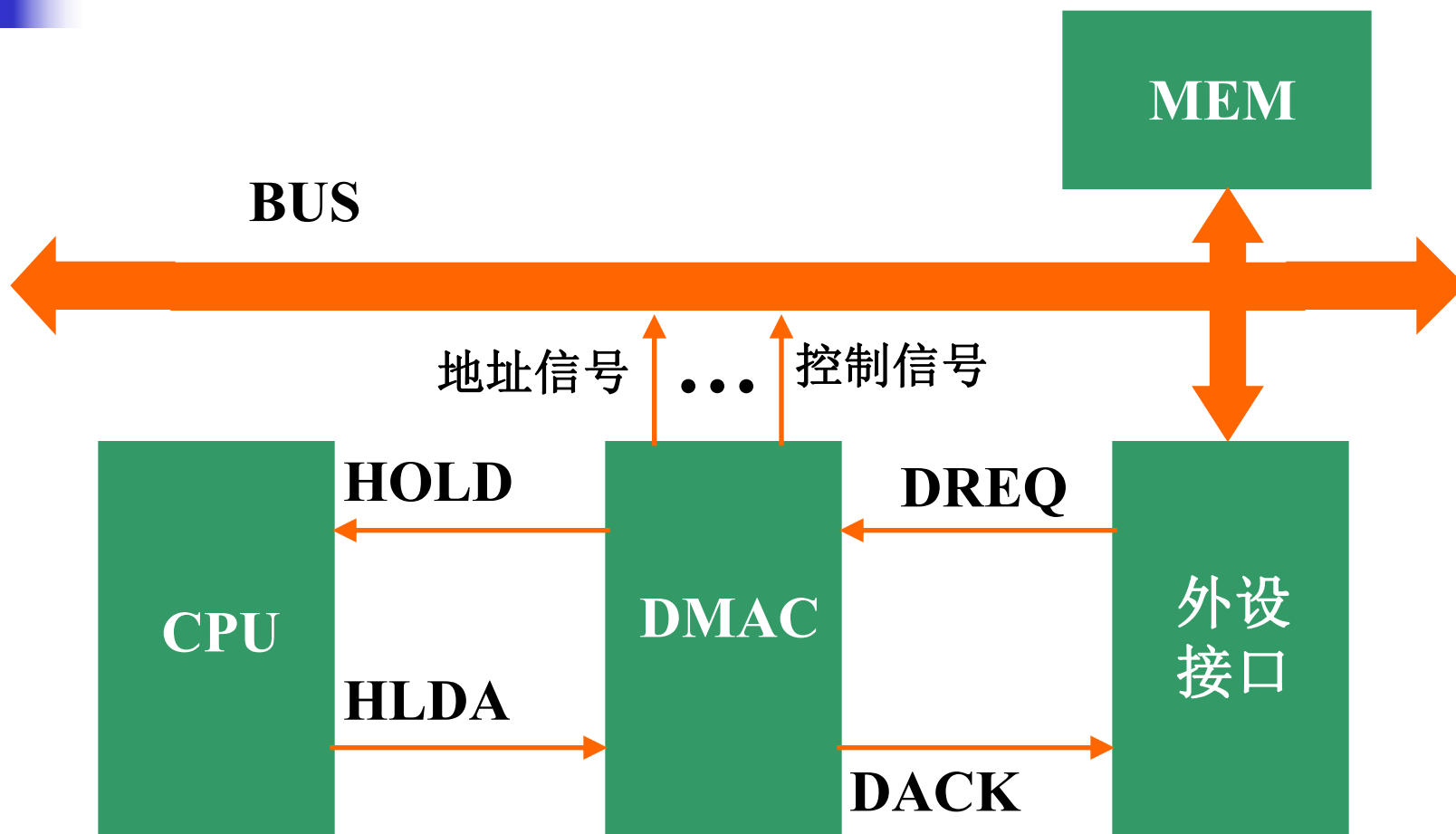


四、DMA控制方式

■ 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DREQ;
- 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控制方式工作过程例

- 例：从外设向内存传送若干字节数据
 - DMAC向I/O接口发出读信号；
 - 向地址总线上发出存储器的地址；
 - 发出存储器写信号和AEN信号；
 - 传送数据并自动修改地址和字节计数器
 - 判断是否需要重复传送操作；
 - 若数据传送完，DMA控制器撤销发往CPU的HOLD信号；
 - CPU检测到HOLD失效后，则撤销HLDA信号，并在下一时钟周期重新开始控制总线。



DMA工作方式

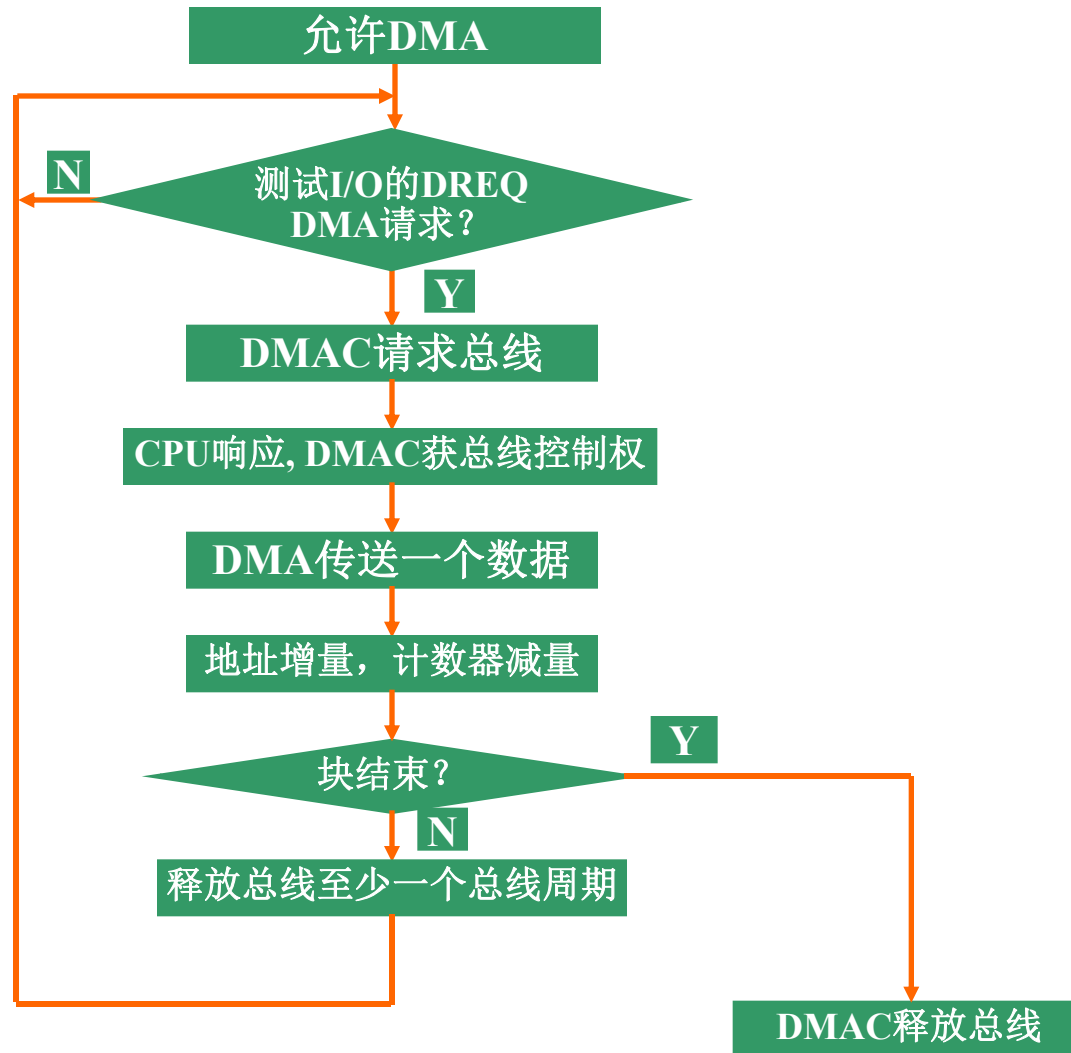
- 周期窃取:

- 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:

- **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。

周期窃取的DMA方式：





DMA控制方式的优缺点

- 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



一、中断的基本概念

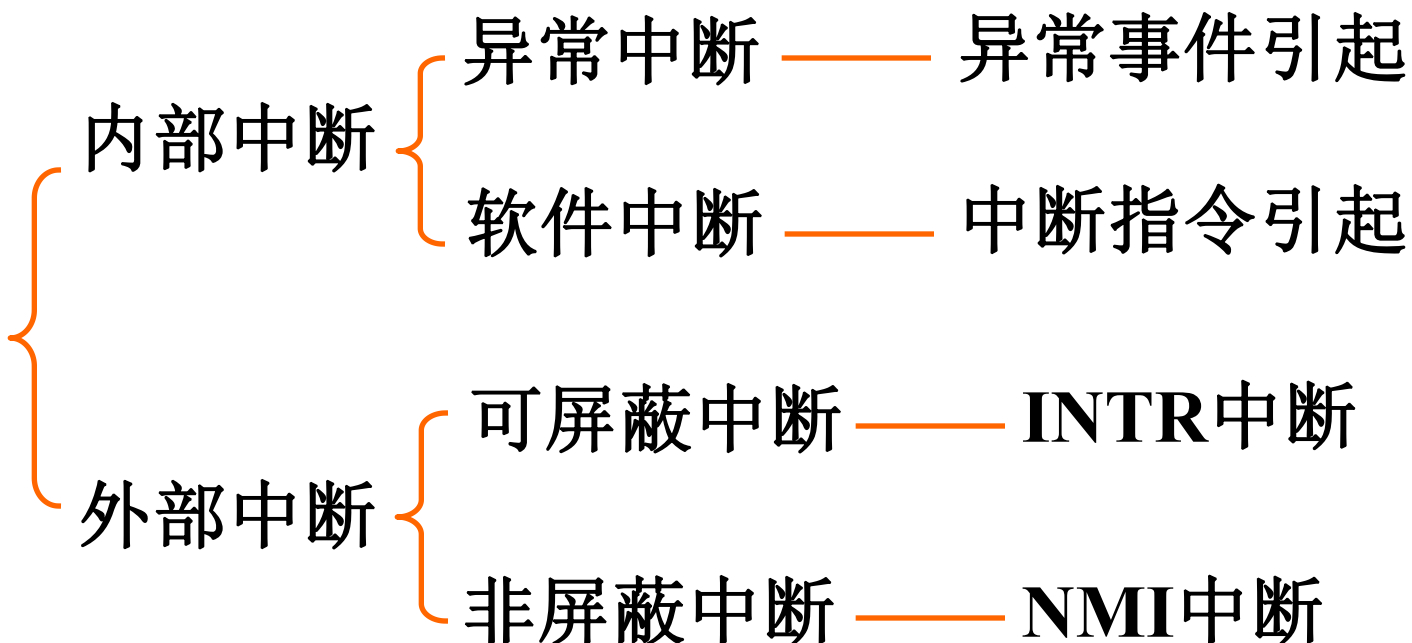
■ 中断：

- **CPU**执行程序时，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(称为中断服务程序或中断处理程序)，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。



中断源

- 引起**CPU**中断的事件，发出中断请求的来源。





引入中断的原因

- 提高数据传输率；
- 避免了**CPU**不断检测外设状态的过程，提高了**CPU**的利用率。
- 实现对特殊事件的实时响应。



二、中断处理的一般过程

- 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：在中断处理程序中查找中断源
- 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。

中断源识别及确定先响应哪个中断请求
(中断判优) 由硬件系统完成



中断判优

- 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。
- 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
- 排队法则
 - 先来先响应



中断嵌套



中断判优方法

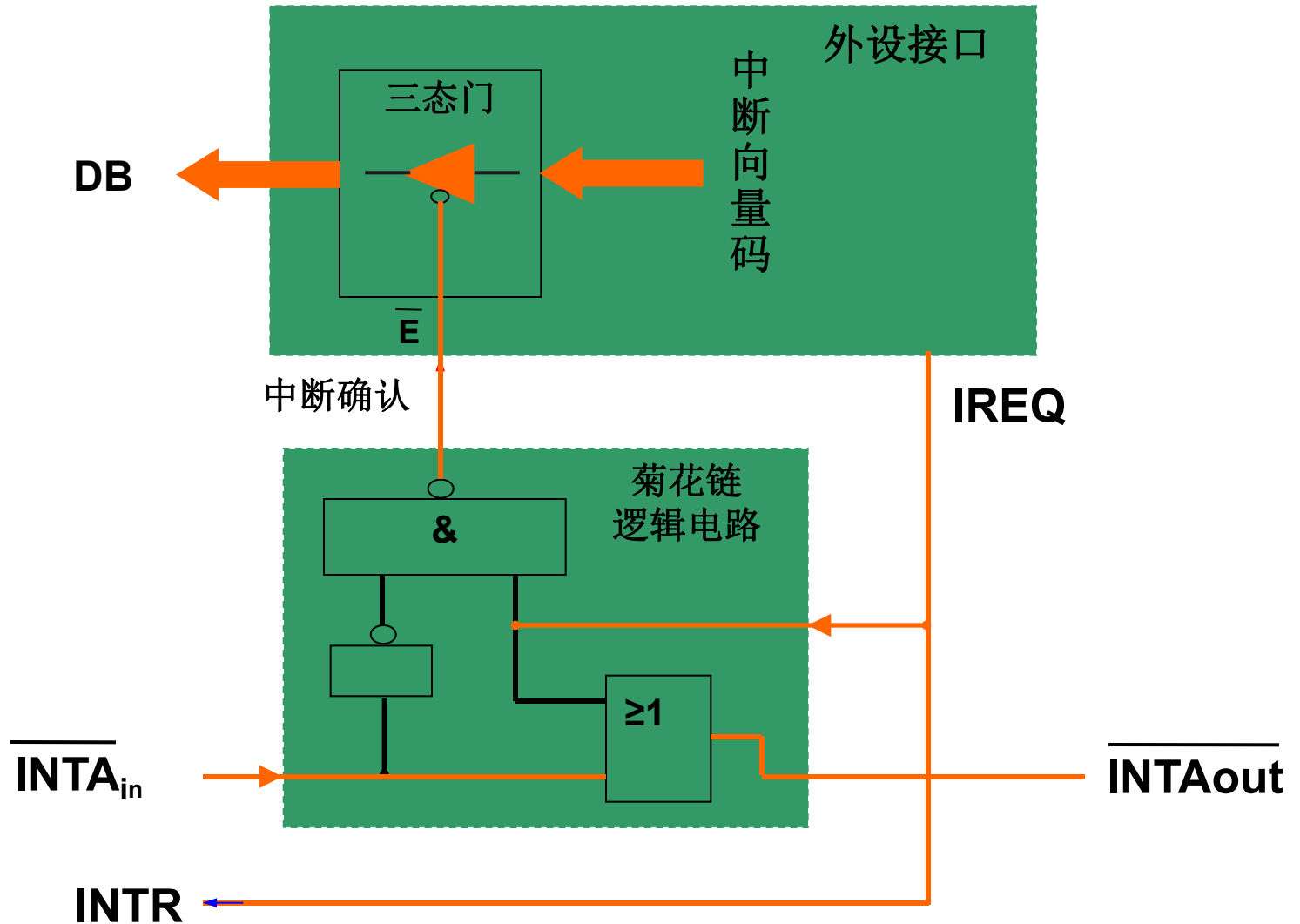
- 软件判优

- 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

- 硬件判优

- 链式判优、并行判优（中断向量法）

菊花链逻辑电路





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



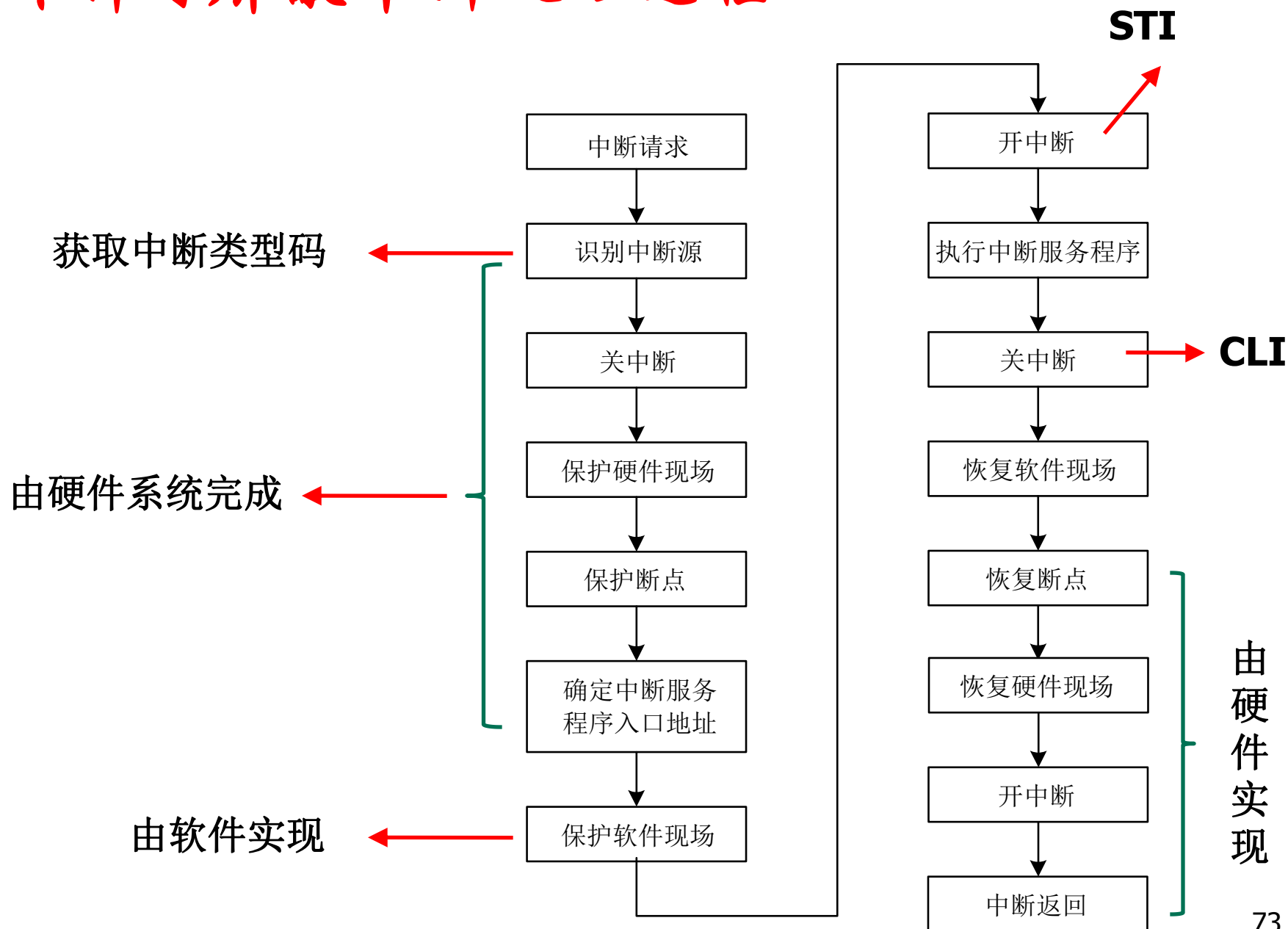
中断返回

- 执行**IRET**指令，使**IP**、**CS**和**FLAGS**从堆栈弹出



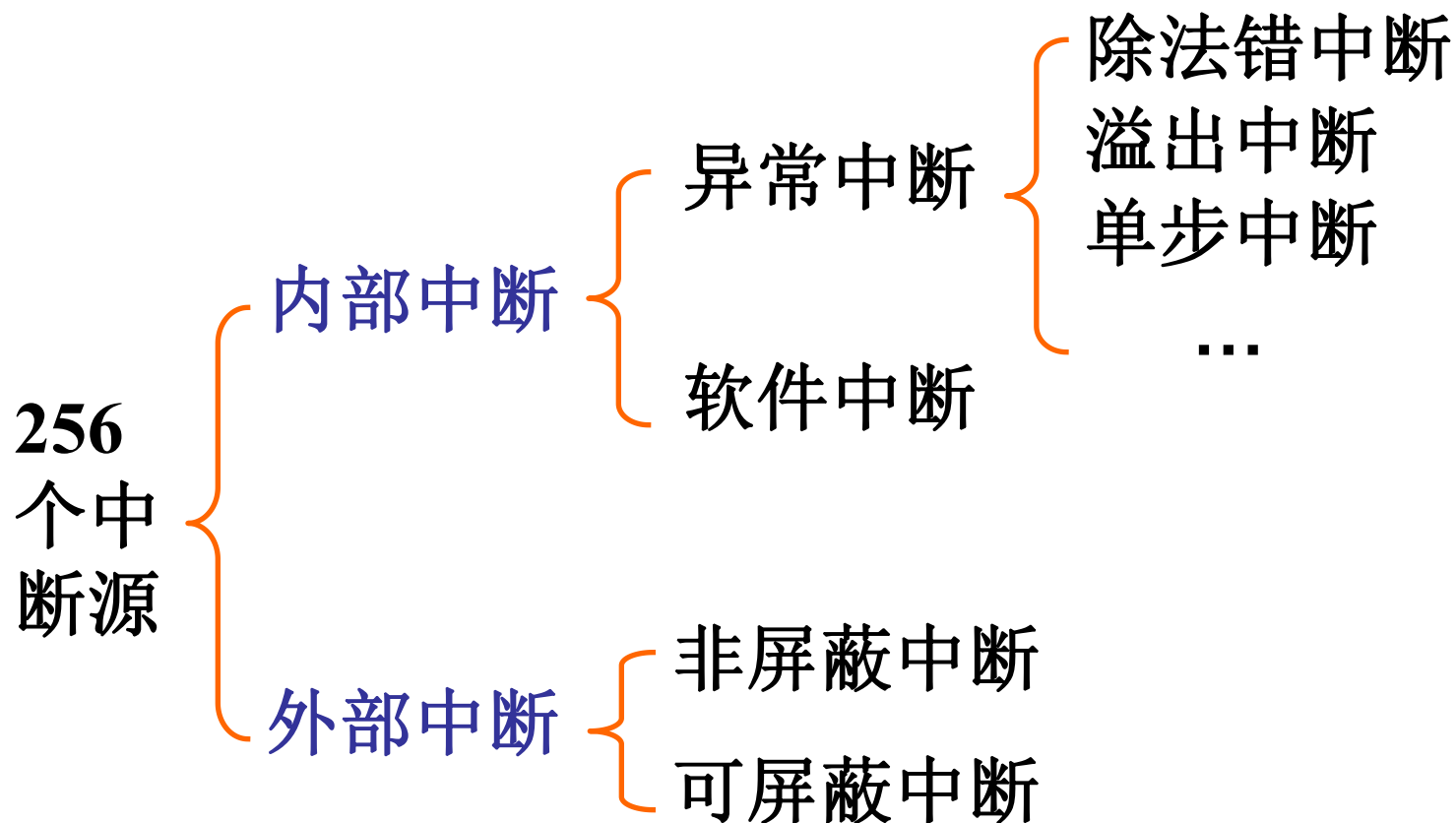
恢复断点和硬件现场

外部可屏蔽中断处理过程

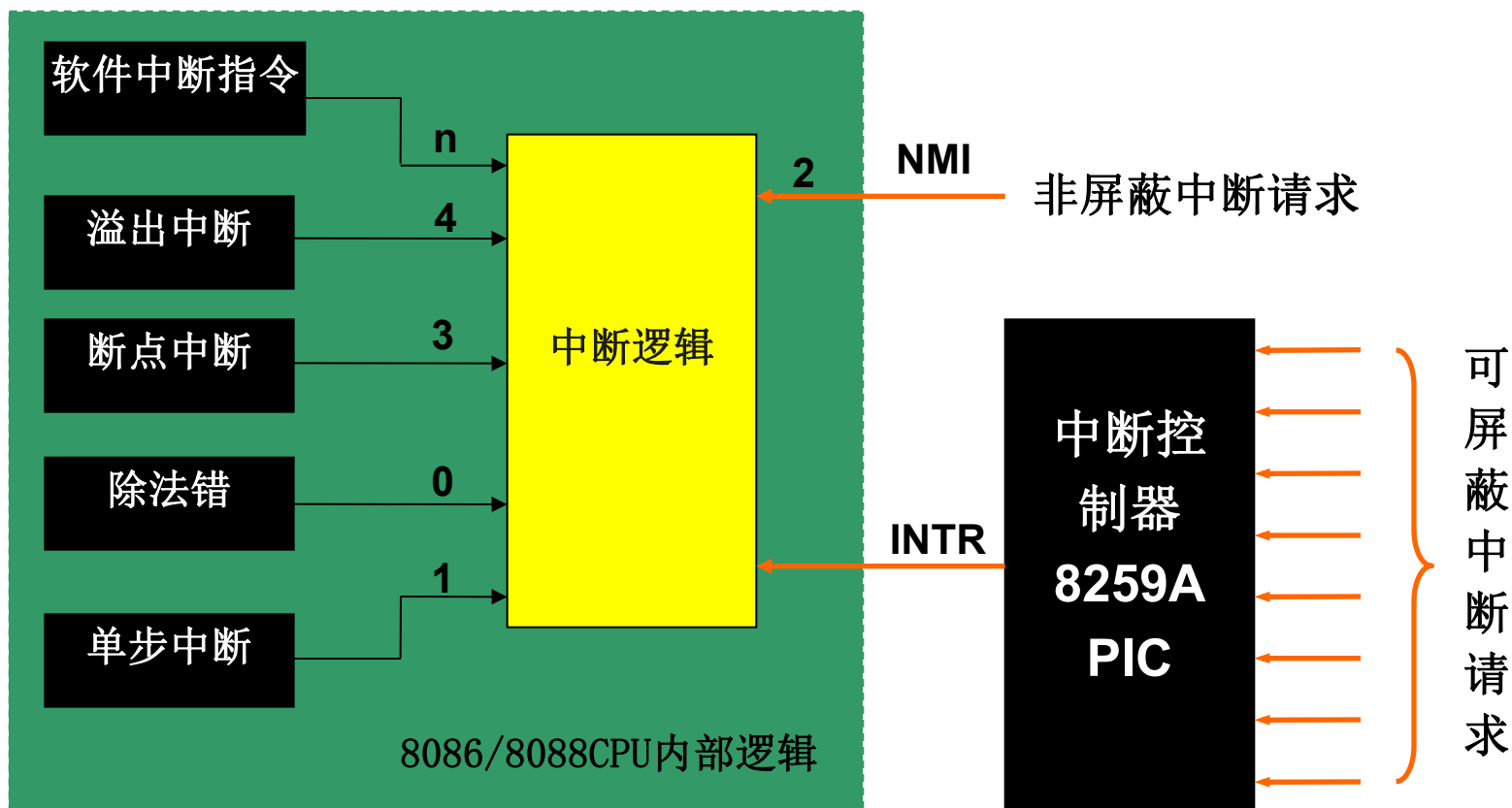




三、8088/8086中断系统



8086/8088中断源类型:





内部中断

中断类型	功能
类型0	除数为0中断例行程序
类型1	单步
类型2	非屏蔽中断， NMI
类型3	设置断点
类型4	溢出处理中断， INTO指令
类型10	显示设备中断
类型20	程序结束中断
类型21	DOS系统功能调用功能程序



外部中断

1、非屏蔽中断

- **NMI** 引脚上出现上升沿触发，
- 不受标志位**IF**的限制
- 类型号：**2**

2、可屏蔽中断

- **INTR**引脚输入，高电平有效
- 受标志位**IF**的限制
- 类型号：**08H~0FH 70H~77H**

中断向量表

00000H

003FFH

1KB

内存空间

00000H

IP偏移地址

CS段基地址

0型中断向量

00004H

IP偏移地址

CS段基地址

1型中断向量

00008H

IP偏移地址

CS段基地址

2型中断向量

0000CH

IP偏移地址

CS段基地址

3型中断向量

00010H

IP偏移地址

CS段基地址

4型中断向量

00014H

⋮

⋮

00080H

IP偏移地址

CS段基地址

类型码为32
的中断向量

⋮

⋮

003FFH

IP偏移地址

CS段基地址

类型码为255
的中断向量



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**，低字为段内偏移，高字为段基址；
- 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：
 - 将中断类型码为**48H**的服务程序入口地址放入向量表
- p274 教材中使用了**DOS**功能调用将服务程序入口地址写入向量表



中断向量表的初始化

- 将类型码为**48H**的中断服务子程序**TIMER**的中断向量用**MOV**指令放入向量表
 - **MOV AX, 0000H**
 - **MOV DS, AX**
 - **MOV SI, 0120H; 48H*4**
 - **MOV BX, OFFSET TIMER**
 - **MOV [SI], BX**
 - **MOV BX, SEG TIMER**
 - **MOV [SI+2], BX**

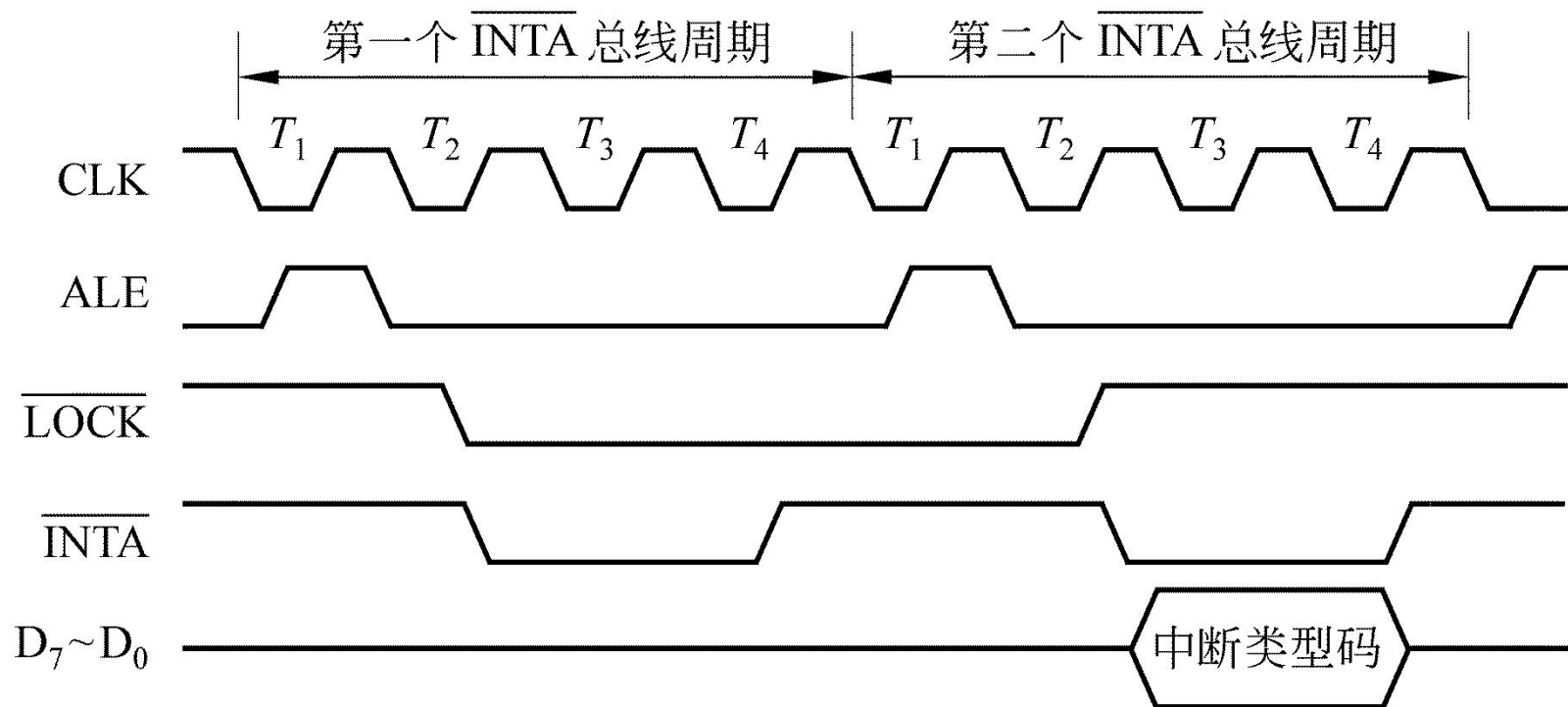


8088内部中断与NMI中断

特点:

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

可屏蔽中断的类型号的获取



8086 的中断响应和处理流程

