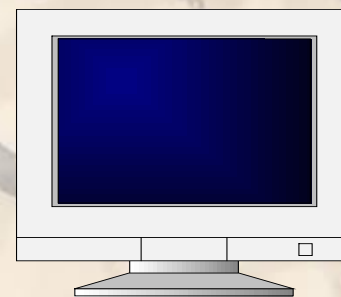


第6章

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（自学）**



§ 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

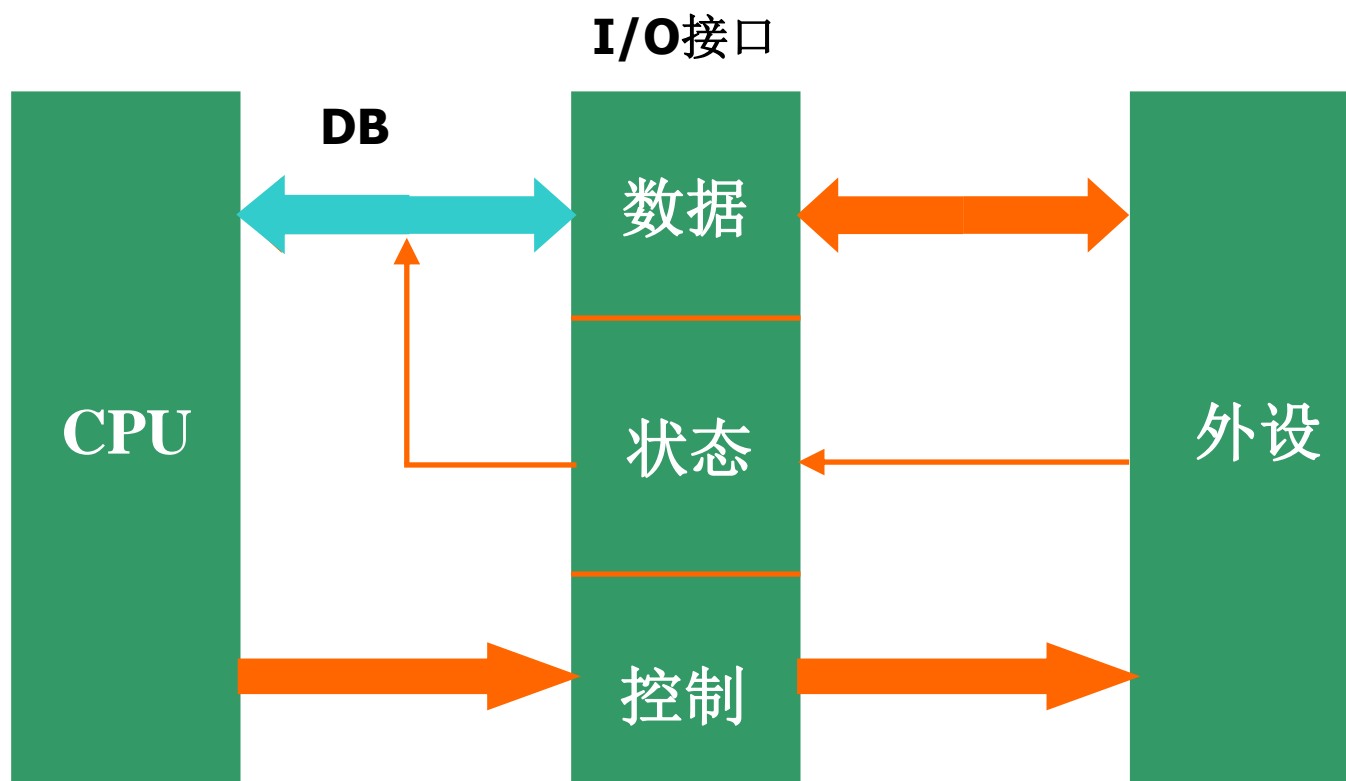
- 有关I/O系统的概念、特点，以及I/O接口的功能等，请参阅教材描述，自行学习。
- 总体上，I/O接口应具备以下功能：
 - 数据的缓冲与暂存
 - 信号电平与类型的转换
 - 增加信号的驱动能力
 - 对外设进行监测、控制与管理，中断处理



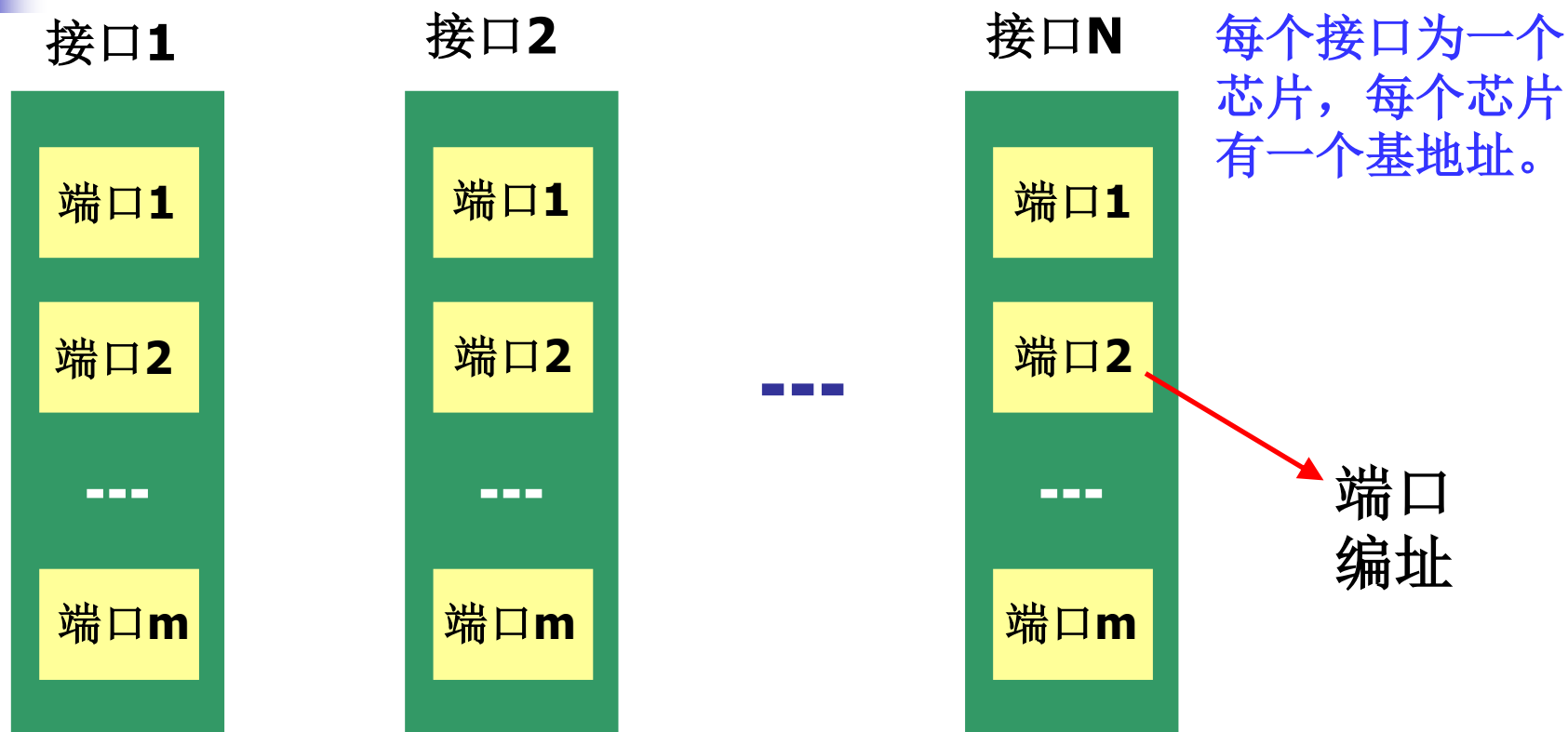
三、I/O端口

- **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的地址



端口地址 = 芯片地址（高位地址） + 片内地址
类似于学号 = 学院(专业)编号 + 班号 + 班内编号



四、I/O端口的编址方式

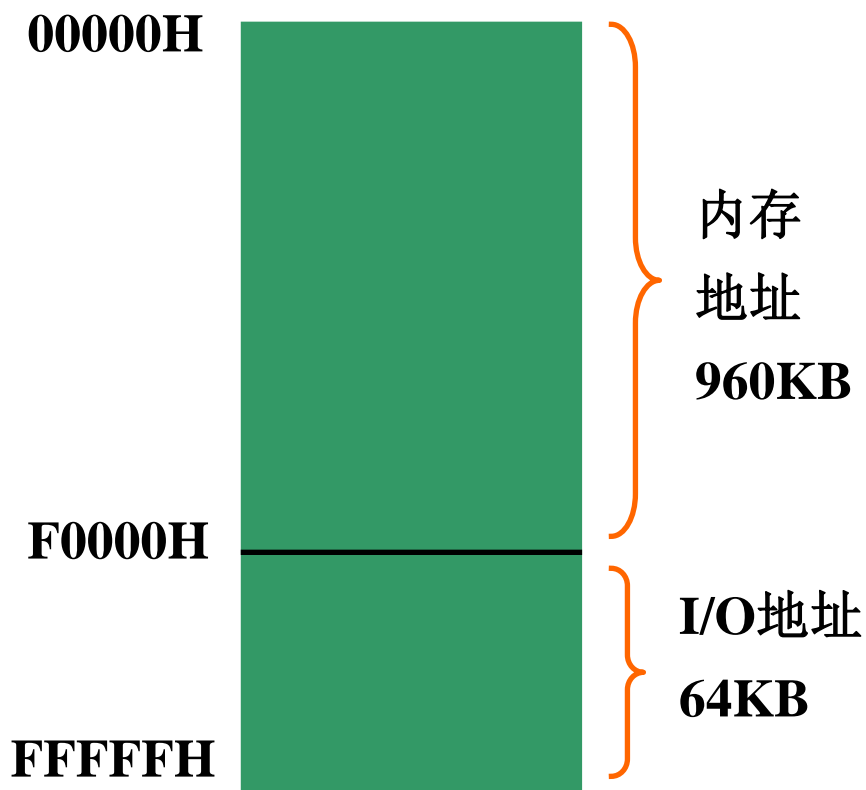
- 编址方式:
 - 与内存统一编址
 - 独立编址

1. I/O端口与内存统一编址

特点:

- 指令及控制信号统一
- 内存地址资源减少

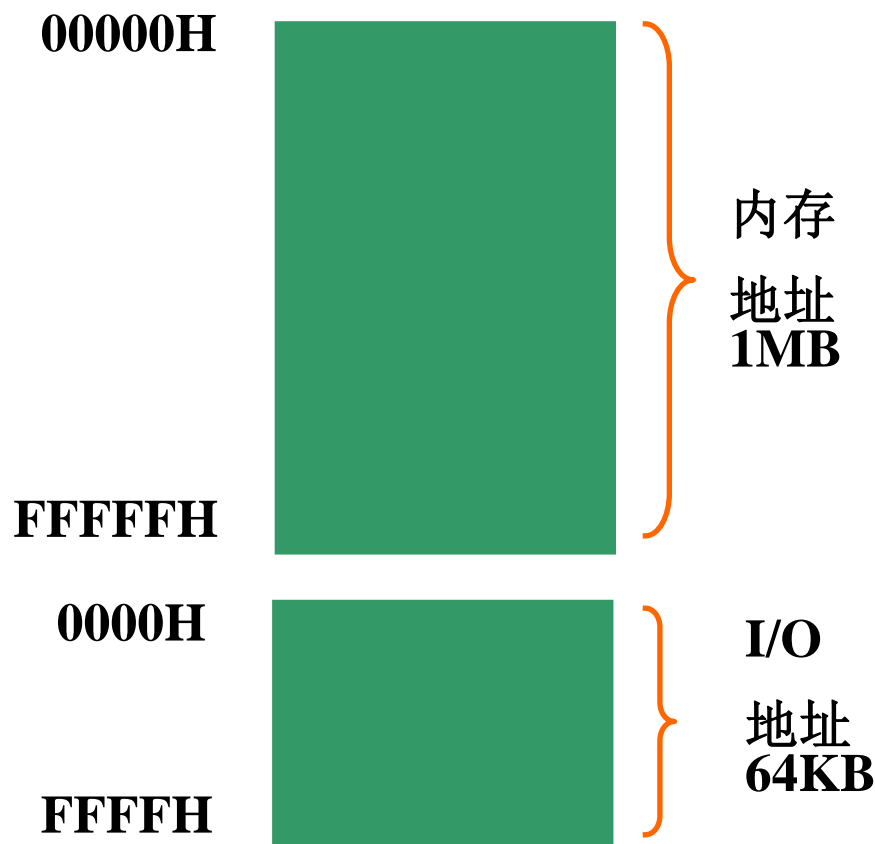
例如，系统地址空间**1MB**



2. I/O端口独立编址

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端口独立编址方式(但地址线与存储器共用)
- 地址线上的地址信号用 $\text{IO}/\overline{\text{M}}$ ($\overline{\text{IO}}/\text{M}$) 来区分
- I/O操作只使用20根地址线中的**16根**: $\text{A}_{15} \sim \text{A}_0$
- 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)



五、I/O地址译码

- 目的
 - 确定端口的地址
- 寻址端口的信号
 - $\overline{\text{IOR}}$ 、 $\overline{\text{IOW}}$ (最大模式); $\overline{\text{RD}}$ 、 $\overline{\text{WR}}$ (最小模式)
 - A15 ~ A0
- 参加译码的信号:
 - $\overline{\text{IOR}}$, $\overline{\text{IOW}}$ (或 $\overline{\text{RD}}$ 、 $\overline{\text{WR}}$), 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



全地址译码与部分地址译码

- 一个接口电路中可以有多个端口。
- **全地址译码：**
 - 全部16位I/O地址信号参与译码
 - 当接口中只有一个端口时，16位地址线应全部参与译码，译码输出直接选择该端口中；
 - 当接口中有多个端口时，则16位地址线的高位参与译码（决定接口的基地址），最低几位直接输入到接口芯片，用于寻址接口中的各个端口。
- **部分地址译码：**
 - 仅用部分地址（比如16位中低10位）信号参与译码
 - 含多个端口的接口，最低的几位直接连到接口芯片，用于寻址接口中的各个端口。



I/O地址译码示例

I/O系统中，因地址资源丰富，多采用部分地址译码。

- **例：某外设接口有4个端口，地址为2F0H——2F3H，由 $A_{15} \sim A_2$ 译码得到，而 A_1 、 A_0 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有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位为接口芯片地址，即片选地址信号。



■ 地址范围:

A11.....A1A0

■ × × × × 0 0 1 0 1 1 1 1 0 0 0 0

.....

.....

.....

■ × × × × 0 0 1 0 1 1 1 1 0 0 1 1



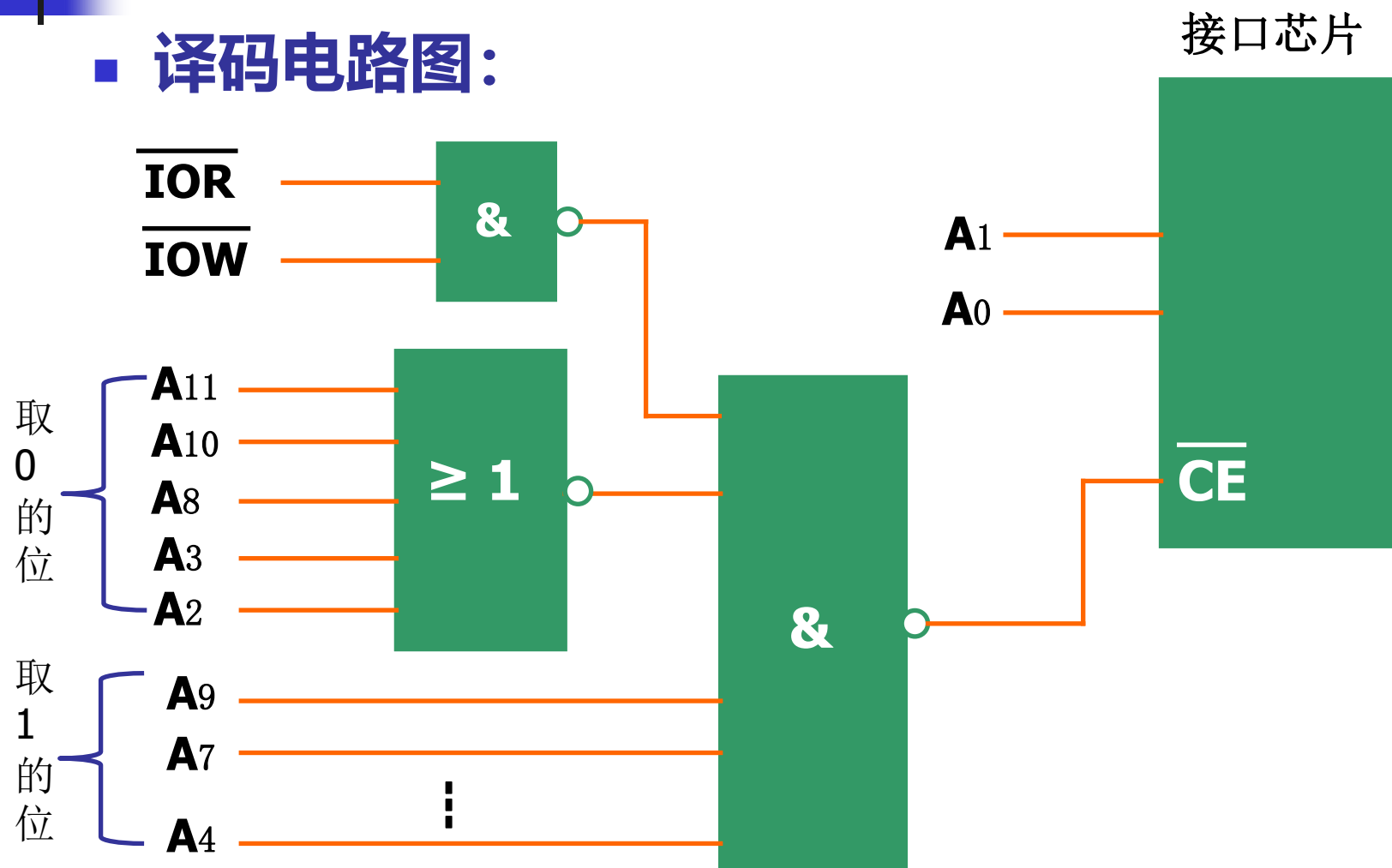
任意状态

片内地址

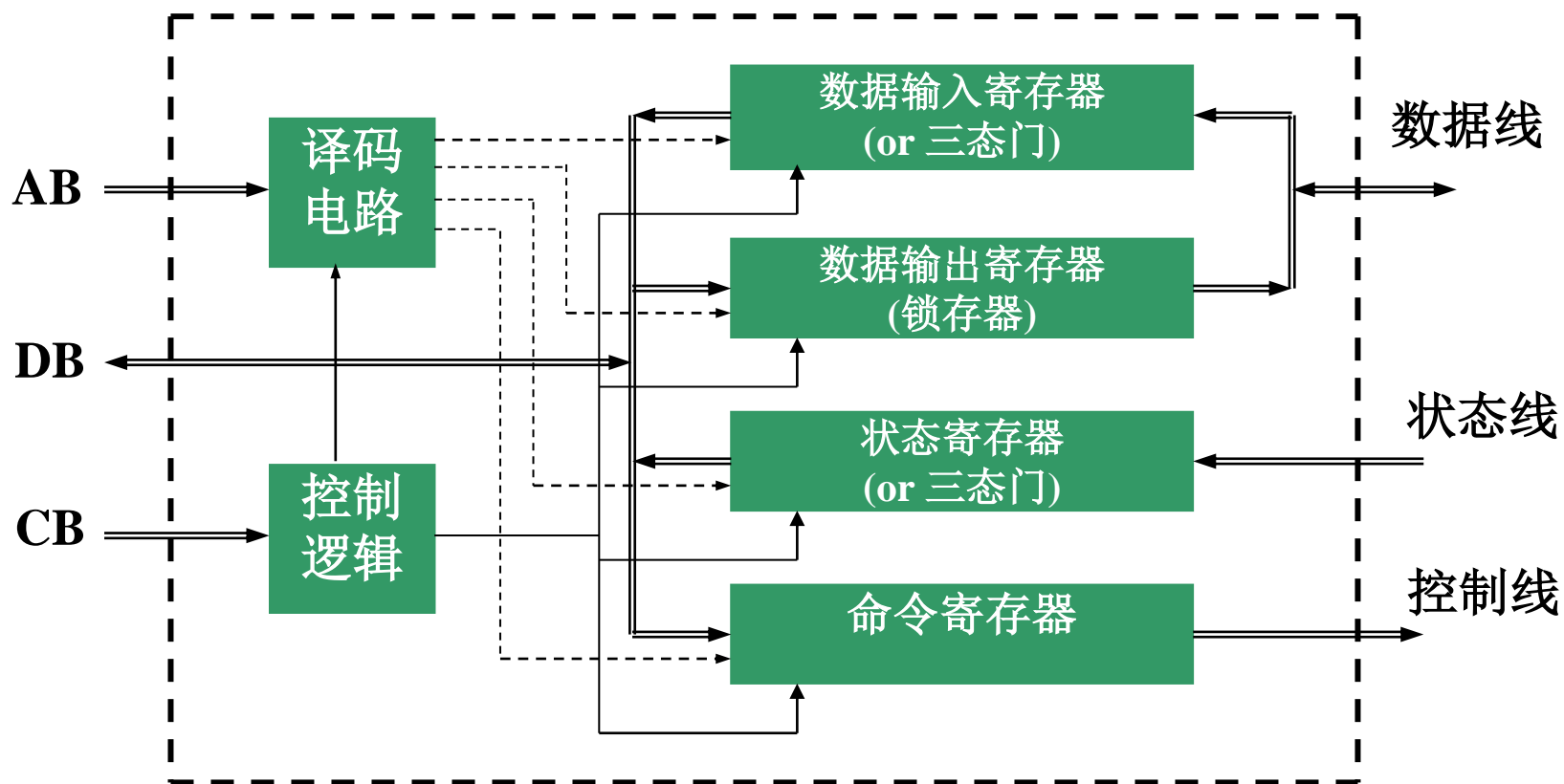
图中不接入

I/O地址译码例

译码电路图：



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:

- 要求对数据具有输出控制能力（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



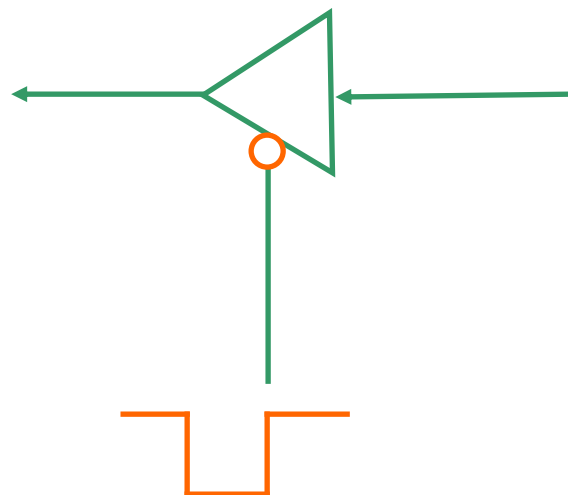
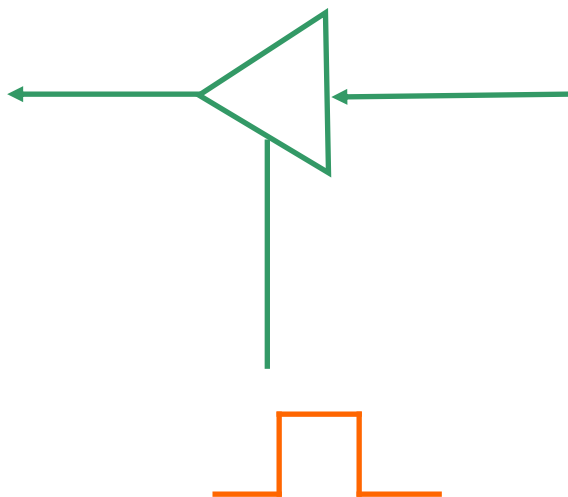
§ 6.2 简单接口电路

掌握：

三态门和锁存器两类简单接口芯片的应用

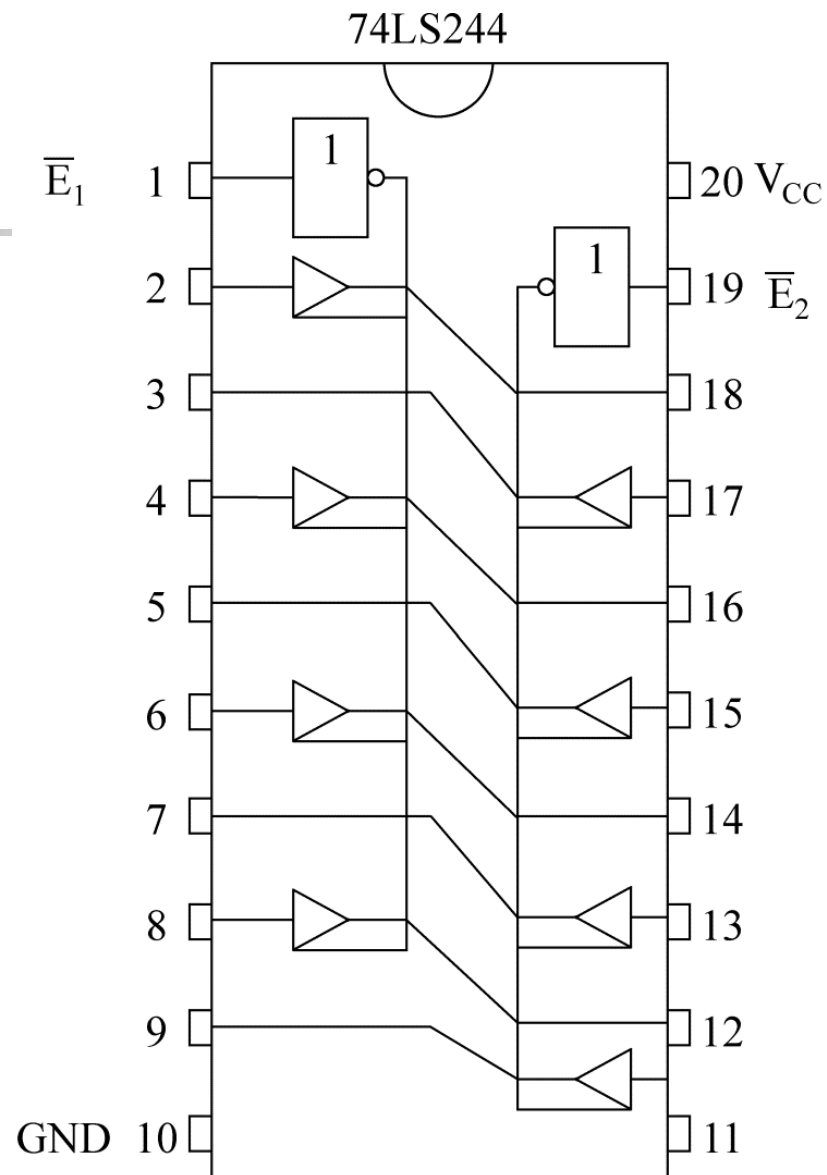
一、三态门接口（回顾）

- 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



例6-1 编程判断图中的开关状态，若全闭合则转**NEXT1**，否则转**NEXT2**。



接口地址:

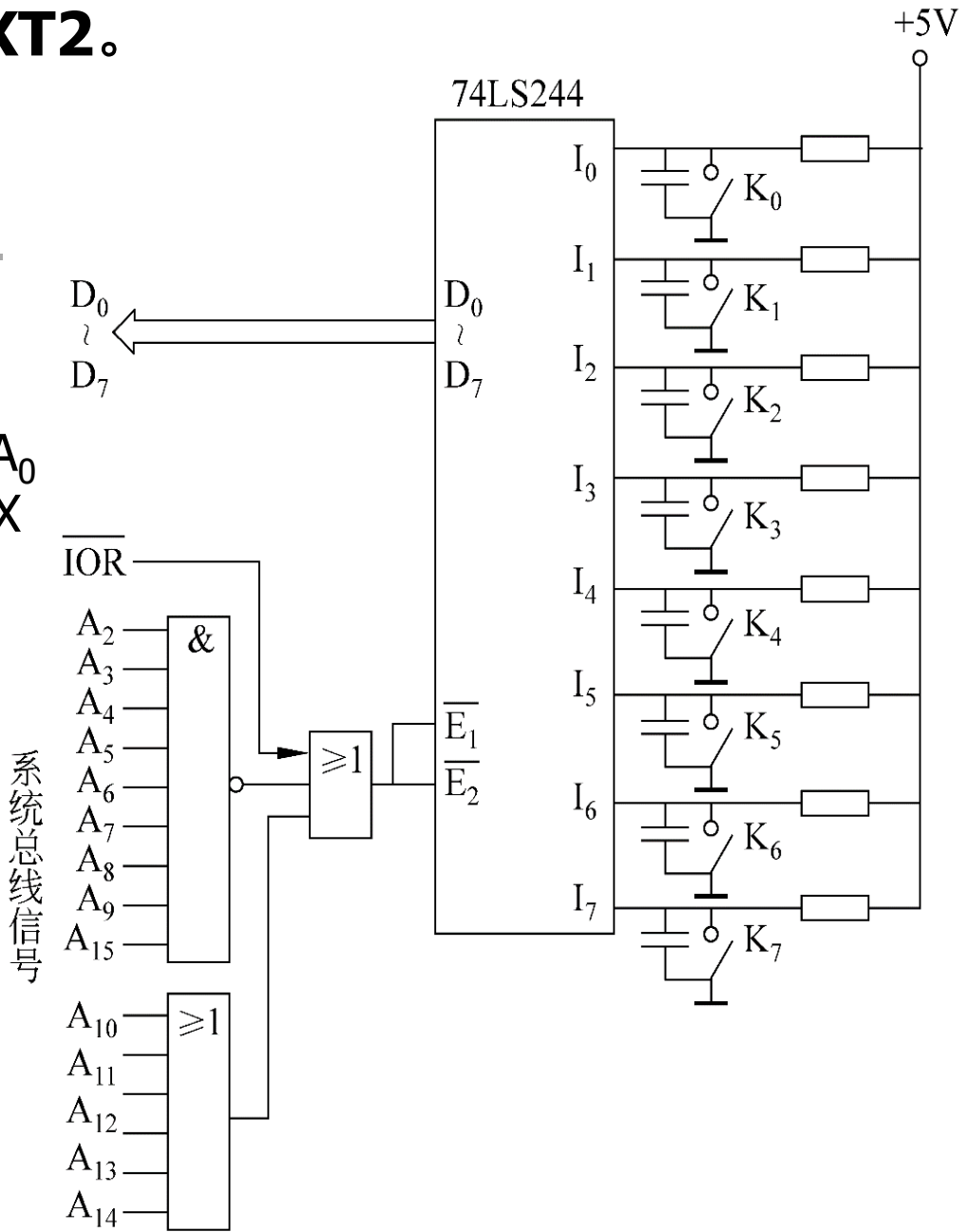
A ₁₅	A ₁₄	...	A ₉	...	A ₂	A ₁	A ₀
1	0	0	0	0	1	1	X

地址范围: **83FCH~83FFH**

可以任选其中一个地址如
83FCH作为该接口地址

实现程序

```
MOV    DX, 83FCH
IN     AL, DX
AND    AL, 0FFH
JZ     NEXT1
JMP    NEXT2
```





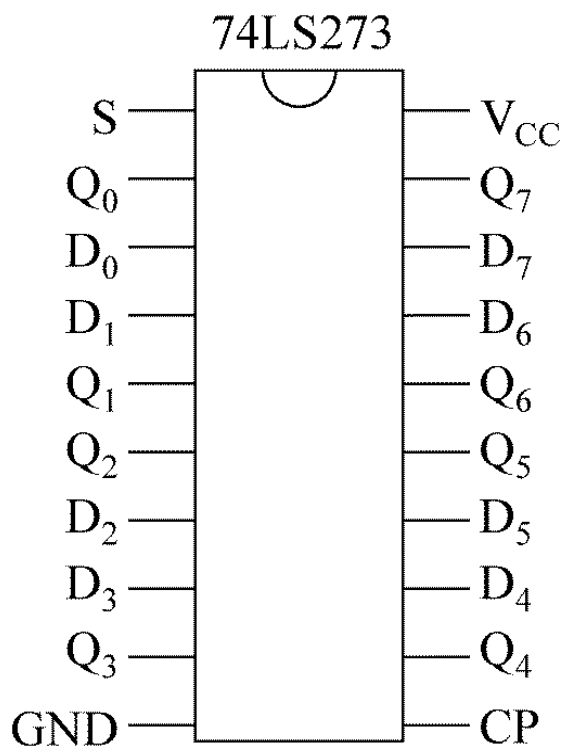
二、锁存器接口

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

常用锁存器芯片

■ 74LS273

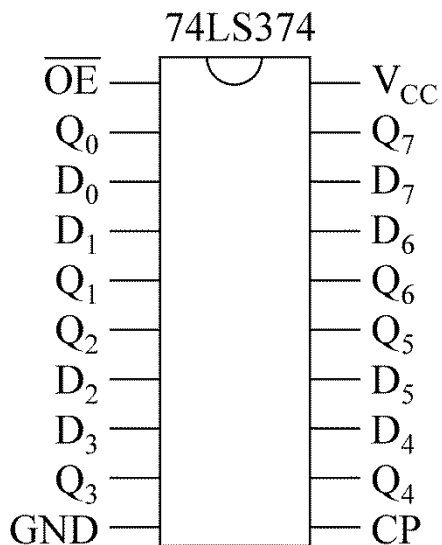
■ 8D触发器



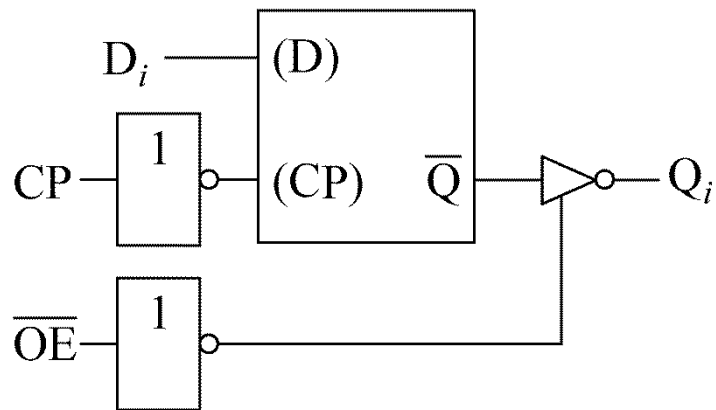
S	CP	D_i	Q_i
0	X	X	0
1	↑	1	1
1	↑	0	0

■ 74LS373和74LS374

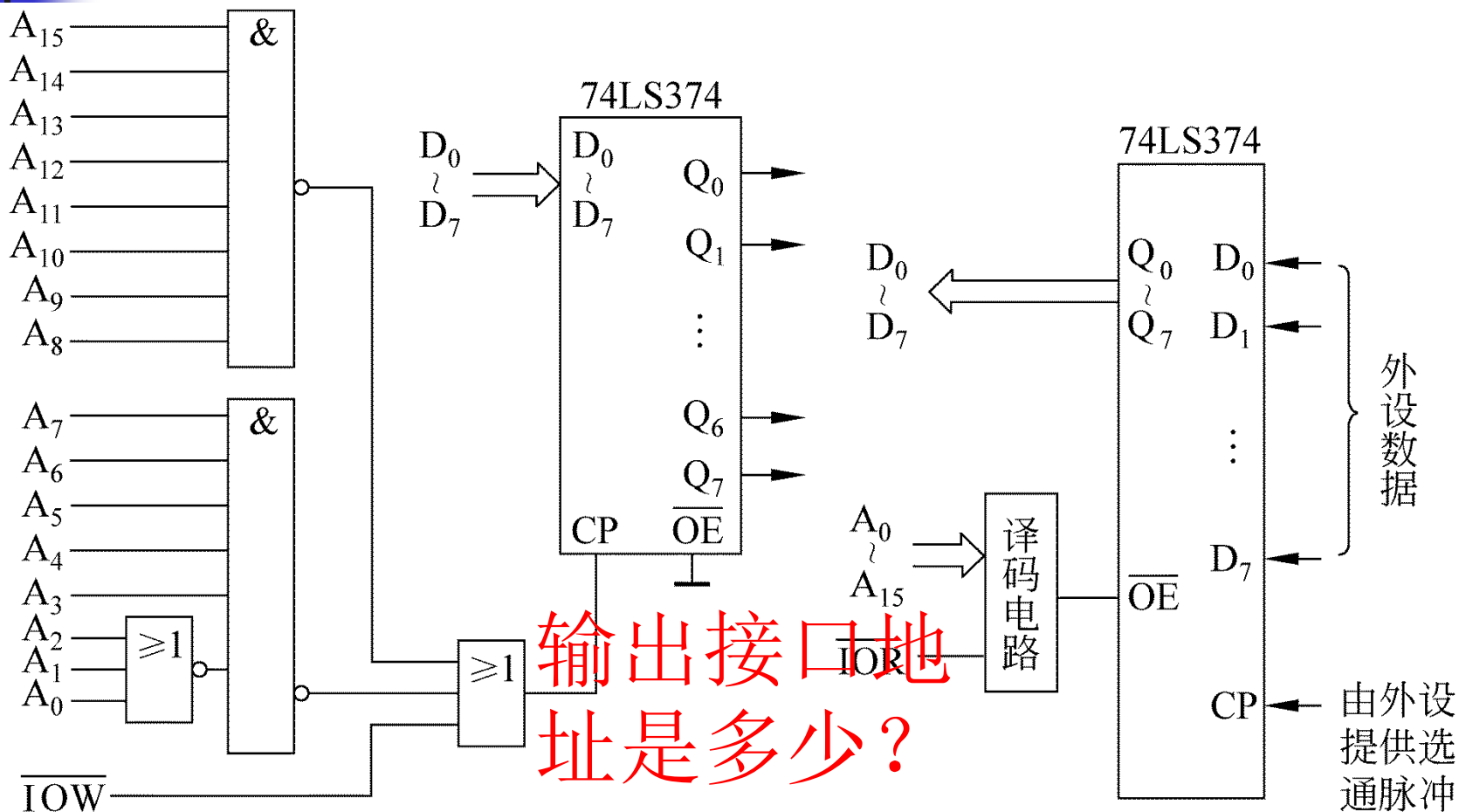
- 三态输出的8D触发器，并具有对数据的输出控制能力。
- 既可以做输入接口，也可以做输出接口。



D _i	CP	$\overline{\text{OE}}$	Q _i
1	↑	0	1
0	↑	0	0
X	X	1	高阻



74LS374作输入输出接口示例

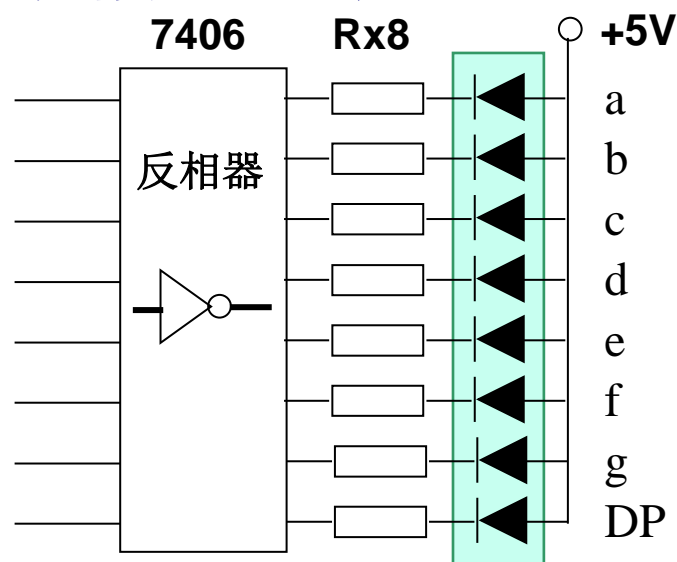
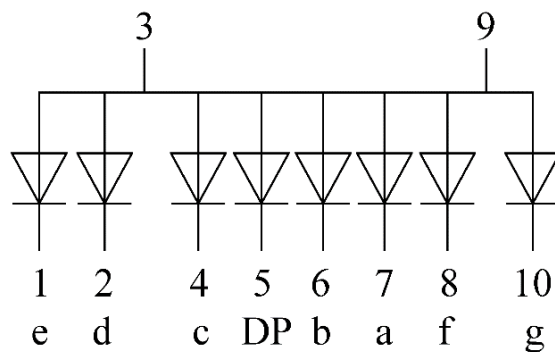
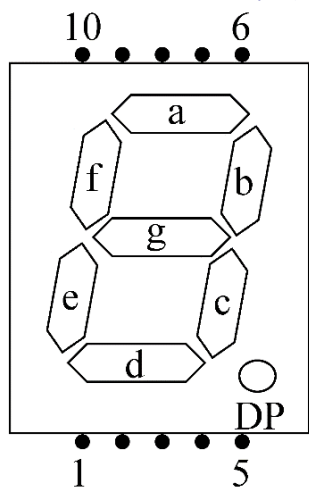


(a) 74LS374 用作输出接口

(b) 74LS374 用作输入接口

I/O接口综合应用例

















- 根据4个开关的状态控制7段数码管的显示
- 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'
- 设显示接口的地址为F0H，开关接口地址为F1H



共阳极数码管结构示意图

数码管驱动电路

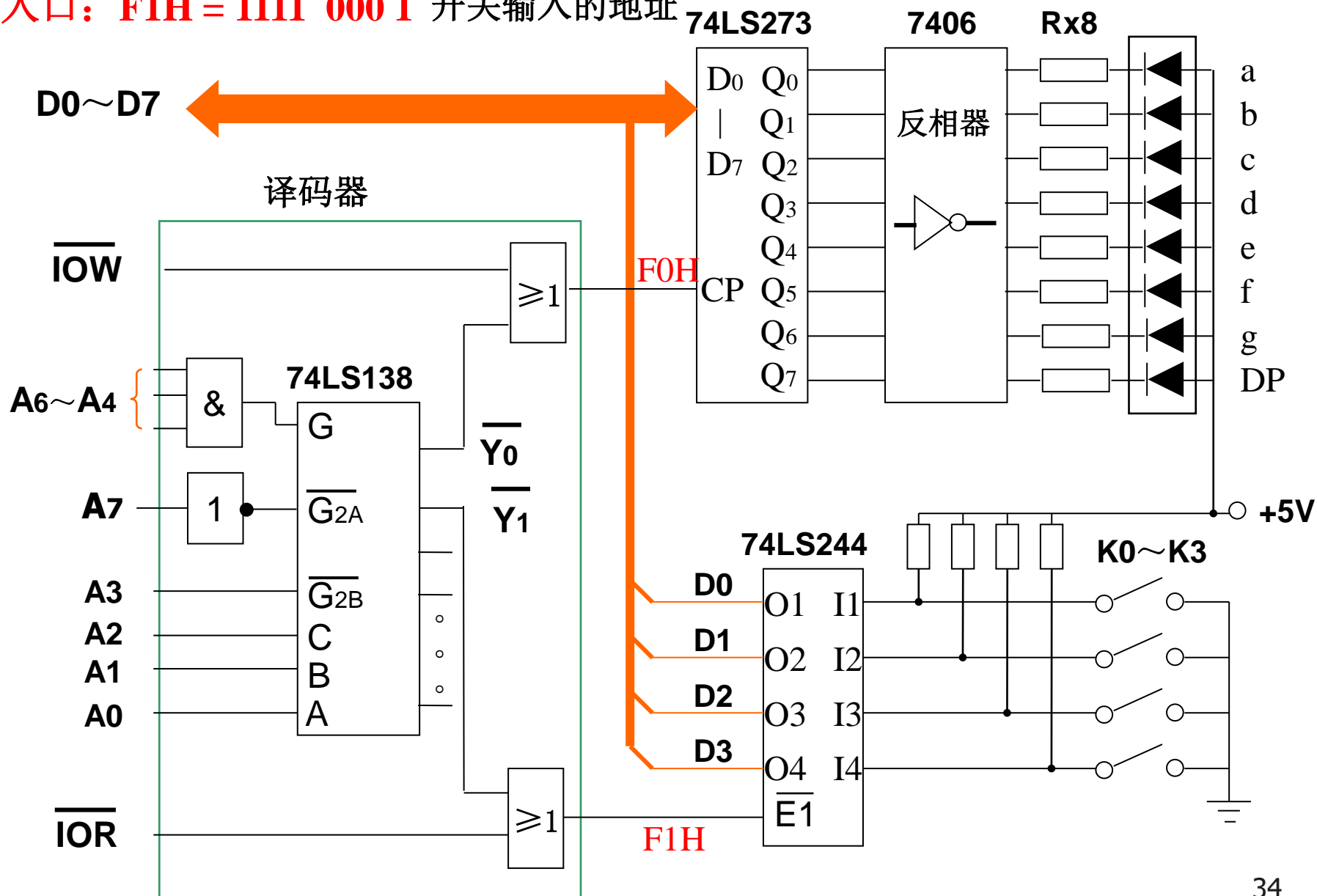
显示符号与输出数据对应表

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

A7.....A1A0

输出口: **F0H = 1111 000 0** 显示输出地址

输入口: **F1H = 1111 000 1** 开关输入的地址



I/O接口综合应用例 —— 程序段

.DATA

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

.CODE

**LEA EBX, Seg7
XOR EAX, EAX**

**GO: IN AL, 0F1H
AND AL, 0FH
MOV ESI, EAX
MOV AL, [EBX+ESI]
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

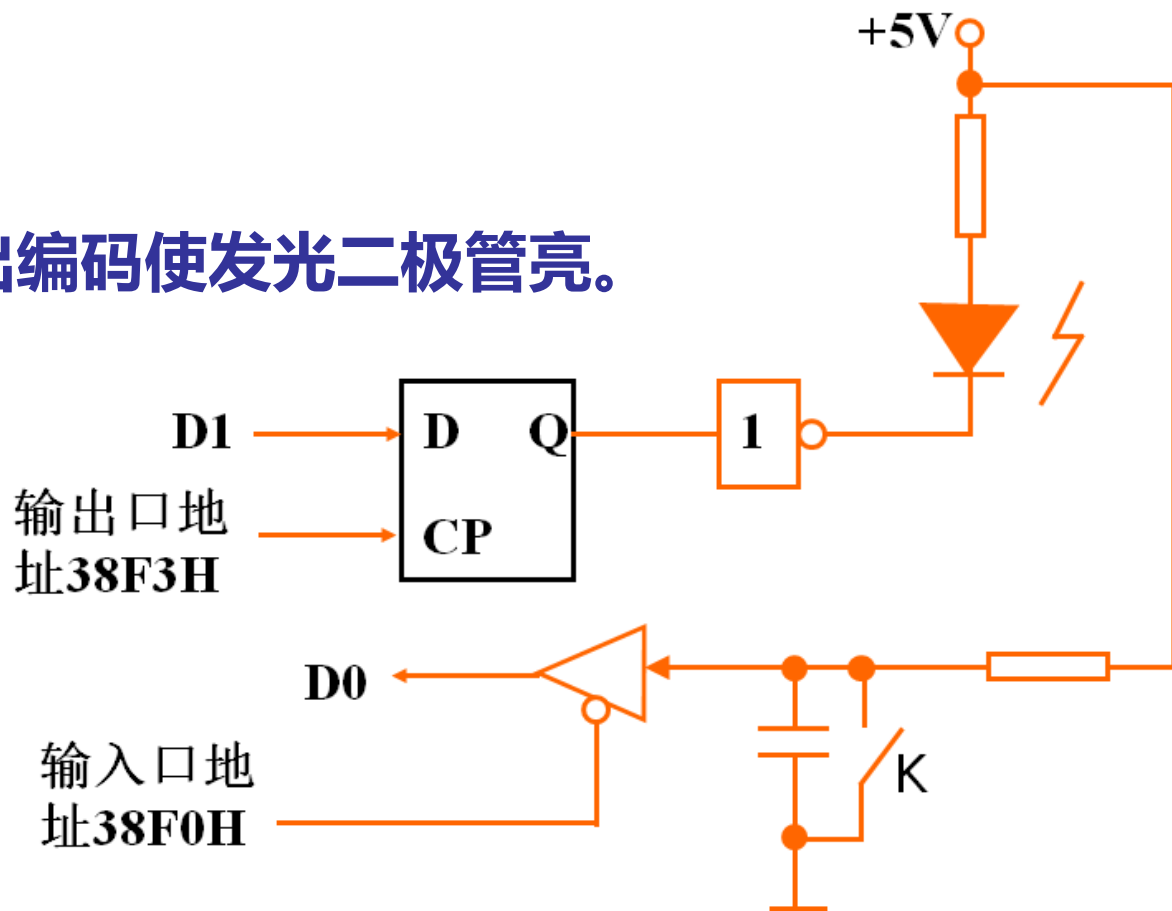


一、无条件传送

- 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k的状态;
- 当开关闭合时, 输出编码使发光二极管亮。

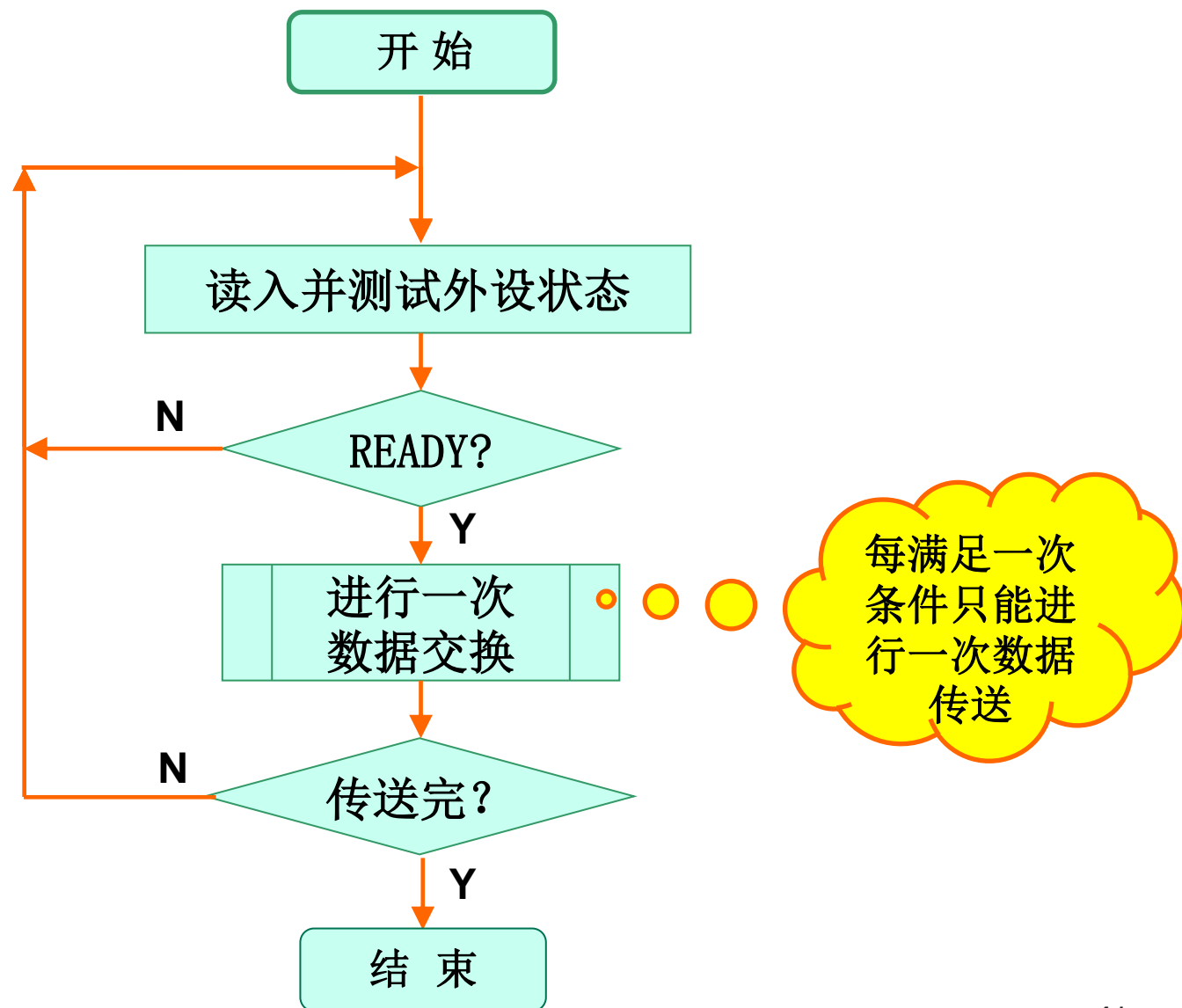


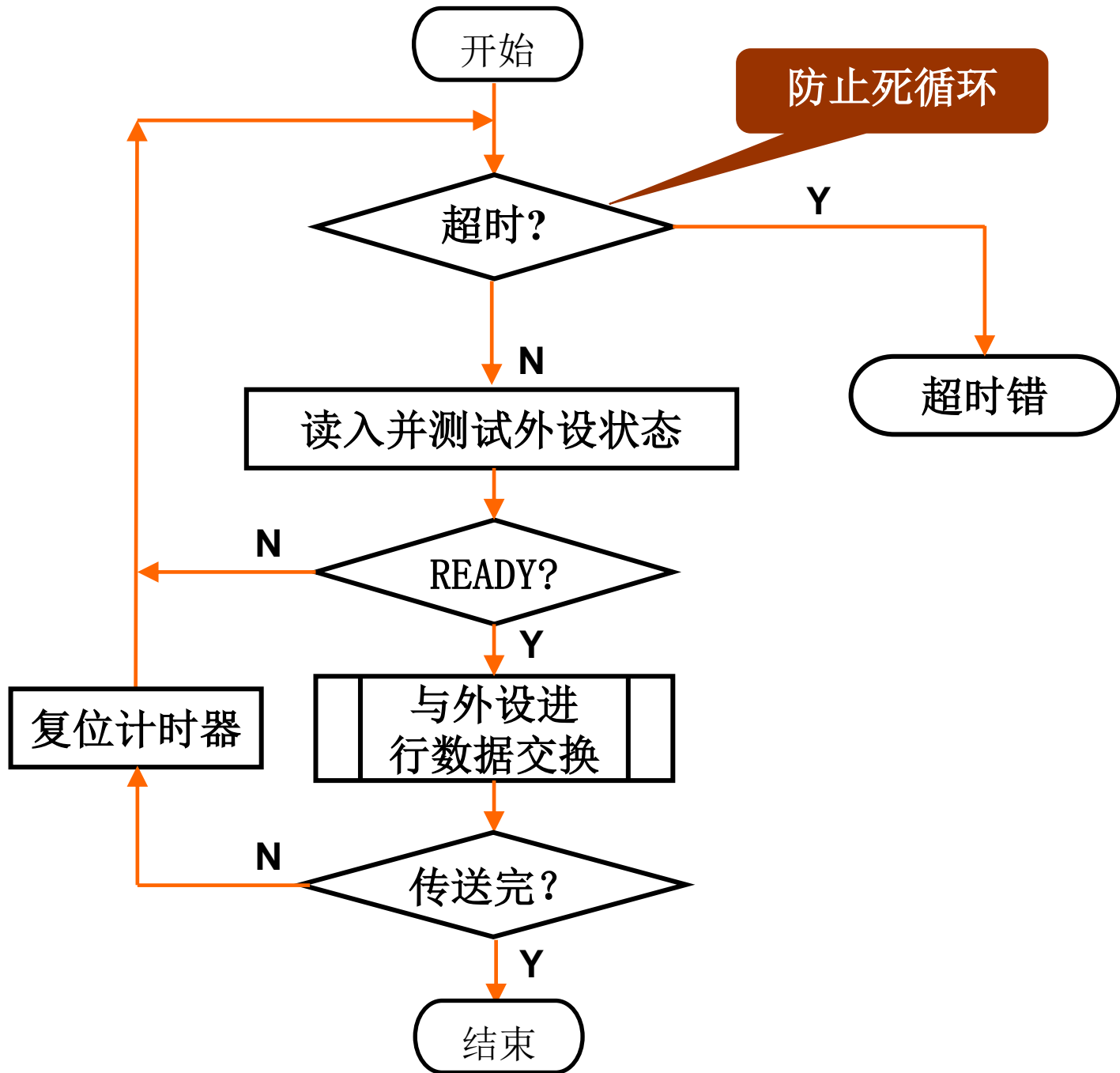


二、查询工作方式

- 仅当条件满足时才能进行数据传送;
- 每满足一次条件一般只进行一次数据传送。
- 适用场合:
 - 外设并不总是处于“准备好”状态
 - 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

单个外设的查询工作方式流程图





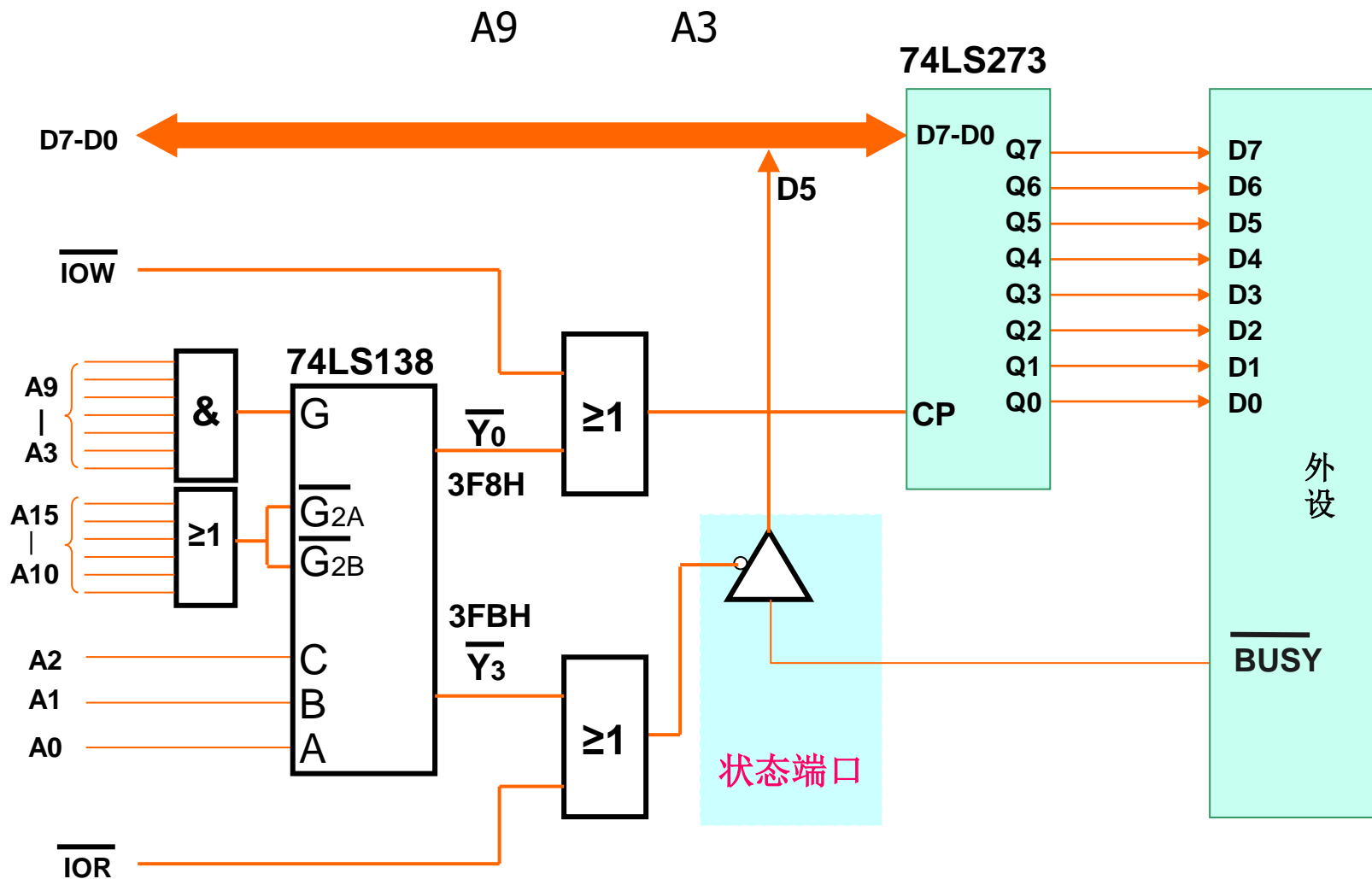


查询工作方式的例子—数据输出

- 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，CPU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，并将内存BUF中的100个字节数据输出。

状态端口地址: **0000 0011 1111 1011** **3FBH**

数据端口地址: **0000 0011 1111 1000** **3F8H**



控制程序

.....

LEA ESI,BUF

MOV ECX,100

AGAIN : MOV DX,03FBH

WAITT: IN AL,DX

TEST AL,20H

JNZ WAITT

MOV DX,03F8H

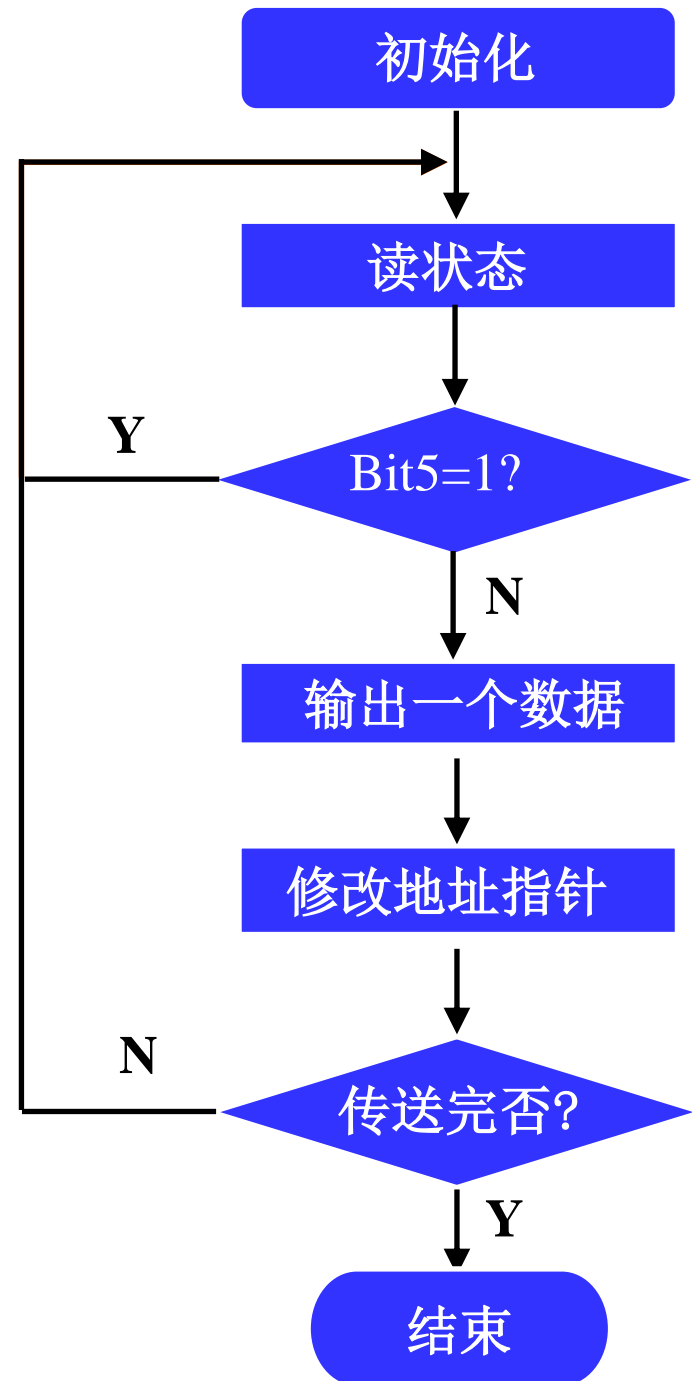
MOV AL, [ESI]

OUT DX, AL

INC ESI

LOOP AGAIN

.....



多个外设时 查询工作方式

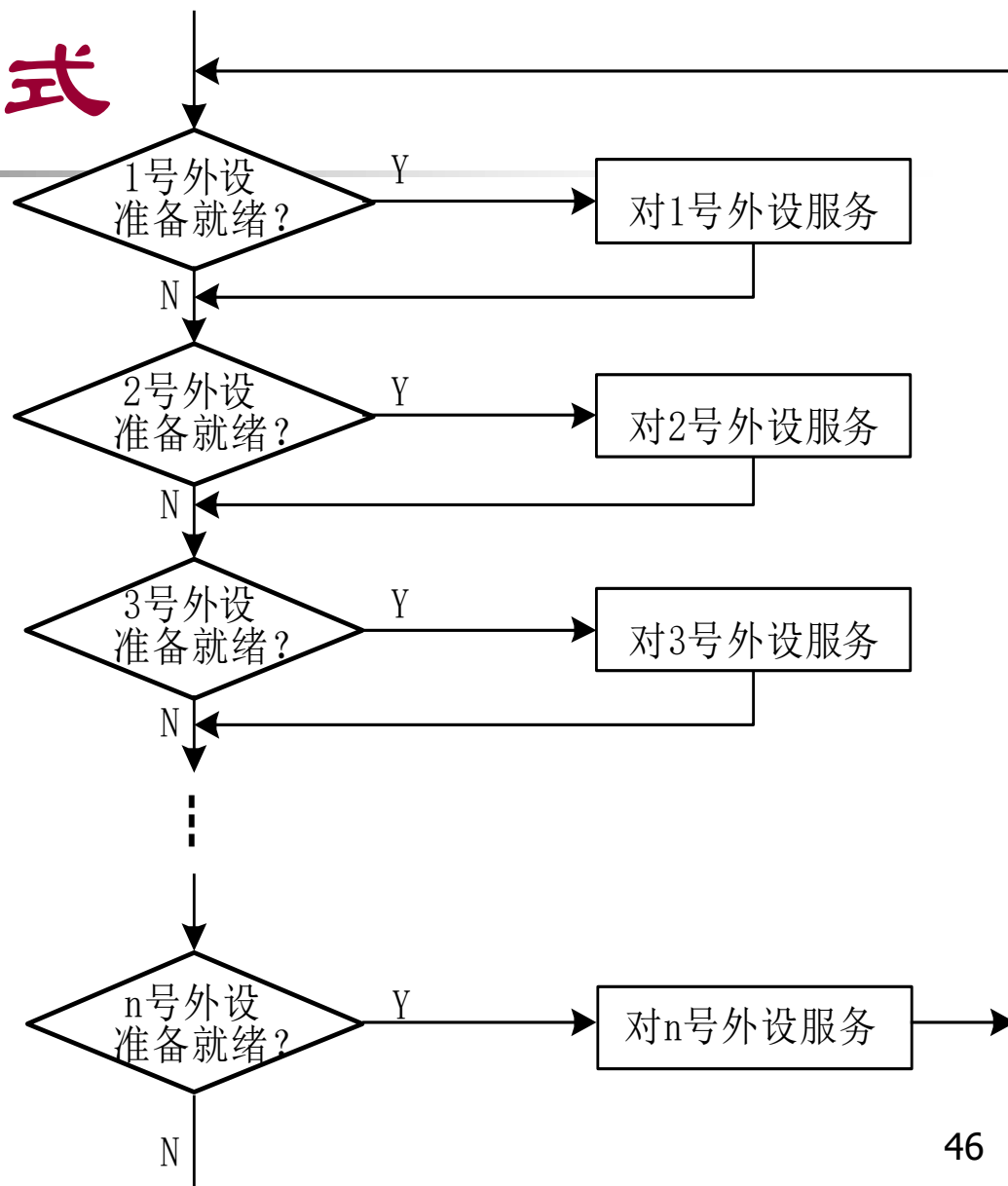
一般工作流程

■ 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





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，速度较快。

■ 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

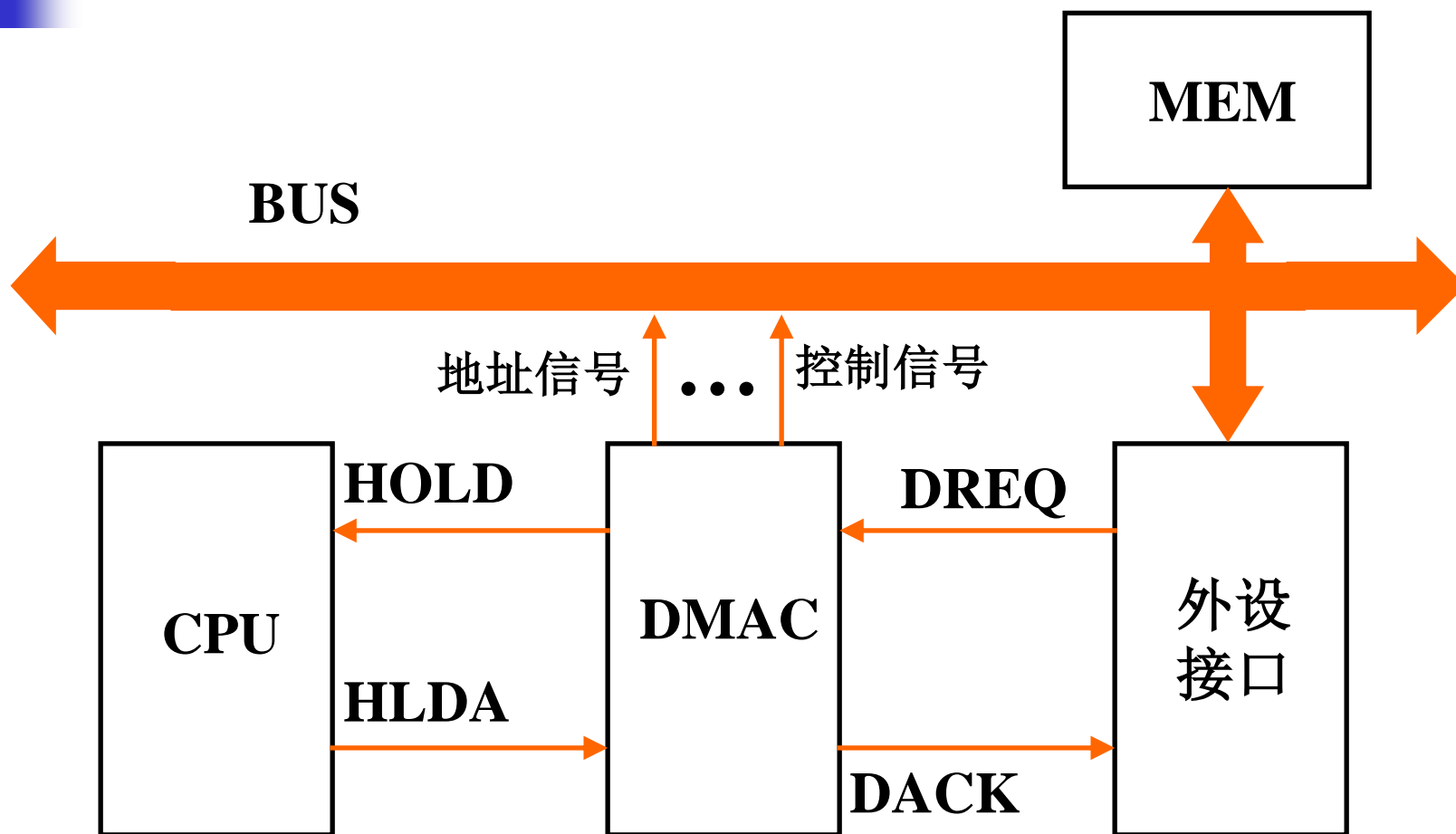


四、DMA控制方式

■ 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

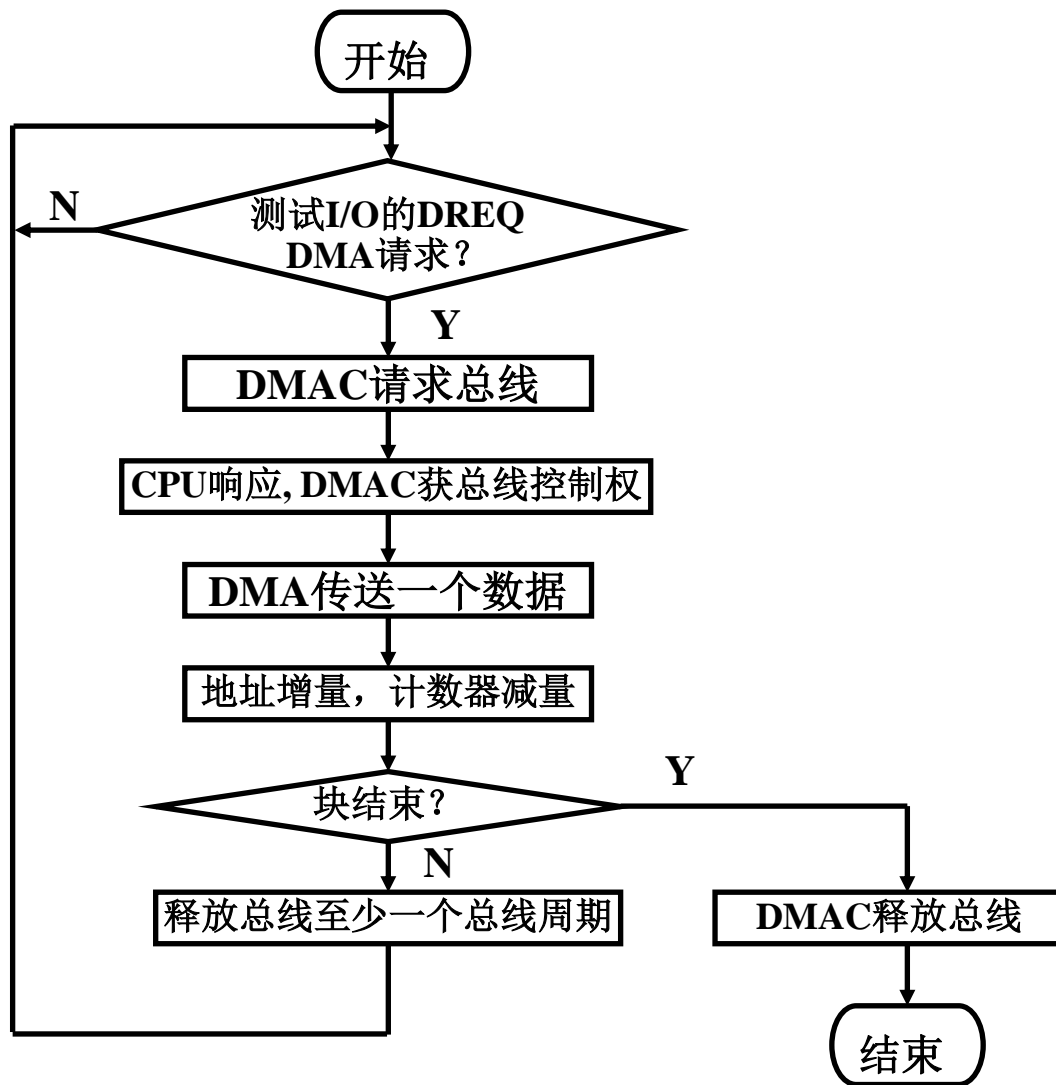
- **周期窃取：**

- **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

- **优点:**

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。

- **缺点:**

控制复杂，硬件成本相对较高。



§ 6.4 中断技术

主要内容

- ◆ 中断的基本概念
- ◆ 中断处理的一般过程
- ◆ **8088/8086**中断系统
- ◆ **IA-32**中断系统



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**
- **IA-32中断描述符表的构成及访问方法**



一、中断的基本概念

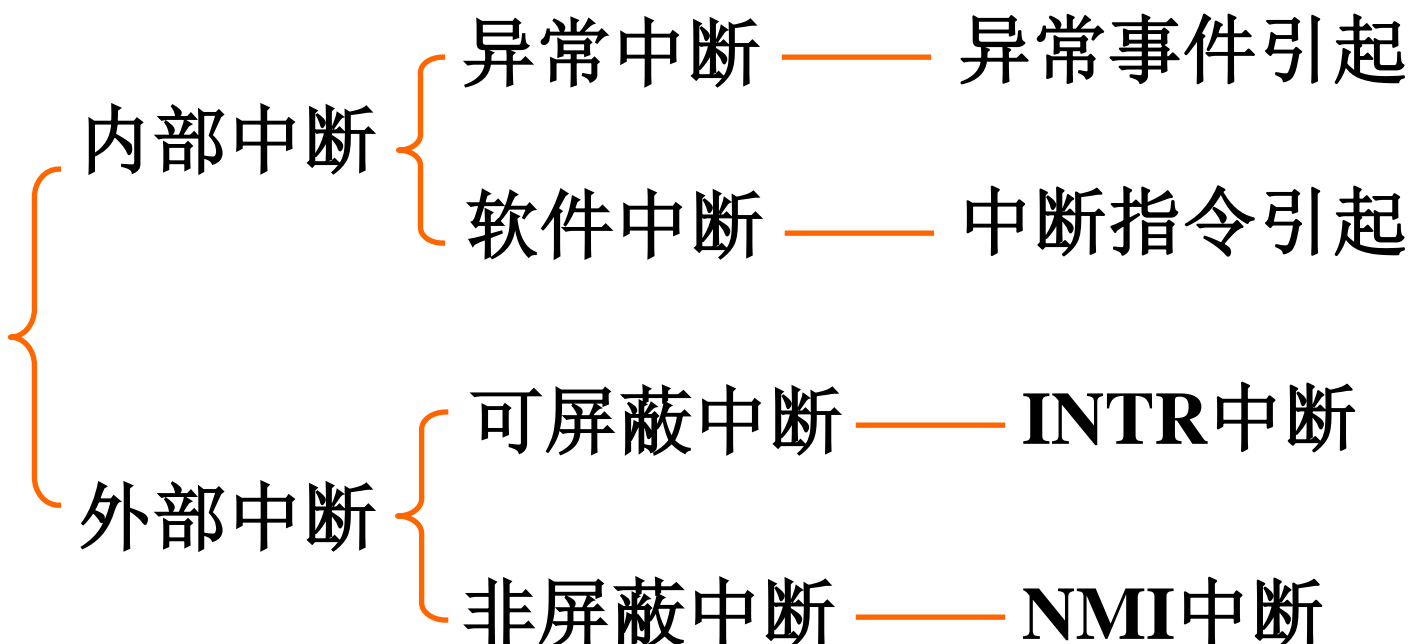
■ 中断:

- **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

包括 { 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—硬件识别**
 - **由中断源提供中断类型号，CPU根据类型号确定中断源。**
 - **中断源识别及中断判优（确定先响应哪个中断请求）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
 - 同时出现或等待的多个中断源，优先级最高的被响应。
 - 低优先级的中断程序允许被高优先级的中断源所中断
 - 也可以设置成禁止中断嵌套



中断嵌套



中断判优方法

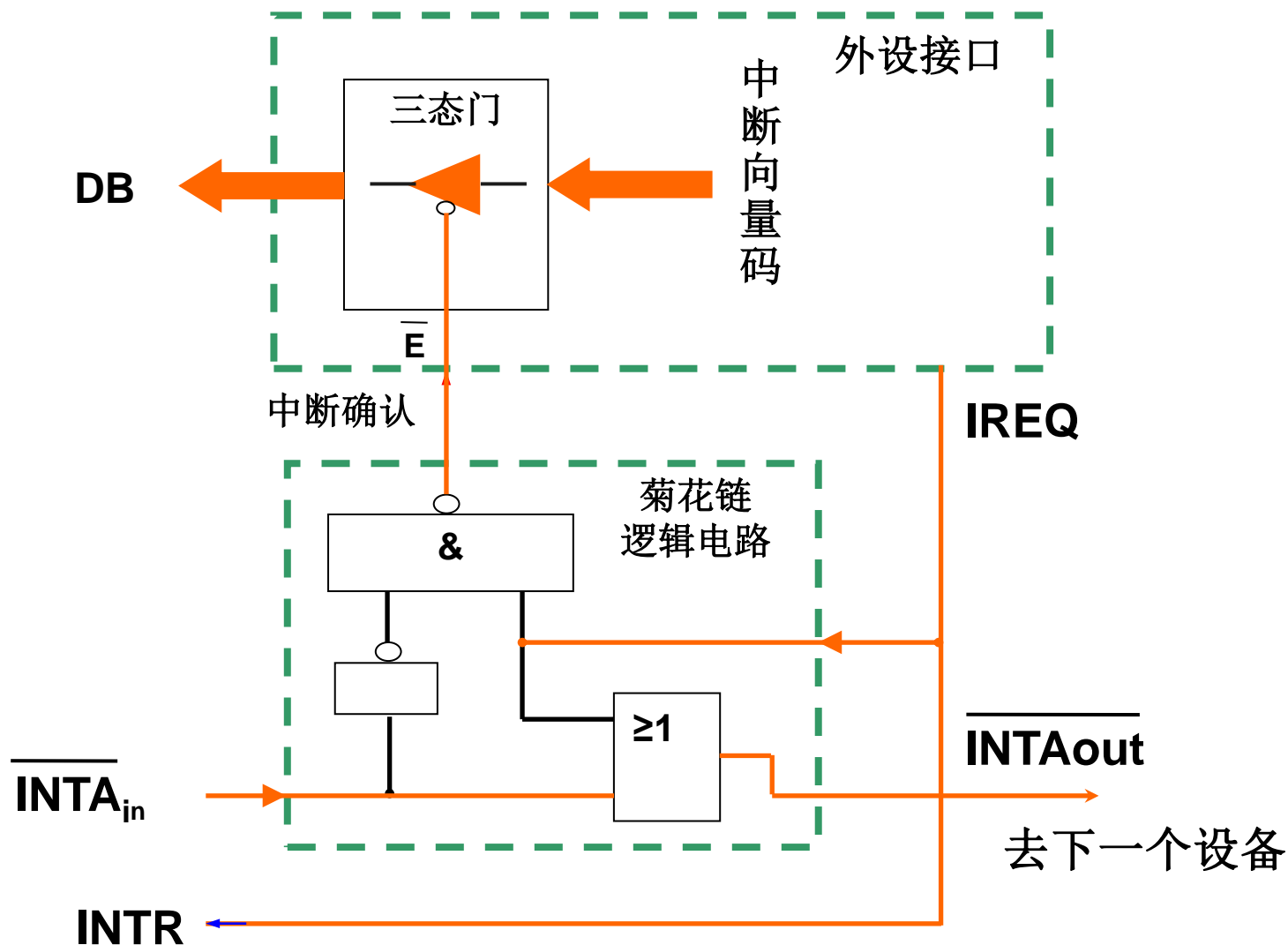
- **软件判优**

- **顺序查询中断请求，先查询的先服务**
 - **即先被查询的中断源优先级别高**

- **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

中断响应包含以下几个操作：

- 向中断源发出中断响应信号 \overline{INTA}
- 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

- **执行中断服务程序**
- **中断服务程序的特点：**
 - **中断服务程序要定义为“远过程”**
 - **结束时要使用IRET指令返回**



中断服务程序完成的工作

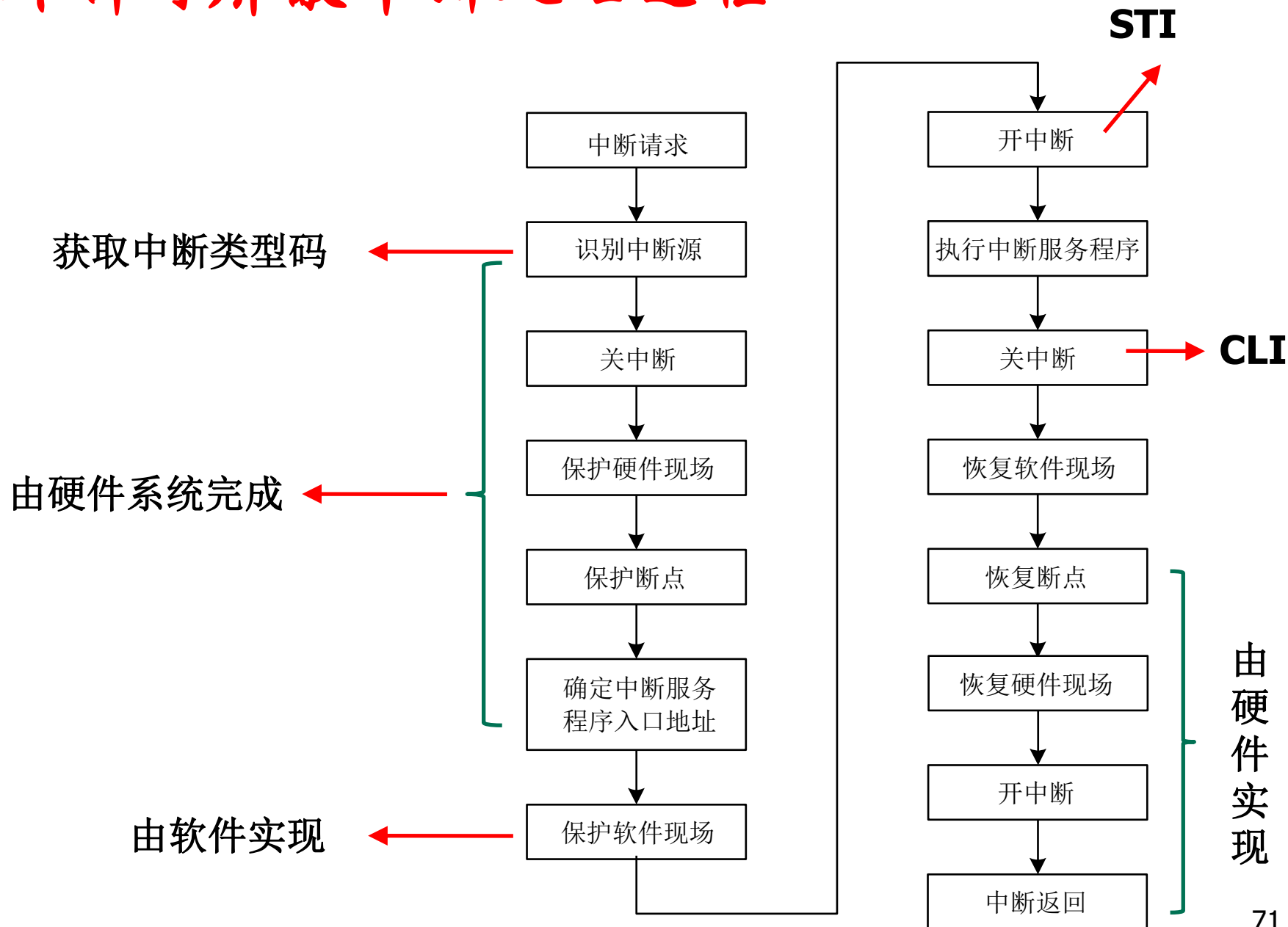
- 在前面的中断响应阶段：关中断，保护硬件现场，保护断点
- 保护软件现场（参数）
- 开中断（STI）--允许中断嵌套
- 中断处理—具体的处理
- 关中断（CLI）
- 恢复软件现场
- 中断返回-执行IRET



中断返回

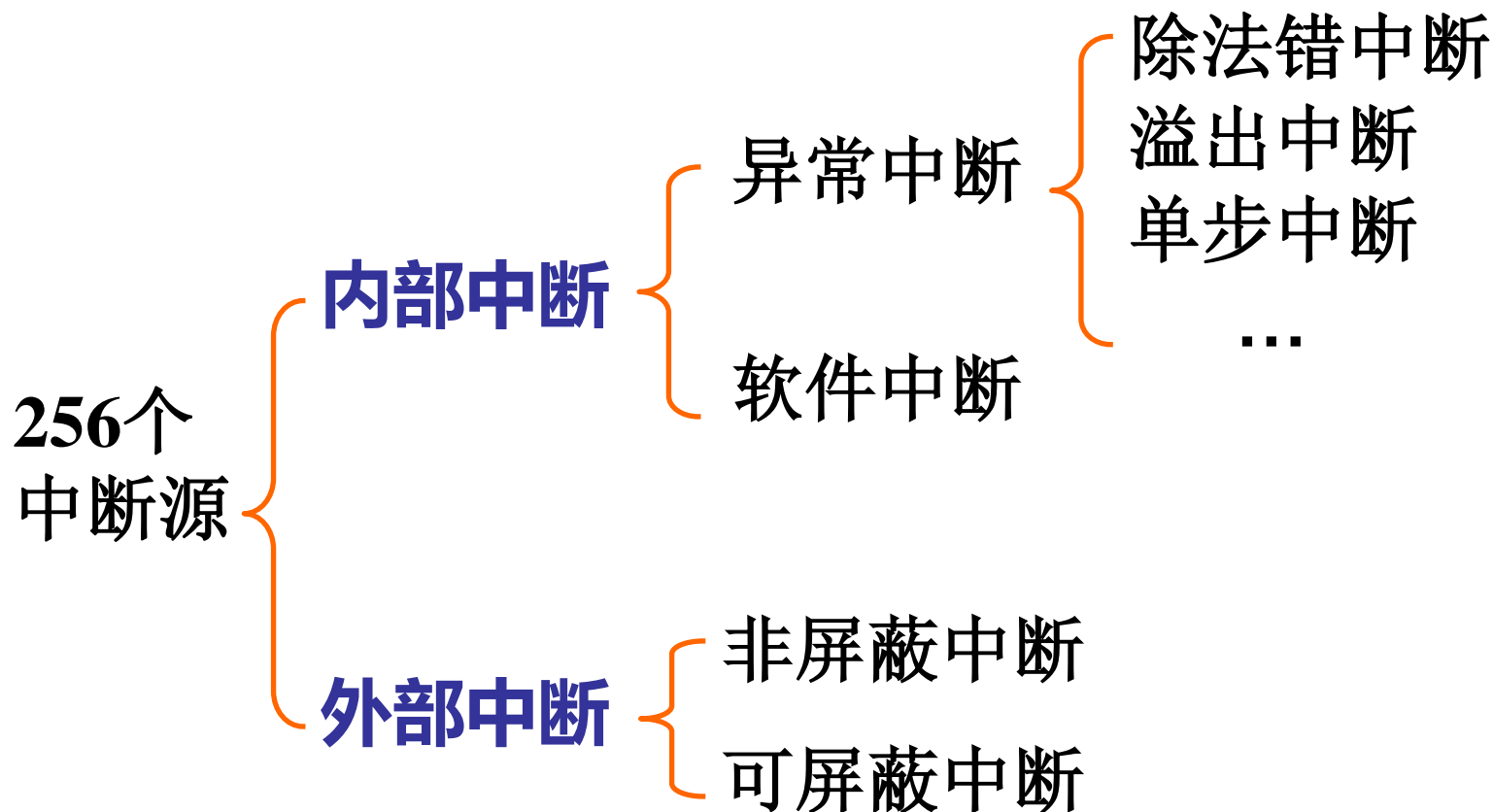
- 执行IRET指令，包括下面的操作：
 - 使IP、CS和FLAGS从堆栈弹出
 - 开中断
- 恢复断点和硬件现场
- 中断返回后才能响应中断请求

外部可屏蔽中断处理过程



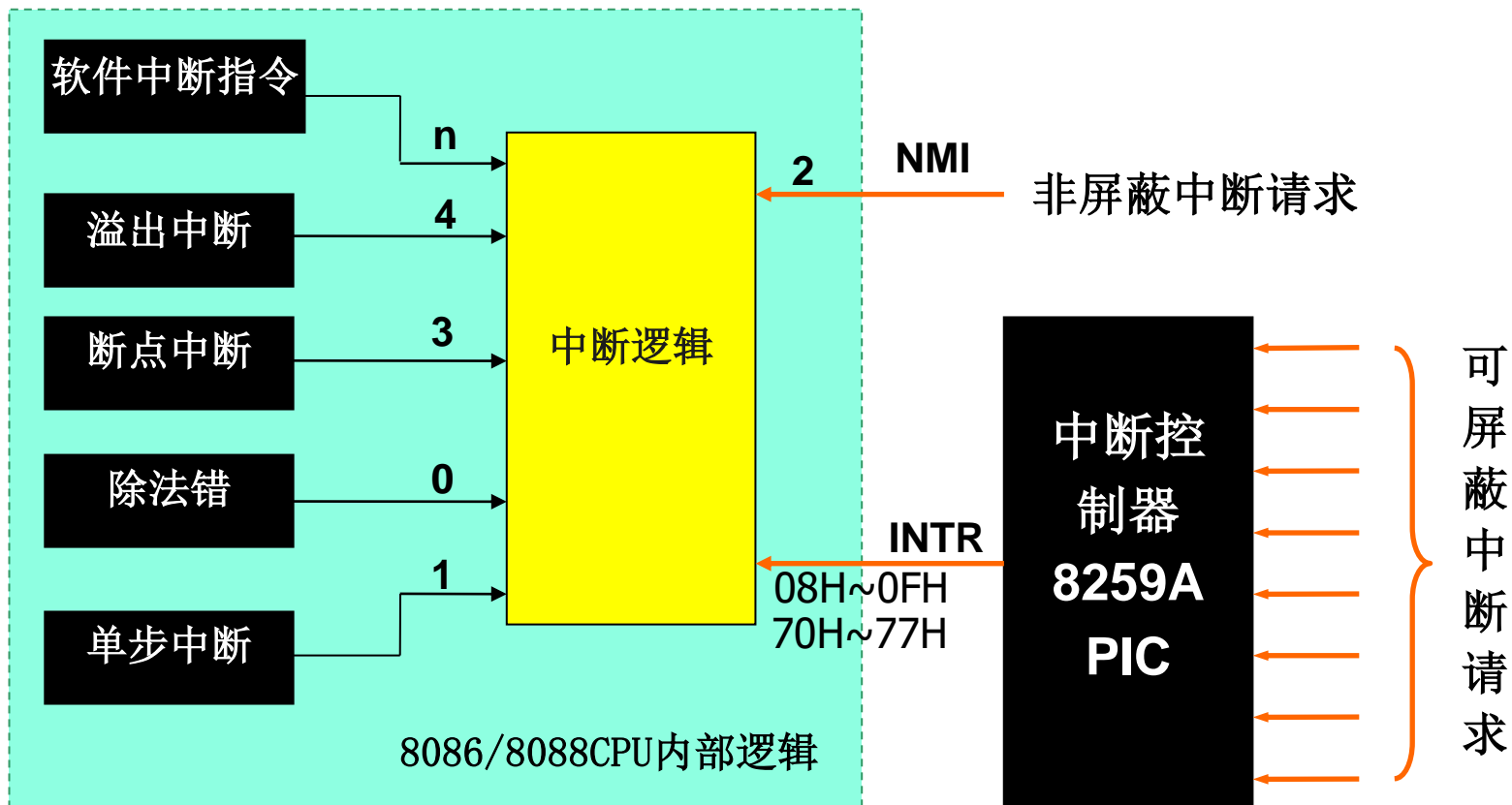


三、8086或实模式的中断系统



中断源的类型号

所有的中断源都统一分配了不同的类型号





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10H	显示设备中断
类型20H	程序结束中断
类型21H	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，即不可以屏蔽。
- 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

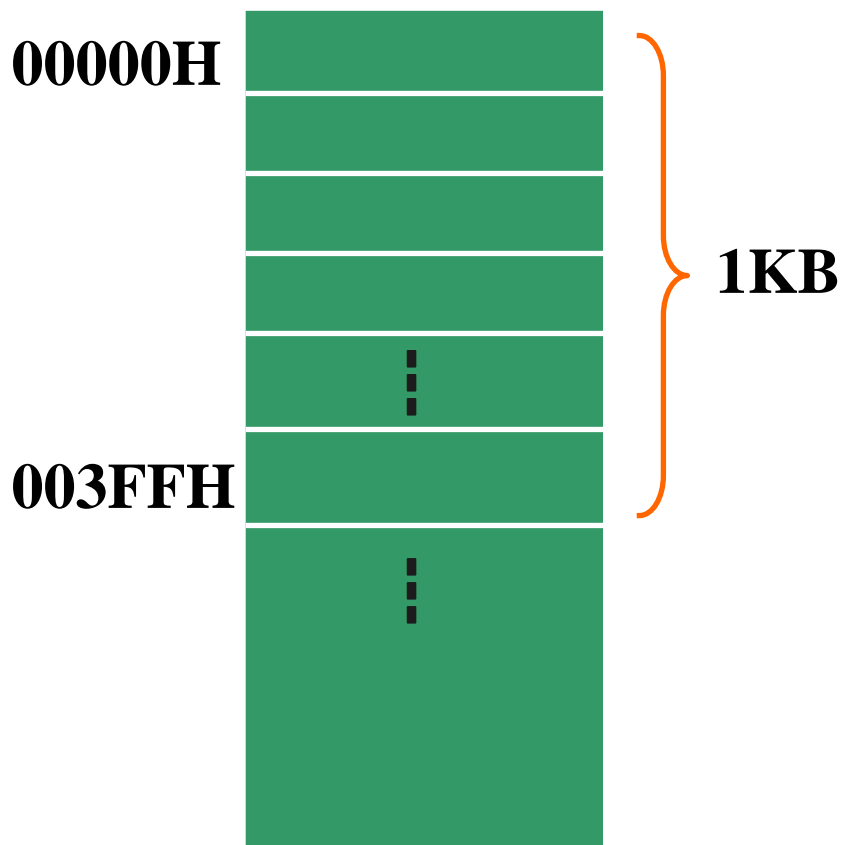


中断向量表IVT

- 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。



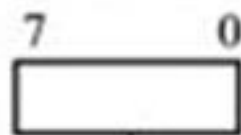
中断向量表位于内存最低**1KB**



内存空间		
00000H	IP偏移地址	0型中断向量
	CS段基地址	
00004H	IP偏移地址	1型中断向量
	CS段基地址	
00008H	IP偏移地址	2型中断向量
	CS段基地址	
0000CH	IP偏移地址	3型中断向量
	CS段基地址	
00010H	IP偏移地址	4型中断向量
	CS段基地址	
00014H	⋮	⋮
00080H	IP偏移地址	类型码为32 的中断向量
	CS段基地址	
	⋮	⋮
003FFH	IP偏移地址	类型码为255 的中断向量
	CS段基地址	

访问IVT的过程

中断类型号



×4

00000

IVT

段: 偏移

003FF

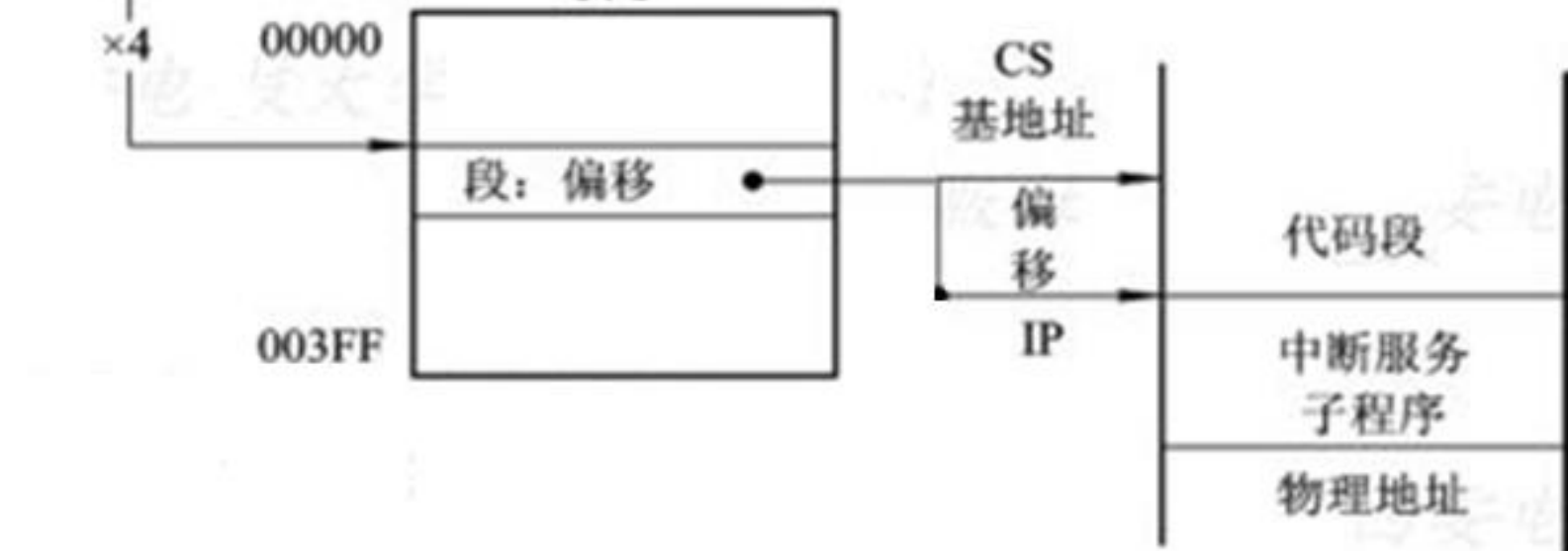
CS
基地址

偏移
IP

代码段

中断服务
子程序

物理地址



中断向量表的初始化

- 系统启动时已经把默认的中断向量写入IVT
 - 用户需将自定义的中断服务程序入口地址放入向量表
 - 注意点:
 - 向量表所在的段基址=0
 - 存放中断服务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
- 教材中使用了25H号DOS功能调用将服务程序入口地址写入向量表

旧p274

新p280



中断向量表的初始化

- 下面的程序用MOV指令将类型码为48H的中断服务程序TIMER的中断向量放入向量表

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

TIMER PROC FAR

.....

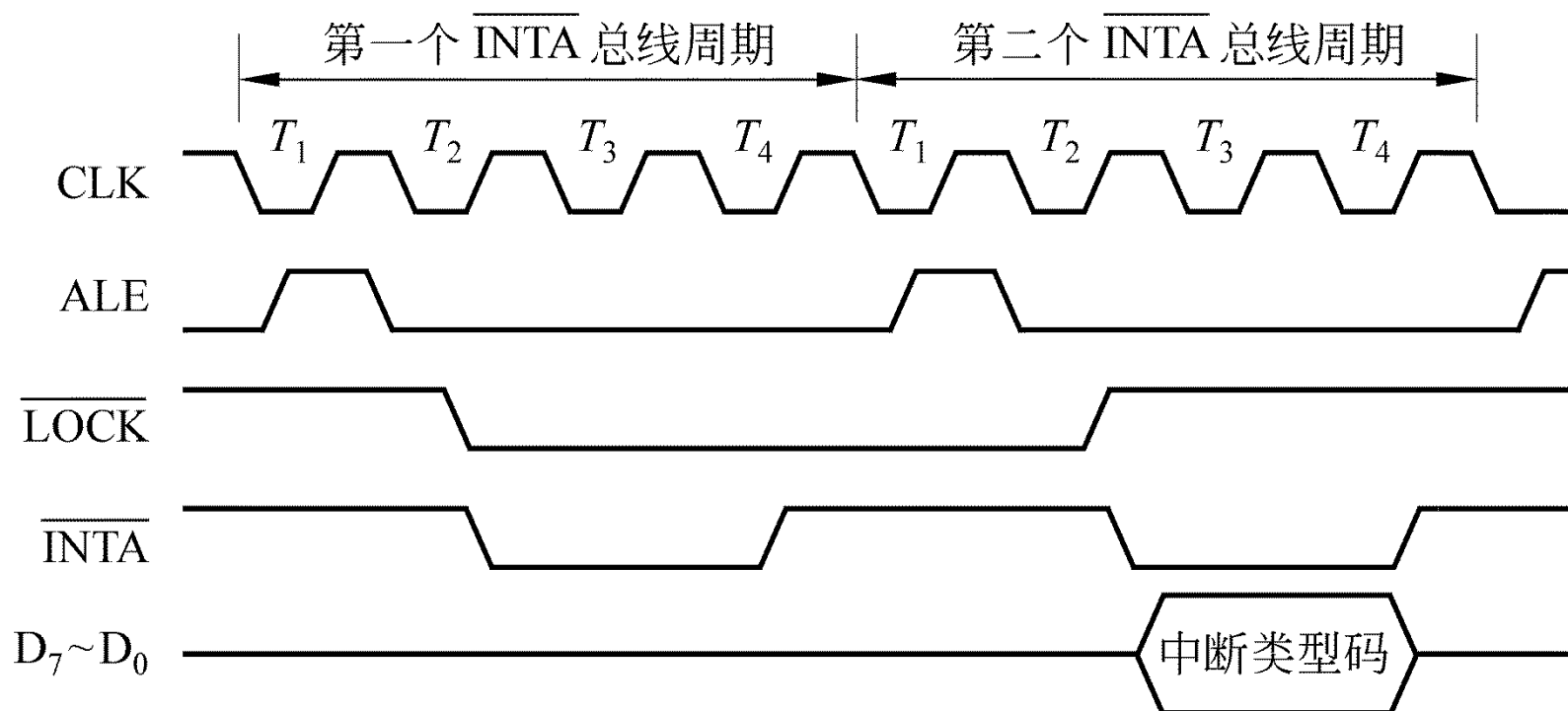
.....

IRET

TIMER ENDP

可屏蔽中断的类型号的获取时序

对可屏蔽中断的响应需要两个总线周期



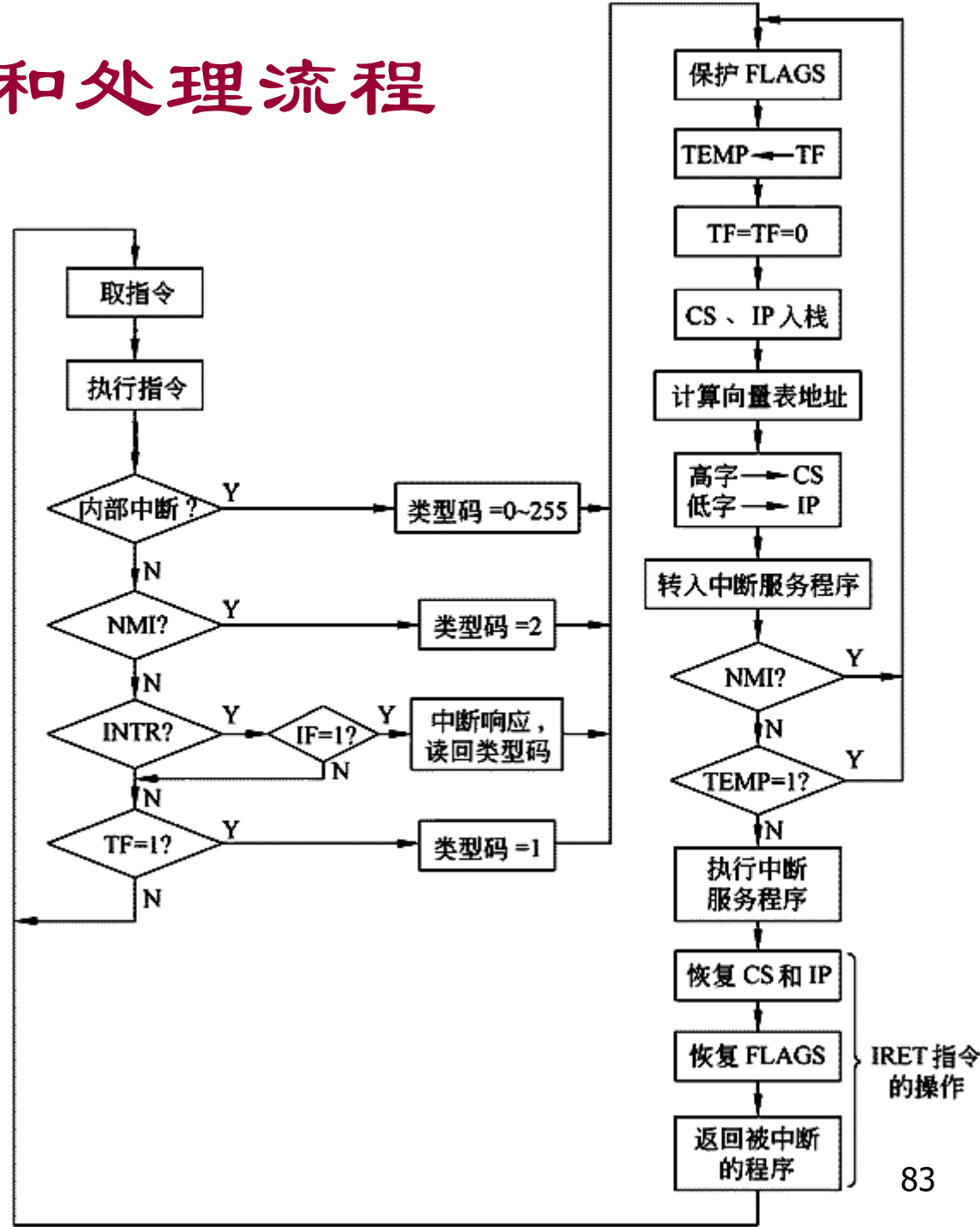


CPU内部中断与NMI中断

特点:

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

CPU的中断响应和处理流程



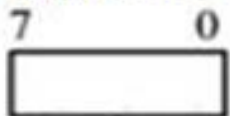
四、IA-32保护模式的中断系统

保护模式使用中断描述符表IDT获取中断服务程序入口地址

- 共有256个表项，对应256个中断类型号中的一个。每个表项占8字节，IDT表长2KB。
 - 每一表项内容为中断门（或陷阱门，或任务门）描述符。
 - IDT在内存的位置由中断描述符表寄存器IDTR指示。
- 以中断类型号乘以8作为访问IDT的偏移，读取相应的中断门（或陷阱门，任务门）描述符表项。
- 门描述符中16位的选择符被装入CS寄存器，32位偏移量装入EIP。
- 以CS寄存器内容去访问GDT或LDT才能得到段的基地址，EIP做偏移量找到中断服务程序入口。

访问IDT的过程

中断类型号



×8

IDTR

IDT

中断门/
自陷门

选择符

属性

偏移

CS

GDT/LDT

段描述符

基地址

属性

边界

基地址

偏移

EIP

代码段

中断服务
子程序

线性地址

