《微型计算机原理与接口技术》
第5版

第8章
中断和可编程中断控制器8259A

本章主要内容：

§8.1 中断

§8.2 8259A的工作原理

§8.3 8259A应用举例

§8.1 中断

8.1.1 中断概念和分类

8.1.2 中断的响应与处理过程

8.1.1 中断概念和分类

1.中断的定义和功能

- **中断定义：**

  计算机在执行正常程序过程中，暂时中止当前程序的运行，转到中断处理程序去处理临时发生的事件，处理完后又恢复原来程序的运行，这个过程称为中断(Interrupt)。

1. 中断的定义和功能

- **中断功能:**
- 使CPU和外设在部分时间内并行工作，大大提高CPU的利用率;
- 在实时控制系统中，现场数据可及时接收处理，避免丢失;
- 故障的处理，如电源掉电、奇偶校验错、运算中溢出错等;
- 利用中断指令，直接调用大量系统已编写好的中断服务程序，实现对硬件的控制。

2. 中断源和中断分类

- 引起中断的原因或能发出中断请求的来源称为中断源。
- 8086有两种中断源，中断分为两大类：

1）外部中断或硬件中断，从不可屏蔽中断引脚NMI和可屏蔽中断引脚INTR引入；

2）内部中断或软件中断，是为解决CPU运行过程中出现的一些意外事件或便于程序调试而设置的。

2. 中断源和中断分类

图8.1 IBM PC机中8086的中断分类和中断源

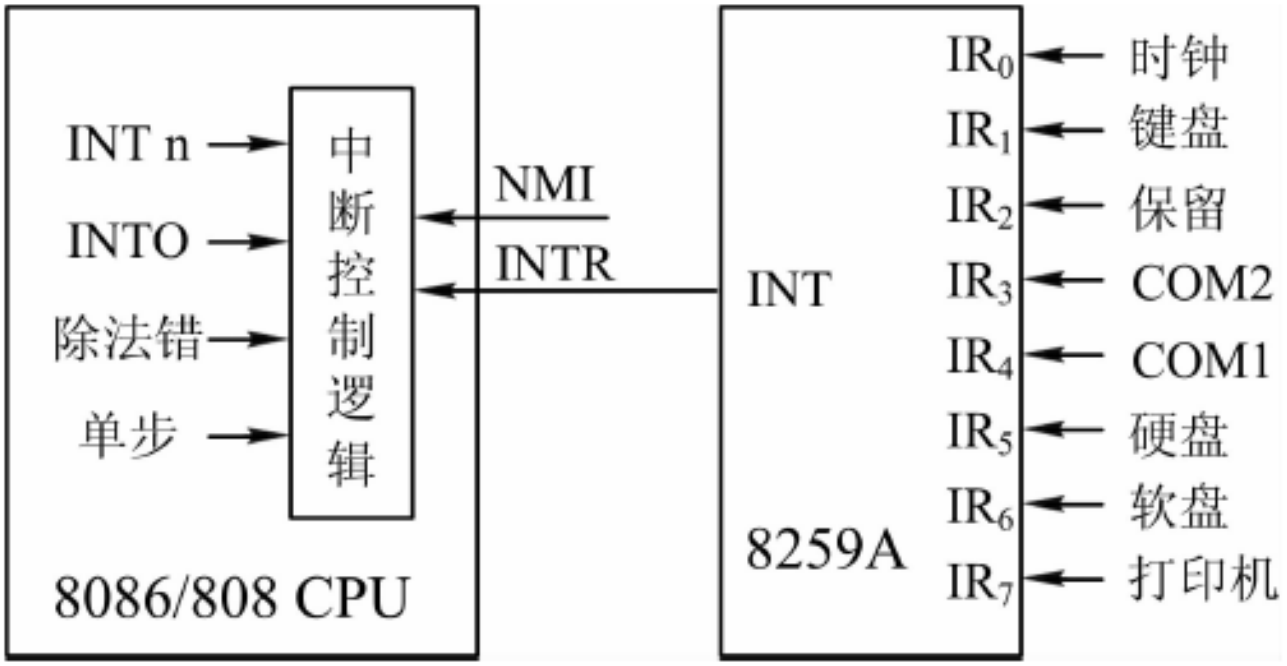


图 8.1 PC 机中 8086 的中断分类和中断源

2. 中断源和中断分类

1）外部中断

- 不可屏蔽中断NMI，处理较紧急的情况，如存储器或I/O校验错、掉电、协处理器异常中断请求等，不受中断标志IF的影响。
- 可屏蔽中断由8259A的INT引脚输出，连到CPU的INTR上。只有当CPU的FLAGS的IF=1时，才允许响应此类中断请求。
- 8259A的输入引脚IR_0~IR_7可引入8级中断：时钟、键盘、串行通信口COM1和COM2、硬盘、软盘、打印机。经芯片内部判别后，将优先级高的中断请求信号送到INT引脚。

2. 中断源和中断分类

2）内部中断

（1）除法错中断

- 执行除法运算指令时，如除数为0或商超过了结果寄存器能容纳的范围，则产生除法错中断。

（2）单步中断

- 单步标志TF置1，指令执行完后，产生单步中断。结果是将CPU的内部寄存器和有关存储器的内容显示出来，便于跟踪程序的执行过程，实现动态排错。

2. 中断源和中断分类

- 8086没有直接使TF标志置1或清0的指令，如何使TF标志置1或清0？使TF标志置1的程序段：

```
PUSHF                    ；标志寄存器FLAGS入栈
POP     AX               ；AX←FLAGS内容
OR      AX，0100H        ；使AX（即标志寄存器）的
                         ；D8=1，其余位不变
PUSH    AX               ；AX入栈
POPF                     ；FLAGS寄存器←AX
```

- 用类似方法将标志寄存器与FEFFH相与，可使TF标志清0，从而禁止单步中断。

2. 中断源和中断分类

（3）溢出中断

- 溢出标志=1，则可由溢出中断指令INTO产生中断类型号为4的溢出中断。若OF=0，不会产生中断。
- 带符号数加、减指令后应安排一条INTO指令，一旦溢出就能及时向CPU提出中断请求，CPU响应中断后可进行相应的处理。

2. 中断源和中断分类

（4）软件中断指令INT n

- 中断类型号n=0~255。它可以安排在程序的任何位置上。

（5）断点中断

- 程序运行到断点时便产生中断，像单步中断一样，查看各寄存器和有关存储单元的内容。
- 断点可设在程序任何地方并可以设多个断点，设置的方法是插入一条INT 3指令。利用断点中断可以调试一段程序，比单步中断的调试速度快得多。

3. 中断向量表

1）中断响应和返回

- CPU响应中断时，首先要把CS、IP寄存器的值（断点）以及标志寄存器FLAGS的值推入堆栈保护。
- 然后找到中断服务程序的入口地址，转去执行相应的中断服务程序。
- 中断服务程序结束时，执行中断返回指令IRET返回正常程序继续执行。
- 如何寻找中断服务程序的入口地址，是中断处理过程中的一个重要环节。

3. 中断向量表

2）中断向量表

- 中断向量表用来存放中断服务程序的入口地址。
- 8086可处理256（0~FFH）类中断，每类中断有一个入

部分中断号所对应的中断如下：

0型中断	除法错中断
1型中断	单步或陷阱中断
2型中断	非屏蔽硬件中断
3型中断	断点中断
4型中断	溢出中断
5型中断	屏幕打印
08H～0FH型中断	可屏蔽硬件中断
10H～1FH型中断	BIOS中断
20H～3FH型中断	DOS中断

3. 中断向量表


专用中断
(5个)
系统保留中断
(27个)
用户自定义中断
(224个)
00000H
00003H
00004H
00007H
00008H
0000BH
0000CH
0000FH
00010H
00013H
00014H
0007FH
00080H
003FCH
类型0中断入口地址
(除法出错)
类型1中断入口地址
(单步中断)
类型2中断入口地址
(NMI)
类型3中断入口地址
(断点中断)
类型4中断入口地址
(溢出中断)
类型5中断入口地址
类型31中断入口地址
类型32中断入口地址
类型255中断入口地址
IP内容
CS内容

3. 中断向量表

例如，对n=2的NMI中断，其中断服务程序的入口地址放在00008~0000BH单元中，入口地址的CS存放在0000AH开始的字单元中，IP存放在00008H开始的字单元中。

? 在PC机中，在8259A的中断输入端IR_0~IR_7引入的中断类型号为08~0FH，如何求它们的中断服务程序入口地址？存放在中断向量表中的什么位置上？

3. 中断向量表

- 举例说明中断类型号n与中断向量表的关系。

例8.1 类型号n=44H的中断服务程序的入口地址为3600：2000H，它们在中断向量表中应如何存放？

- 中断类型号n=44H=01000100B

  它的中断服务程序的入口地址应放在44H×4开始的4个字节单元中，乘4操作只要将类型号n左移2位，右边补2个0即可。

- 44H×4=01 0001 0000B= 0110H

  从0110H开始存放3600：2000H

如图8.3 ➔

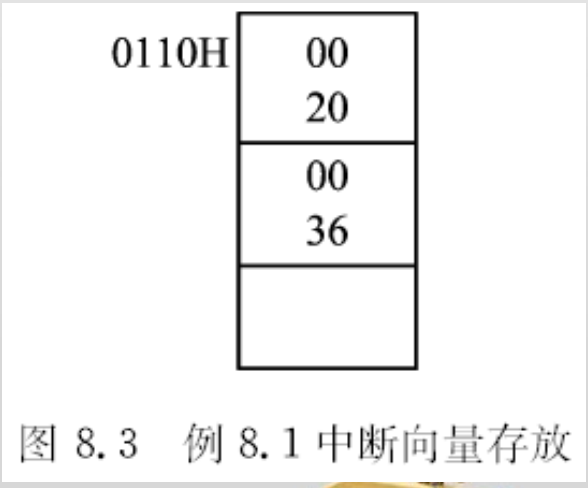


图 8.3　例 8.1 中断向量存放

3. 中断向量表

例8.2　若在中断向量表中，（0040H）= 240BH，而（0042H）= D169H，如图8.4所示，试问这4个单元中的内容对应的中断类型号n=？该中断服务程序的起始地址是什么？

- 中断服务程序的入口地址从0040H单元开始存放，其类型号n=40H/4=10H（右移2位）
- 由图可知，中断类型号为10H的中断服务程序的入口地址=D169：240BH。

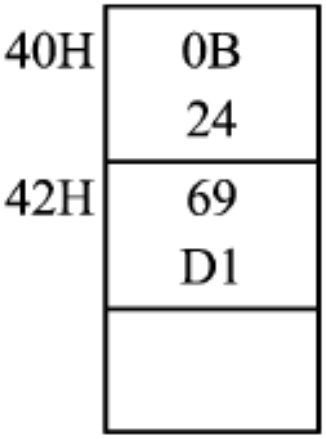


图 8.4 例 8.2 中断向量存放

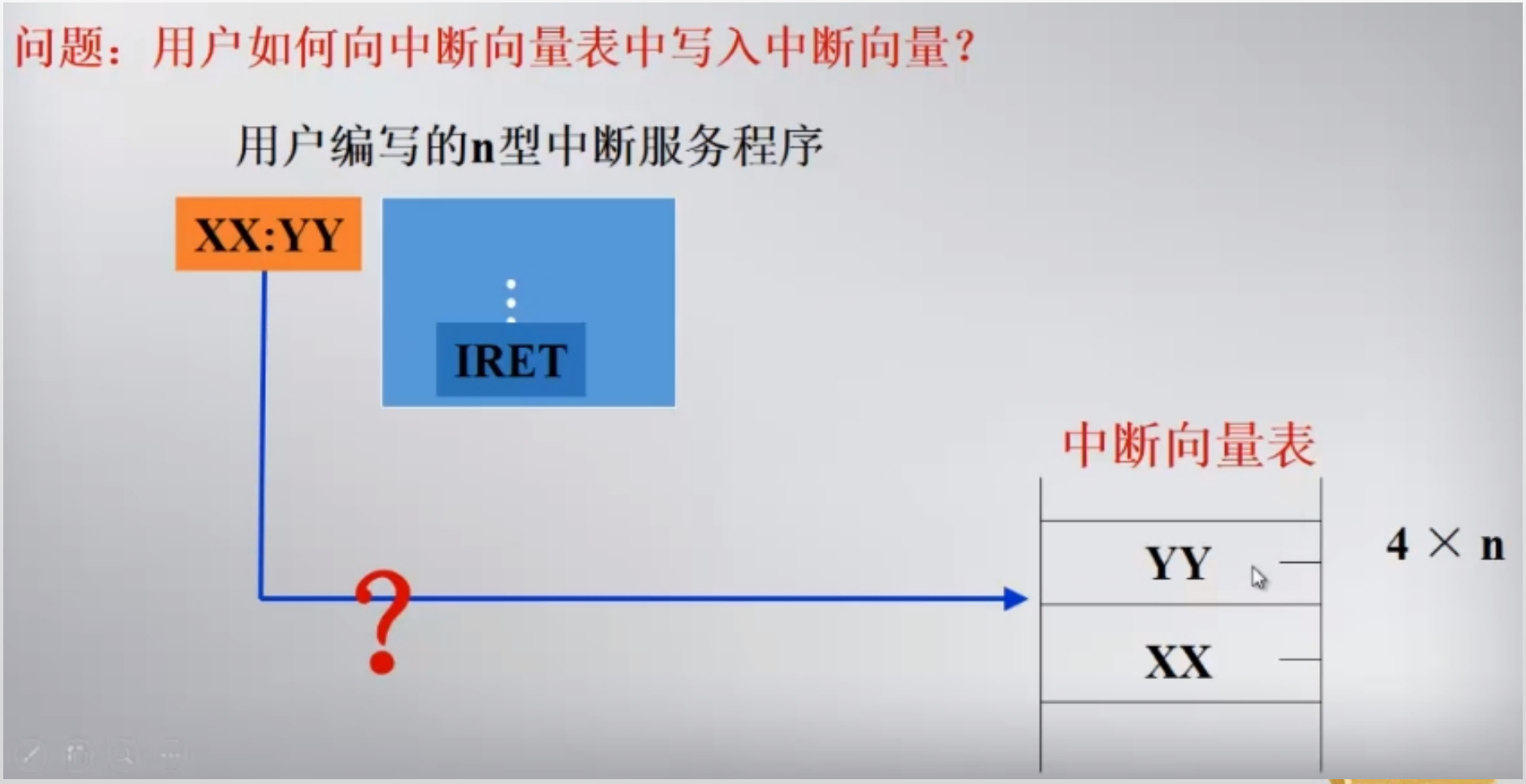
8.3.2 中断向量的设置和中断处理程序设计实例

- 中断向量的设置
- PC机对256类中断，有些已分配了固定功能，规定了中断服务程序的入口地址。
- 如类型号n=0~4为专用中断，n=5为打印屏幕中断，n=08~FH分配给8259A。
- 还有一些保留给用户使用，必须在中断向量表中建立相应的中断向量。常用以下两种方法设置中断向量。

中断向量表的初始化

- 由BIOS设计的中断服务程序，如INT 16H，INT 10H，其中断向量在加电时由BIOS负责写入中断向量表。
- 由DOS设计的中断服务程序，如INT 21H，其中断向量在加电时由DOS负责写入中断向量表。
- 用户程序开始的中断服务程序，由用户程序写入中断向量表。

问题：用户如何向中断向量表中写入中断向量？



1）用指令直接进行设置

- 这种方法利用MOV指令，直接将类型为N的中断服务程序的入口地址送到中断向量表的相应单元中去。
- 具体做法：将中断服务程序的入口地址的偏移地址，送到4×N开始的字单元中，并将其基地址送到（4×N+2）开始的字单元中。

例8.15　设中断服务程序的入口地址名为INTR_AD，类型号为N，要求将INTR_AD的CS:IP置入0000：（4×N）开始的单元中去，试编写汇编语言程序。

方法一，直接用MOV指令设置

```
        MOV  AX，0
        MOV  ES，AX                          ；目的地址基址
        MOV  BX，N*4                         ；目的地址偏移量
        MOV  AX，OFFSET  INTR_AD
        MOV  ES：[BX]，AX                    ；置入偏移地址
        MOV  AX，SEG  INTR_AD
        MOV  ES：[BX+2]，AX                  ；置入段基地址
           ┆
INTR_AD：                                    ；中断服务程序
           ┆
                IRET
```

- 设n型中断服务程序的名字是SERVICE,如何将SERVICE的入口地址写入对应的中断向量表，方法如下:

```
CLI
PUSH    DS
MOV     AX, 0000H
MOV     DS,AX
MOV     BX,4*n
MOV     AX,OFFSET SERVICE
MOV     [BX],AX
MOV     AX,SEG SERVICE
MOV     [BX+2],AX
POP     DS
STI
```

方法二，用字符串操作指令STOSW和MOV指令设置

```
   MOV  AX，0          ；目的地址基址为ES，其值为0
   MOV  ES，  AX
   MOV  DI，  N*4      ；DI←N×4，即目的地址偏移量
   MOV  AX，OFFSET  INTR_AD
                       ；AX←INTR_AD的偏移地址（IP）
   CLD                 ；方向标志清0
   STOSW               ；（ES：DI）←中断服务程序的IP
   MOV  AX，CS
   STOSW               ；后两个字节单元←中断服务程序的CS
     ┆
INTR_AD：              ；中断服务程序
    PUSH  AX           ；保护现场
    PUSH  BX
     ┆                 ；中断处理
    POP   BX           ；恢复现场
    POP   AX
    IRET               ；中断返回
```

2）利用DOS功能调用设置

- DOS功能调用专门提供了在中断向量表中设置和取得中断向量的手段，功能号为25H和35H。

设置中断向量

入口参数 DS：DX=中断向量（中断服务程序入口地址）
AL=中断类型号N
AH=25H（DOS功能号）
执行 INT 21H指令
结果 将AL中指定的中断类型号为N的中断向量（DS：DX）置入中断向量表中。

取得中断向量

入口参数 AL=中断类型号N
AH=35H（DOS功能号）
执行 INT 21H 指令
结果 N号中断的中断向量从中断向量表中取到ES：BX中

例8.16 利用DOS功能调用，编写设置和取得中断向量的程序段。程序段如下：

```
MOV     AL，N           ；中断类型号N
MOV     AH，35H
INT     21H             ；N号中断向量取到ES：BX
PUSH    ES              ；将原中断向量送堆栈保存
PUSH    BX
PUSH    DS              ；保存DS
MOV     AX，SEG  INTR_AD
MOV     DS，AX          ；DS←用户新中断向量段基址
MOV     DX，OFFSET  INTR_AD
                        ；DX←用户新中断向量偏移量
MOV     AL，N           ；新中断向量类型号
MOV     AH，25H
INT     21H             ；设置新中断向量
```

例8.16（续）

```
    POP     DS                  ；恢复DS
      ┆
    POP     DX                  ；恢复原中断向量
    POP     DS
    MOV     AL，N
    MOV     AH，25H
    INT     21H
    RET
INTR_AD
        ┆                       ；用户编写的中断服务程序
    IRET
```

例8.17 将例8.14中，中断类型号N=31H的中断向量1000：2000H，设置到中断向量表中。

程序如下：

```
MOV     AX，1000H
MOV     DS，AX              ；DS←段基地址
MOV     DX，2000H           ；DX←偏移地址
MOV     AL，31H             ；中断类型号N
MOV     AH，25H             ；DOS功能号
INT     21H                 ；设置中断向量量
```

- 例8.14中，中断类型号N=35H、44H、45H的中断向量也可用类似方法设置。

- 练习：编写中断服务程序实现在屏幕上显示字符串“This is a Interruption Service Program!”。设中断类型号取60H，采用DOS功能调用法置中断服务程序入口地址，通过软件中断指令INT 60H实现中断服务程序的调用。
- 程序设计如下：
- DATA SEGMENT
- MESG DB ‘This is a Interruption service Program!$’
- DATA ENDS

```
CODE SEGMENT
        ASSUME  CS:CODE, DS:DATA
START:
        MOV   AX,DATA
        MOV   DS,AX
        PUSH   DS
        MOV   DX,OFFSET  DISP60
        MOV   AX,SEG  DISP60
        MOV   DS,AX
        MOV   AH,25H
        MOV   AL,60H
        INT    21H
        POP     DS
```

```
INT   60H
MOV AH,4CH
INT 21H
DISP60   PROC FAR
        MOV DX,OFFSET MESG
        MOV AH,09H
        INT 21H
        IRET
DISP60   ENDP
CODE ENDS
        END START
```

4. 中断优先级和中断嵌套

1）中断优先级

- 按中断源的重要性和实时性，排出响应中断的次序，这种次序称为中断优先级。
- 在8086中，中断优先级从高到低的次序为：

- 除法错、INT n、INTO ；最高级，同一行的；有同等优先级
- NMI ；次高级
- INTR ；较低级（由8259A引入）
- 单步中断 ；最低级

4. 中断优先级和中断嵌套

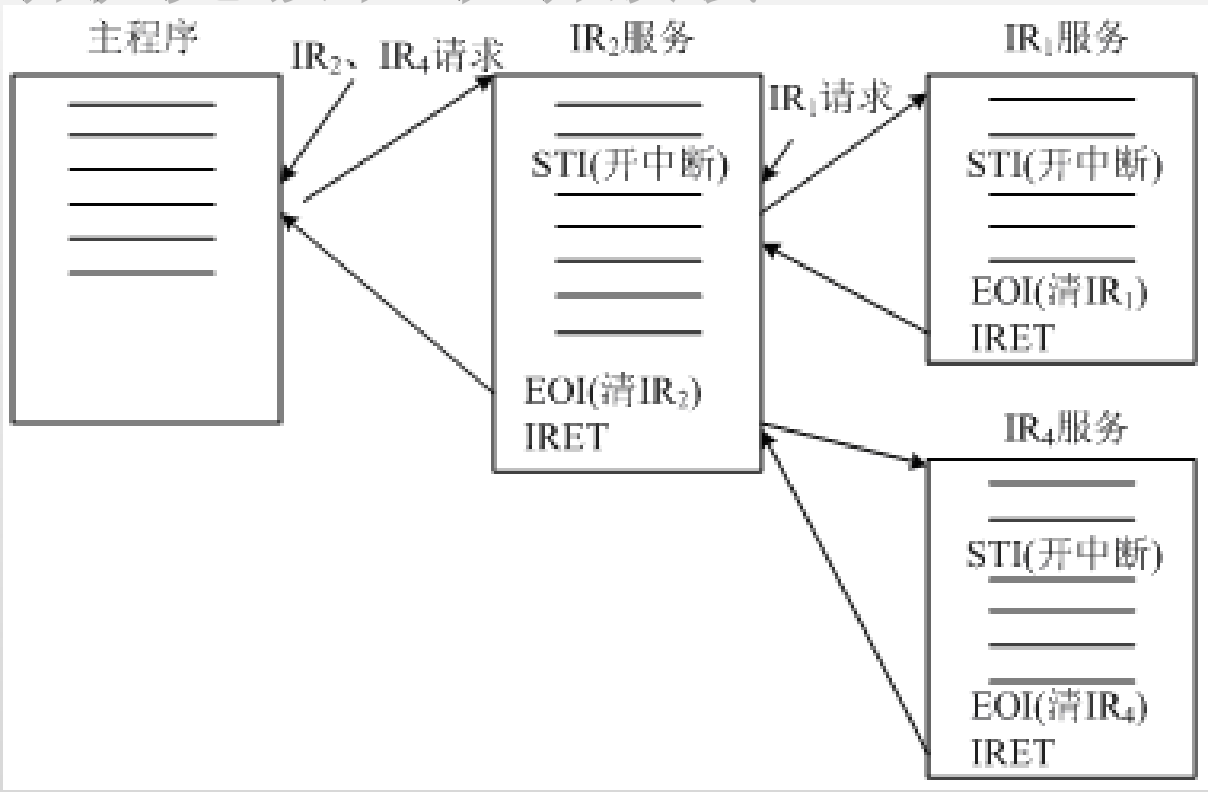
2）中断嵌套

- CPU响应中断时，一般先响应优先级高的，后响应优先级低的中断请求。
- 如CPU正在执行中断服务程序时，有优先级较高的中断源提出请求，则将正在处理的中断暂时挂起，先为高级中断服务，服务完后再返回较低级中断，称为中断嵌套。

4. 中断优先级和中断嵌套

- **如何实现中断嵌套？**
- 机器进入中断服务程序后，硬件会自动关中断。
- 只有用STI指令将中断打开后，才允许高级中断进入，实现中断嵌套。
- 中断服程序结束前，必须用EOI命令结束该级中断，并用IRET指令返回到中断前的断点处去继续执行原程序。

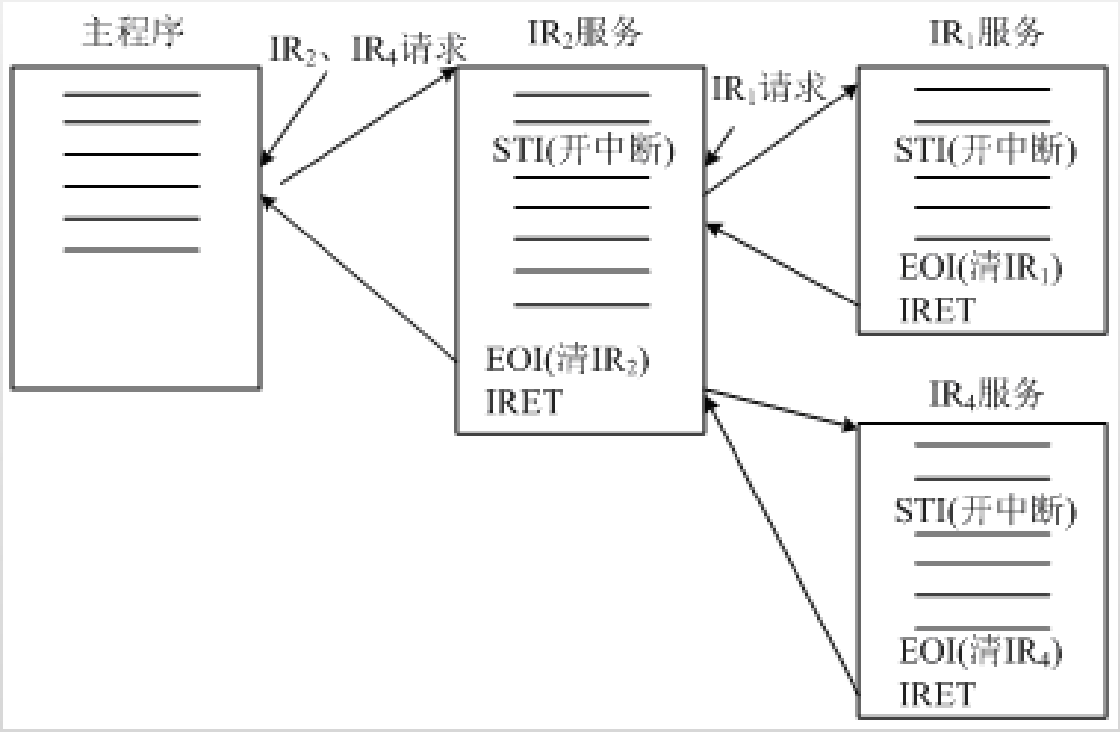
4. 中断优先级和中断嵌套



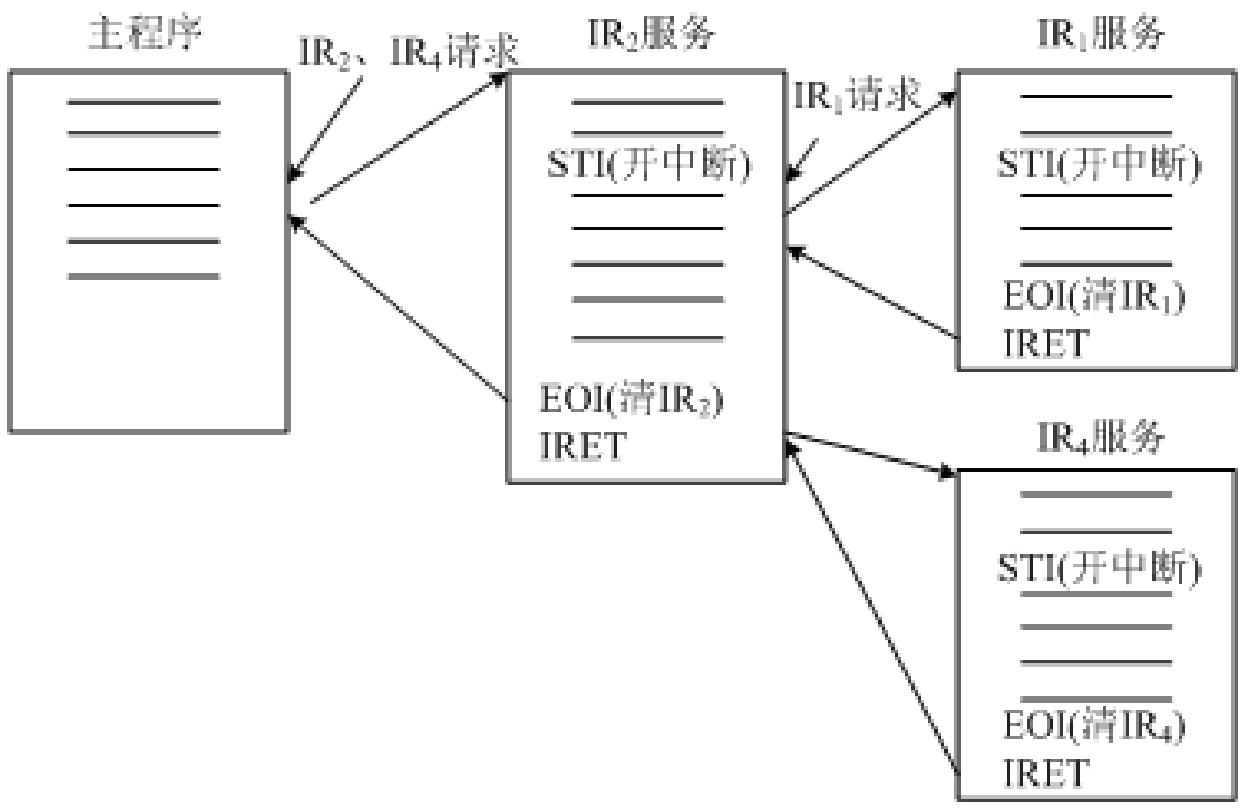
- 可屏蔽中断从8259A的8个输入端引入，一般情况下，优先级从高到低排列的次序为IR_0~IR_7。
- 中断嵌套的示意图如图所示，说明见下页。

4. 中断优先级和中断嵌套

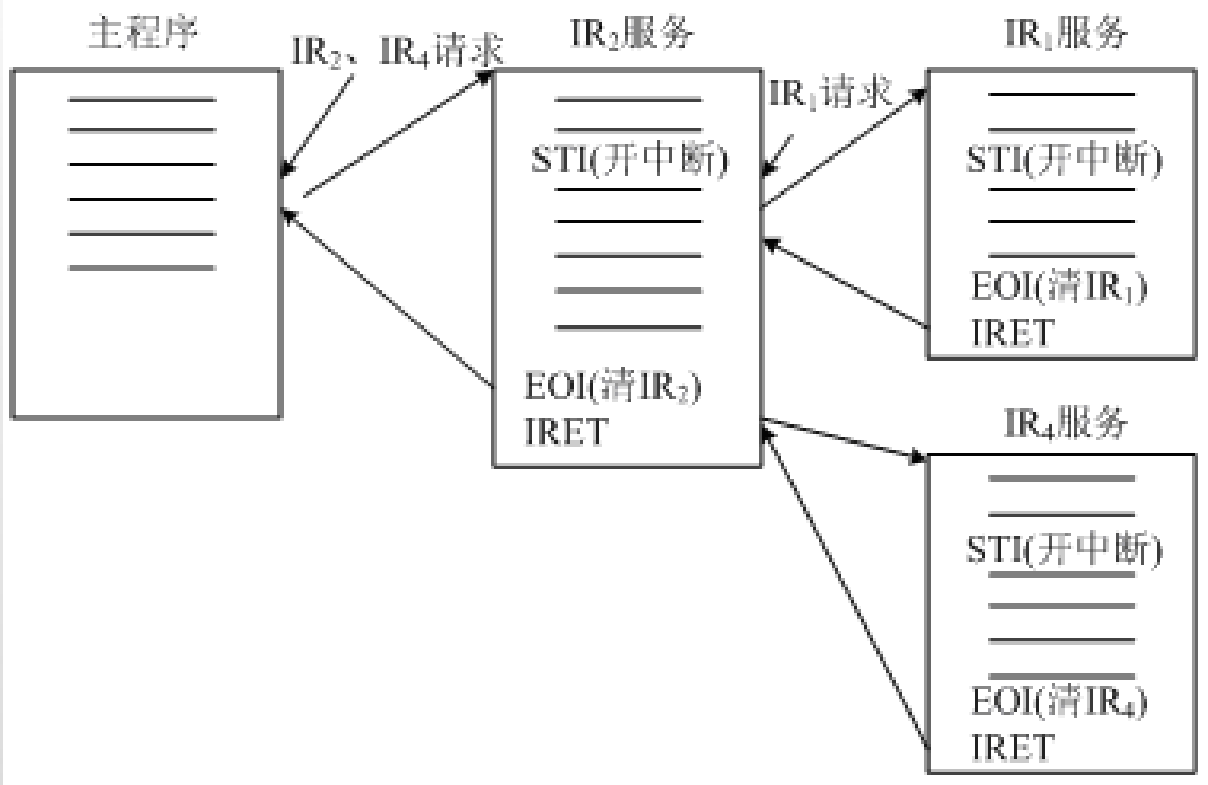
- 图中的中断嵌套情况的执行过程：



- 主程序运行中，IR_2、IR_4同时提出中断请求。IR_2优先级高，先为IR_2服务。在IR_2的服务程序中，要用STI指令开中断，允许更高级中断进入。



- IR$_2$服务时, IR$_1$提出请求, IR$_2$被挂起,为IR$_1$服务。IR$_1$结束前用EOI指令清除IR$_1$的服务寄存器,结束IR$_1$中断, 并用IRET指令返回IR$_2$服务程序, 继续运行。
- 运行至EOI命令时结束IR2的服务,响应IR$_4$的中断。



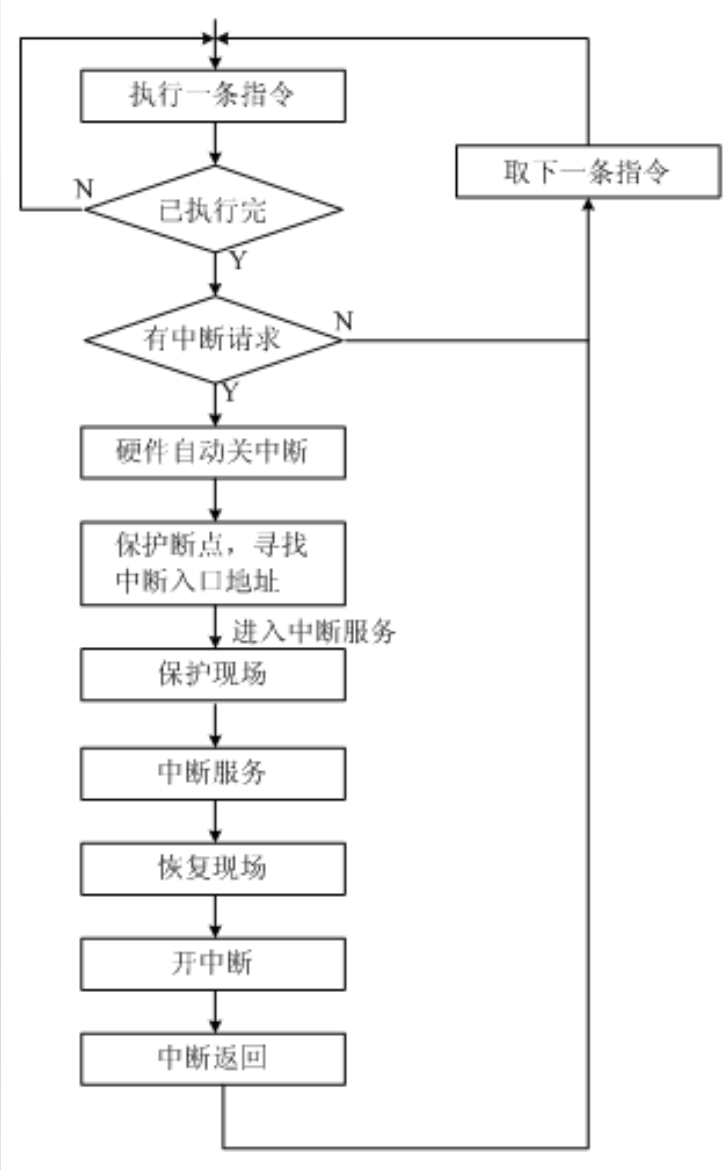
- **IR_4结束后，由IRET指令返回IR_2服务程序，最后从IR_2返回主程序。**
- **这样就完成了多重嵌套中断程序的执行过程。在中断服务程序中，如不安排开中断指令STI，则高级中断不能打断低级中断，也就不能实现中断的嵌套。**

§8.1 中断

8.1.1 中断概念和分类

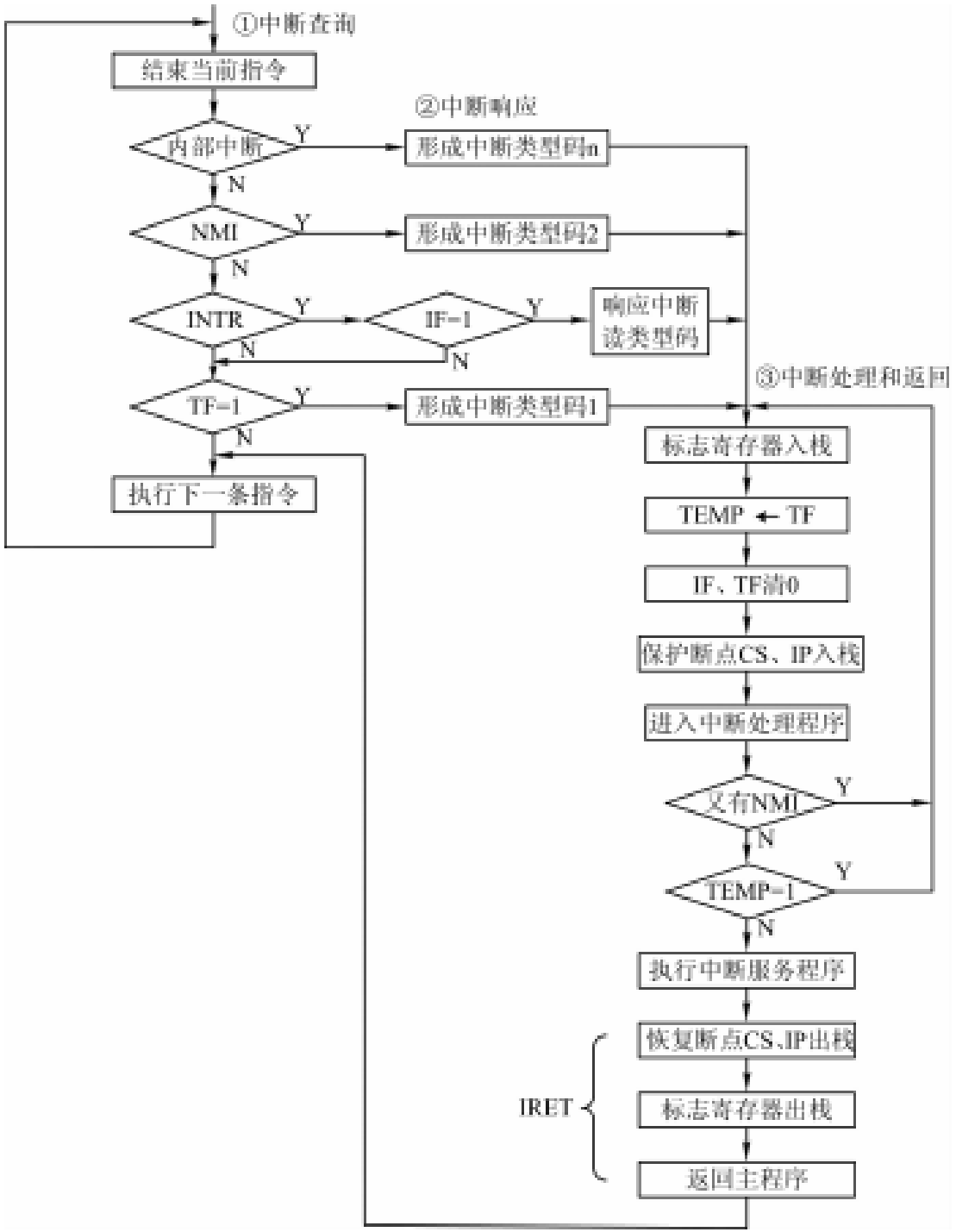
8.1.2 中断的响应与处理过程

8.1.2 中断的响应与处理过程



1.中断响应过程

- 如图，可屏蔽中断的响应和处理流程。
  - 每条指令执行完，CPU都要查NMI和INTR脚上是否有中断请求，若无，继续取下一条指令执行。如有则响应中断，硬件自动完成关闭中断和保护断点操作，将下条要执行指令的CS和IP（即断点）推入堆栈；
  - 然后寻找中断服务程序的入口地址，找到后就转入相应的中断服务程序。
  - 恢复现场，开中断，返回断点。



2. 8086的中断响应与处理

- **图8.7：8086的中断响应与处理的流程**
  - 中断响应和请求流程分为：中断查询、中断响应、中断处理和返回三个部分。

2. 8086的中断响应与处理

1）中断查询

- CPU在执行完每条指令后，会顺序检查是否有软中断、NMI、INTR和单步中断，如有，则进入中断响应周期。

2）中断响应

- 根据不同的中断源形成不同的中断类型码n，再在中断向量表中，根据n找到各自的中断服务程序的入口地址，转入相应的中断处理程序。
- 例如n=2，则形成中断类型码2，再在中断向量表中找到n×4=2×4=8开始的连续4个字节单元，从中取出CS：IP，转去执行可屏蔽中断。

2. 8086的中断响应与处理

- 重点介绍从INTR引脚引入的可屏蔽中断的类型码是如何形成的。CPU响应INTR中断后，要执行两个连续的中断响应INTA总线周期，其时序图如图8.8 ⬇

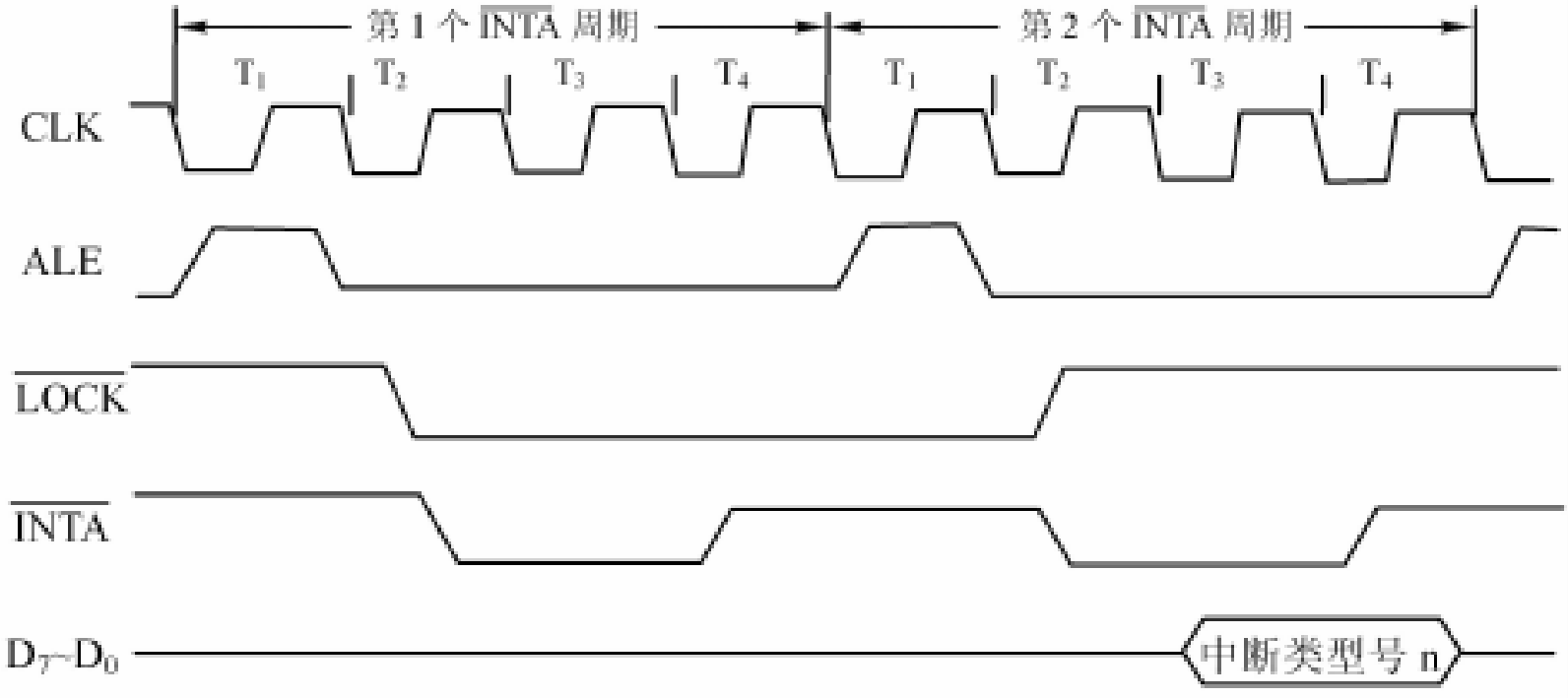


图 8.8　可屏蔽中断响应总线周期的时序图

2. 8086的中断响应与处理

- 第一个 $\overline{INTA}$ 周期，CPU使数据线D_7~D_0浮空，

  $\overline{INTA}$

  T_2~T_4期间向8259A发第1个中断响应信号

  表示CPU已响应此中断，禁止其它总线控制器

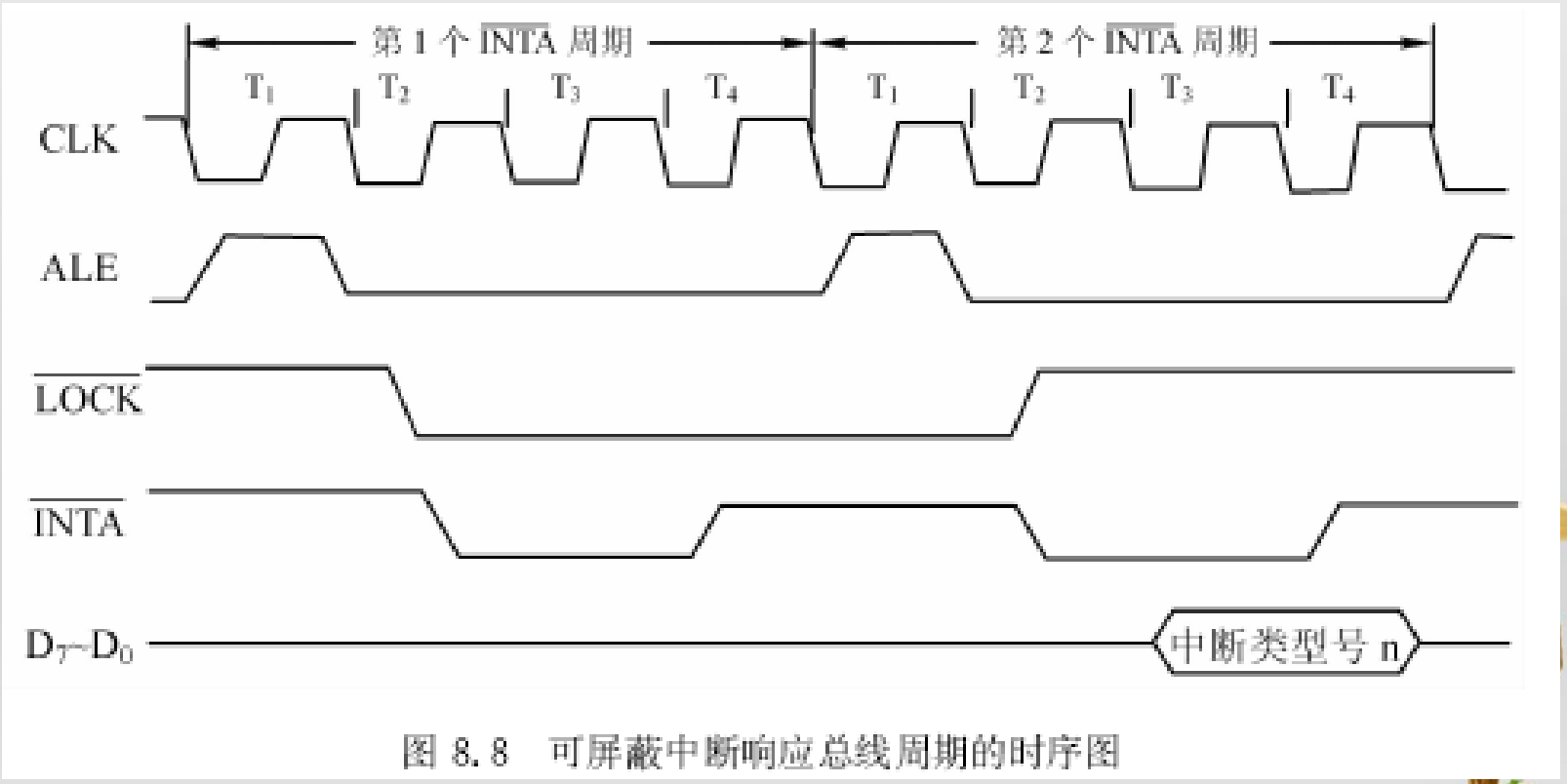


图 8.8　可屏蔽中断响应总线周期的时序图

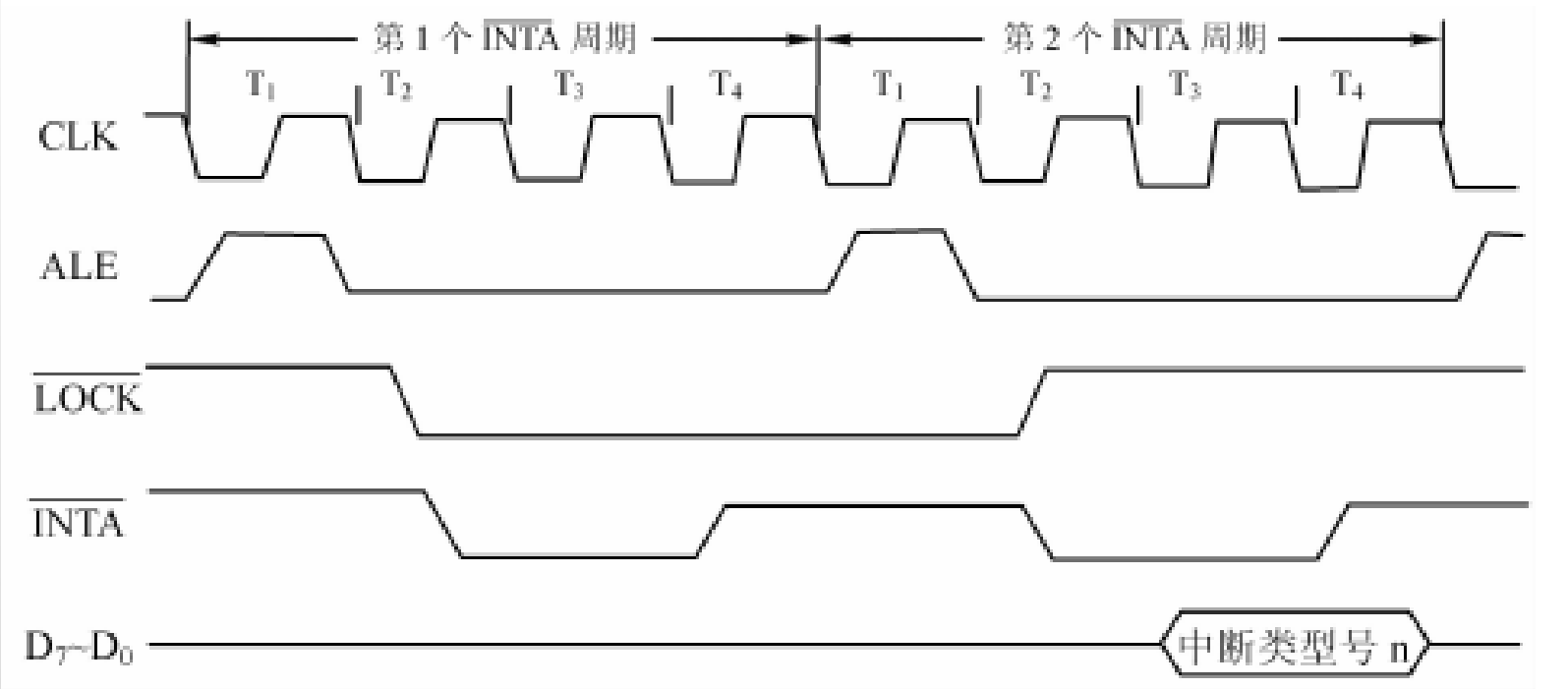


图 8.8　可屏蔽中断响应总线周期的时序图

- 第二个 $\overline{\text{INTA}}$ 周期，8259A收到第二个 $\overline{\text{INTA}}$ 信号后，将中断类型号n置于数据总线上，由此找到中断服务程序的入口地址。

2. 8086的中断响应与处理

从8259A的IR_7~IR_0上可引入8级中断，形成8个8位中断类型码，其中高5位D_7~D_3由用户通过对8259A编程来确定，在PC/XT机中为00001，低3位D_2~D_0由IR_7~IR_0的序号来决定，见表8.1。高5位确定后，8级中断类型码就确定了。

表 8.1　8259A 的 8 级中断类型码的确定

D_7～D_3	D_2～D_0	中断类型号	中断输入引脚	中断源
00001	0 0 0	08H	IR_0	系统时钟
00001	0 0 1	09H	IR_1	键盘
00001	0 1 0	0AH	IR_2	保留
00001	0 1 1	0BH	IR_3	串口 2（COM2）
00001	0 1 0	0CH	IR_4	串口 1（COM1）
00001	1 0 1	0DH	IR_5	硬盘
00001	1 1 0	0EH	IR_6	软盘
00001	1 1 1	0FH	IR_7	打印机

2. 8086的中断响应与处理

3）中断处理和返回

（1）由硬件自动完成以下工作

- ♦ FLAGS的内容入栈；
- ♦保□□步标志TF；
- ♦清IF标志, 关中断, 中断处理过程中禁止其它中断进入；
- ♦清TF标志，使CPU不会以单步形式执行中断处理程序；
- ♦保□断点，CS：IP入栈。

2. 8086的中断响应与处理

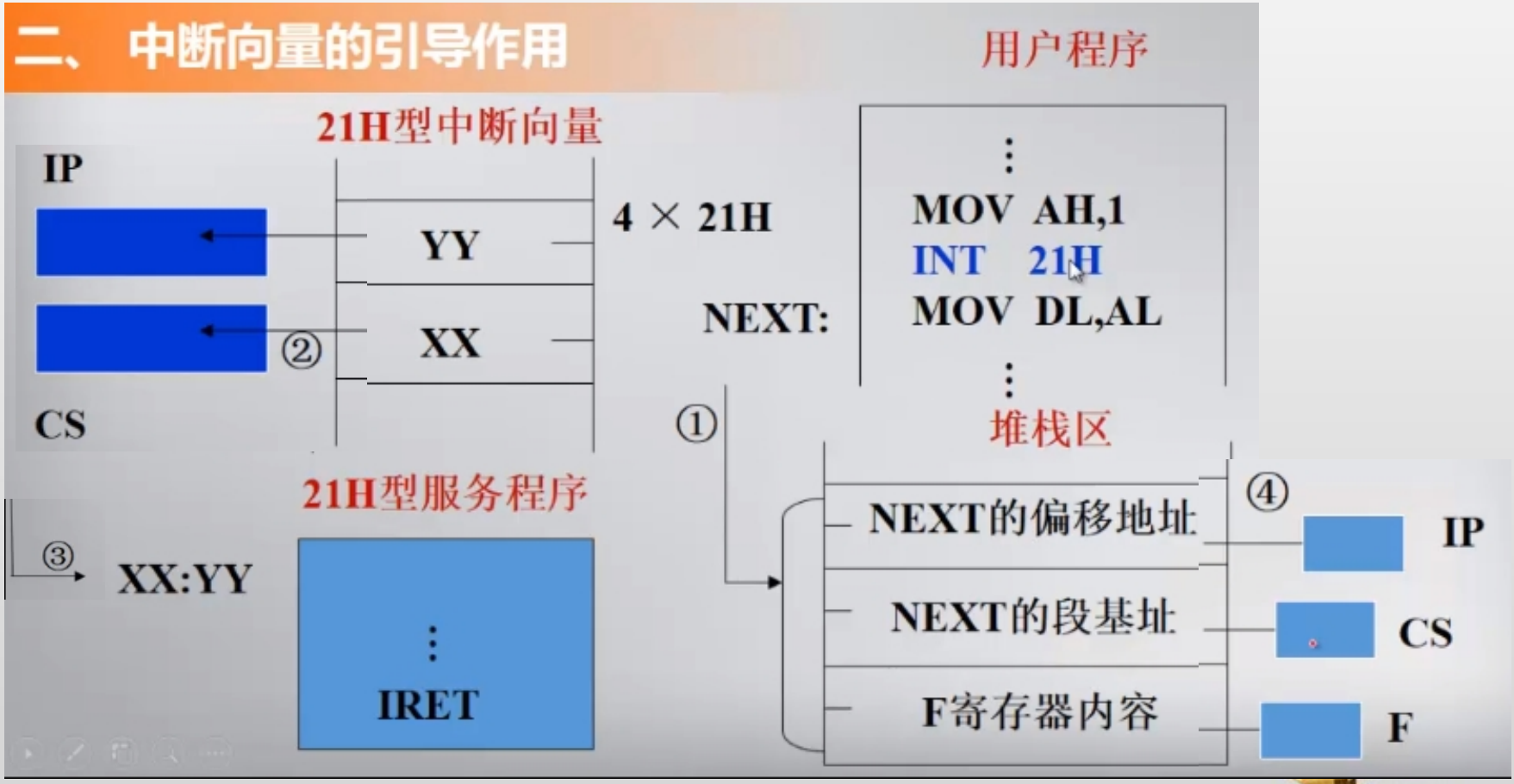
（2）进入中断服务

- 进入中断处理后，　若在处理过程中又有NMI进入，NMI中断处理后，会清除CPU中锁存的NMI请求信号，使加在CPU上的NMI只会被CPU识别一次；
- 接下来执行用户编写的中断服务程序，包含保护现场，中断处理和恢复现场程序。

（3）执行用户编写的中断返回指令IRET

- CS：IP出栈，恢复断点，恢复FLAGS的内容，返回主程序，继续执行下一条指令。

二、 中断向量的引导作用


用户程序
21H型中断向量
IP
YY
4 × 21H
MOV AH,1
INT 21H
NEXT:
MOV DL,AL
②
XX
CS
①
堆栈区
21H型服务程序
④
NEXT的偏移地址
IP
③
XX:YY
NEXT的段基址
CS
IRET
F寄存器内容
F