

本节主题

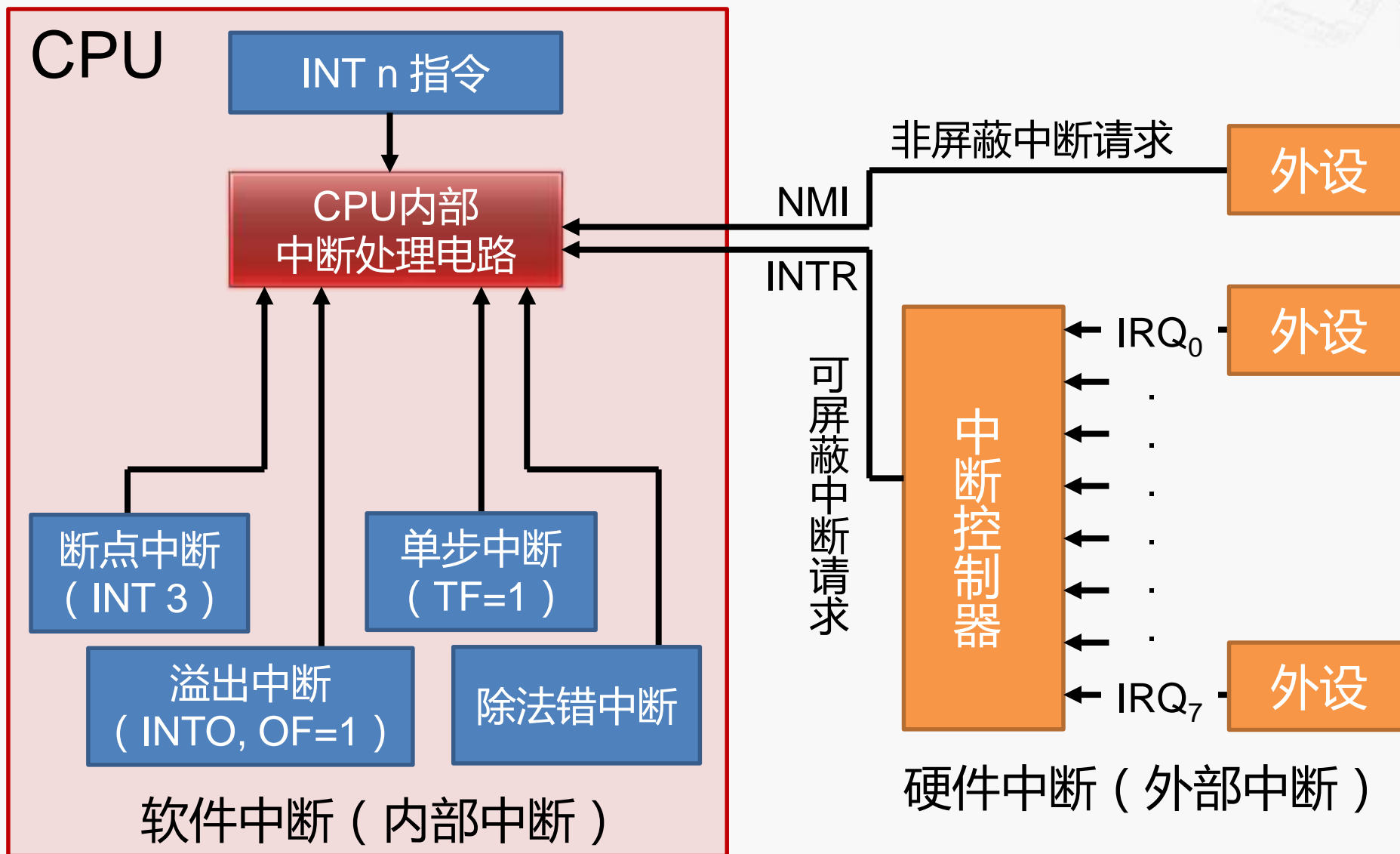


中断的处理过程

北京大学·慕课
计算机组成
制作人：陆俊林



中断的检测（示例：x86实模式系统）



中断处理过程



1. 关中断

- CPU关闭中断响应，即不再接受其它外部中断请求

2. 保存断点

- 将发生中断处的指令地址压入堆栈，以使中断处理完后能正确地返回

3. 识别中断源

- CPU识别中断的来源，确定中断类型号，从而找到相应的中断服务程序的入口地址

中断处理过程（续）



4. 保护现场

- 将发生中断处的有关寄存器（中断服务程序中要使用的寄存器）以及标志寄存器的内容压入堆栈

保护现场的作用

主程序片段：

```
...  
MOV    CX, [2CH]  
MOV    BX, [3AH]  
DIV    BX ;假设此时发生中断  
CMP    CX, BX  
JNZ    NOT_EQUAL  
...
```

中断服务程序片段1：

```
...  
ADD    CX, DX  
...  
IRET; 中断返回后，CX内容已改变
```

中断服务程序片段2：

```
...  
PUSH   CX  
...  
ADD    CX, DX  
...  
POP    CX  
IRET; 中断返回后，CX内容没有变化
```

中断处理过程（续）



4. 保护现场

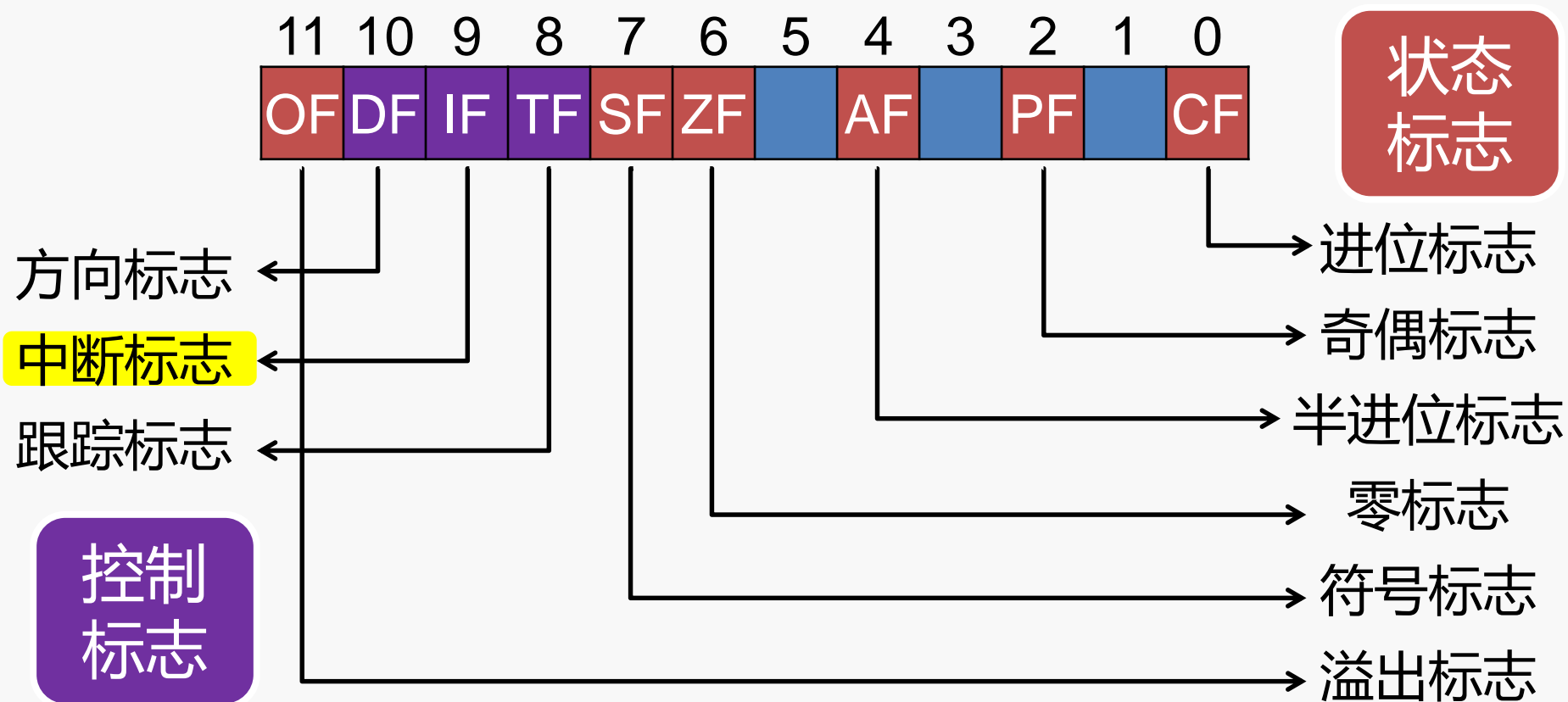
- 将发生中断处的有关寄存器（中断服务程序中要使用的寄存器）以及标志寄存器的内容压入堆栈

5. 执行中断服务程序

- 转到中断服务程序入口开始执行，可在适当时刻重新开放中断，以便允许响应较高优先级的外部中断

软件开放和关闭中断响应的方法

标志寄存器（FLAGS）中的IF标志位



控制标志的功能简述（IF）



🎯 中断标志IF（Interrupt Flag）

- 控制对可屏蔽中断的响应
 - 若 $IF=1$ ，则允许CPU响应可屏蔽中断请求
 - 若 $IF=0$ ，则不允许CPU响应可屏蔽中断请求
- 可以用指令设置IF标志位
 - STI ；把中断标志IF置1
 - CLI ；把中断标志IF清0
- IF对非屏蔽中断和内部中断不起作用

中断处理过程（续）



4. 保护现场

- 将发生中断处的有关寄存器（中断服务程序中要使用的寄存器）以及标志寄存器的内容压入堆栈

5. 执行中断服务程序

- 转到中断服务程序入口开始执行，可在适当时刻重新开放中断，以便允许响应较高优先级的外部中断

6. 恢复现场并返回

- 把“保护现场”时压入堆栈的信息弹回原寄存器，然后执行中断返回指令，从而返回主程序继续运行

如何从中断服务程序中返回

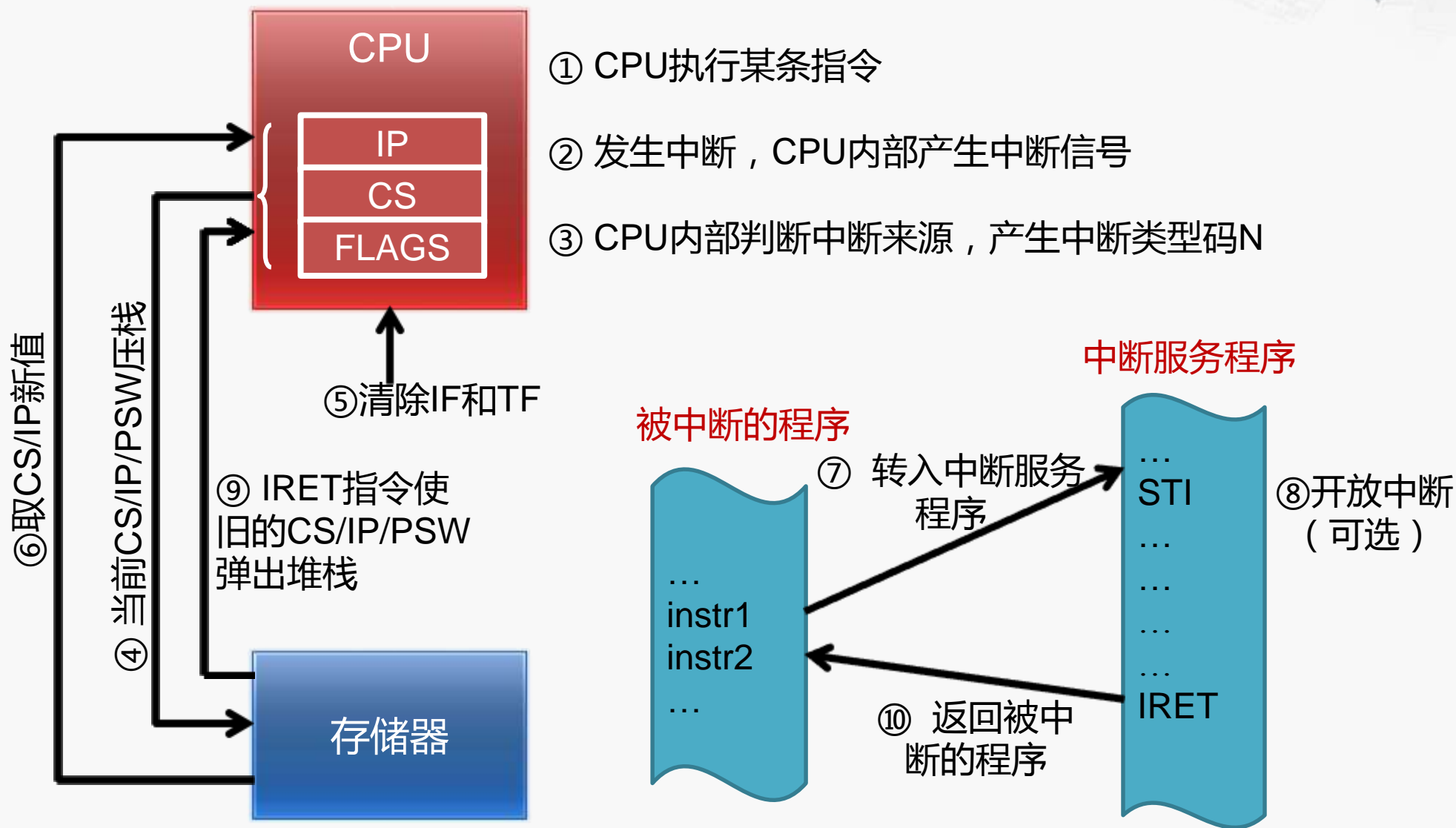


IRET指令（中断返回）

- ❏ 格式：IRET
- ❏ 操作
 - 从栈顶弹出3个字，分别送入IP、CS和FLAGS寄存器
- ❏ 说明
 - 放在中断服务程序的末尾
 - 按中断调用时的逆序恢复现场
 - 返回到程序发生中断处继续执行

扩展：IRETD指令，IRETQ指令

中断处理过程（以内部中断为例）



中断处理过程的任务分工

通常的软硬件分工

- 前3项通常由处理中断的硬件电路完成
- 后3项通常由软件（中断服务程序）完成

1

• 关中断

2

• 保存断点

3

• 识别中断源

4

• 保护现场

5

• 执行中断服务程序

6

• 恢复现场并返回

中断处理过程的任务分工

🔍 注意事项

- 针对具体系统，中断服务程序设计者必须清楚软硬件的具体分工
- 硬件完成了哪些操作？如标志寄存器是否已被压入堆栈
- 需要软件完成哪些操作？

1

• 关中断

2

• 保存断点

3

• 识别中断源

4

• 保护现场

5

• 执行中断服务程序

6

• 恢复现场并返回

本节小结



中断的处理过程

北京大学·慕课
计算机组成
制作人：陆俊林

