

中断与异常

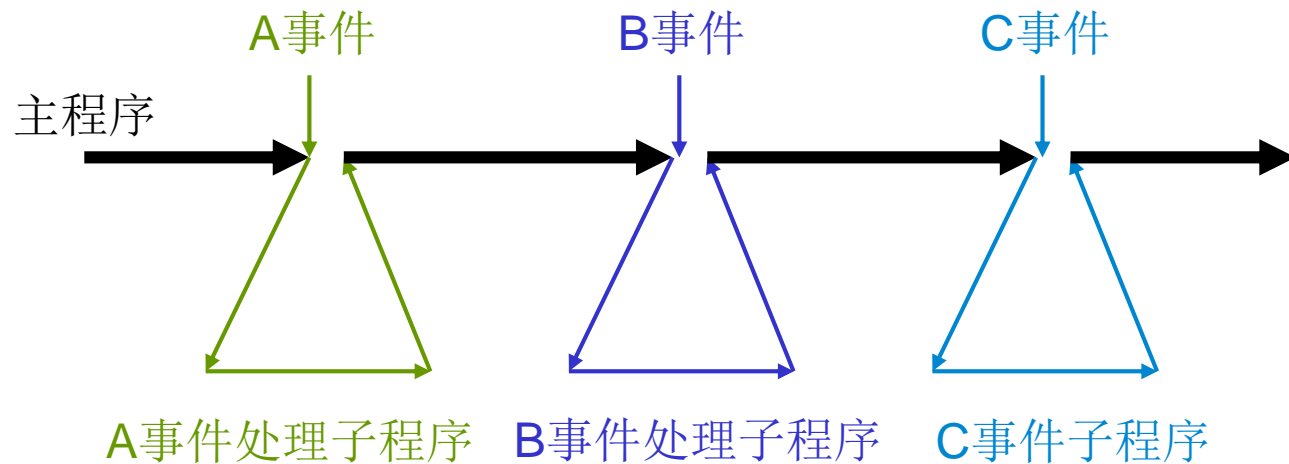


上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

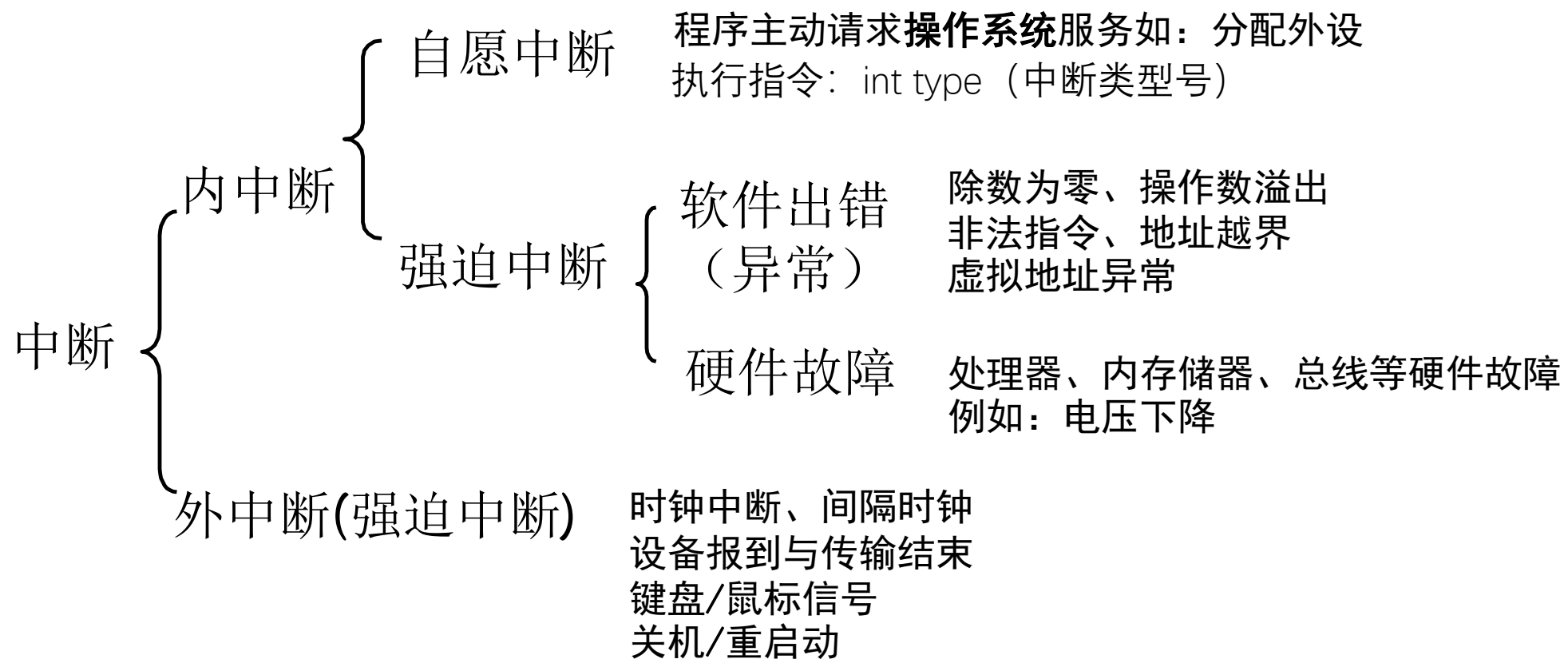
“中断”或“异常”



- 在程序运行时，系统外部、内部或现行程序本身出现需要特殊处理的“事件”
- CPU立即强行中止现行程序的运行，启动相应的程序来处理这些“事件”
- 处理完成后，CPU恢复原来的程序运行
- 这些“事件”被称为“中断”或“异常”



中断的类型



中断的意义

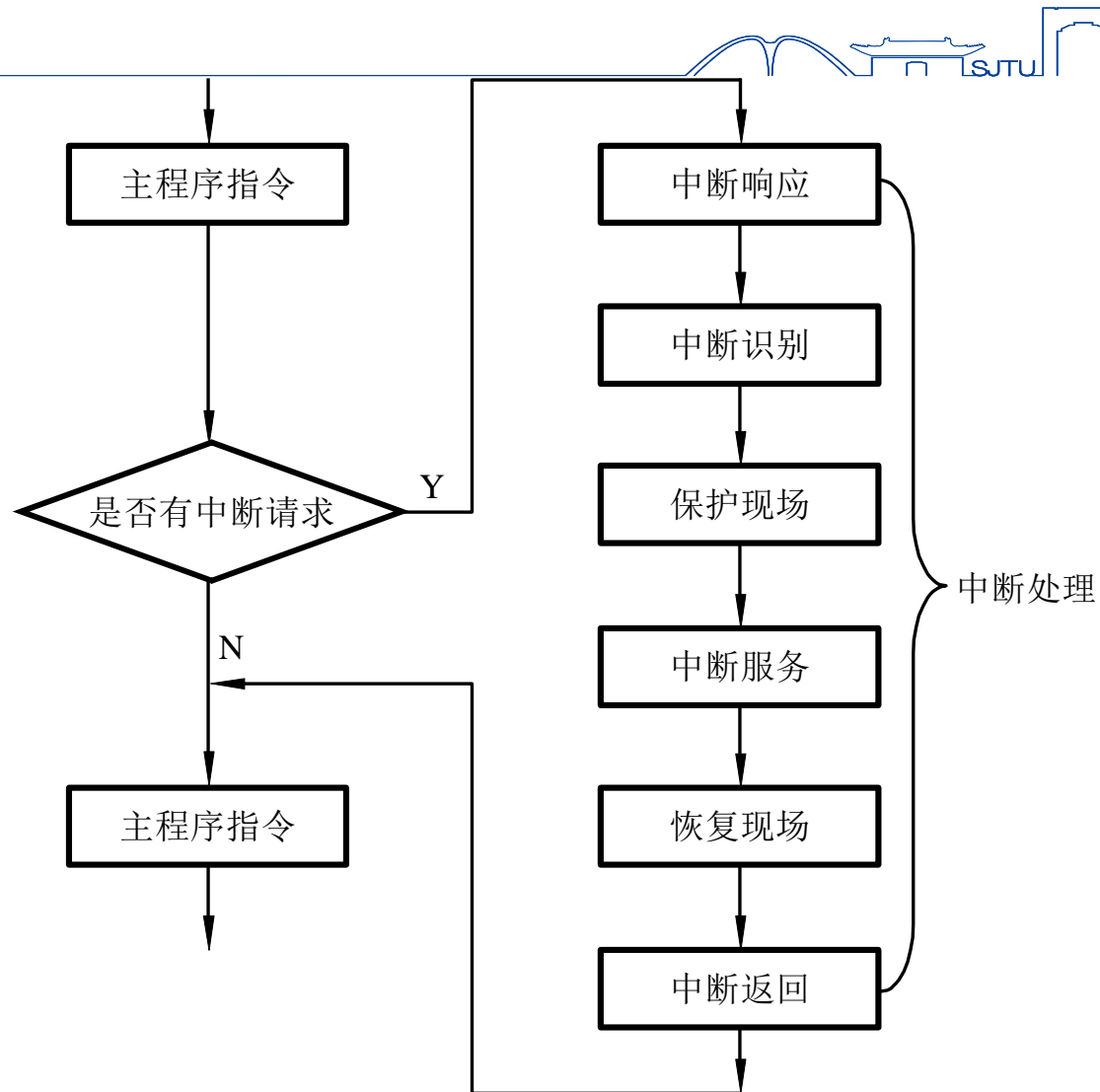


中断技术赋予计算机应变能力，将有序的运行和无序的事件统一起来，增强了系统的处理能力。

中断处理

中断处理过程

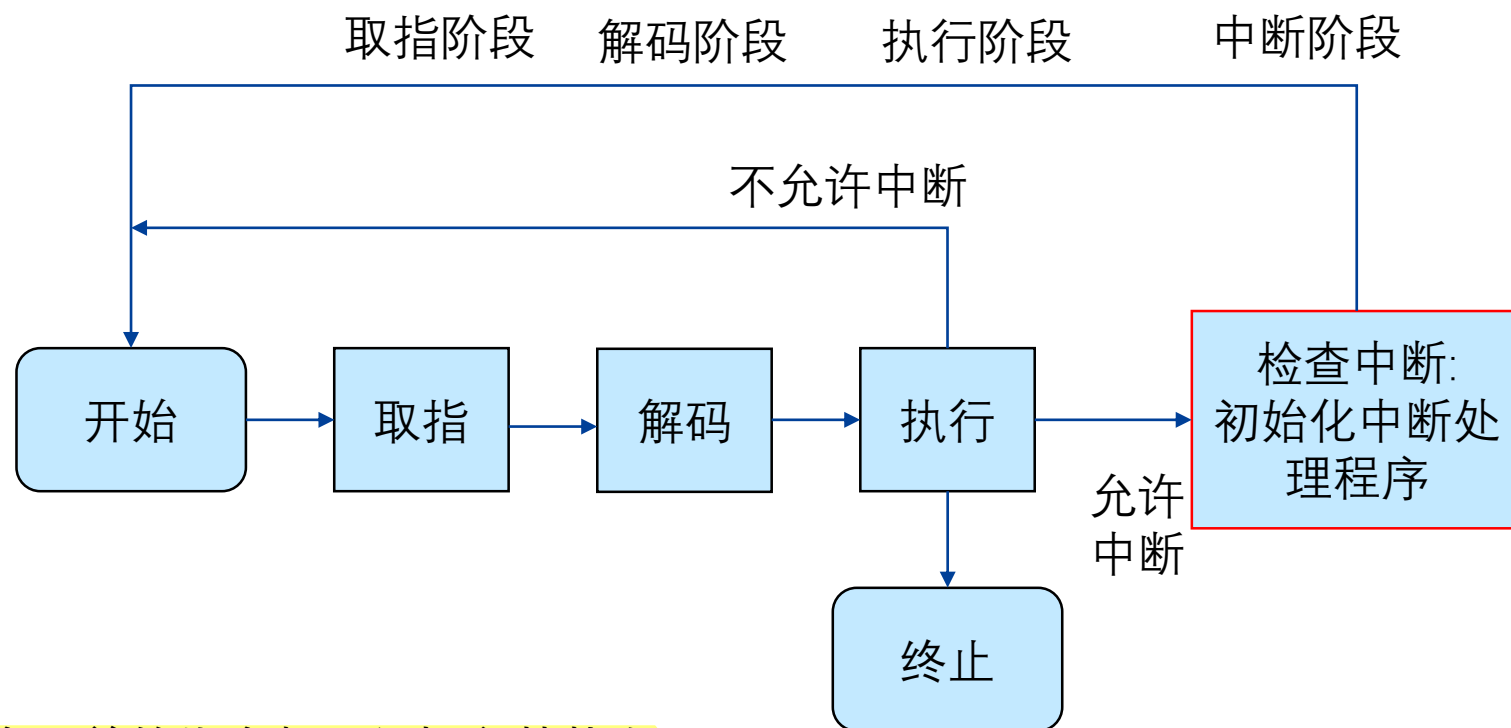
- 一条指令结束时切换
- 保存断点，保护上下文（现场）
- 恢复现场，返回断点
- 保证程序的完整性



精确中断响应



在指令执行周期的最后一个阶段增加一个“检查中断”的操作，以响应中断



“精确”中断：

- (1) 所有该指令之前的指令都已经提交其状态
- (2) 所有后续指令（包括中断的指令）没有改变任何机器的状态



中断处理中的问题



- 中断源识别方式
- 中断处理程序入口地址的形成
- 中断裁决
- 中断屏蔽

问题1：中断源的识别



CPU轮询

相应的设备做出回答

原因寄存器

中断向量

中断源识别：原因寄存器



原因寄存器

记录中断的原因

由硬件自动设置

■ MIPS的异常原因代码

- 0 (INT) => 外部中断
- 4 (ADDRL) => 地址错误 (load或取指)
- 5 (ADDRS) => 地址错误 (store)
- 6 (IBUS) => 取指时总线错误
- 7 (DBUS) => 取数据时总线错误
- 8 (Syscall) => 系统调用
- 9 (BKPT) => 断点
- 10 (RI) => 保留的指令码
- 12 (OVF) => 算术溢出

Cause 寄存器



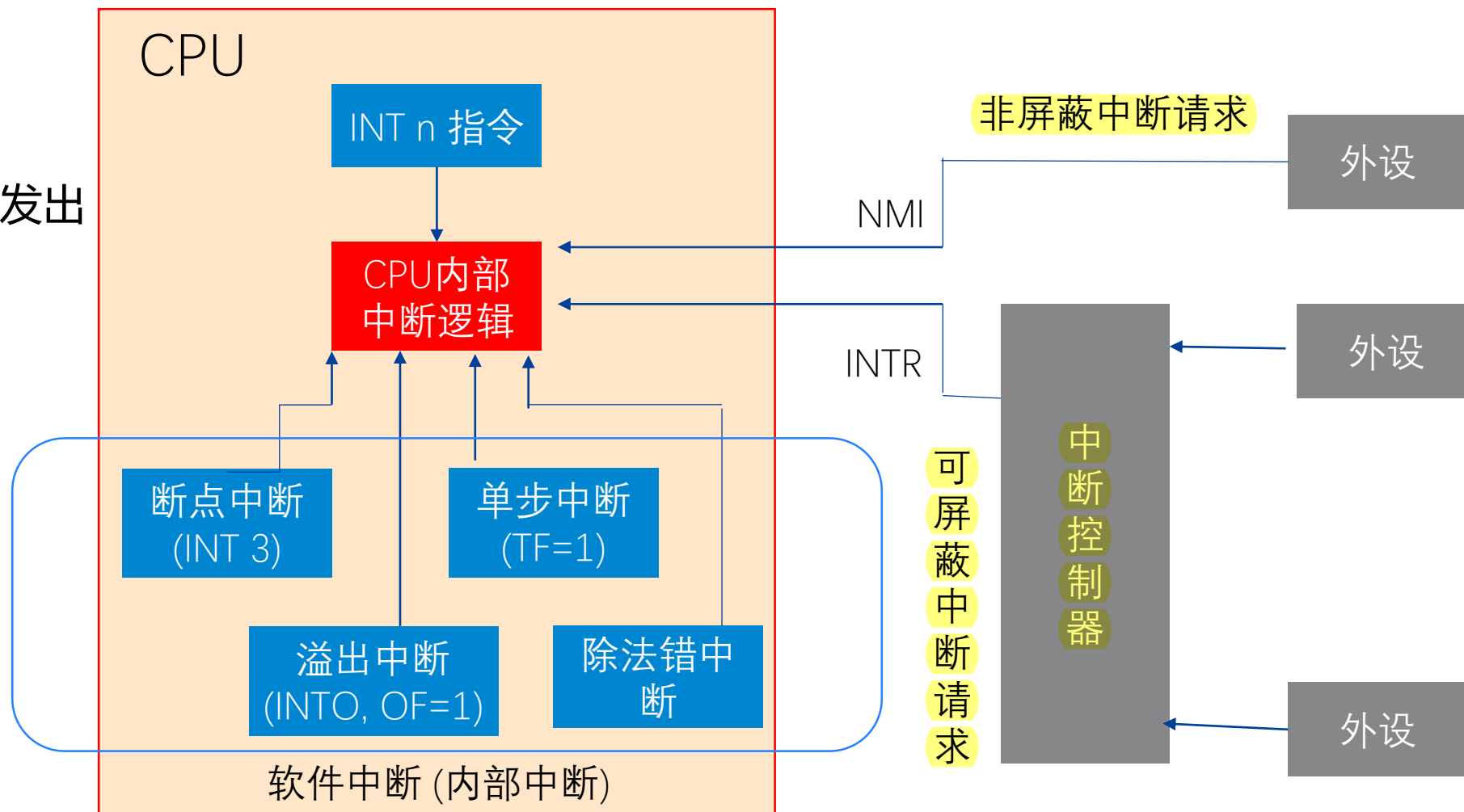
Exception Code



中断源识别：中断向量

中断向量

- 由设备主动向CPU发出识别信息



问题2：中断处理程序入口地址的形成

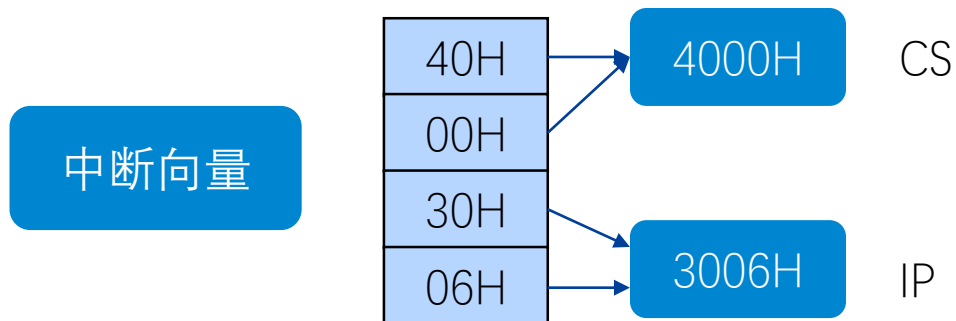


- 非向量中断
 - 响应时执行查询程序查询中断源;
 - 根据中断原因（偏移量），计算服务程序入口地址
- 向量中断
 - 将服务程序入口(中断向量)组织在中断向量表中;
 - 响应时由硬件直接产生相应向量地址,
 - 按向量地址查表，取得服务程序入口，转入相应服务程序

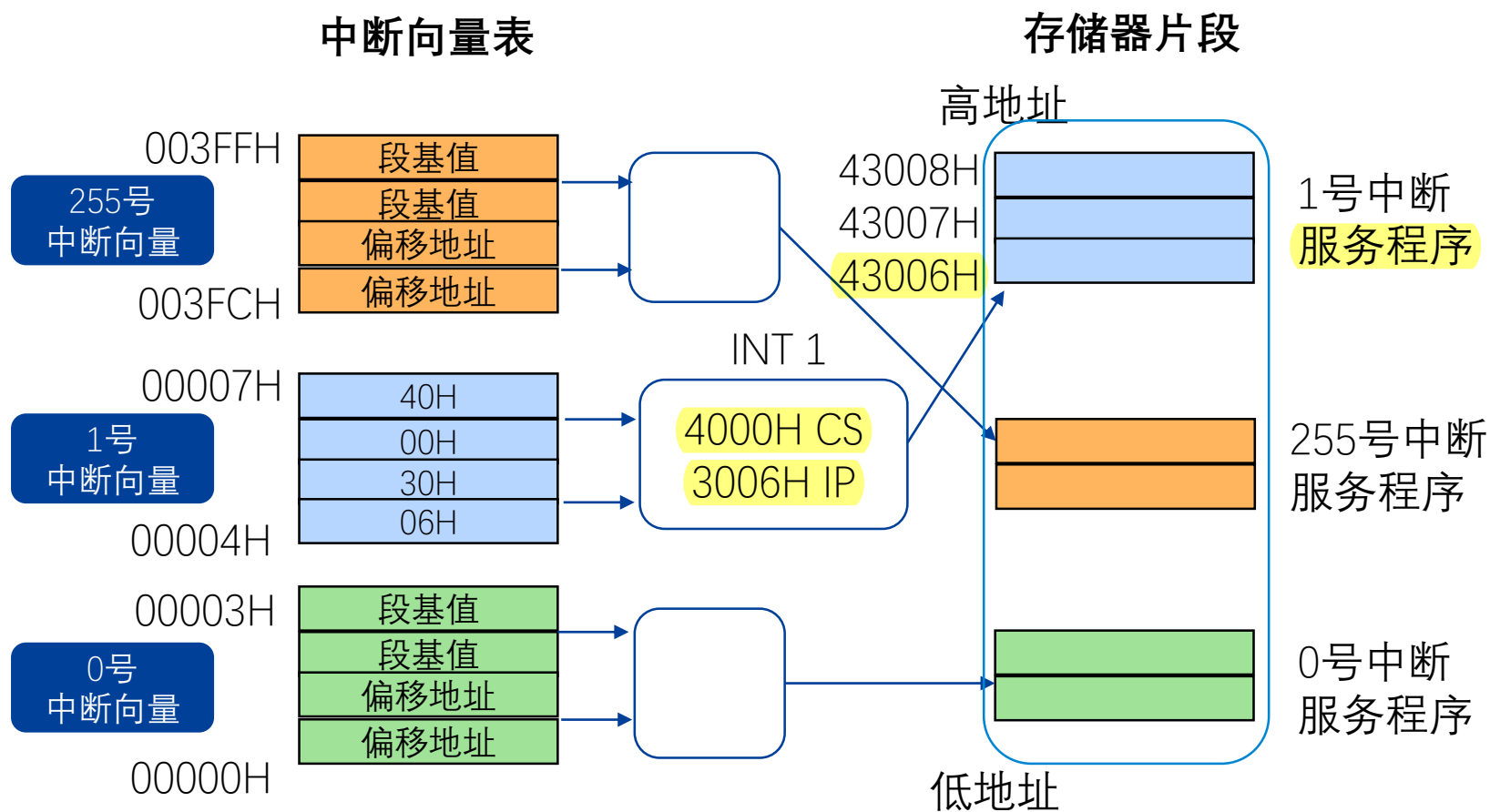
中断向量



- 中断向量表区的地址范围：00000H~003FFH
- 存放256个中断服务程序的入口地址（也称中断向量）
- 每个中断向量占4个字节单元
 - 前两个字节单元存放中断服务程序入口地址的偏移量（IP）
 - 后两个字节单元存放中断服务程序入口地址的段基值（CS）
 - 低字节在前，高字节在后



中断向量和中断服务程序



8086的中断向量表



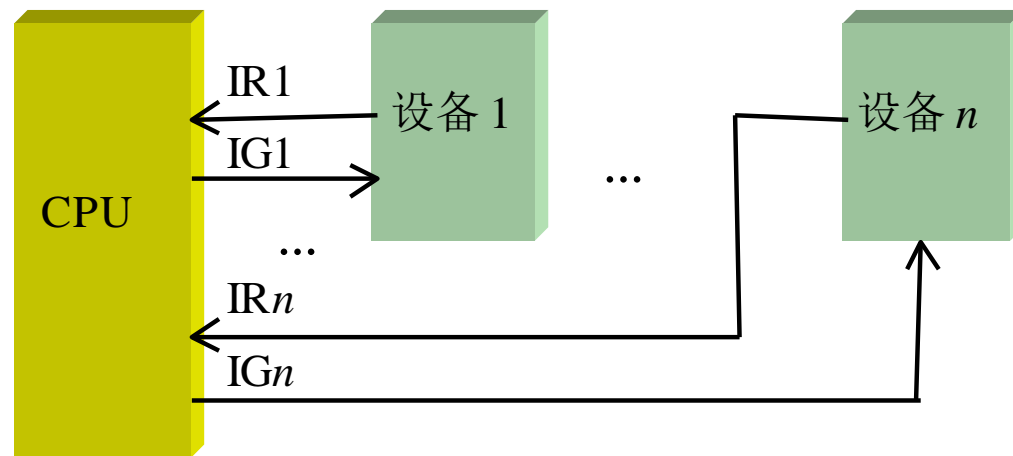
| 中断用途 | 类型号 | 说明 |
|--------------------|-------|-----|
| 供用户定义的中断 (224个) | 类型255 | |
| | | |
| | 类型32 | |
| 保留的中断 (27个) | 类型31 | |
| | | |
| | 类型5 | |
| 专用的中断 (5个) | 类型4 | 溢出 |
| | 类型3 | 断点 |
| | 类型2 | 非屏蔽 |
| | 类型1 | 单步 |
| | 类型0 | 除法错 |



问题3：中断裁决

三种方式：

- 链式查询
- 独立请求
- 分组链式



IRx: 中断请求

IGx: 中断许可

问题4：中断屏蔽：单重中断

简单的中断处理过程：

- 关中断
- 保存现场
- 识别中断
- 形成服务程序入口地址
- 执行服务程序
- 恢复现场
- 开中断

问题4：中断屏蔽：多重中断

多重中断

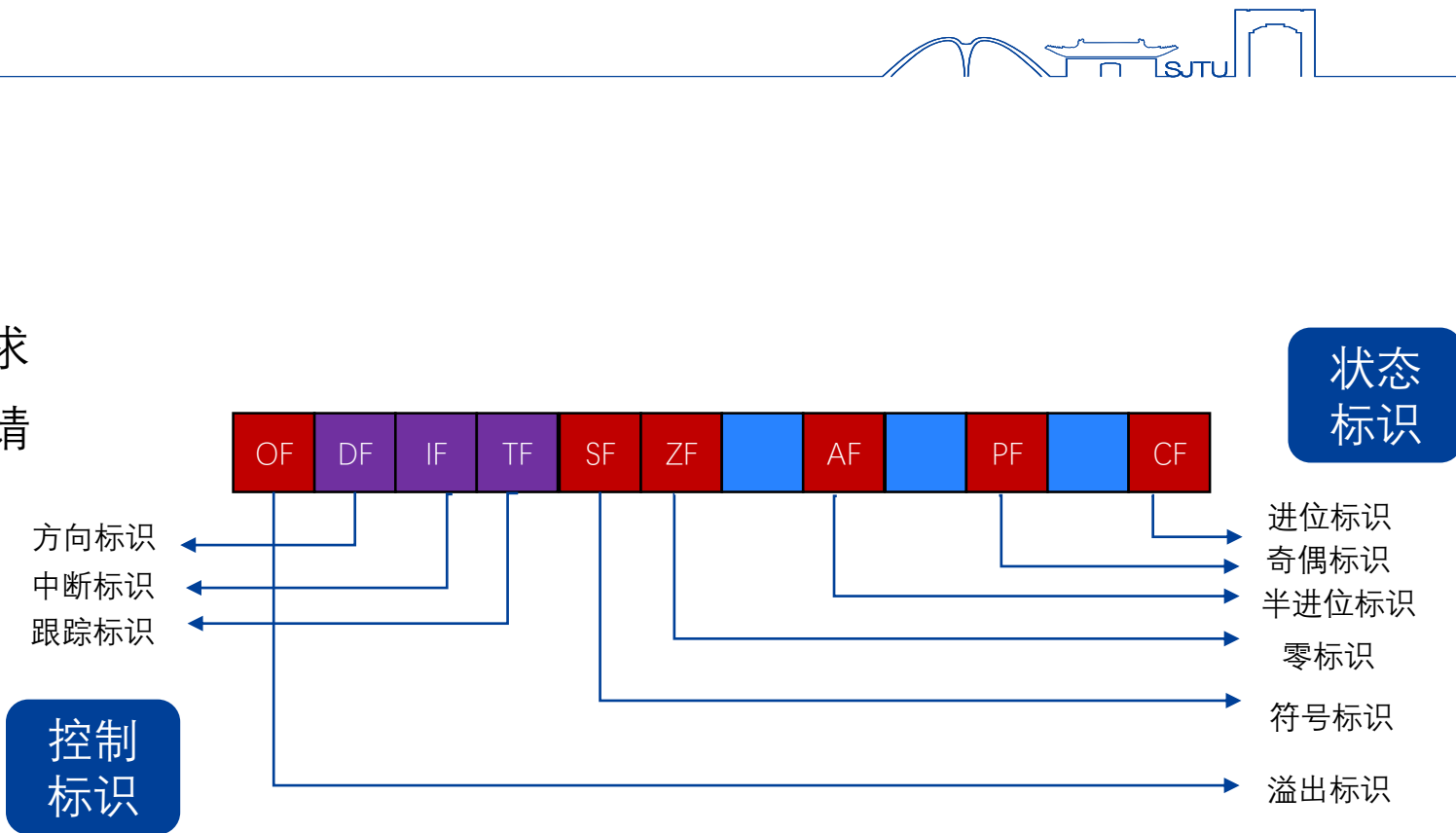
- 中断服务程序也可以被中断
- 中断嵌套

中断处理过程：

- 关中断
- 保存现场
- 识别中断
- 形成服务程序入口地址
- **开中断**
- 执行服务程序
- **关中断**
- 恢复现场
- 开中断

关闭中断响应的方法

- 标志寄存器 (FLAGS) 中中断标志IF
- 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



如何从中断服务程序中返回



- IRET指令（中断返回）
- 格式：IRET
- 放在中断服务程序的末尾
- 操作
 - 从栈顶弹出3个字，分别送入IP、CS和FLAGS寄存器
 - 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4项通常由软件完成

中断处理过程：

- 关中断
- 保存现场
- 识别中断
- 形成服务程序入口地址
- 执行服务程序
- 恢复现场
- 开中断

谢谢！

