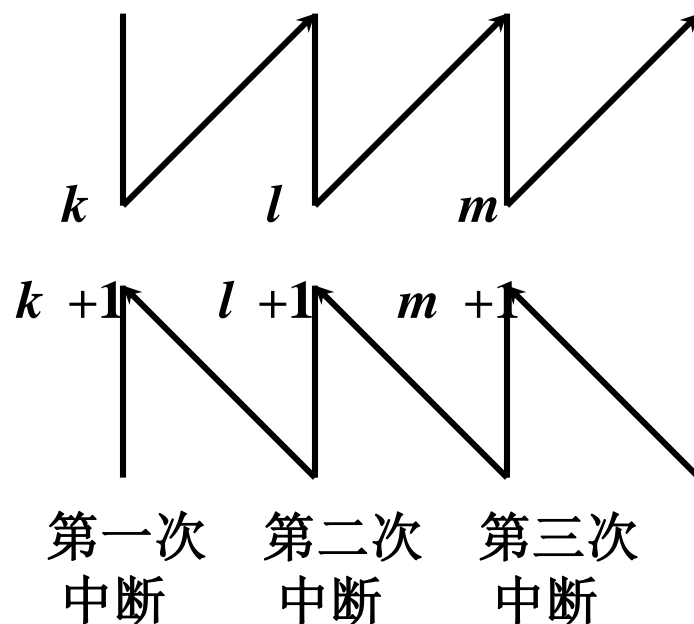


## 1. 多重中断的概念(中断嵌套)



程序断点  $k+1$ ,  $l+1$ ,  $m+1$

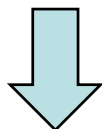
(7) 处理中断的过程中又 出现新的中断 怎么办 ?

## 2. 实现多重中断的条件



### (1) 提前设置开中断

中断隐指令响应，存断点之后关中断  
中断服务结束之前开中断



中断隐指令响应，存断点之后关中断  
中断服务开始之后开中断

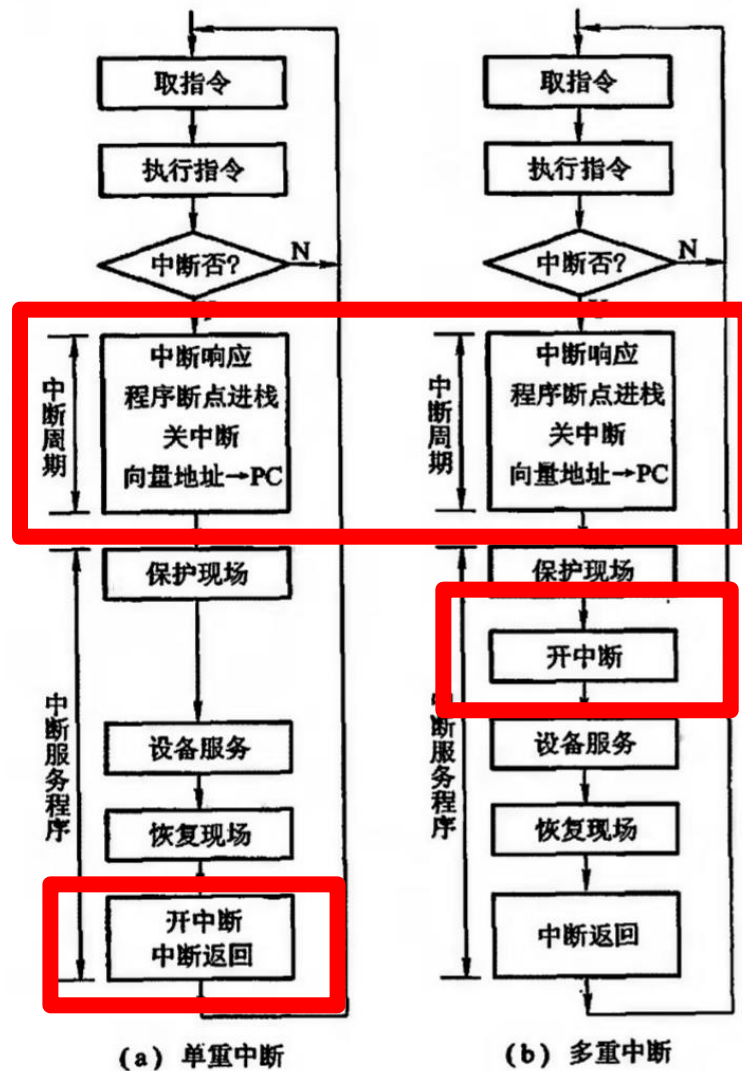


图 5.43 单重中断和多重中断服务程序流程

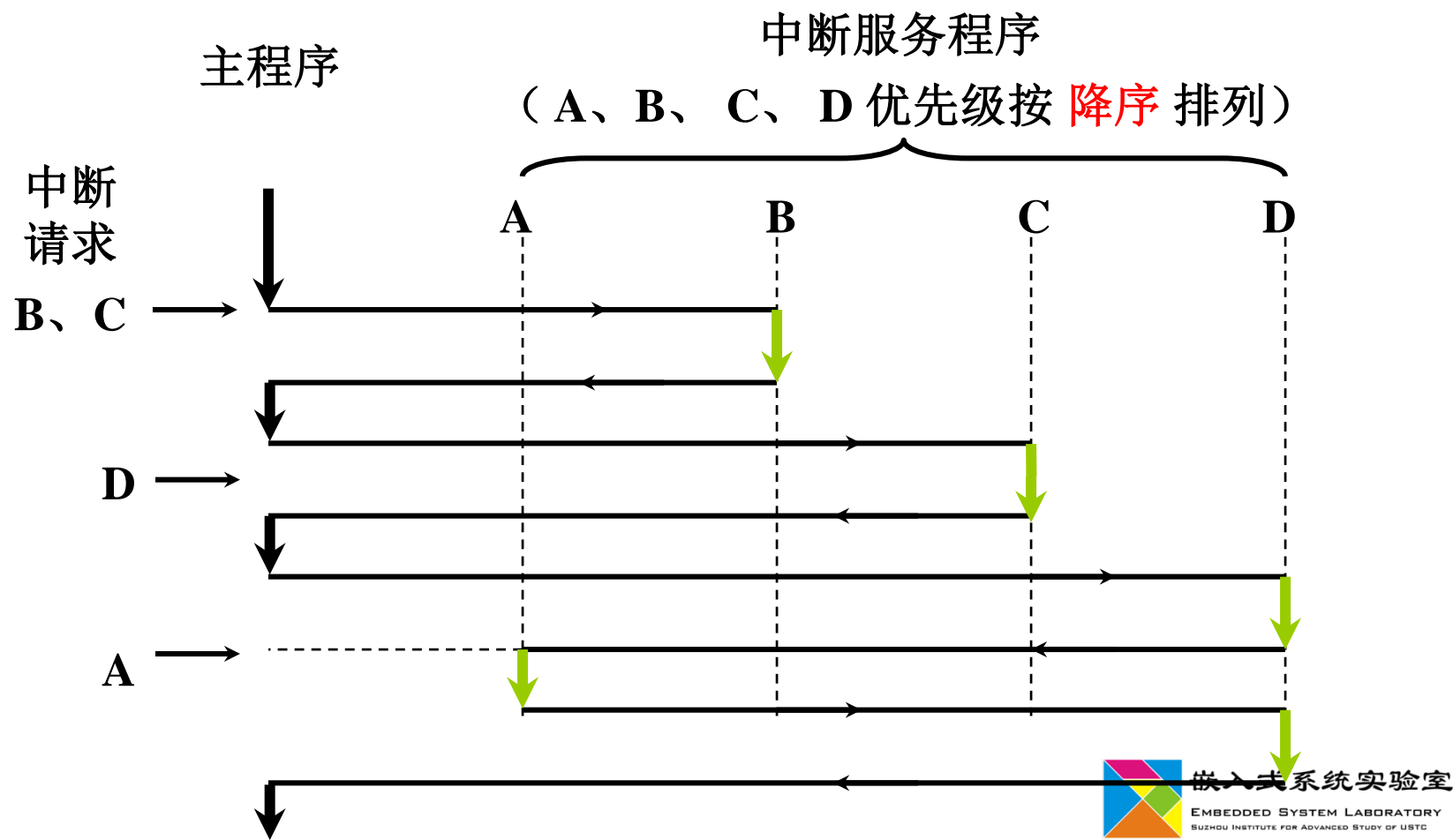
## 2. 实现多重中断的条件



(1) 提前设置开中断

优先级如何设定？

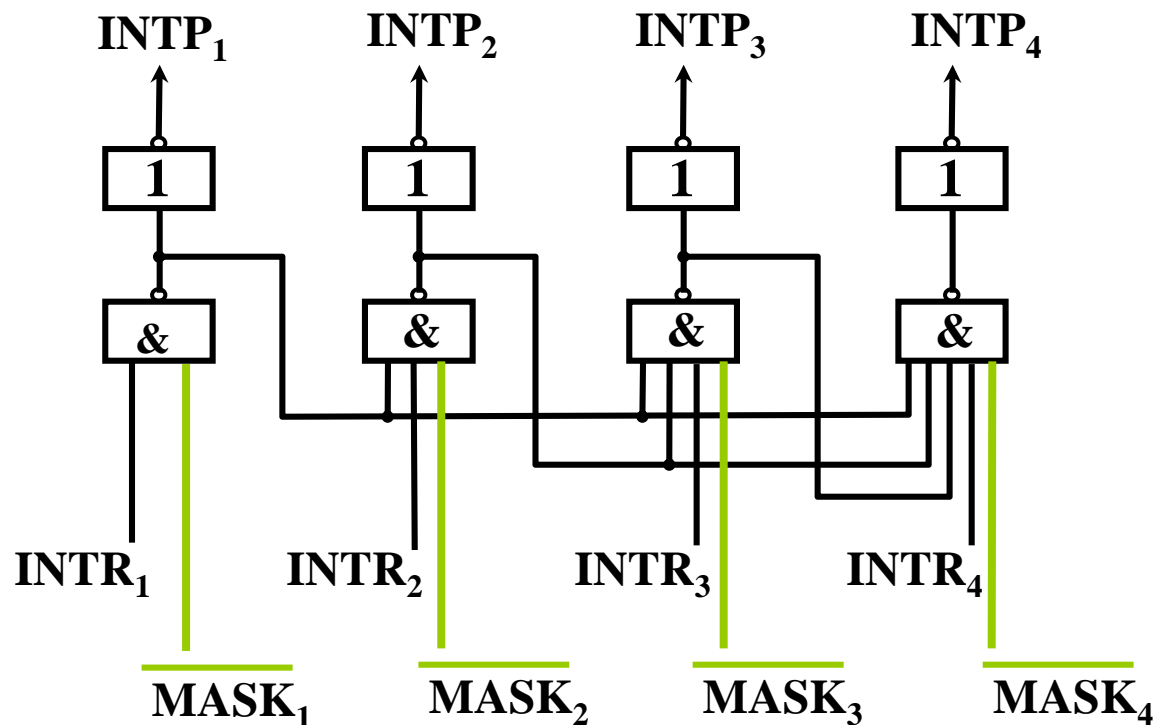
(2) 优先级别高的中断源 有权中断优先级别低的中断源



### 3. 屏蔽技术



#### (1) 屏蔽触发器的作用-动态优先级设定



**MASK = 0** (未被屏蔽)

**MASK<sub>i</sub> = 1** (被屏蔽)

**INTR** 被置 “1”

**INTP<sub>i</sub> = 0** (不能被排队选中)

## (2) 屏蔽字



16个中断源 1, 2, 3 ... 16 按 **降序** 排列，每个对应16位屏蔽字

在**ISR**中设置屏蔽字，屏蔽对应中断源

优先级	屏 蔽 字															
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
5	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
6	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
⋮	⋮															
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

### (3) 新屏蔽字的设置-ISR

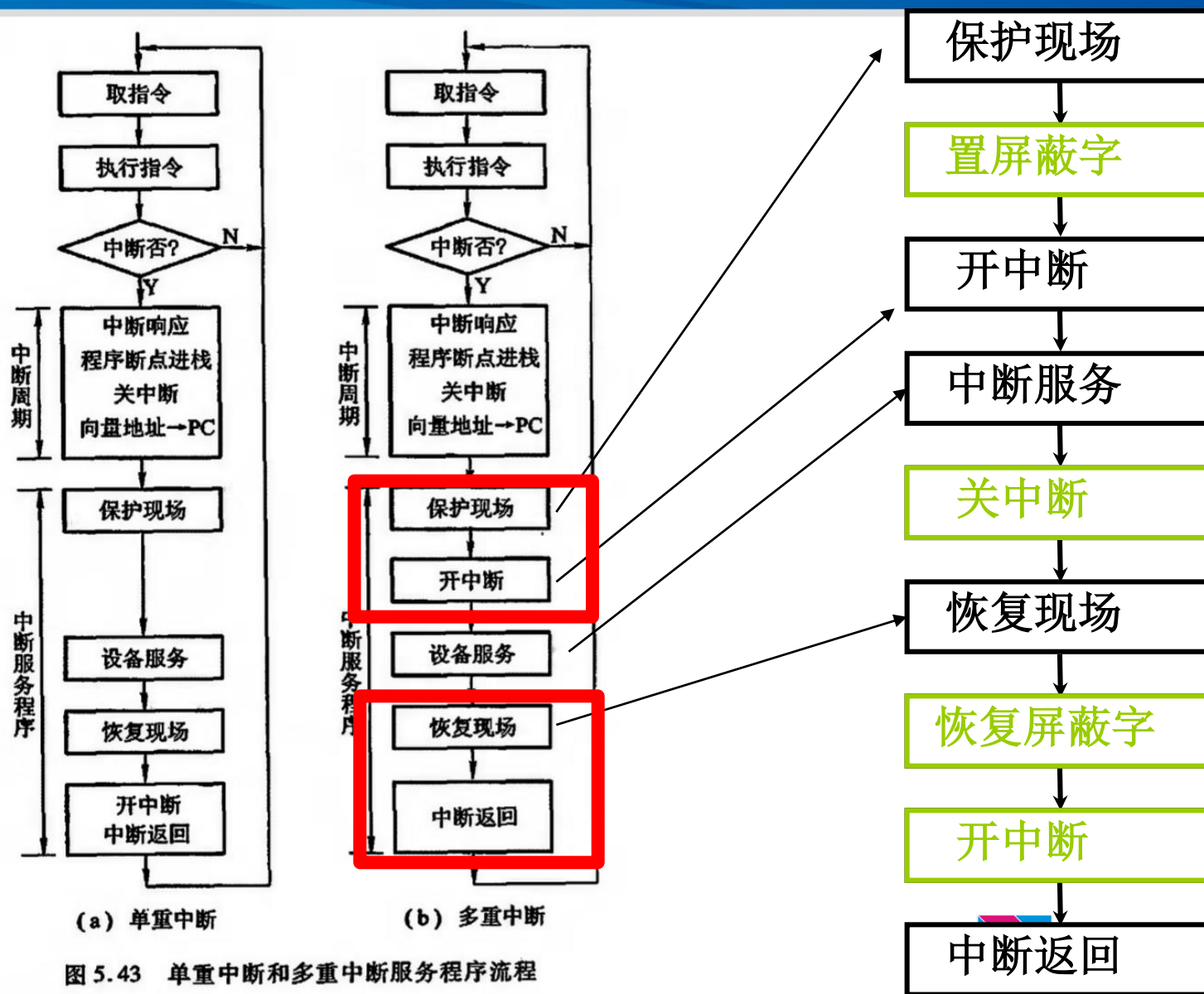


图 5.43 单重中断和多重中断服务程序流程

## (4) 屏蔽技术可改变处理优先等级



响应优先级

不可改变

为何不改变响应优先级？

处理优先级

可改变（通过重新设置屏蔽字）

中断源	原屏蔽字	新屏蔽字
A	1 1 1 1	1 1 1 1
B	0 1 1 1	0 1 0 0
C	0 0 1 1	0 1 1 0
D	0 0 0 1	0 1 1 1

响应优先级 **A→B→C→D** 降序排列

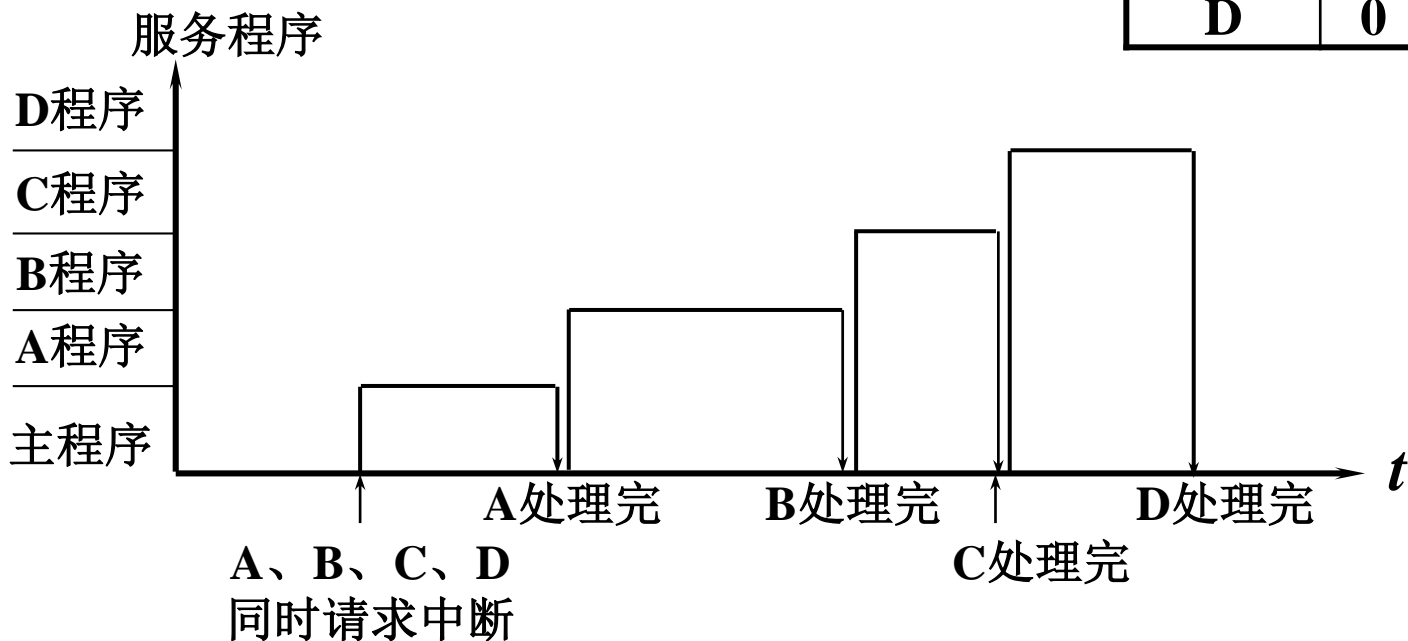
处理优先级 **A→D→C→B** 降序排列



## (4) 屏蔽技术可改变处理优先等级



中断源	原屏蔽字	新屏蔽字
A	1 1 1 1	1 1 1 1
B	0 1 1 1	0 1 0 0
C	0 0 1 1	0 1 1 0
D	0 0 0 1	0 1 1 1

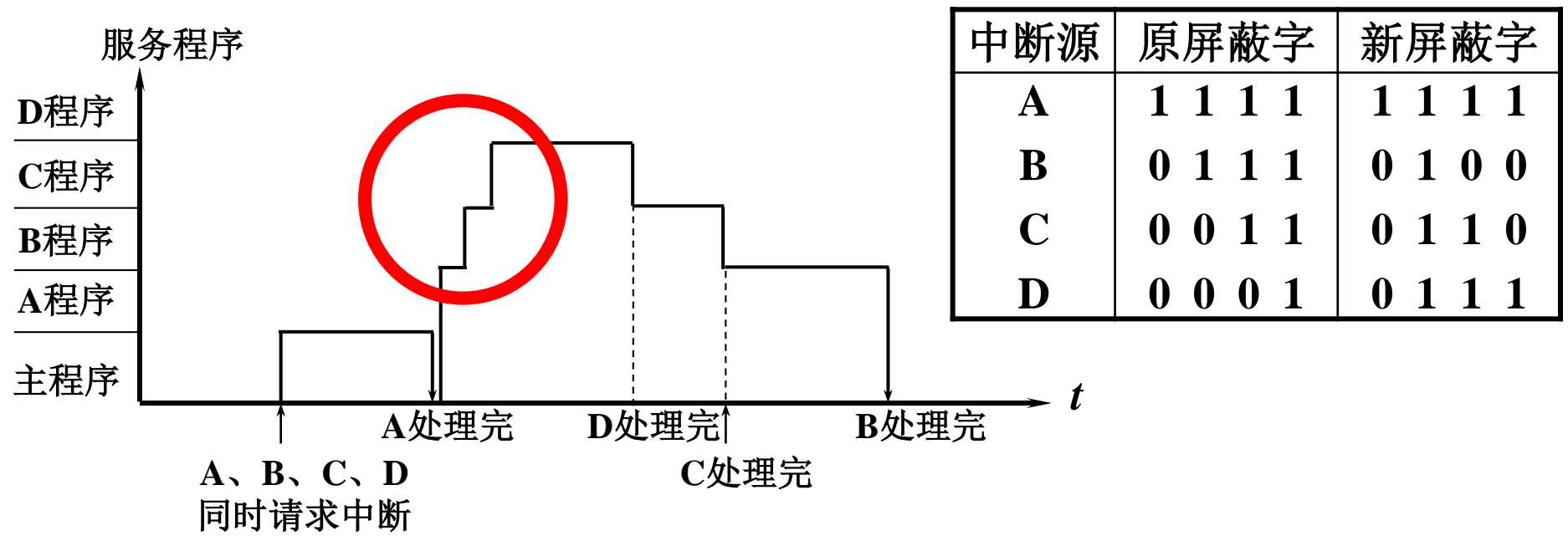


CPU 执行程序轨迹（原屏蔽字）





# (4) 屏蔽技术可改变处理优先等级



CPU 执行程序轨迹（新屏蔽字）

## (4) 屏蔽技术的其他作用

可以 **人为地屏蔽** 某个中断源的请求，便于程序控制

问：结合屏蔽字设置的响应和处理流程，分析红色圈中的“锯齿”是从何来的？