

四、 中断控制指令

四、中断指令

■ 中断的概念

- 某种异常或随机事件使处理器暂时停止正在运行的程序，转去执行一段特殊处理程序，并在处理结束后返回原程序被中断处继续执行的过程。

中断源

■ 中断指令：

- 引起CPU产生一次中断的指令

中断与过程调用：

■ 相似点：

- 从一个正在执行的过程转向另一个过程（处理程序），并在执行完后返回原程序继续执行

■ 区别：

- 中断是随机事件或异常事件引起，调用是事先已在程序中安排好；
- 调用指令在指令中直接给出子程序入口地址，中断指令只给出中断向量码，入口地址则在向量码指向的内存单元中。
- 调用可以是近过程调用或远过程调用，中断处理程序均为远过程；
- 响应中断请求不仅要保护断点地址，还要保护 FLAGS 内容。

1. 中断指令

- 格式:

INT n

中断类型码
n=0 ~ 255

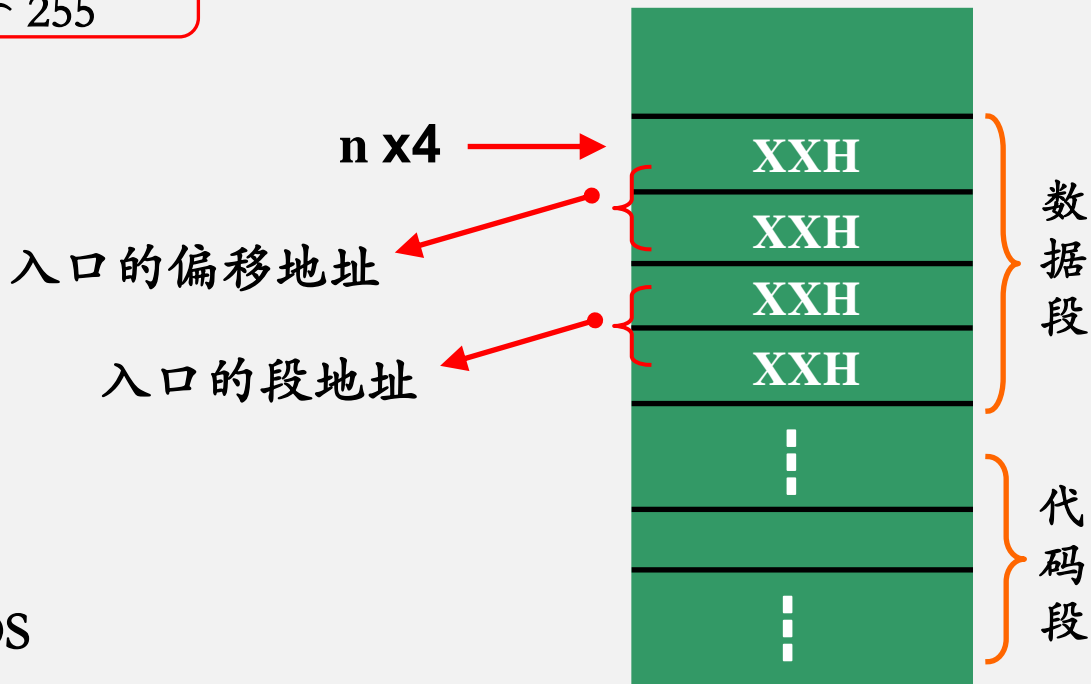
- 说明:

- nx4



存放中断服务子程序入口
地址的单元的偏移地址

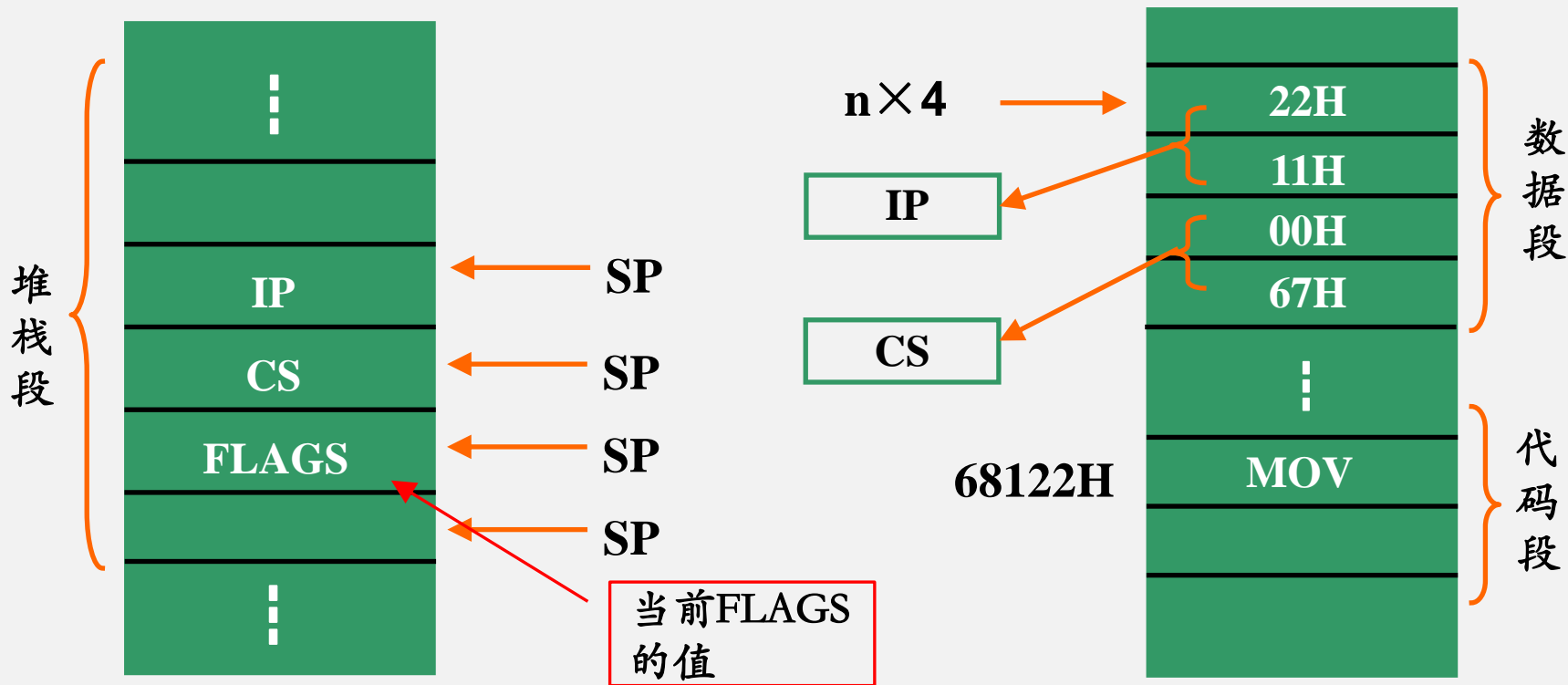
该单元在数据段, 段地址=DS



中断指令的执行过程

- ① 将FLAGS压入堆栈；
- ② 将INT指令的下一条指令的CS、IP压栈；
- ③ 由 $n \times 4$ 得到存放中断向量的地址；
- ④ 将中断向量（中断服务程序入口地址）送CS和IP寄存器；
- ⑤ 转入中断服务程序。

中断指令的执行过程



中断指令例

执行程序段：

CS IP

⋮

6200H:010DH MOV SP, 1200H

6200H:0110H INT 21H

6200H:0112H MOV AX, BX

⋮

执行INT
指令后

SP=11FAH

SP=11FCH

SP=11FEH

SP=1200H

12H

01H

00H

62H

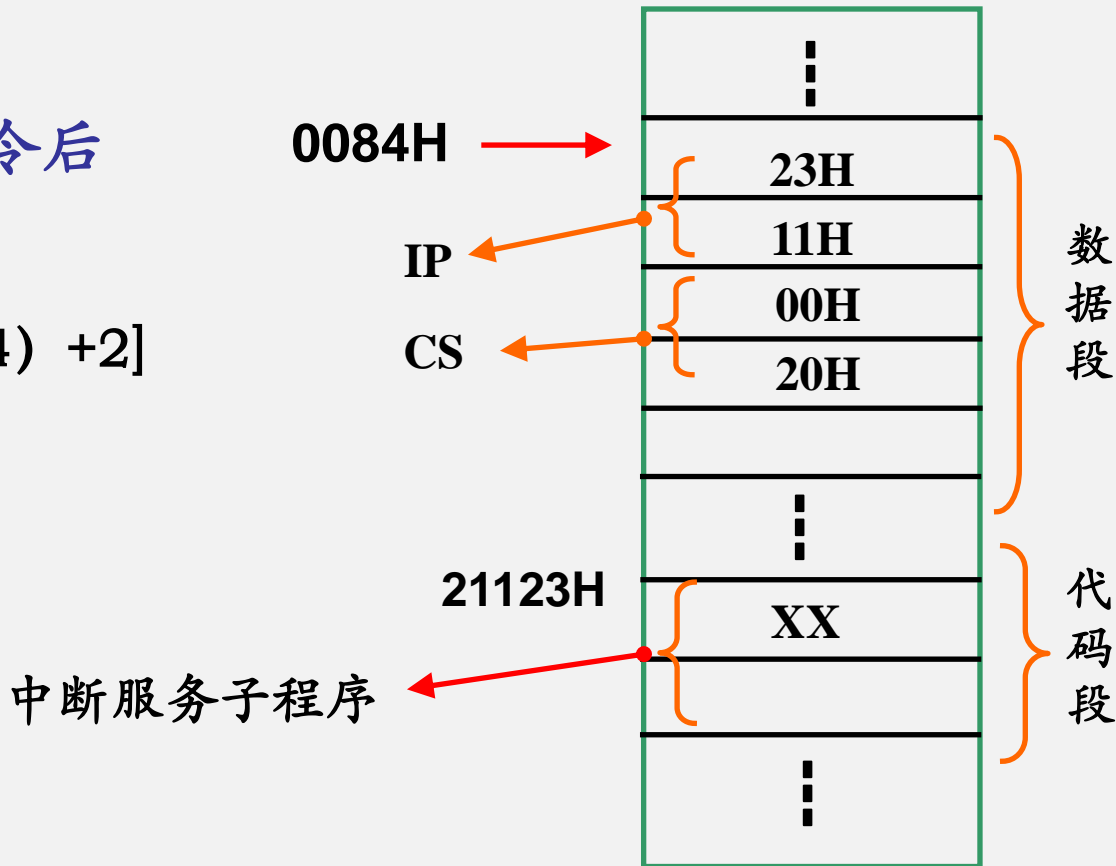
FLAGS

堆
栈
段

中断指令例

■ 执行INT 21H指令后

- $IP = [21H \times 4]$
- $CS = [(21H \times 4) + 2]$



2. 中断返回指令

- 格式：
 - IRET
- 中断服务程序的最后一条指令，负责：
 - 恢复断点
 - 恢复标志寄存器内容

