

5.4 中断方式与接口

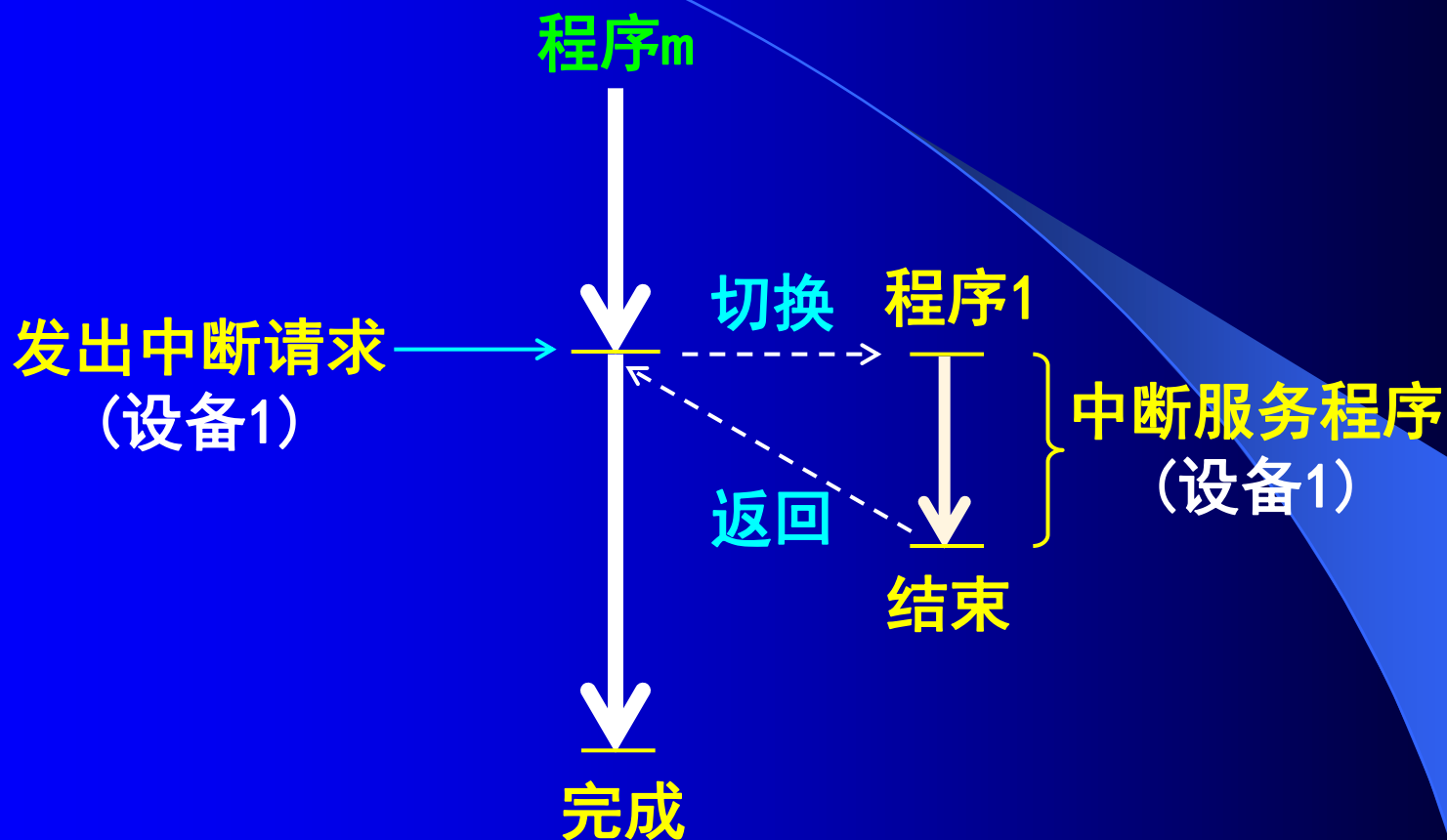
5.4.1 中断基本概念

1. 定义

在程序运行过程中, 如果发生某种随机事态, CPU暂停当前程序(被中断), 转而执行该事态对应的服务程序, 结束后再恢复原程序的执行。

特征: 程序切换+随机性

中断的基本原理



CPU: 程序m → 程序1 → 程序m

中断的实质与特点

(1) 实质

程序切换

方法：保存断点，保护现场；
恢复现场，返回断点。
时间：一条指令结束时切换。
保证程序的完整性。

(2) 特点

随机性

随机发生的事态(按键、故障)
有意调用, 随机请求与处理的事态(调用打印机)
随机插入的事态(软中断指令插入程序任何位置)

注意中断与子程序调用的区别。

※两者的区别:

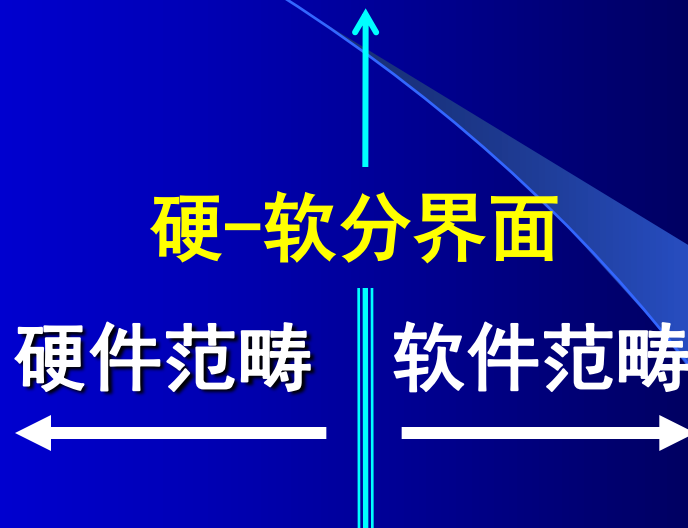
- 子程序的执行由程序员事先安排，而中断服务程序的执行则是由随机中断事件触发。
- 子程序的执行受主程序或上层程序控制，而中断服务程序一般与被中断的现行程序无关。
- 一般不存在同时调用多个子程序，但可能发生多个外设同时向CPU发出中断服务请求的情况。

2. 中断的典型应用

- (1) 管理中、低速I/O操作, 实现主-外并行工作
- (2) 以软中断方式来处理系统调用, 如读写磁盘
- (3) 故障处理
- (4) 实时处理, Real-Time Processing
对各事件以足够快的速度进行处理, 并在允许的时间尺度内作出反应。
- (5) 人机对话
- (6) 多机通信

3. 中断系统的软硬件组织

(1) 软件：中断服务程序、中断向量表



(2) 硬件 { 接口方面：请求、屏蔽、传递、判优等逻辑
CPU方面：对中断请求的响应逻辑

【例子】一种典型的中断组织方法

(1) 列出系统中的各种中断请求，如：

外部中断源：

IREQ0-系统时钟，如日历钟；**IREQ1**-实时时钟，供实时处理用；**IREQ2**-通信中断，组成多机系统或联网时用；**IREQ3**-键盘；**IREQ4**-CRT显示器；**IREQ5**-硬盘；**IREQ6**-软盘；**IREQ7**-打印机

内部中断源：

掉电、溢出中断、校验错误等引起的中断。

软中断：**INT 1~INT n**，可以根据需要进行扩充，作为系统的功能调用命令。

(2) 为各中断源编制**中断服务程序**；

(3) 将中断服务程序的入口地址写入**中断向量表**；

4. 中断方式的分类

(1) 硬件中断与软中断

硬中断: 由**硬件**请求信号引发的中断

软中断: 由**软件**触发的中断

(2) 内中断与外中断

内中断: 中断源来自**主机内部**

外中断: 中断源来自**主机外部**

(3) 强迫中断与自愿中断

强迫中断: 由故障和外部源引起，非程序中的安排

自愿中断: 自中断或软中断，在程序中有意安排的

(4) 可屏蔽中断与非屏蔽中断

可屏蔽中断: 可以通过屏蔽字屏蔽该类请求, 关中断时不响应该类请求。

非屏蔽中断: 该类请求与屏蔽字无关; 请求的响应与开/关中断无关。

(5) 向量中断与非向量中断

向量中断: 由硬件方式来确定服务程序入口地址;

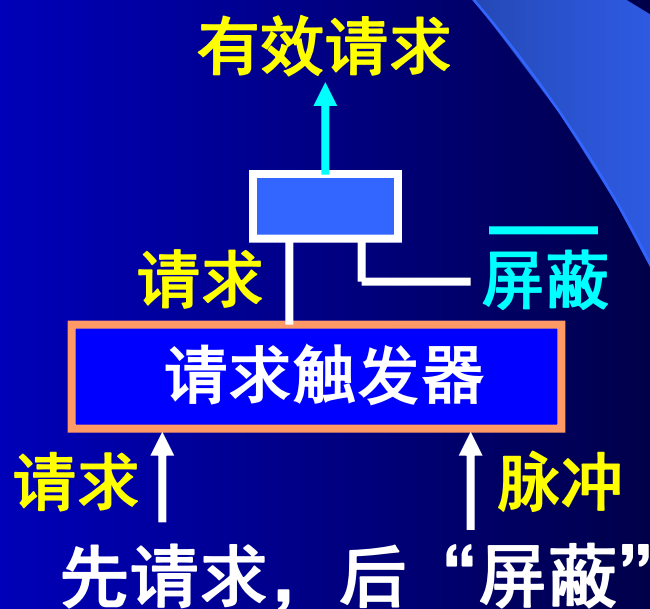
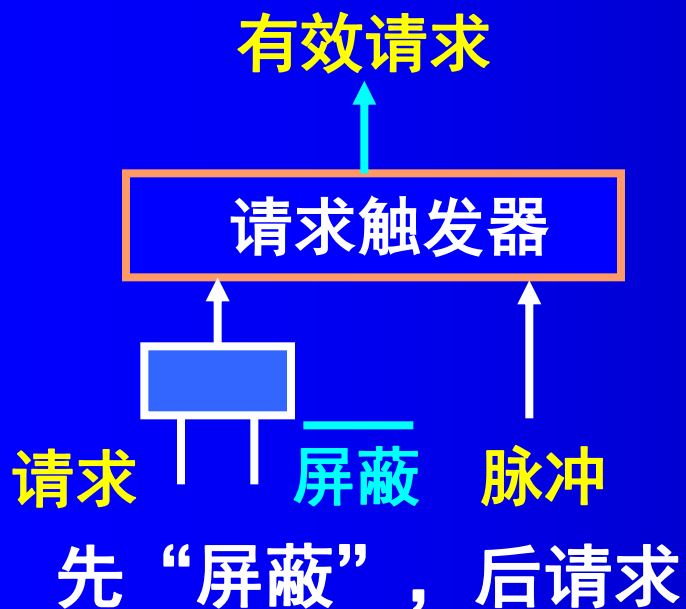
非向量中断: 由软件查询确定服务程序入口地址;

5.4.2 中断全过程（外中断）

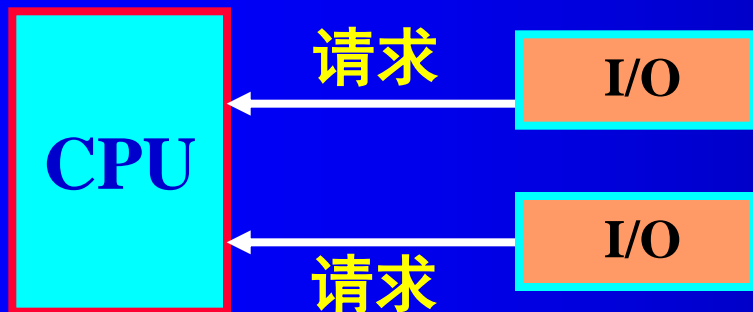
1. 中断请求的提出与传递

(1) 能产生中断请求的前提条件？

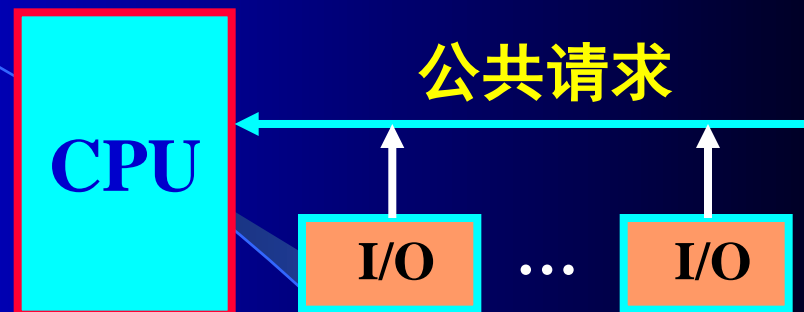
- 外设有中断请求需要：例如**准备就绪**、**任务完成**等；
- 该中断请求未被屏蔽：例如**屏蔽标志为0**



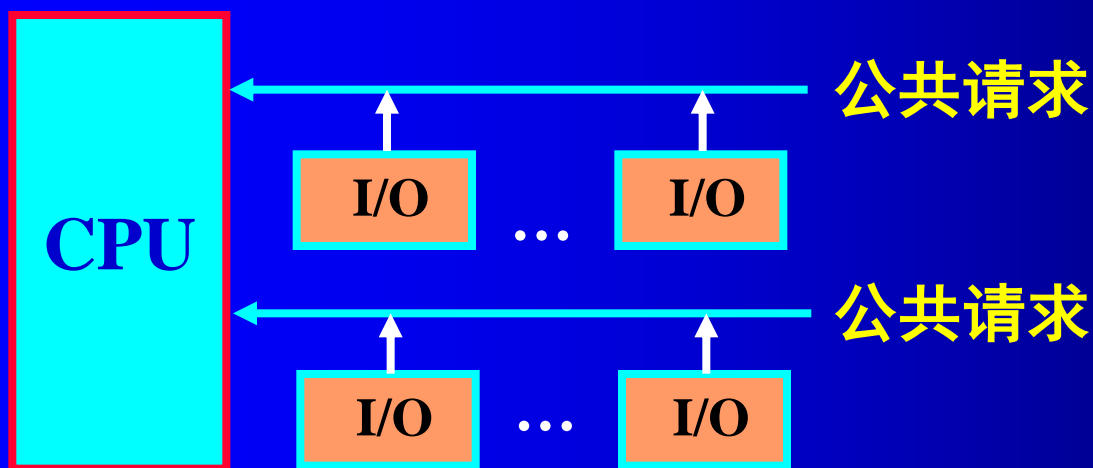
(2) 如何传送中断请求?



1) 使用单独请求线



2) 使用公共请求线



3) 混合方式传送

2. 中断请求优先级的判断

(1) 中断源优先级顺序的安排

故障 > DMA > 外部设备引起的中断

基本原则：高速操作优先于低速操作，输入优先于输出。

(2) CPU当前程序与外设请求的判优

- 先查询CPU的允许中断标志 $\begin{cases} =1, & \text{开中断} \\ =0, & \text{关中断} \end{cases}$
- 再分析当前程序的优先级（PSW字段）

现行程序的优先级 $\begin{cases} < \text{外设请求优先级, 响应} \\ \geq \text{外设请求优先级, 不响应} \end{cases}$

(3) 多个中断请求之间的判优

1) 软件判优

由程序查询顺序确定优先级。

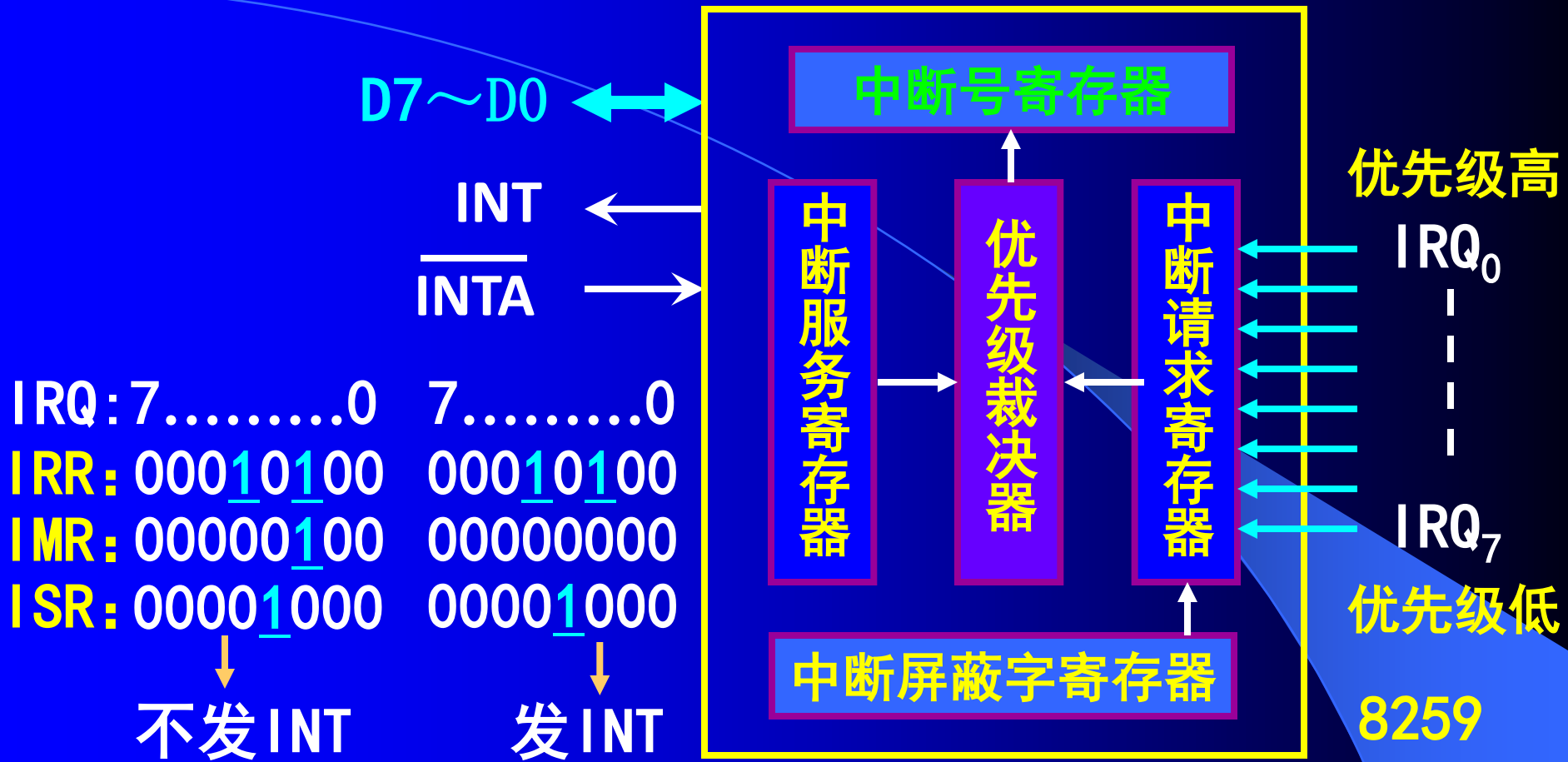
可灵活修改优先级。

2) 硬件判优

[例] 通过中断控制器判优

中断控制器 (如8259) 集中解决请求信号的接收、屏蔽、判优、编码等问题。

(如后图)



中断请求 → 8259 (屏蔽、判优, 生成相应中断号备用)
 → 公共请求INT → CPU 中断源的序号
 (CPU响应后, 取回中断号, 转入相应服务程序)

3. CPU对中断请求的响应

(1) 响应条件

- ✓有未被屏蔽中断请求到达；
- ✓CPU处于开中断模式；
- ✓中断源优先级比当前程序的优先级更高；
- ✓CPU刚执行完一条指令（非停机）；

(2) 如何形成中断服务程序的入口地址？

取决于中断方式是 { 向量中断
非向量中断

1) 非向量中断(通过程序)

将所有中断源的中断服务程序入口地址组织在**公共查询程序**中；CPU响应时执行此查询程序，确定中断源对应的服务程序入口地址。

2) 向量中断(通过硬件)

将所有中断源的中断服务程序入口地址(**中断向量**)组织在**中断向量表**中；CPU响应时由硬件产生**向量地址**，据此查**中断向量表**确定服务程序入口地址。

- **中断向量**：中断服务程序的入口地址+状态字PSW；
- **中断向量表**：用来存放中断向量的表(**一段存储区**)
- **向量地址**：用来访问向量表的地址(也叫**中断指针**)

【例1】模型机向量表

(从主存2#单元开始安排)
M按字编址，一个入口地址
16位，占一个编址单元。

向量地址 = 2# + 中断号 × 1
(单元地址)

向量表	
2#	入口地址0
3#	入口地址1
	⋮

0号中断源
1号中断源

【例2】IBM PC的向量表

(从主存0#单元开始安排)

M按字节编址。一个入口地址
32位，占4个编址单元。

向量地址 = 0# + 中断号 × 4

向量表	
0#	入口偏移0
	入口基址0
4#	入口偏移1
	入口基址1
	⋮

0号中断源
1号中断源

4、中断的响应过程（向量中断方式）

执行中断隐指令 → 中断服务程序

※中断隐指令



硬件
自动完成

※中断服务程序的处理

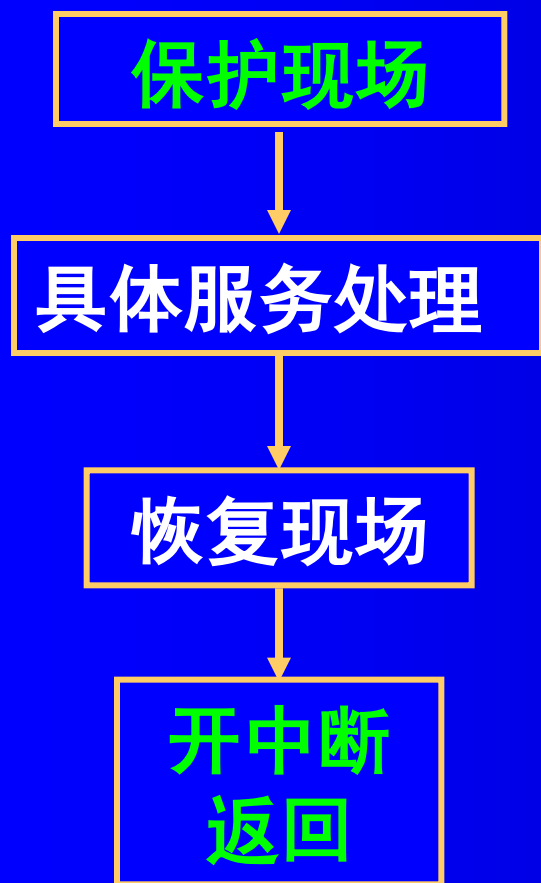
主要任务：CPU执行中断服务程序。

(1) **单级中断**：CPU响应后只处理一个中断源的请求，处理完毕后才能响应新的中断请求。

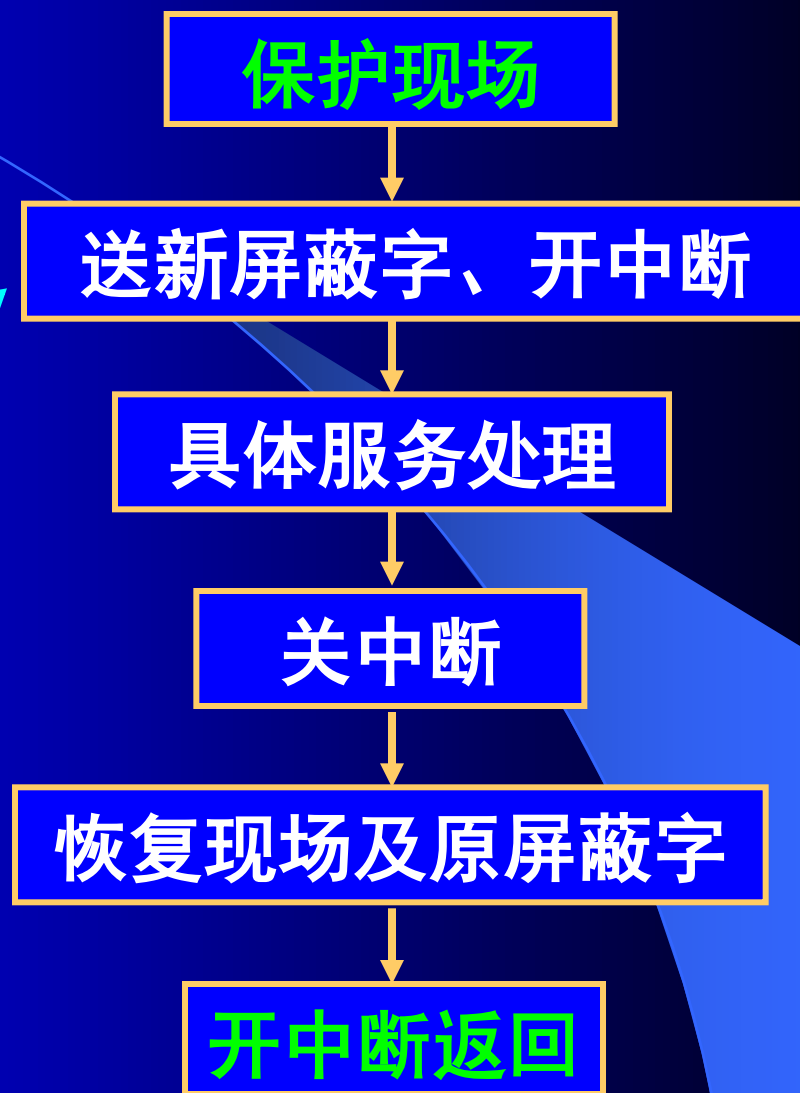
(2) **多重中断**：在某次中断服务过程中，允许响应处理更高优先级的中断请求。

	单级中断	多重中断
中断隐指令	关中断、保存断点及PSW等，取服务程序入口地址及新的PSW	关中断、保存断点及PSW等，取服务程序入口地址及新的PSW
中断服务程序	如后图所示	如后图所示

单级中断



多重中断



外中断:
数据传送

禁止同级或
更低级的中
断请求, 允
许更高级的
请求

【注意】在响应过程、保护现场、恢复现场等关键阶段, 应关中断响应以防止被打扰。

5.4.3 中断接口模型

1. 组成（寄存器级）

（1）寄存器选择

对接口中的寄存器寻址。

（2）命令字寄存器

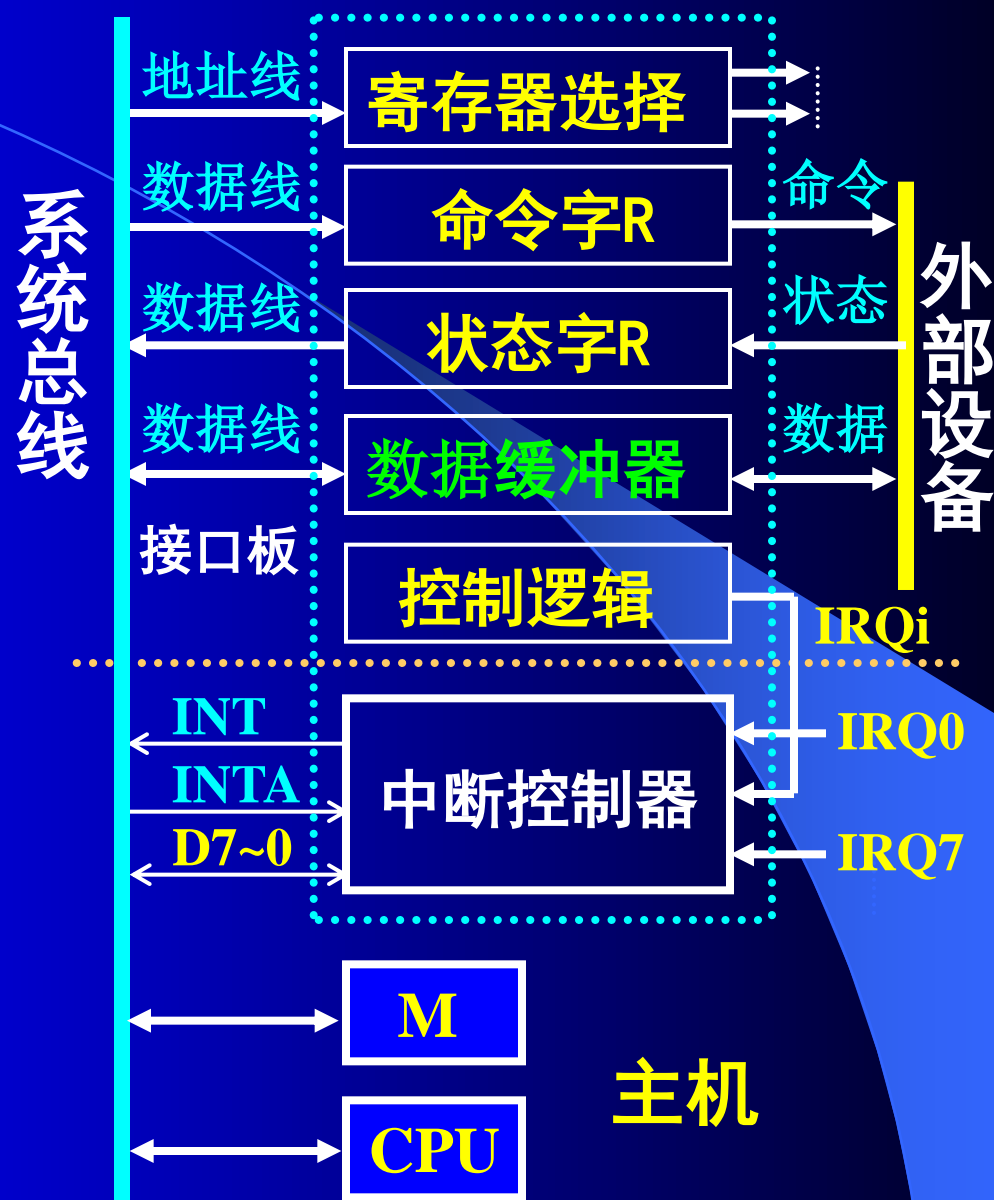
接收CPU发向外设的命令字，转换为相应操作命令送外设。

命令字格式：

用代码表示各种命令

（3）状态字寄存器

反映设备和接口的各种运行状态，用代码表示状态



(4) 数据缓冲器

传送数据，实现缓冲

(5) 控制逻辑

请求信号产生逻辑

电平转换逻辑

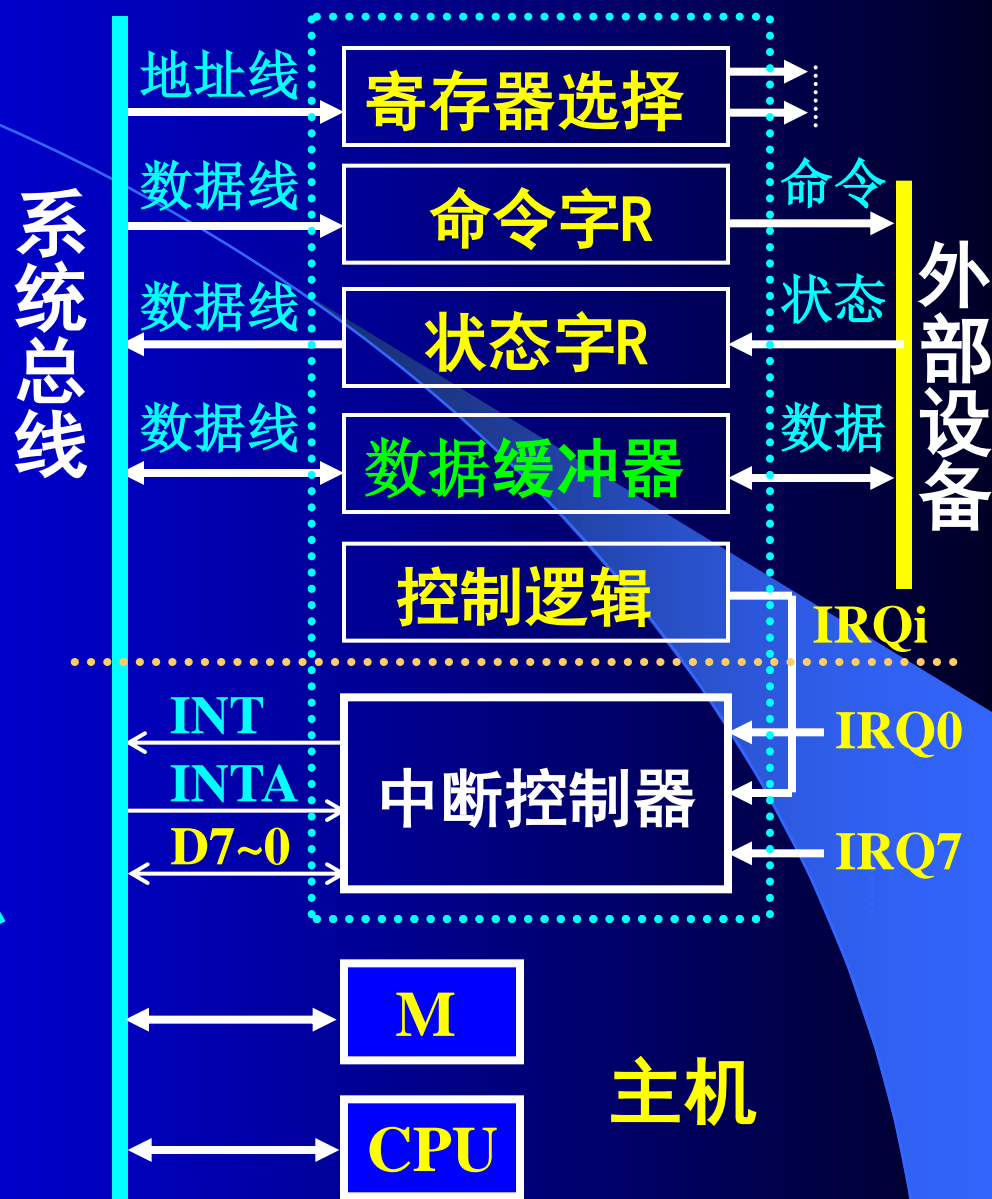
串-并转换逻辑(串口)

扩展中断源数量

(6) 公用中断控制器

接收外设请求、屏蔽、判优、送出公共请求；

接收中断批准，送出中断号
(中断类型码)。



2. 工作过程（外中断）

(1) 初始化：设置工作方式，送屏蔽字，分配中断类型码。

(2) 发启动命令（送命令字），启动设备。

(3) 设备就绪，申请中断。

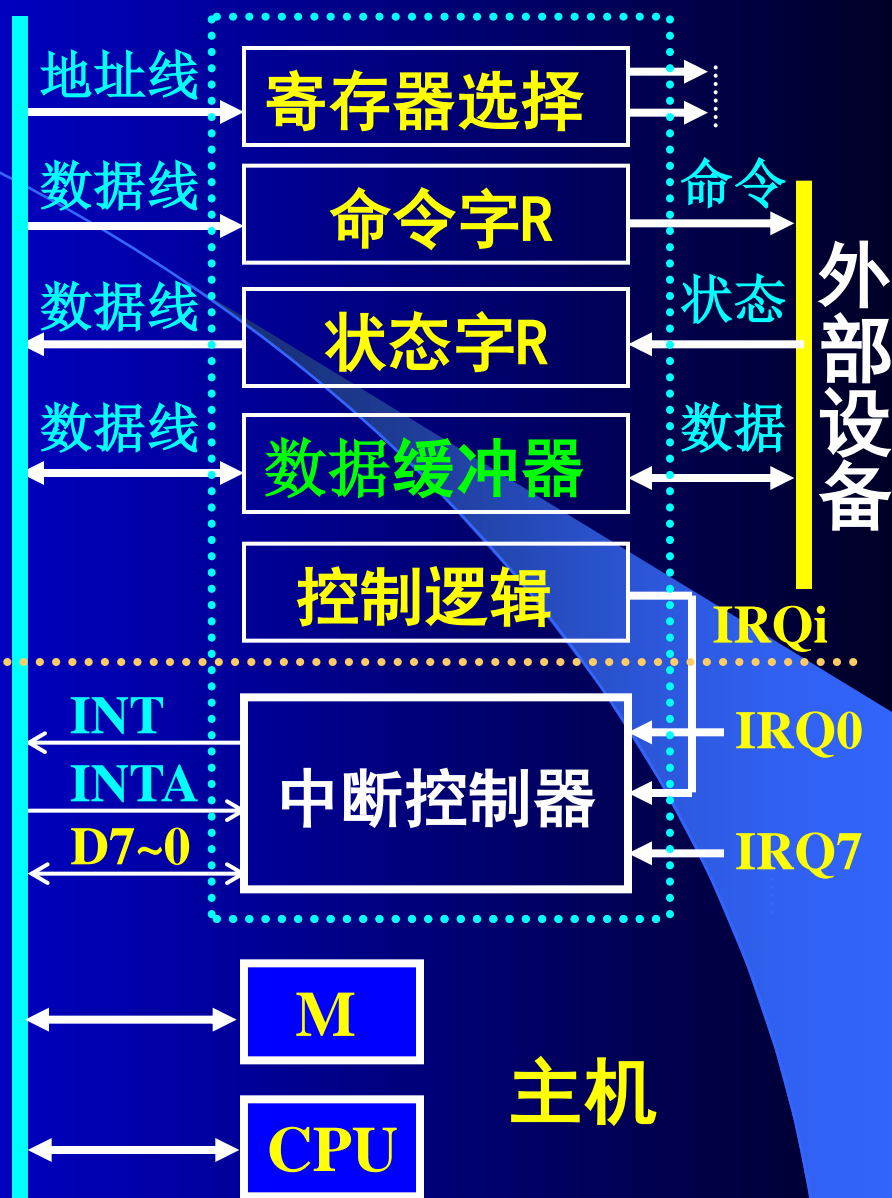
(4) 中断控制器汇集各请求，经屏蔽、判优，形成中断号，向CPU发出INT。

(5) CPU响应，发批准INTA。

(6) 中断控制器送出中断号。

(7) CPU执行中断隐指令操作，准备执行服务程序。

系统总线



3. 中断接口的设计

【例子】模型机需扩展两个外中断源，共用一个中断号。

主机发向外设的命令包括：启动、停止、数据选通；

外设的状态包括：忙、完成、出错。

【任务】请为这两个外中断源设计中断接口方案。

【分析】结合中断系统的工作场景，考虑设计此中断接口时，需要解决哪些方面的问题？

- ✓ 使用哪个中断连接端口？
- ✓ 控制命令、状态信息如何编码？
- ✓ 接口的寄存器级结构模型？
- ✓ 分析其工作过程？

(1) 确定中断端口

两个中断源共用一个端口：

IRQ2



(2) 拟定各命令和状态字的编码

5	4	3	2	1	0
启动1	停止1	选通1	启动2	停止2	选通2

命令字格式：

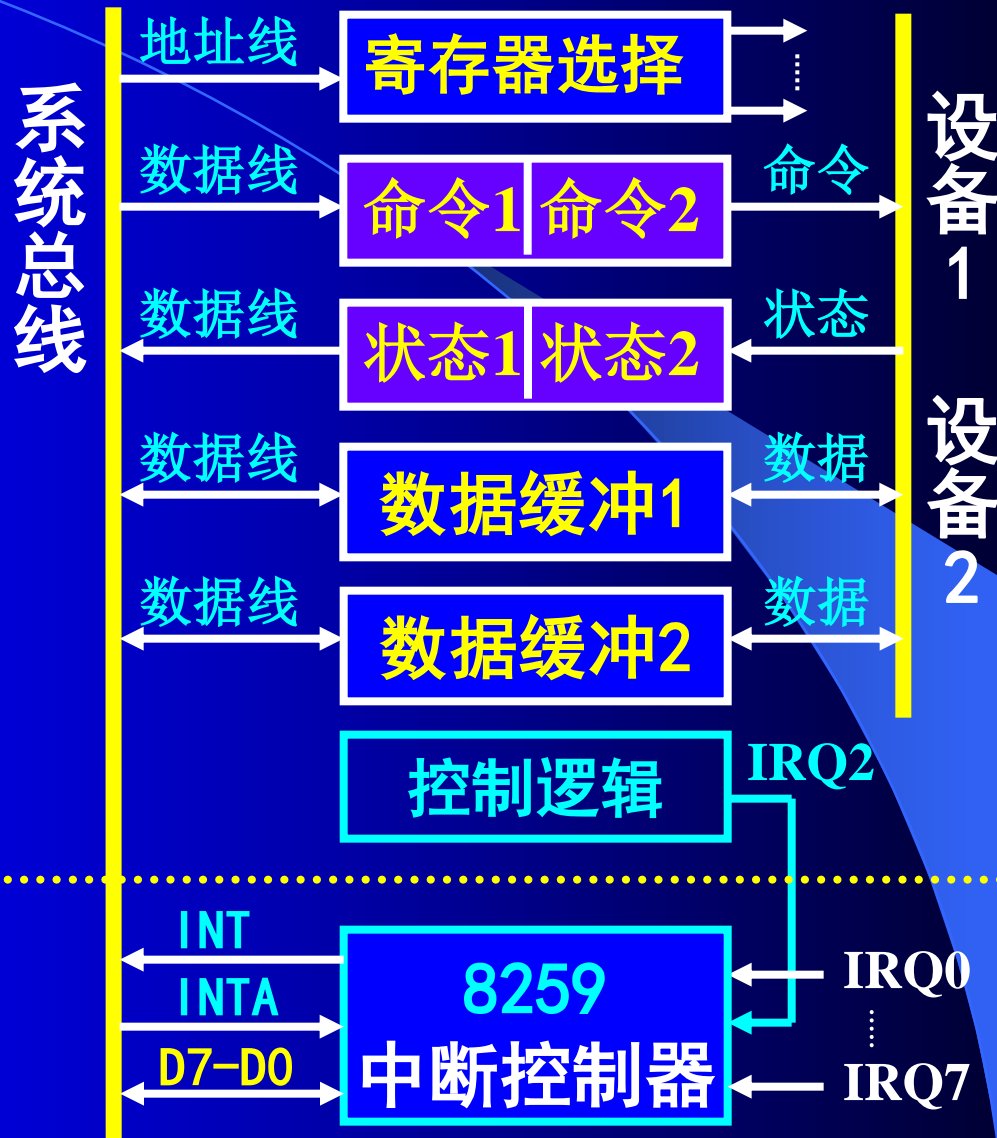
忙1	完成1	出错1	忙2	完成2	出错2
----	-----	-----	----	-----	-----

状态字格式：

【注意】如寄存器宽度为16，则其第15-6位(即高10位)未被使用(待扩展)。

(3) 画出接口方案模型

接口方案如右图



(4) 分析工作过程

向量中断+非向量中断

请求1与请求2在控制逻辑中形成公共请求**IRQ2**，送入8259进行屏蔽和判优；

CPU响应后均执行**IRQ2**的中断服务程序(向量中断方式)

CPU在**IRQ2**的中断服务程序中查询各设备状态，确定中断源，计算其入口地址，准备转入执行该中断服务程序(非向量中断方式)

系统总线

