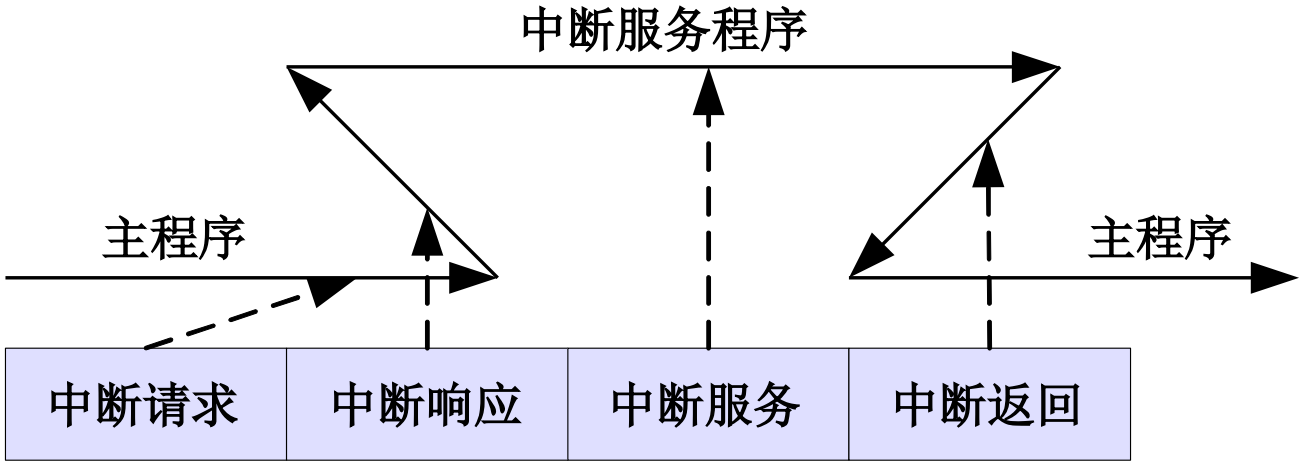
2、中断过程

- 中断过程包含4个阶段



2、中断过程

① 中断申请

I. 对于外中断，外设或其他中断源通过 CPU 的中断请求引脚向 CPU 发中断请求信号， CPU 在每条指令执行完后，监测是否有中断请求，有效则转入中断响应阶段。

II. 对于内中断，则无需中断请求，直接可以根据中断类型号转入相应的中断服务程序。

2、中断过程

① 中断申请

III. 需解决的主要问题是：

a）**中断屏蔽**：对那些 CPU 目前不准备响应的中断源， CPU 如何禁止它们产生中断请求？

b）**中断请求信号的传递**：当系统中有多个中断源时，各中断源如何向 CPU 提出中断请求？

c）**CPU 对中断请求信号的监测**： CPU 如何监测到有中断请求？

2、中断过程

② 中断响应

I. CPU 首先通过硬件保存程序断点（PC）及标志寄存器，以便中断返回，由于该过程对软件设计者是透明的，因此又称为 CPU 执行了中断隐指令。然后进入中断响应周期，或者通过向量方式或者通过软件查询方式得到中断服务程序入口，并置入 PC 。

II. 需解决的主要问题是：

a）**中断优先级的判别**：如果同一时刻有多个中断源向 CPU 申请中断， CPU 首先响应那个中断？

b）**中断源的识别**： CPU 如何知道当前响应的是哪个中断源？即：转入哪个中断源的中

2、中断过程

③ 中断服务

I. 中断服务程序中，首先保护现场，将有关寄存器的内容压栈，然后进行 I/O 操作，实现数据传送。最后，恢复现场，并执行中断返回指令。

II. 需解决的主要问题是中断嵌套。如果 CPU 在执行某个中断服务程序的过程中，又发生新的中断请求，那么 CPU 如何处理？

④ 中断返回

中断返回指令的功能是：将中断隐指令保存的程序断点和标志读出并送入 PC 和标志寄存器，从而回到 CPU 原来的程序断点处继续执

3、中断的作用

① 实现 CPU 和多台 I/O 设备并行工作

② 具有处理应急事件的能力

③ 进行实时处理

④ 实现人机通信

⑤ 实现多道程序运行和分时操作

⑥ 实现应用程序和操作系统（管态程序）的联系

⑦ 实现多机系统中各处理机间的联系

二、中断请求与判优

1. 中断请求信号的产生与监测
2. 中断屏蔽
3. 中断请求信号的传递
4. 中断请求的排队判优

1、中断请求信号的产生与监测

- 计算机的多个中断源随机向CPU发出中断请求，计算机为每个中断源设置一个触发器，称为**中断请求触发器INTR**，当某个中断源有中断请求时，其相应的INTRi=1。中断请求信号锁存在中断请求触发器中，等到CPU响应这个中断请求后才清除。
- 由多个中断请求触发器构成一个**中断请求寄存器IRR**，IRR每一位对应一种中断源。中断寄存器的内容称为**中断字**，中断字中为“1”的位表示对应的中断源存在中断请求。

1、中断请求信号的产生与监测

- CPU 在每条指令执行完毕后，通过检测 CPU 的中断请求引脚是否有效来达到监测目的。
- 大多数 CPU 具有若干个中断请求引脚，用以监测是否有中断发生。譬如，80X86CPU 有 INTR 和 NMI 两条中断引脚。

①INTR 是可屏蔽的中断请求引脚，受程序状态字 Flags 的 IF 位（中断使能标志）的影响：IF=0，CPU 禁止响应 INTR 引脚上的中断请求；IF=1，CPU 允许响应 INTR 引脚上的中断请求。

②NMI 是不可屏蔽的中断请求引脚，不受 IF 的影响，一旦从该引脚引入的中断源有中断请求，CPU 将会立即响应。

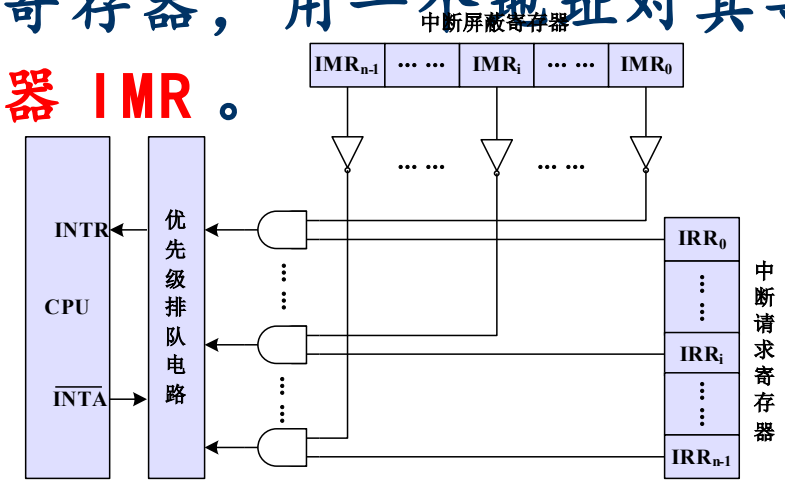
2、中断屏蔽

- 一般中断系统中，对应每一个中断源设置一个中断屏蔽触发器 $INTM_i$，以实现对单个中断源的屏蔽控制。

  ①$INTM_i$=1，则中断源 i 被屏蔽。

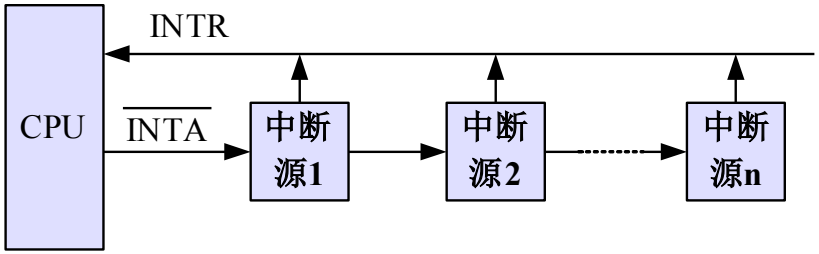
  ②当 $INTM_i$=0，则中断源 i 被开放。

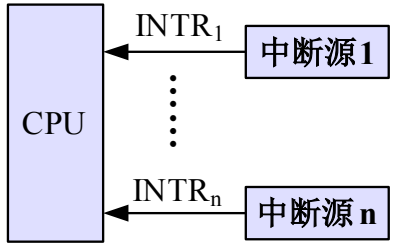
- 将中断系统中的所有中断源的屏蔽触发器放在一起，形成一个寄存器，用一个地址对其寻址，称为屏蔽寄存器 IMR。



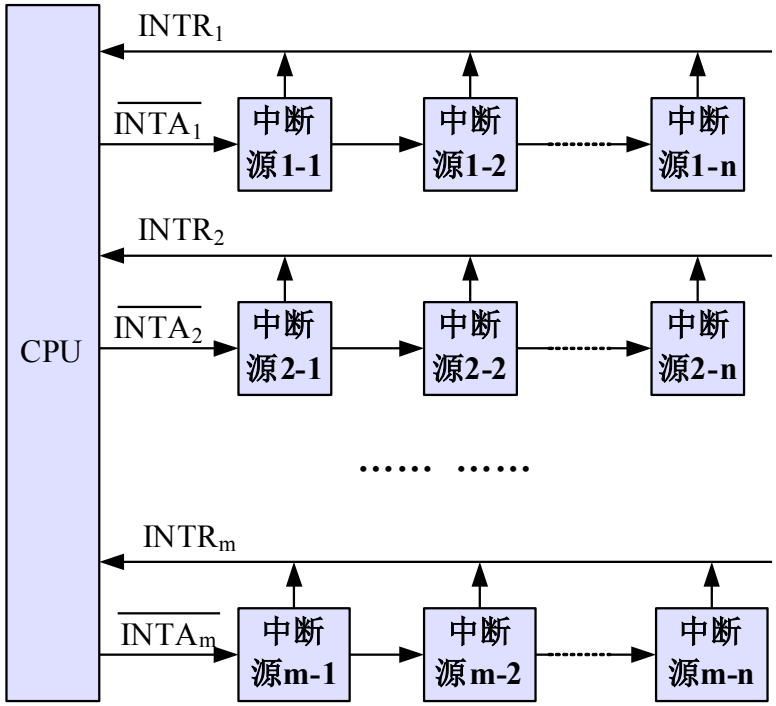
3、中断请求信号的传递

公共中断请求线





独立中断请求线



二维结构中断请求

4、中断请求的排队判优

- 中断的优先级是指有多个中断同时发生时，CPU 对中断源响应的次序。
- 确定中断优先级的原则是：

①对一旦提出请求需要立刻响应处理，否则就会造成严重后果的中断源，规定最高的优先级

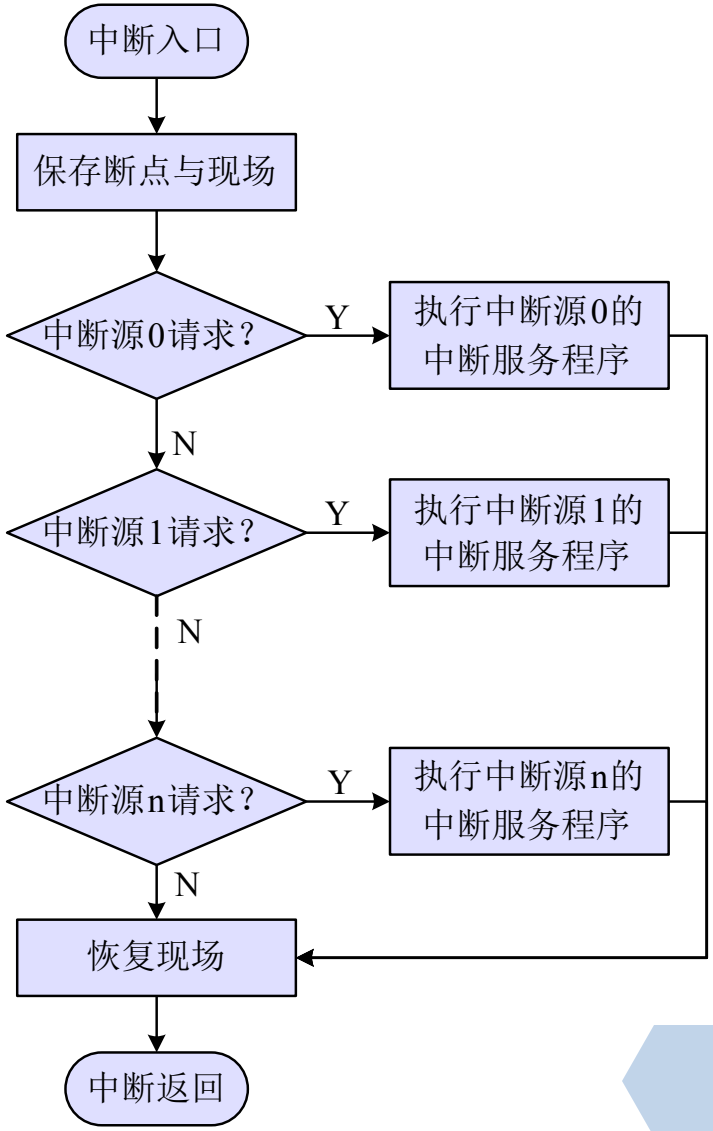
②对可以延迟响应和处理的中断源，规定较低的优先级。

③一般，把硬件故障引起的中断优先级定为最高，其次是软件故障中断和 I/O 中断。

4、中断请求的排队判优

- 中断请求的排队判优，常用的方法有两种：软件查询和硬件排队电路

① 软件查询法：用程序来判断优先级，这是最简单的中断判优方法。软件查询法用于一根公共请求线的情况

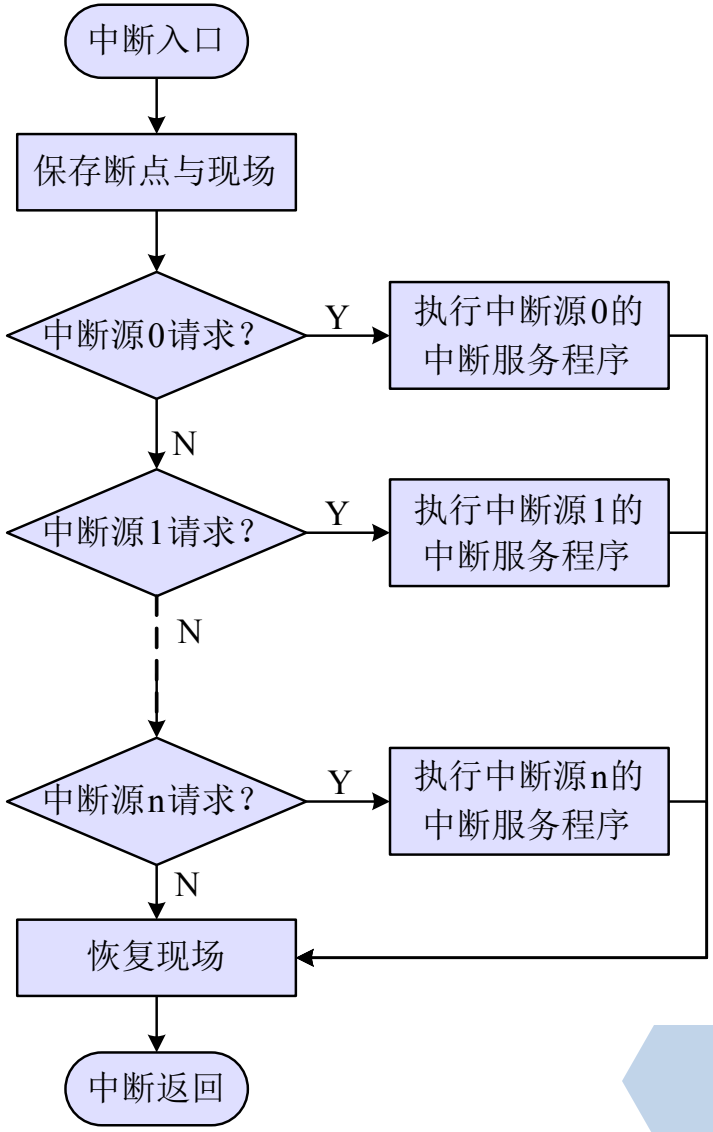


4、中断请求的排队判优

① 软件查询法

I. 优点：灵活地修改中断源的优先级别，硬件电路实现简单；

II. 缺点：查询、判优完全靠程序实现，需要占用 CPU 时间，中断响应较慢，优先级较低的设备被响应的等待时间较长。



4、中断请求的排队判优

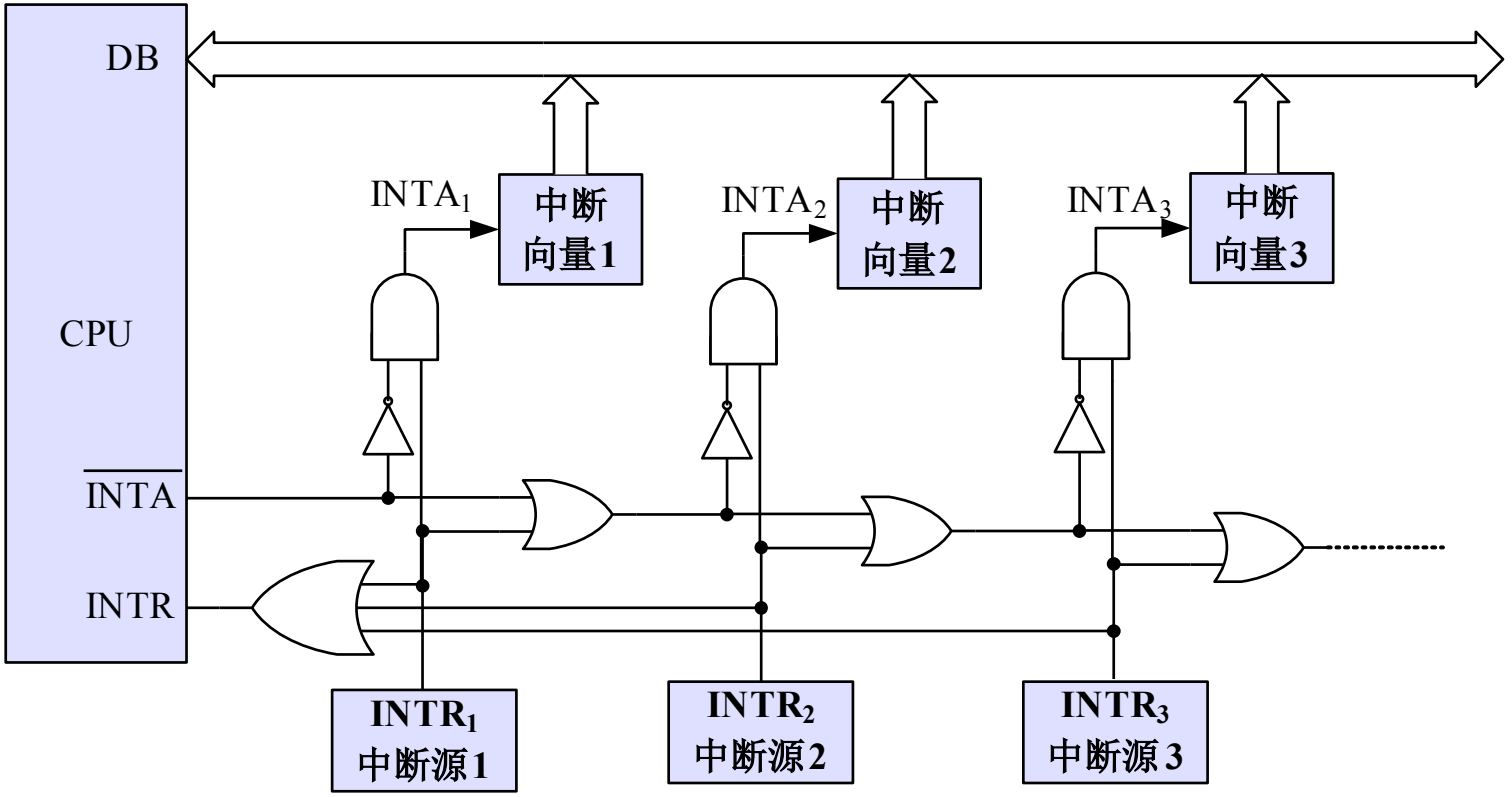
② 硬件排队电路

优先级别高的中断请求将自动封锁优先级别低的中断请求的处理。硬件排队电路一旦设计连接好之后，将无法改变其优先级别。

I. 串行排队链与向量中断

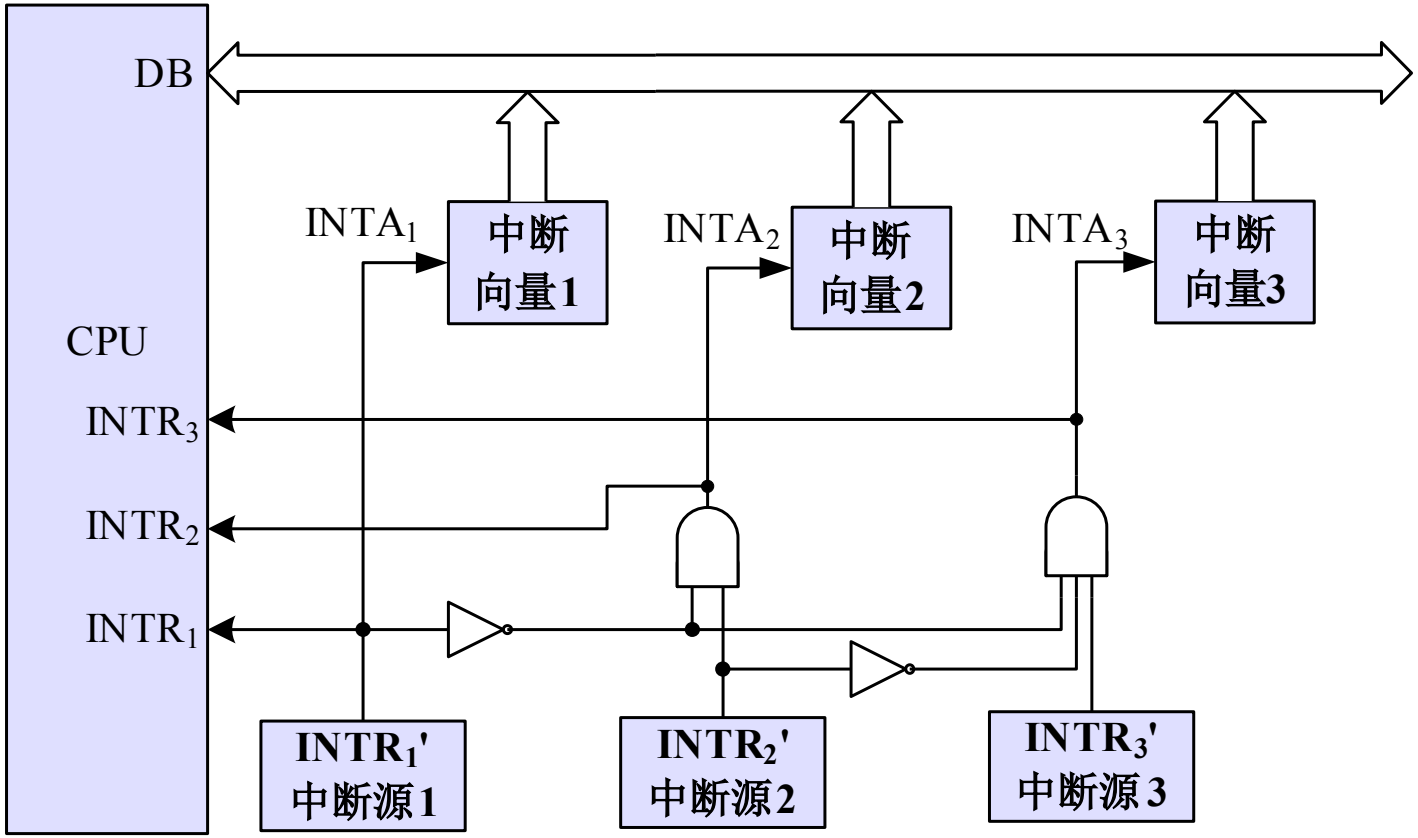
适用于向量中断方式，中断响应信号逐级传送，先到达的设备，其优先级高于中断响应信号后到达的设备，即电路中距离 CPU 最近的中断源优先级最高，这里距离远近是指电气上的信号传递顺序。这种方法实现时电路较简单，但优先级固定，取决于固定的硬件连接，不够灵活，不易于改变或调整优先级。

单线请求的串行排队判优电路



4、中断请求的排队判优

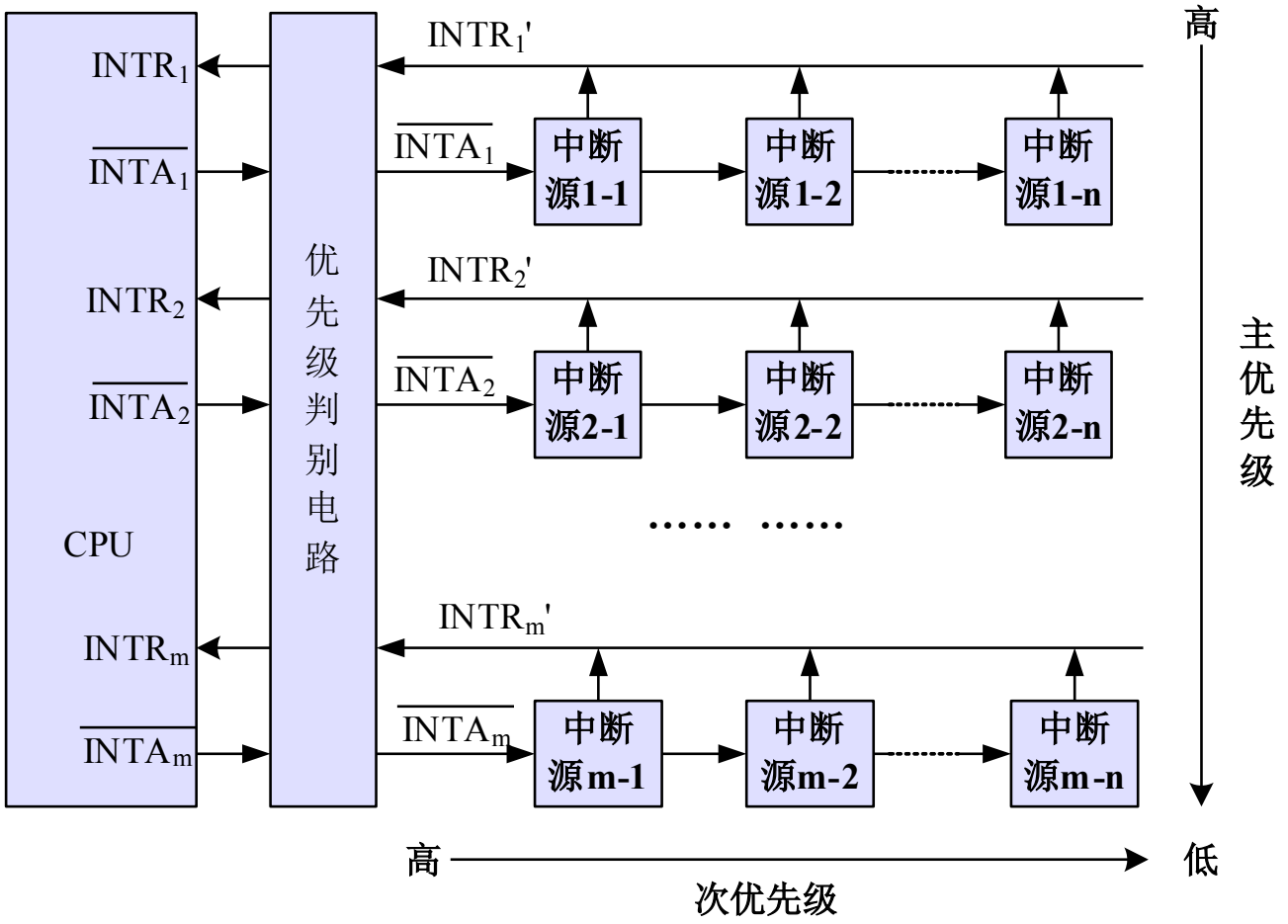
II. 独立请求优先级排队电路



多线请求的串行排队判优电路

4、中断请求的排队判优

III. 二维结构优先排队判优电路



二维结构的中断优先级排队电路

三、中断响应

1. CPU 响应中断的条件

- CPU 的中断使能触发器开放（ IE=1 ，允许中断）；
- 在规定的时间内， CPU 的中断请求引脚有效；
- 该中断未被屏蔽；
- 本条指令执行完；

2. CPU 中断响应的过程

- 关中断

当 CPU 响应中断后，立即自动关中断（把内部的中断使能触发器 IE 清零），禁止接

三、中断响应

2. CPU 中断响应的过程

- 保存断点

  断点信息包括两部分： PC 和程序状态字 PSW 。断点通常保存在堆栈中，有些计算机中将断点保存在特殊的中断返回寄存器中。

- 识别中断源，转入服务程序入口地址

  中断源识别的方法有两种：向量中断和软件查询

三、中断响应

- **向量中断**

  I. 中断服务程序入口地址被称为**中断向量**。各中断源的中断向量存放在内存一片连续的单元中，形成一张**中断向量表**，表的内容是相应的**中断服务程序入口地址**，存放中断向量的单元地址称为中断向量地址，简称为**向量地址**。

三、中断响应

- **向量中断**

II. 当 CPU 响应中断时，由硬件（外设接口或者中断控制器）自动产生一个指定的地址（向量地址）或者代码（中断类型号），它们与该中断源的中断向量有一一对应关系。由向量地址或中断类型号指出每个中断源设备的中断向量（中断服务程序入口地址），这种使用向量识别中断源的中断系统称为向量中断。

三、中断响应

② 软件查询

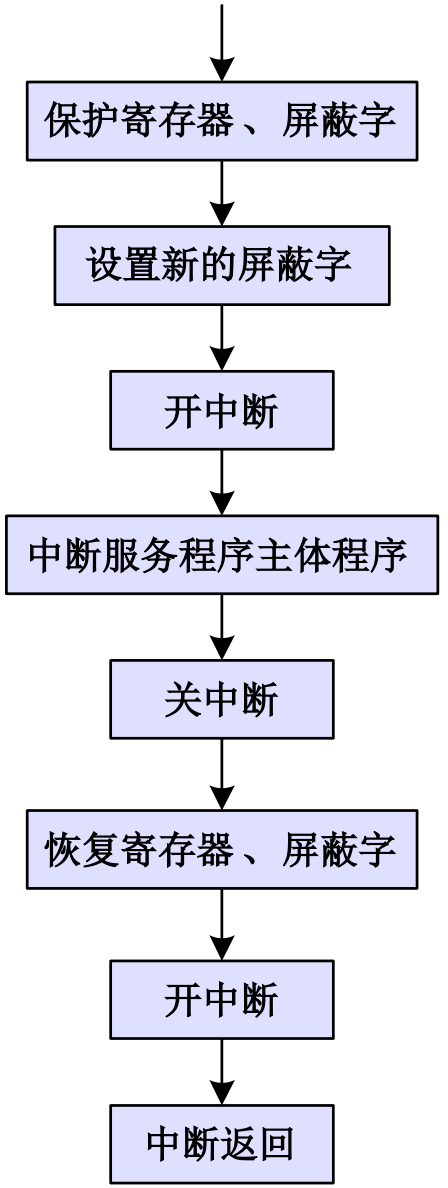
由 CPU 执行一个公共的中断处理程序，逐个询问外设接口有否发出中断请求（测试中断请求触发器），若有中断请求，则转入其中断服务程序的入口开始执行。

四、中断服务与返回

1. 中断服务程序
2. 中断服务程序与子程序的区别
3. 中断嵌套
4. 利用中断屏蔽技术修改中断优先级

1、中断服务程序

- 预处理部分
- 中断服务
- 恢复部分
- 中断返回



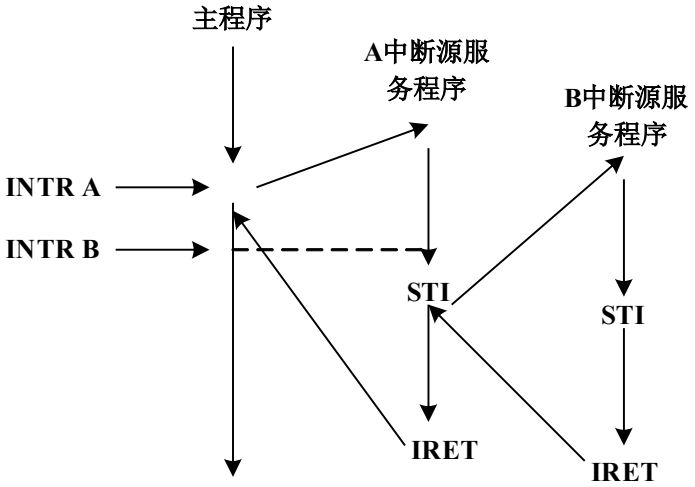
2、中断服务程序与子程序的区别

- 子程序的执行是由程序员事先安排好的（由一条调用子程序指令转入），而中断服务程序的执行则是由随机的中断事件引起的；
- 子程序的执行受到主程序或其上层程序的控制，而中断服务程序一般与被中断的现行程序毫无关系；
- 不存在同时调用多个子程序的情况，而有可能发生多台 I/O 设备同时请求 CPU 为自己服务的情况。

3、中断嵌套

- 中断嵌套是指CPU在执行某个中断服务程序的过程中允许再响应更高级别的中断请求，也称为多重中断。如果正在执行的中断服务程序中禁止再响应其他中断请求，就称为单重中断。

- 中断嵌套技术的实现，关键是在中断处理程序中必须适时开放中断（STI指令），并且使用堆栈的“先进后出”特性保证中断的逐级返回。



4、利用中断屏蔽技术修改中断优先

- 当正在进行某个中断处理时，与它同级或比它优先级低的中断请求不能被响应，只有比它优先级高的中断请求才可能被响应。
- 中断的优先级一般由硬件排队电路决定，不便于改动。但是利用中断屏蔽技术可以巧妙地改变各中断源的优先级，使计算机适应各种场合的需要。例如在图 8.12 中的中断服务程序中，通过更改中断源的屏蔽字，能动态地改变中断处理的优先级。
- 通常把屏蔽字看成软件排队，通过程序修改屏蔽字的方法可以方便地改变中断源得到 CPU 服务的先后次序，实现灵活的优先级排队。

本章小结

1. 外设的编址方式通常有统一编址和独立编址两种。
2. CPU 访问外设，即 CPU 访问外设接口。
3. 主机与外设交换信息的方式有四种：
   - 程序查询方式；
   - 程序中断方式；
   - 直接存储器访问（DMA）方式；
   - 通道与输入输出处理机（IOP）方式。
4. 中断技术实现了 CPU 与外设之间并行工作，提高了输入输出效率，同时它不仅是计算机处理一切随机出现的事件的手段，而且也是实现计算机系统资源管理的重要方法。

本章小结

5. 中断过程包含**中断请求、中断响应、中断服务、中断返回**4个阶段。

- 中断屏蔽技术不仅使得CPU能够禁止或允许某些中断源的中断请求，并且可以灵活地修改中断源的优先级。
- 中断请求优先级的排队方法通常有软件查询和硬件排队判优两种方法。排队后的中断请求信号可以单线、多线或者二维结构的形式传送至CPU。CPU则在本条指令执行完后，监测中断输入引脚有否中断，若有中断请求，则通过中断隐指令来保存断点和识别中断源，并转入中断服务程序执行。
- 中断服务程序的最后，通常以中断返回指令，返回断点处。