

计算机组成原理与系统结构

第八章 输入输出系统

<http://jpkc.hdu.edu.cn/computer/zcyl/dzkjdx/>





第八章 输入输出系统

8.1

概述

8.2

输入输出接口

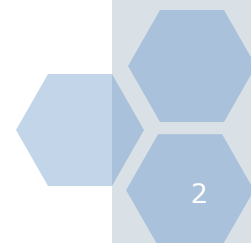
8.3

主机与外设交换信息的方式

8.4

中断系统

本章小结





8.1 概述

一

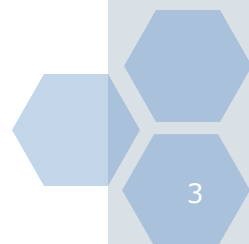
输入输出系统的构成

二

外设与CPU的连接

三

I/O指令格式





一、输入输出系统的构成

1

外设的地位和作用

2

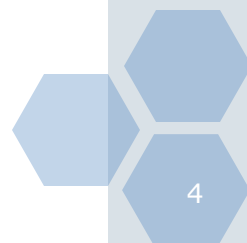
外设的特点

3

外设的分类

4

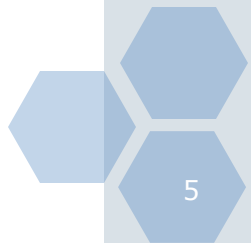
外设的编址方式





1、外设的地位和作用

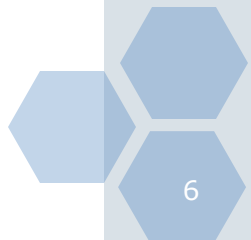
- 外部设备在计算机系统中的作用可以分为四个方面：
 - ① 外部设备是人机对话的重要设备
 - ② 外部设备是完成数据媒体变换的设备
 - ③ 外部设备是计算机系统软件和信息驻在地
 - ④ 外部设备是计算机在各领域应用的重要工具





2、外设的特点

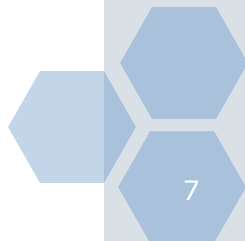
- 外设具有工作速度差异大、结构原理差异大、时序独立、异步性明显等特点，处理的信息从数据格式到逻辑时序一般不可能直接与CPU兼容。
- 计算机与I/O设备间的连接与信息交换不能直接进行，而必须设计一个“接口电路”作为两者之间的桥梁，使CPU和外设协调工作，这种I/O接口电路又叫“I/O适配器”（I/O Adapter）。





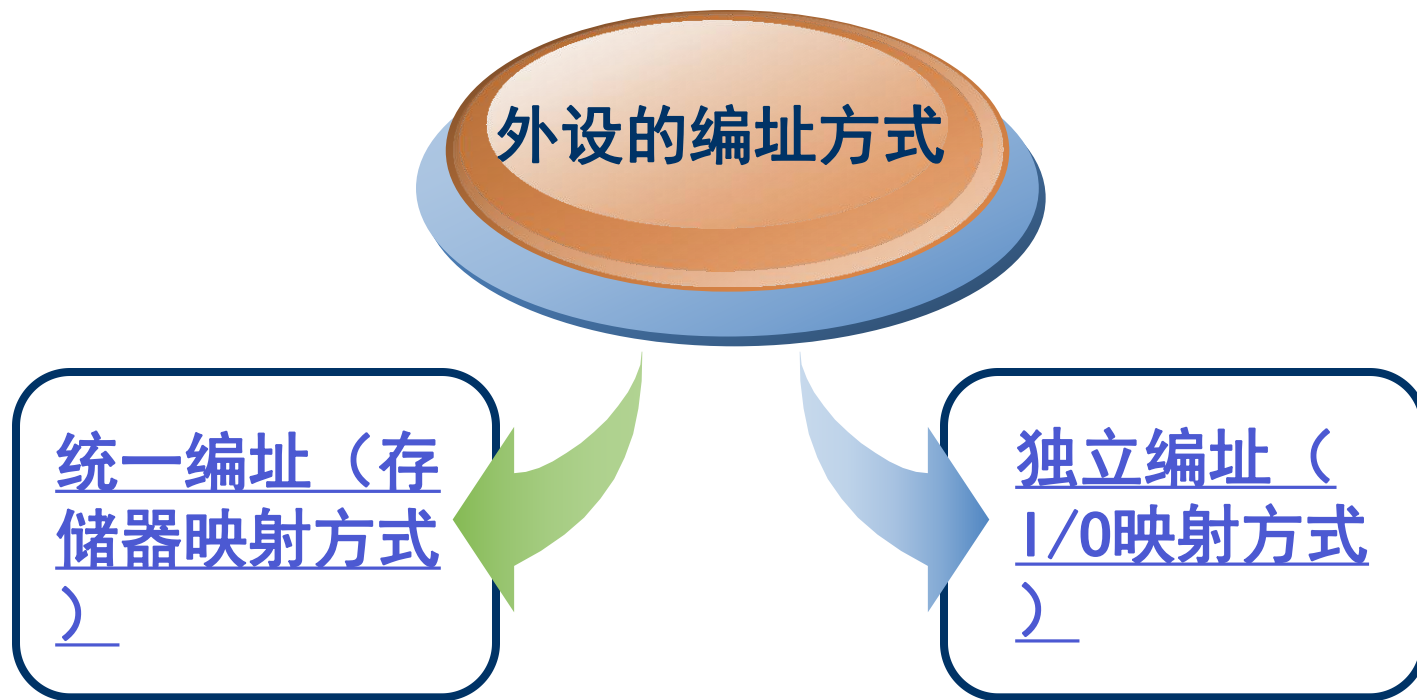
3、外设的分类

- 按照功能可分为输入设备和输出设备两大类：
 - ① **输入设备**用于将各种形式的外部信息转换为计算机所能识别的二进制信息
 - ② **输出设备**则用于将计算机中的二进制信息转换为人或其他机器所能识别的信息形式。
- 按照**外设的工作速度**又可以分为低速设备、中速设备、高速设备。例如，键盘、鼠标是常见的低速设备，磁盘是常见的高速设备。
- 根据**外设计算机系统中所起的作用**，可以分为人-机交互设备、外存储器设备、通信设备。





4、外设的编址方式





统一编址

- 一个I/O端口等同于一个存储器单元。
- **应用：**Motorola 系列、Apple系列微型机及一些单片机和单板机。
- **优点：**
 - ①指令系统中不设置专用的I/O指令，用功能很强的**访存指令**（如LOAD/STORE或者MOV）来访问I/O端口，通过地址来区分访问的是存储器还是I/O端口。
 - ②外设数目或I/O寄存器数几乎**不受限制**。
 - ③微机的读写控制逻辑较为**简单**。

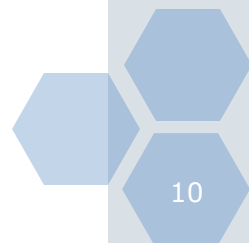




统一编址

■ 缺点：

- ① I/O端口占用部分主存空间，可用主存空间减小；
- ② 访存指令较长，执行速度较慢；
- ③ I/O端口地址译码电路复杂，译码时间较长。





独立编址

- I/O端口地址空间与存储器地址空间相互独立。指令系统中设置了专用的I/O指令，用I/O指令来访问I/O端口，用访存指令来访问存储器，因此，虽然I/O端口地址与存储器地址有部分重叠，但通过指令可以区分。
- 应用：IBM-PC 系列、Z-80系列微型机及一些大型机。
- 优点：
 - ①I/O端口地址不占用存储器地址空间；
 - ②I/O端口数量不多，占用地址线少，地址译码简单，速度较快；
 - ③使用专用I/O命令(IN/OUT)，指令短，执行速度快，可读性强。



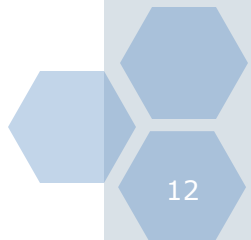
独立编址

■ 缺点：

- ①专用I/O指令增加指令系统复杂性，且I/O指令类型少，程序设计灵活性较差；
- ②要求处理器提供MEMR/MEMW和IOR/IOW两组控制信号，增加了控制逻辑的复杂性。

■ Intel系列微机I/O编址

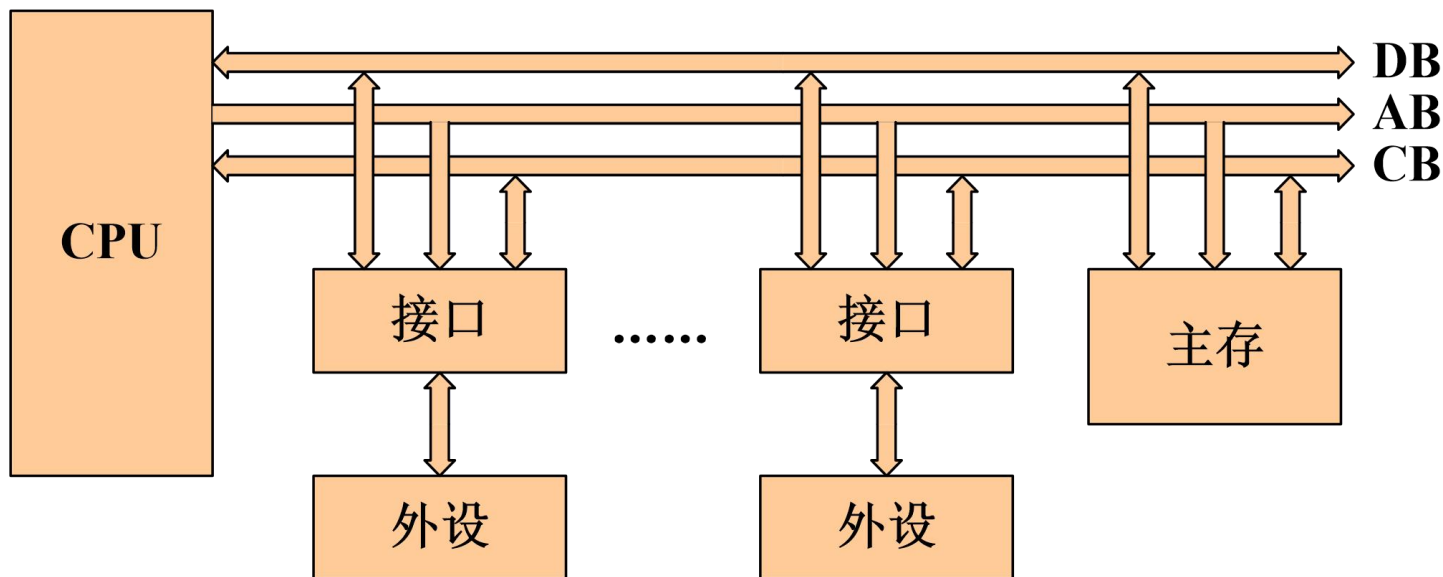
- ①Intel系列微处理器支持I/O独立编址方式和I/O统一编址（存储器映象编址）方式。
- ②Intel系列微机系统仅支持I/O独立编址方式。





二、外设与CPU的连接

外设接口通过总线与CPU连接



1. CPU访问外设的实质是访问外设接口中的寄存器(端口)。
2. 相比存储器的访问，CPU访问外设的过程是完全等同的，不同的是所发送的读写信号有区别。

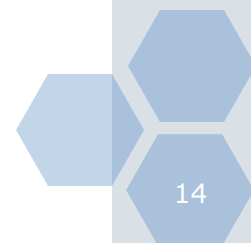




三、I/O指令格式

1. 专用的I/O指令包含操作码、命令码和地址码三部分：

- 操作码用于区分访存指令和I/O指令
- 命令码用于区分I/O操作的种类
- 地址码则指明要访问的外设端口地址以及CPU寄存器号





三、I/O指令格式

2. IBM/PC机的I/O指令只有两条：IN和OUT

- IN指令将外设端口中的数据读入累加寄存器AL (AX) 。
- OUT指令将累加寄存器AL (AX) 的数据写入外设端口。
- 它们的汇编助记符及格式如下：

- 端口地址为8位

IN AL/AX, port; 输入指令
OUT port, AL/AX; 输出指令
Port为8位的端口地址 (0~255) ;

- 端口地址为16 位

MOV DX, port
IN AL/AX, DX; 输入指令
OUT DX, AL/AX; 输出指令
DX内为16位的端口地址port。



三、I/O指令格式

3. 80286以上的CPU还支持I/O端口与内存直接交换数据:

■ 输入操作:

MOV DX, Port

LES DI, Buffer_in

INSB; (将DX所指向的端口地址内的数据输入到由ES: DI所指向的内存单元, 传送一个8位的数据)

INSW; 同上, 传送一个16位的数据

■ 输出操作:

MOV DX, Port

LDS SI, Buffer_out

OUTSB; (将由DS: SI所指向的内存单元内的数据输出到DX所指向的端口地址, 传送一个8位的数据)

OUTSW; 同上, 传送一个16位的数据





8.2 输入输出接口

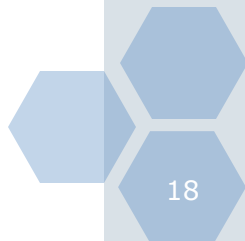




一、I/O接口的功能

1. I/O接口的功能如下：

- ① 实现数据缓冲
- ② 执行CPU的命令
- ③ 返回外设的状态
- ④ 设备选择。
- ⑤ 实现数据格式的转换
- ⑥ 实现信号的转换
- ⑦ 中断管理功能

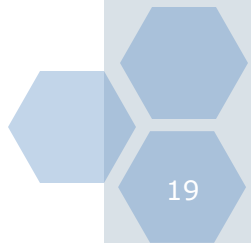




二、I/O接口的组成

1. I/O接口的硬件电路主要包括三部分：

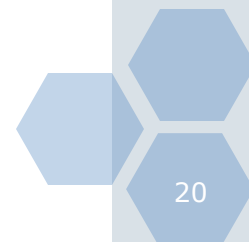
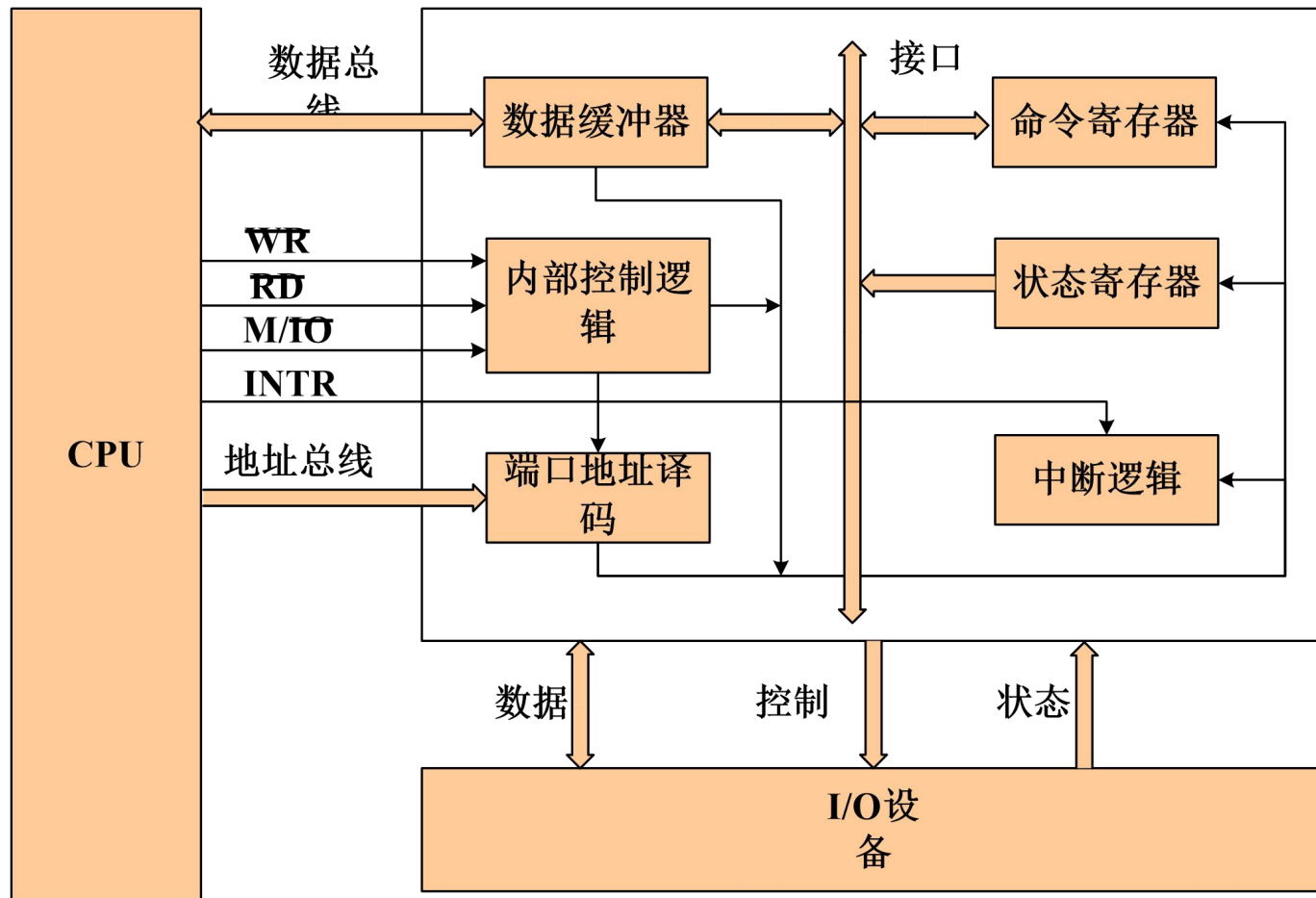
- **基本电路**主要包括寄存器及其控制逻辑。寄存器包括命令寄存器（控制寄存器）及其译码器、数据缓冲寄存器、状态寄存器，分别用以保存CPU的命令、数据信息和外设的状态。
- **端口地址译码电路**对地址总线上的外设地址进行译码，用以决定是否选中设备自身。
- **供选电路**由于接口的功能和结构有很大的区别，因此各接口电路中可能选择使用中断控制逻辑、定时器、计数器、移位器等器件。





二、I/O接口的组成

1. I/O接口组成框图





8.3 主机与外设交换信息的方式

一

程序查询方式

二

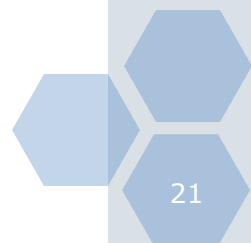
程序中断方式

三

直接存储器访问（DMA）方式

四

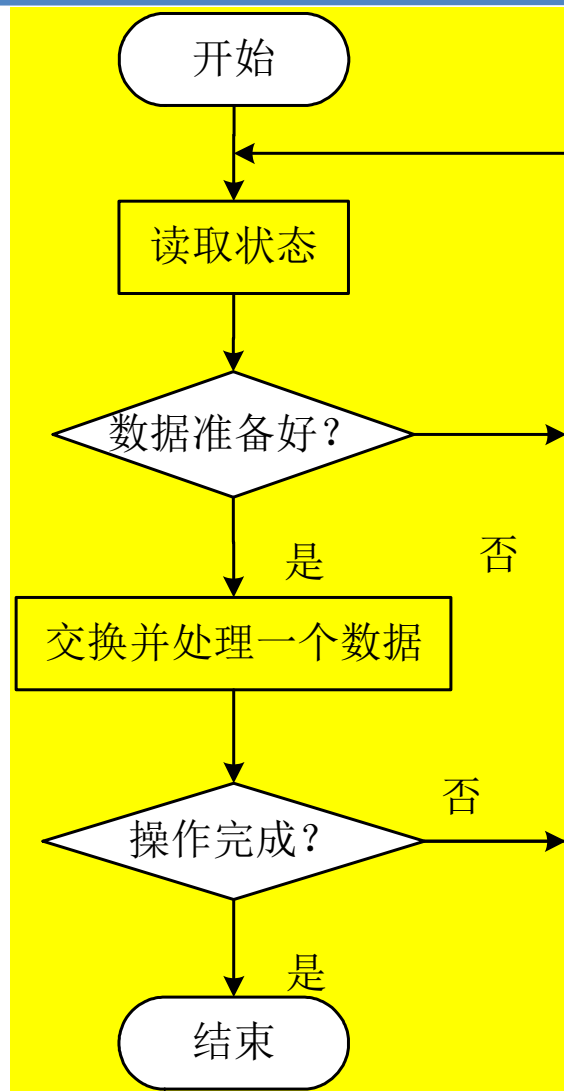
通道与输入输出处理机方式





一、程序查询方式

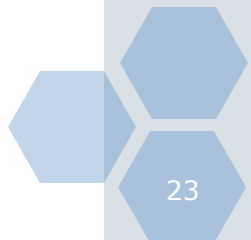
1. **工作原理：** CPU查询外设已准备好后，才传送数据。
2. **特点：** CPU与外设间通过程序同步，CPU被外设独占，CPU效率低下。
3. **要求：** 不需要增加额外的硬件电路。
4. **应用：** 适同在CPU不太忙且传送速度要求不高时。





二、程序中断方式

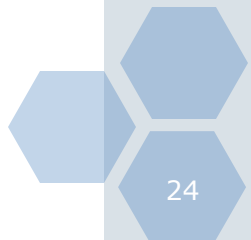
1. **工作原理**：在外设准备数据时，CPU执行与传送数据无关的工作，外设准备好数据后，主动向CPU发送一个中断请求，当CPU**执行完当前指令后**，停止当前程序的执行，自动转向中断服务程序，在中断服务程序中，完成一个数据的传送，之后中断返回至原来的断点处，继续执行。
2. **特点**：在外设准备数据时，CPU与外设**并行工作**，CPU效率有所提高，并且**CPU可以同时被多个外设占用**。
3. **要求**：接口中需要中断控制逻辑支持。
4. **应用**：适用于中低速设备。





三、直接存储器访问（DMA）方式

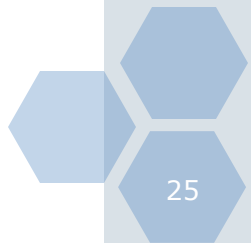
1. **工作原理：** 将I/O过程中，与内存交换数据的操作交由**DMA控制器**来控制，简化了CPU对输入输出的控制，进一步提高了CPU的效率。
2. **特点：** 数据的传送不经过CPU（**由DMAC控制**），而I/O设备管理由CPU控制，简化了CPU对I/O的控制。硬件开销大，结构复杂，但CPU的效率high。
3. **要求：** 需要DMA控制器及相关逻辑支持。
4. **应用：** 适用与高速度大量数据传送时。





四、通道与输入输出处理机方式

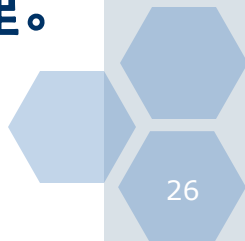
1. **通道**是一个具有特殊功能的**处理器**，它可以实现对外围设备的统一管理和外围设备与内存之间的数据传送。
2. **特点**：能独立地执行用**通道指令**编写的输入输出控制程序，产生相应的控制信号送给由它管辖的设备控制器，继而完成复杂的输入输出过程。
3. **要求**：需要具有特殊功能的处理器，某些应用中称为**输入输出处理器（IOP）**。
4. **应用**：适用与高速度大量数据传送时。





四、通道与输入输出处理机方式

5. 输入输出处理机（IOP）通常称作外围处理机（PPU），它是通道方式的进一步发展。这种PPU基本上独立于主机工作，它的结构更接近一般处理机，甚至就是微小型计算机。
6. **特点：** I/O处理机接管了CPU的各种I/O操作及I/O控制功能，**CPU能与IOP并行工作。**
7. **要求：** 需要IOP支持。
8. **应用：** 高速I/O归IOP管理，低速I/O设备归CPU管理。





8.4 中断系统

一

中断的基本概念

二

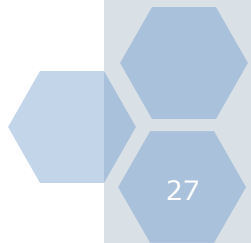
中断请求与判优

三

中断响应

四

中断服务与返回





一、中断的基本概念

1

中断源的分类

2

中断过程

3

中断的作用



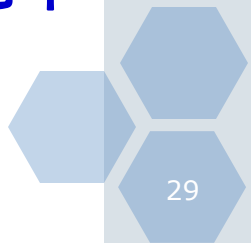


1、中断源的分类

- **中断源**：由于某种原因引起CPU中断的事件或设备。
中断源一般可以分为**硬中断**和**软中断**两类。
- **硬中断**：由外部设备和其他CPU外部事件引起的中断，因此又叫**外中断**。

①常见的外部中断有输入输出请求、实时时钟、计时器、电源故障、设备故障、校验线路等等。
外中断一般通过CPU的**中断请求引脚**引入。

②例如，在80X86系列CPU上，设有**INTR**、**NMI**两个中断请求引脚。

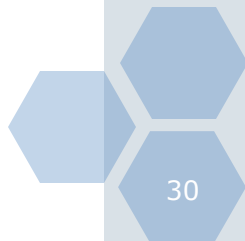




1、中断源的分类

- **软中断**：指CPU内部的指令或程序执行中的突发事件所引起的中断，又叫**内中断**。

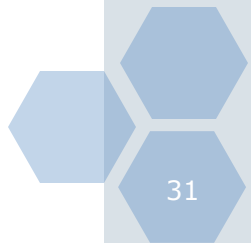
①常见的软中断主要包括指令中断（例如中断指令**INT n**）和程序异常（例如除数为零，运算溢出、指令的单步运行、程序运行至断点处等等）。





1、中断源的分类

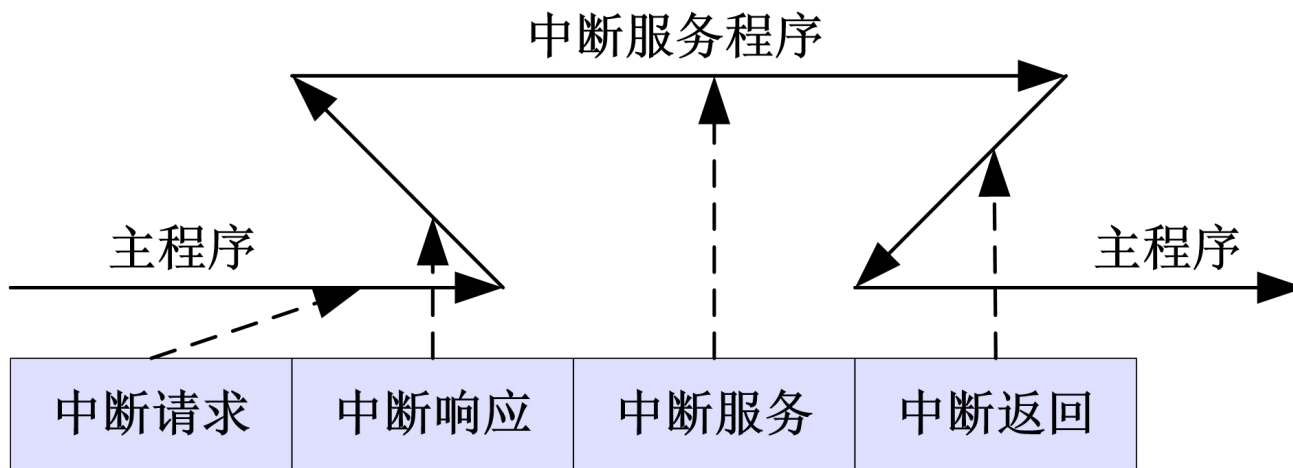
- 在很多中断系统中，对所有的中断源编码，为其分配一个惟一的代号，称为**中断类型号**。例如，80X86有256种中断类型，因此中断类型号为8位二进制（0-255）。中断类型号用于寻找**中断服务程序**的入口地址（**中断向量**），以实现程序转移。





2、中断过程

- 中断过程包含4个阶段

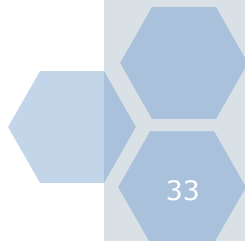




2、中断过程

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。



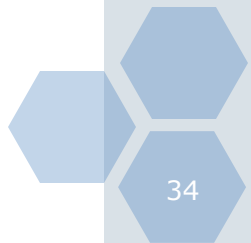


2、中断过程

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？

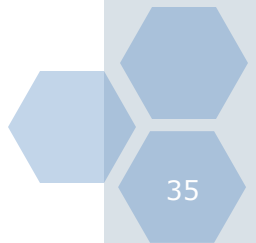




2、中断过程

② 中断响应

- 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。



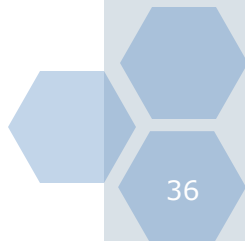


2、中断过程

② 中断响应

- 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断服务程序入口？

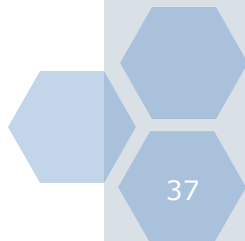




2、中断过程

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？

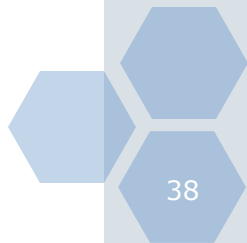




2、中断过程

④ 中断返回

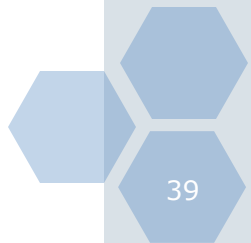
中断返回指令的功能是：将中断隐指令保存的程序断点和标志读出并送入PC和标志寄存器，从而回到CPU原来的程序断点处继续执行。





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

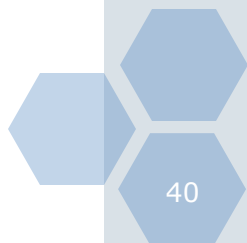
中断屏蔽

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 **INTR_i=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。

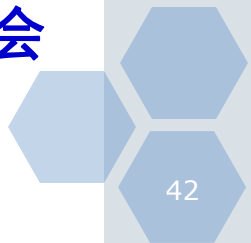


1、中断请求信号的产生与监测

- 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。





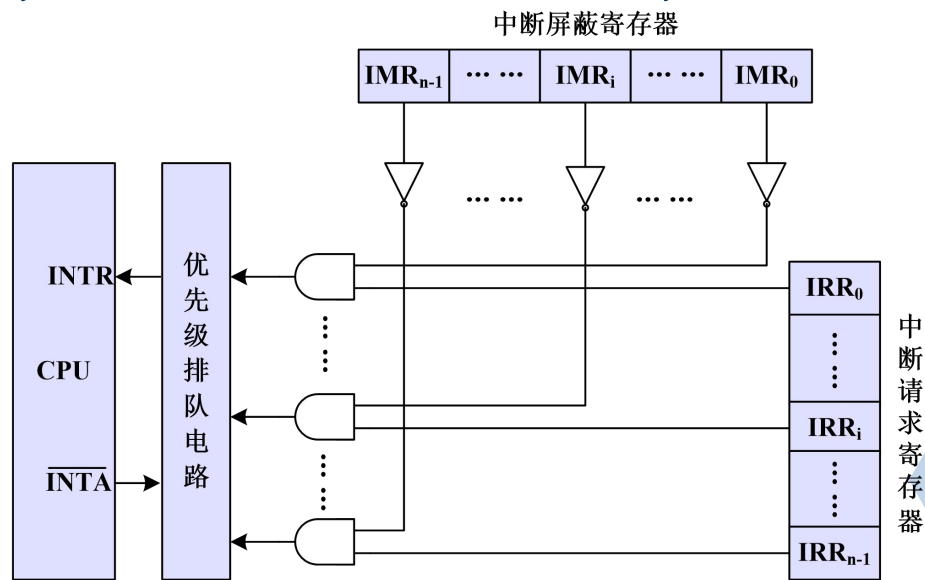
2、中断屏蔽

- 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。

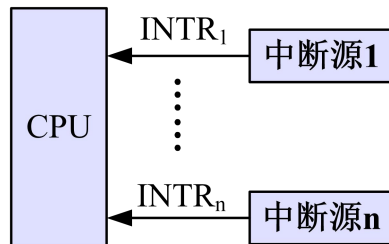
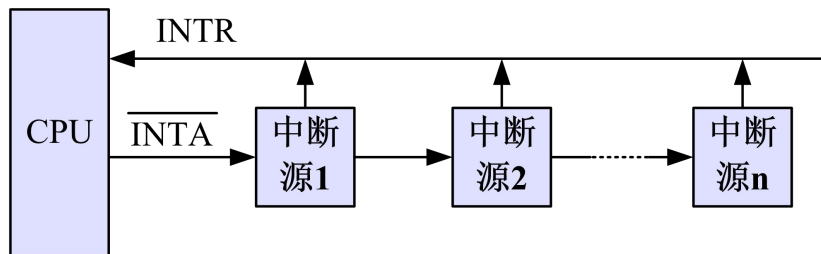
- 将中断系统中的所有中断源的屏蔽触发器放在一起，形成一个寄存器，用一个地址对其寻址，称为 **屏蔽寄存器 IMR**。



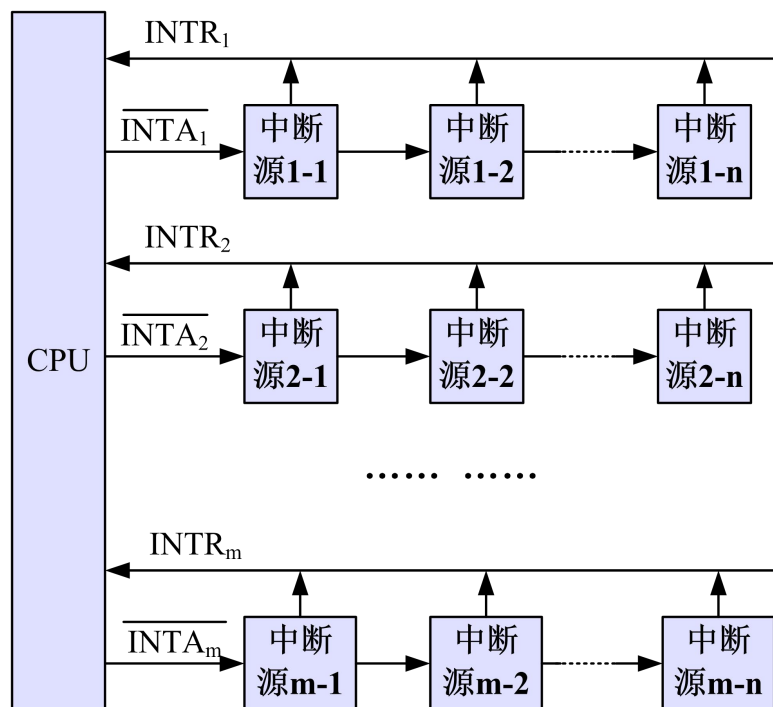


3、中断请求信号的传递

公共中断请求线



独立中断请求线



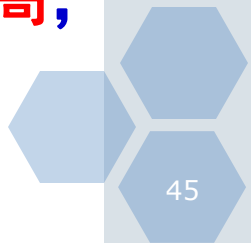
二维结构中中断请求





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。

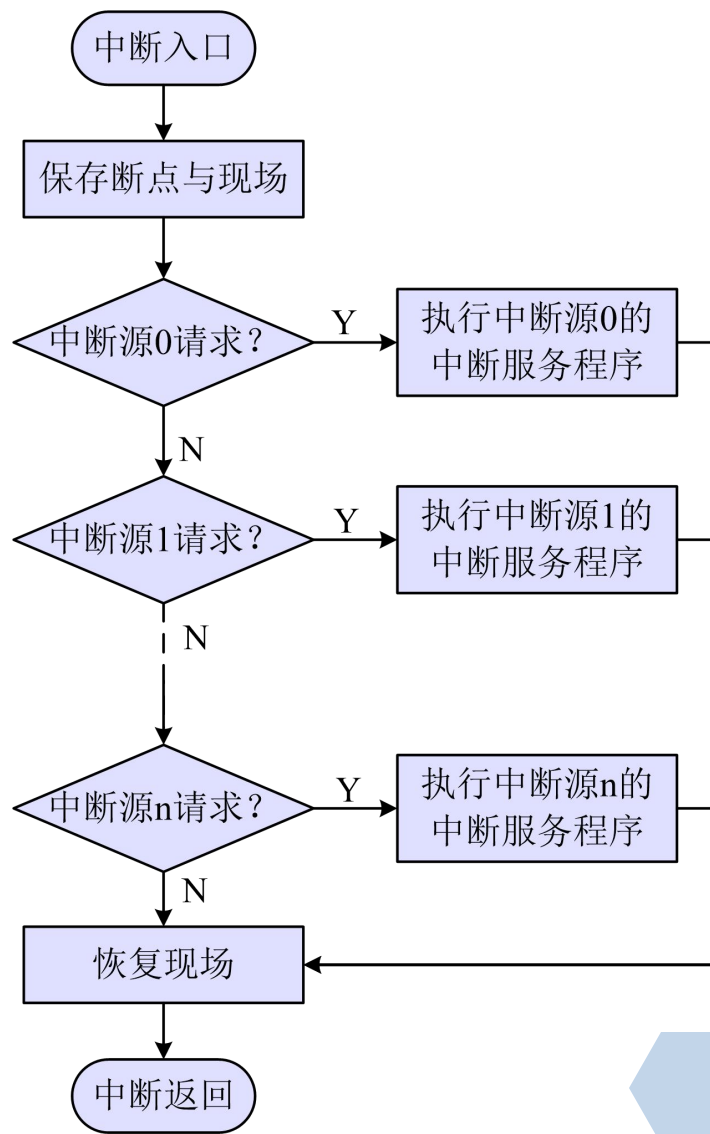




4、中断请求的排队判优

- 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

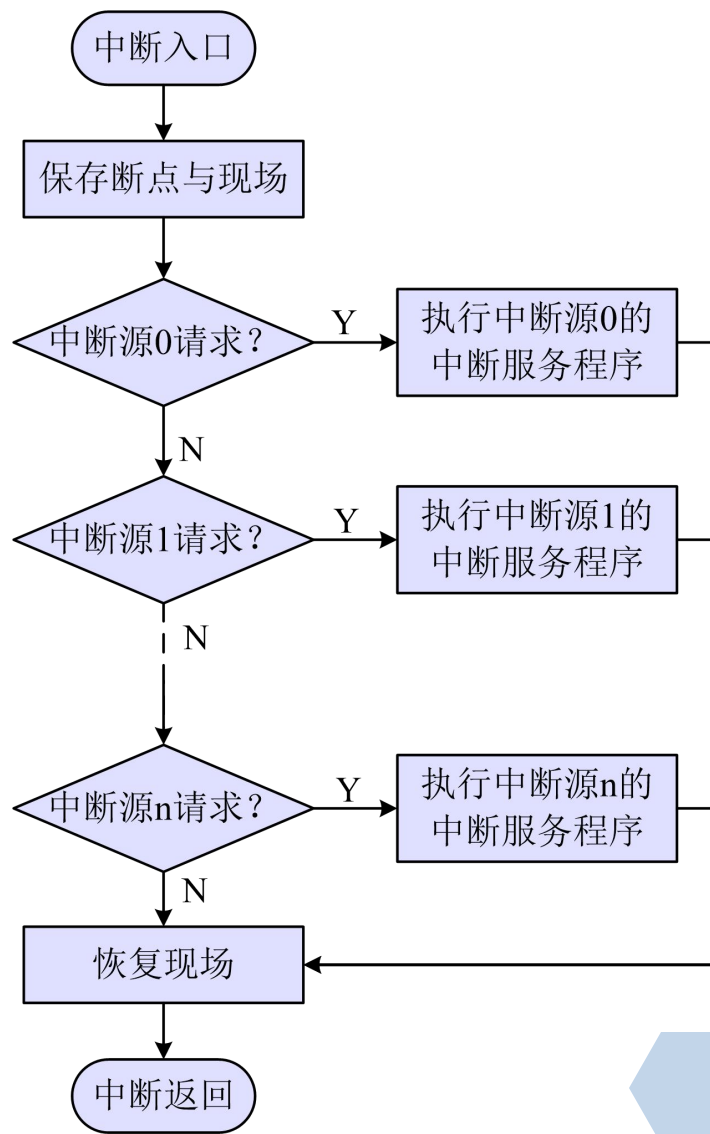




4、中断请求的排队判优

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。

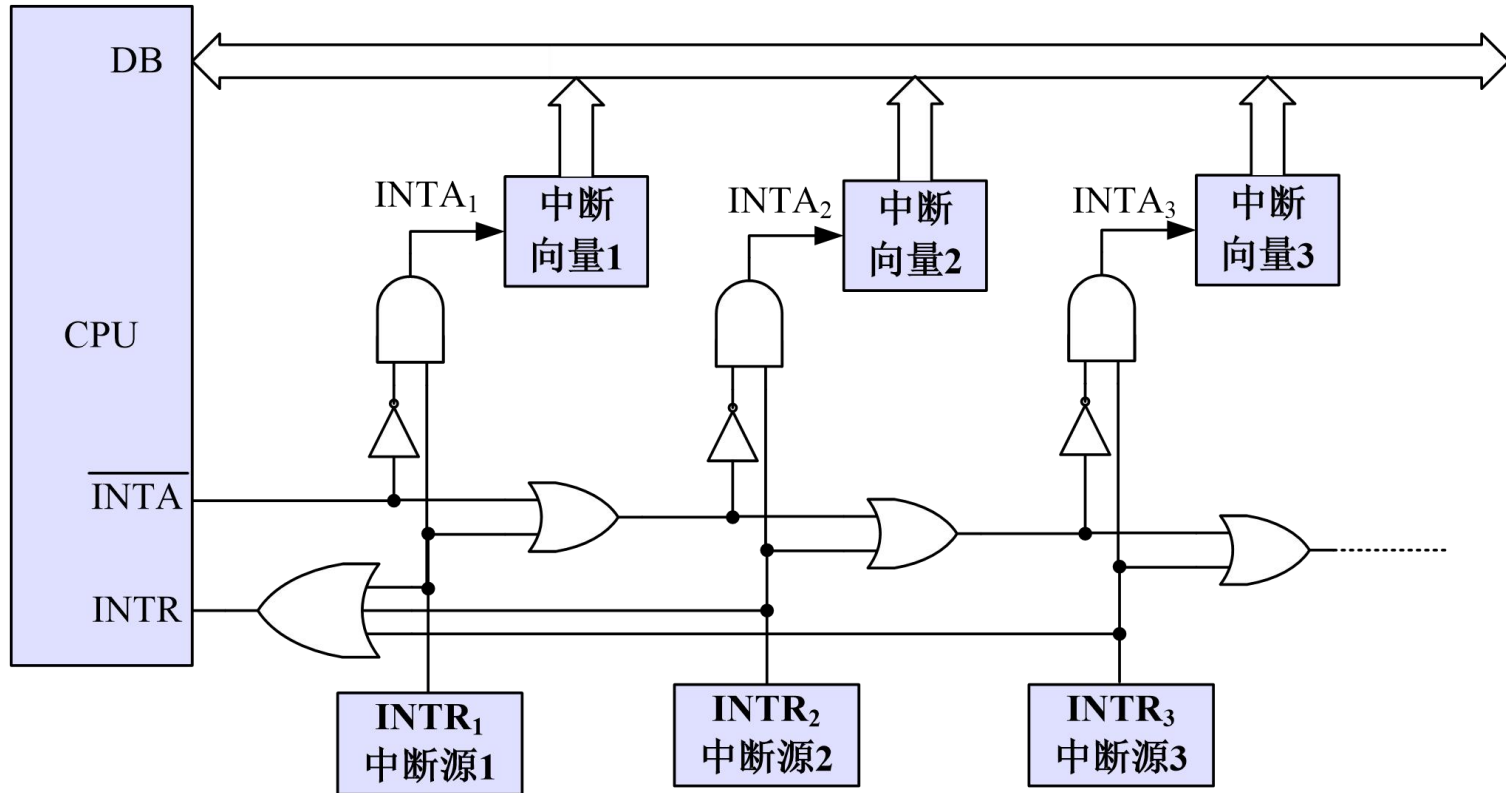
串行排队链与向量中断：

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。





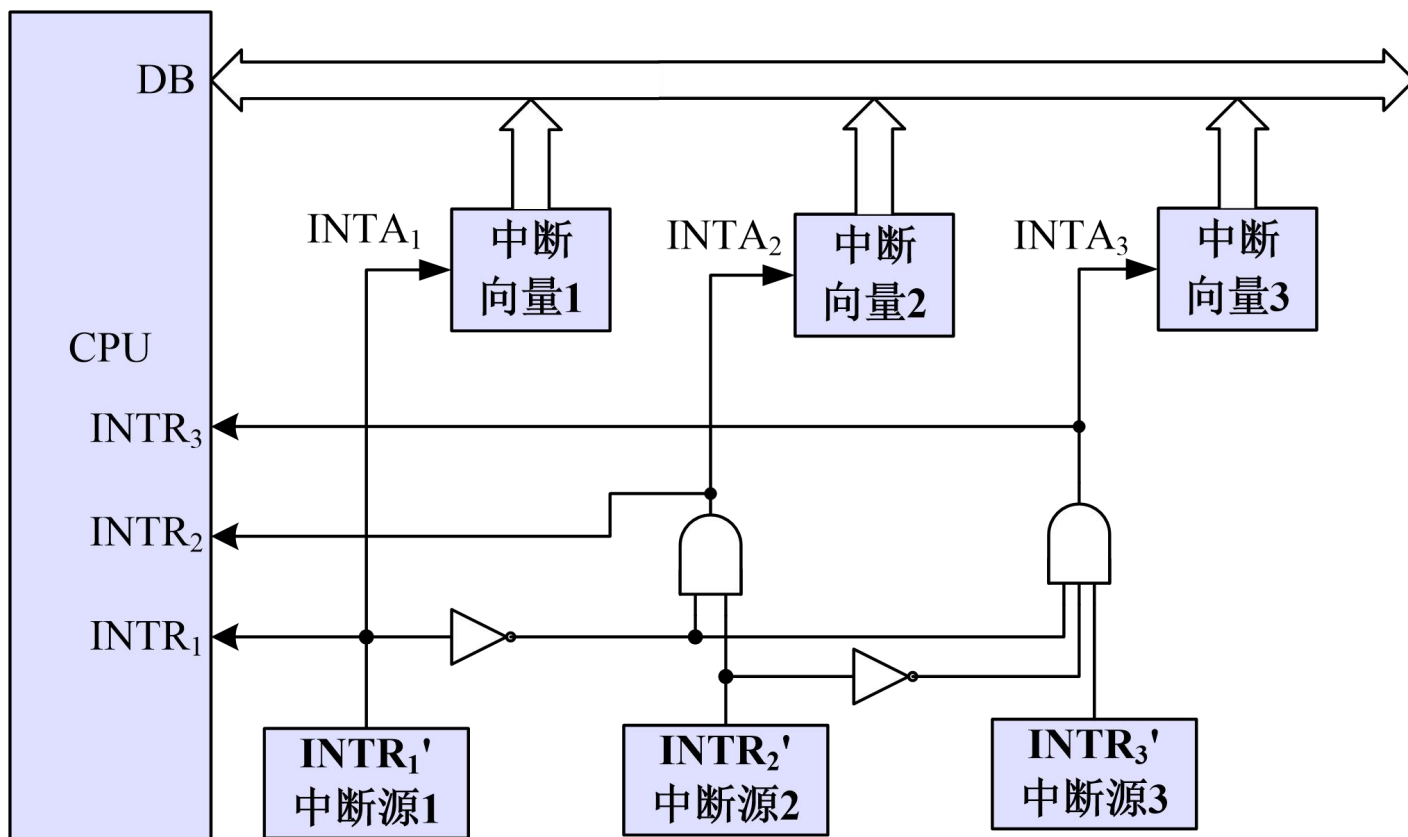
单线请求的串行排队判优电路





4、中断请求的排队判优

II. 独立请求优先级排队电路

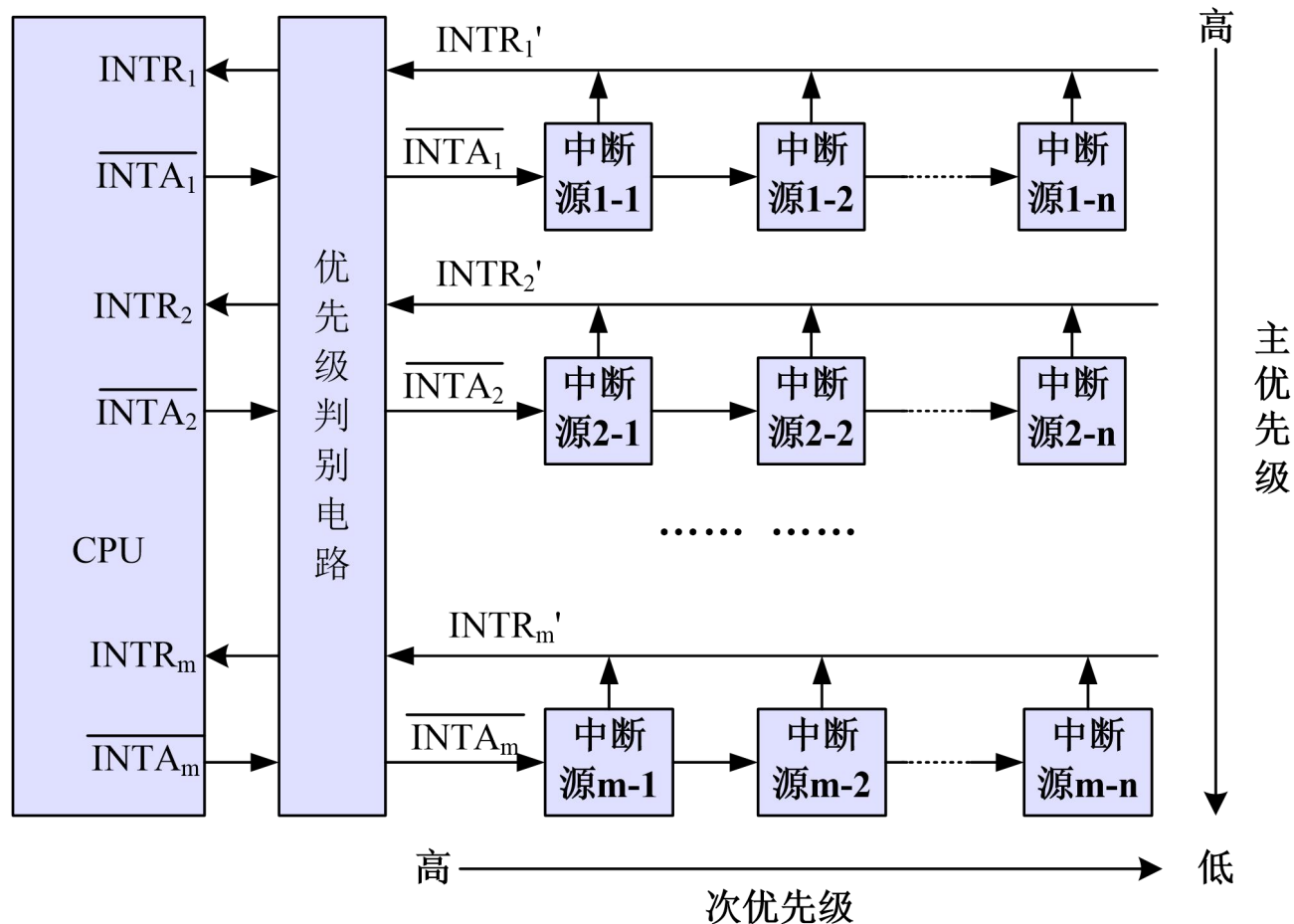


多线请求的串行排队判优电路



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），禁止接收新的中断，以保证接下来中断隐指令操作不被打断。



三、中断响应

2. CPU中断响应的过程

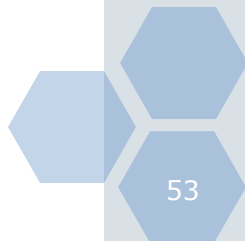
◆ 保存断点

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**

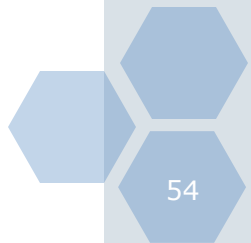




三、中断响应

- **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**。

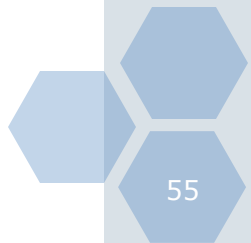




三、中断响应

- **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。

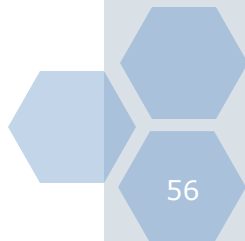




三、中断响应

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

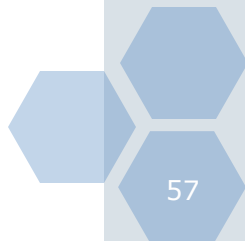
中断服务程序与子程序的区别

3

中断嵌套

4

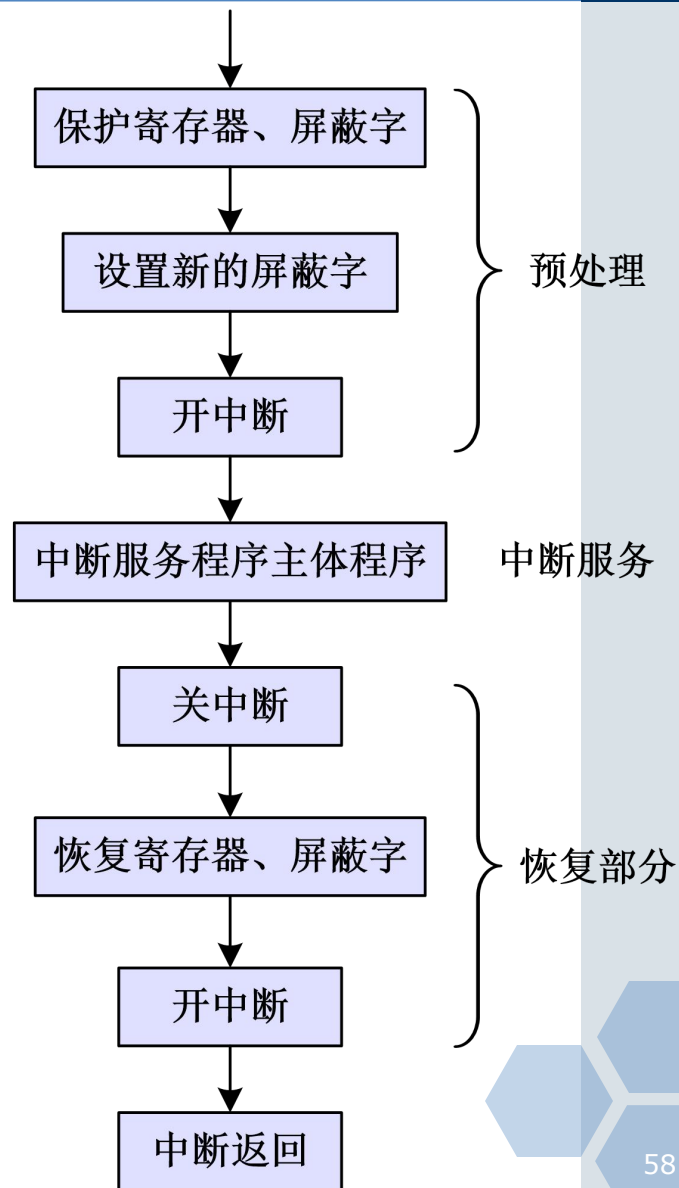
利用中断屏蔽技术修改中断优先级





1、中断服务程序

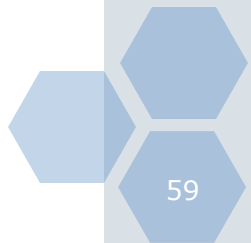
- ◆ 预处理部分
- ◆ 中断服务
- ◆ 恢复部分
- ◆ 中断返回





2、中断服务程序与子程序的区别

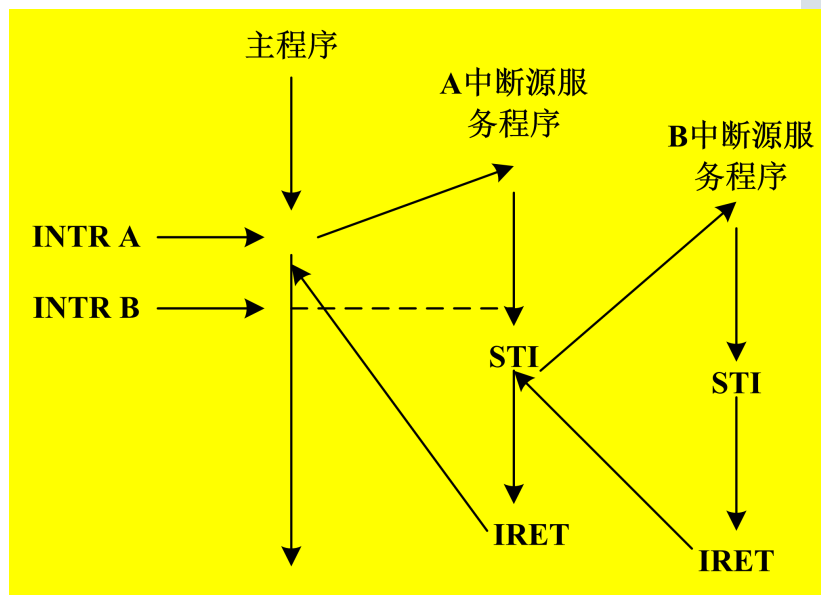
- ◆ 子程序的执行是由程序员事先安排好的（由一条调用子程序指令转入），而中断服务程序的执行则是由随机的中断事件引起的；
- ◆ 子程序的执行受到主程序或其上层程序的控制，而中断服务程序一般与被中断的现行程序毫无关系；
- ◆ 不存在同时调用多个子程序的情况，而有可能发生多台I/O设备同时请求CPU为自己服务的情况。





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级

- 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，通过以中断返回指令，返回断点处。