

第4章 输入输出系统

李峰

fli@sdu.edu.cn

<https://funglee.github.io>

4.1 输入输出原理

4.2 中断系统

4.3 通道处理机

4.4 输入输出处理机

4.1 输入输出原理

- 通常把处理机与主存储器之外的部分统称为输入输出系统，包括输入输出设备、输入输出接口和输入输出软件等。
- 实际上，运算器、控制器、主存储器和总线等也要通过输入输出系统来管理。

4.1.1 输入输出系统的特点

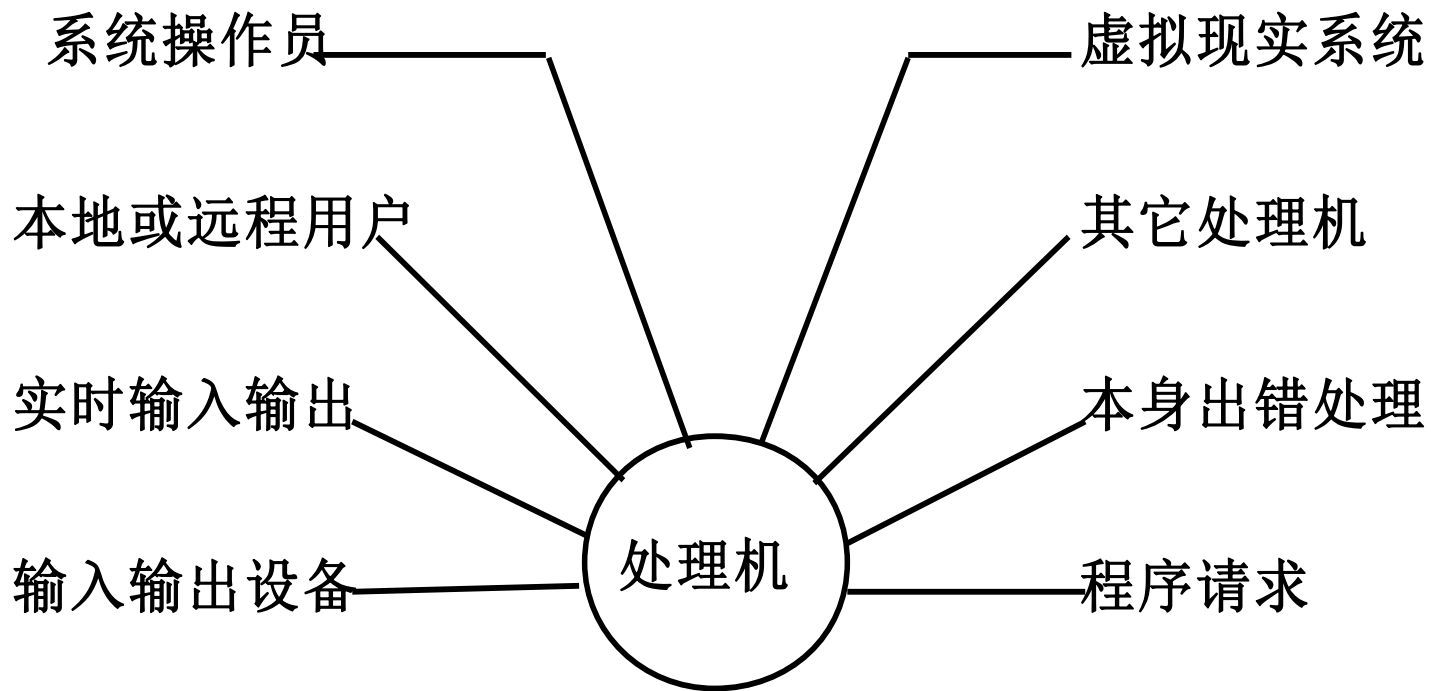
4.1.2 输入输出系统的组织方式

4.1.3 基本输入输出方式

4.1.1 输入输出系统的特点

- 输入输出系统是处理机与外界进行数据交换的通道。
- 输入输出系统是计算机系统中最具多样性和复杂性的部分。
- 输入输出系统涉及到机、光、电、磁、声、自动控制等多种学科。
- 输入输出系统最典型地反映着硬件与软件的相互结合。
- 输入输出系统的复杂性隐藏在系统软件中，用户无需了解输入输出设备的具体细节。

处理机的外部世界包括：



1. 实时性

- 对于一般输入输出设备, 如果处理机提供的服务不及时, 可能丢失数据, 或造成外围设备工作的错误。
- 对于实时控制计算机系统, 如果处理机提供的服务不及时, 可能造成巨大的损失, 甚至造成人身伤害。
- 对于处理机本身的硬件或软件错误: 如电源故障、数据校验错、页面失效、非法指令、地址越界等, 处理机必须及时处理。
- 对不同类型的设备, 必须具有与设备相配合的多种工作方式。

2. 与设备无关性

- 独立于具体设备的标准接口。例如，串行接口、并行接口、SCSI (Small Computer System Interface) 接口等
- 计算机系统的使用者，在需要更换外围设备时，各种不同型号，不同生产厂家的设备都可以直接通过标准接口与计算机系统连接。
- 处理机采用统一的硬件和软件对品种繁多的设备进行管理。
- 某些计算机系统已经实现了即插即用技术。

3. 异步性

- 输入输出设备通常不使用统一的中央时钟，各个设备按照自己的时钟工作，但又要在某些时刻接受处理机的控制。
- 处理机与外围设备之间，外围设备与外围设备之间能并行工作

针对实时性，采用层次结构的方法，

针对与设备无关性，采用分类处理的方法

针对异步性，采用自治控制的方法，

4.1.2 输入输出系统的组织方式

1. 自治控制

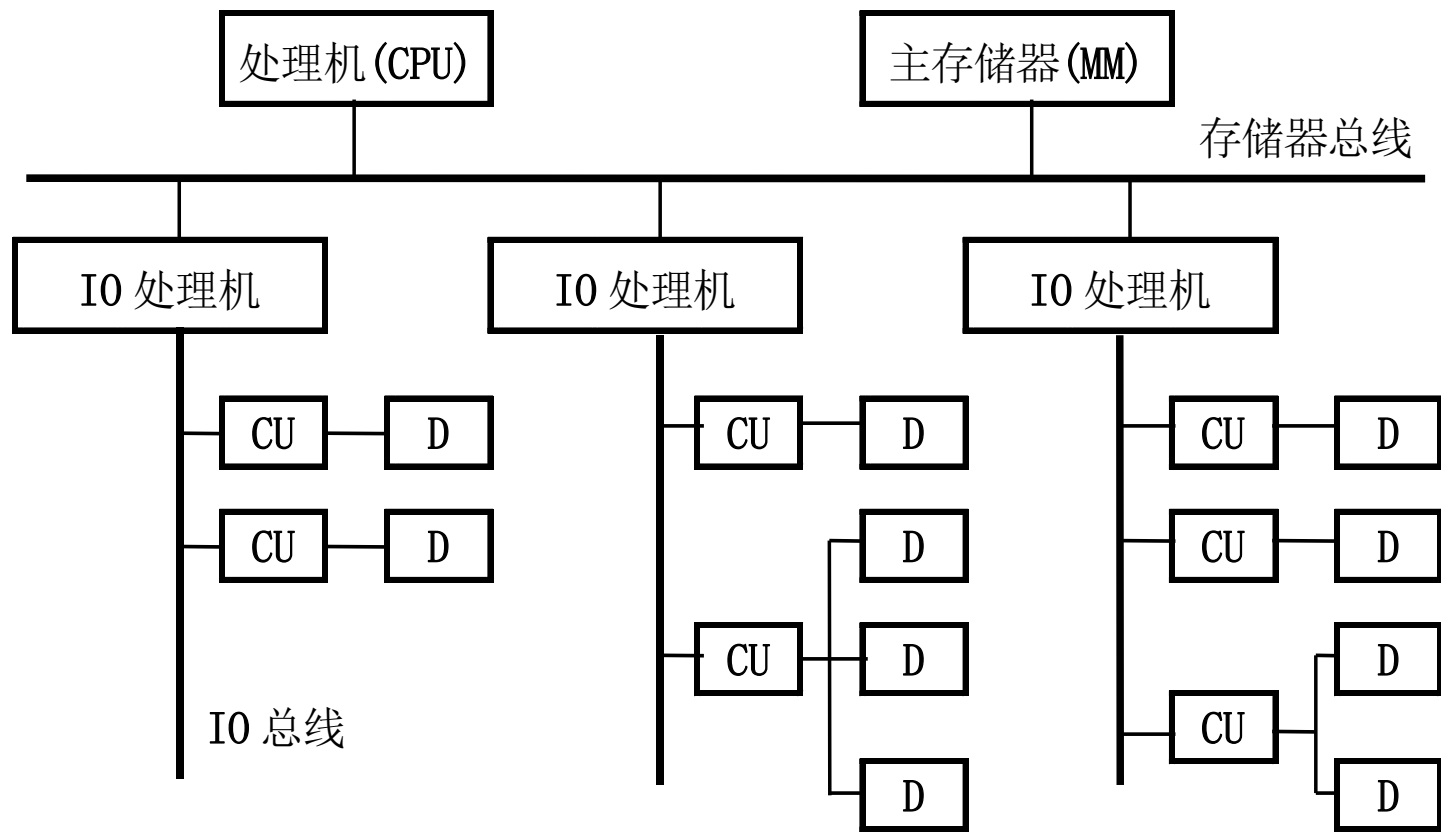
- 输入输出系统是独立于CPU之外的自治系统
- 处理机与外围设备之间要有恰当的分工

2. 层次结构

- 最内层是输入输出处理机、输入输出通道等
- 中间层是标准接口。
- 标准接口通过设备控制器与输入输出设备连接

3. 分类组织

- 面向字符的设备，如字符终端、打字机等
- 面向数据块的设备，如磁盘、磁带、光盘等。



CU 是设备控制器，D 是外围设备
输入输出系统的层次结构

4.1.3 基本输入输出方式

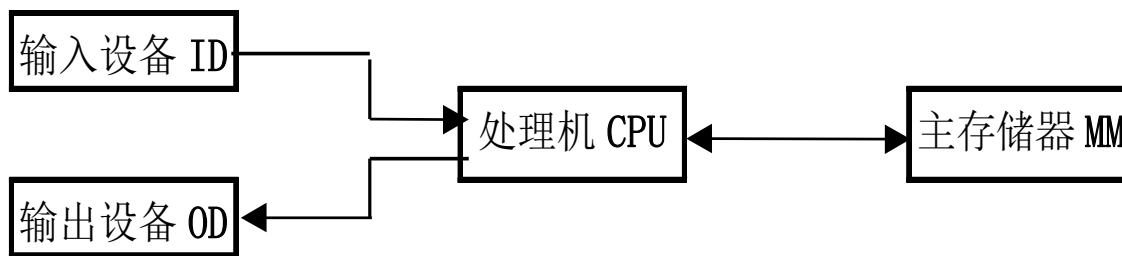
1. 程序控制输入输出方式

- 状态驱动输入输出方式、应答输入输出方式、查询输入输出方式、条件驱动输入输出方式
- 程序控制输入输出方式的4个特点：
 - 1) 何时对何设备进行输入输出操作受CPU控制
 - 2) CPU要通过指令对设备进行测试才能知道设备的工作状态。空闲、准备就绪、忙碌等
 - 3) 数据的输入和输出都要经过CPU
 - 4) 用于连接低速外围设备，如终端、打印机等

例：一个处理机在一段时间内只能管理一台打印机。处理机执行指令的速度为1GIPS，字长32位，打印机每秒钟100个字符。

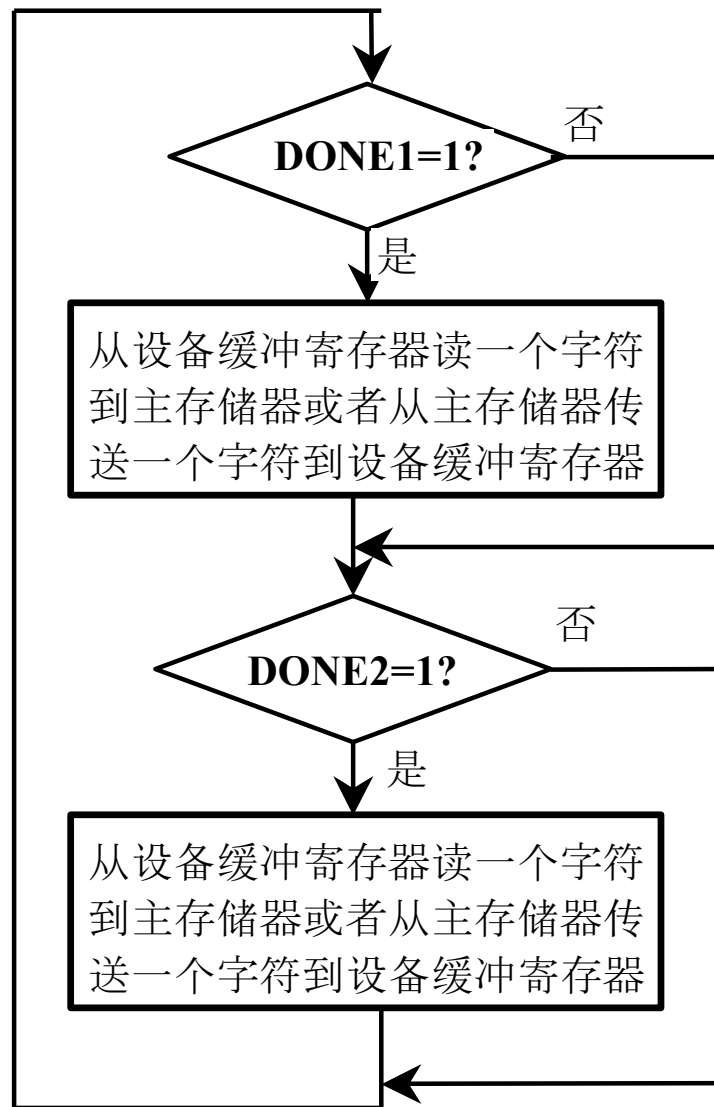
解：处理机用一条指令就能向打印机传送4个字符。因此，处理机的实际利用率只有即**4千万分之一**。

$$100 / (10^9 \times 4) = 0.25 \times 10^{-7}$$



程序控制方式的数据传送过程

- 一个处理机管理多台外围设备。
处理机采用轮流循环测试方法，
分时为各台外围设备服务。
- 优点：灵活性很好。可以很容易地改变各台外围设备的优先级。
- 缺点：不能实现处理机与外围设备之间并行工作。



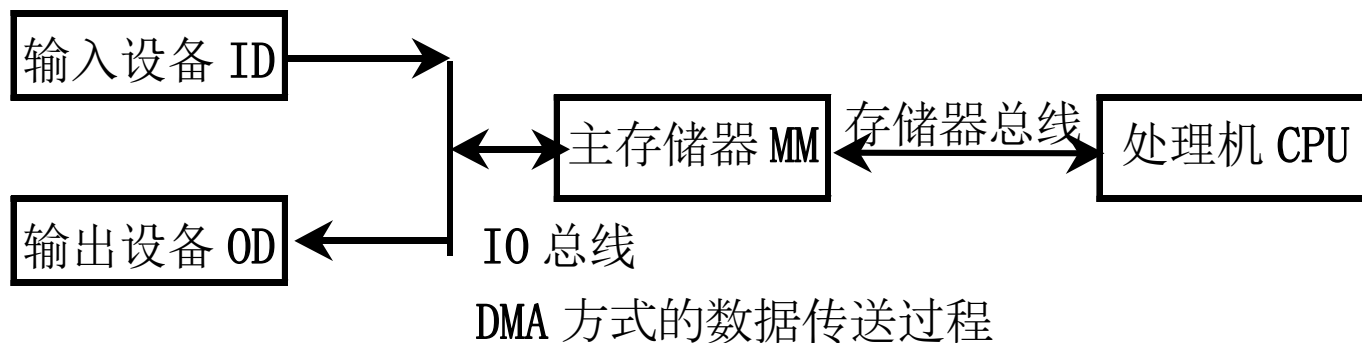
在程序控制方式中一个处理机管理多台外部设备的程序流程图

2. 中断输入输出方式

- 定义：当出现来自系统外部，机器内部，甚至处理机本身的任何例外的，或者虽然是事先安排的，但出现在现行程序的什么地方是事先不知道的事件时，CPU暂停执行现行程序，转去处理这些事件，等处理完成后再返回来继续执行原先的程序。
- 特点：
 - 1) CPU与外围设备能够并行工作。
 - 2) 能够处理例外事件。
 - 3) 数据的输入和输出都要经过CPU。
 - 4) 用于连接低速外围设备。

3. 直接存储器访问方式

- 直接存储器访问方式(DMA: Direct Memory Access)，主要用来连接高速外围设备。如磁盘存储器，磁带存储器、光盘辅助存储器，行式打印机等。



DMA方式具有如下特点：

- 1) 外围设备的访问请求直接发往主存储器，数据的传送过程不需要CPU的干预。
- 2) 全部用硬件实现，不需要做保存现场和恢复现场等工作。
- 3) DMA控制器复杂，需要设置数据寄存器、设备状态控制寄存器、主存地址寄存器、设备地址寄存器和数据交换个数计数器及控制逻辑等。
- 4) 在DMA方式开始和结束时，需要处理机进行管理。

DMA输入设备的工作流程如下：

- 从设备读一个字节到DMA控制器中的数据缓冲寄存器BD（DMA控制器中）。
- 若一个字没有装配满，则返回到上面；若校验出错，则发中断申请；
- 若一个字已装配满，则将BD中的数据送主存数据寄存器。
- 把主存地址寄存器BA（DMA控制器中）中的地址送主存地址寄存器，并将BA中的主存地址增值。
- 把DMA控制器内的数据交换个数计数器减 1 。
- 若交换个数为0，则DMA数据传送过程结束，否则回到上面。

DMA输出设备的工作流程如下：

- 把主存地址寄存器BA（DMA控制器中）中的地址送入主存地址寄存器，并启动主存储器，同时将BA中的主存地址增值。
- 将主存储器数据寄存器中的数据送DMA控制器的数据缓冲寄存器BD。
- 把BD中的数据写到输出介质上（可能要逐个字符输出）。
- 把DMA控制器内的数据交换个数计数器中的内容减 1 。
- 若交换个数为0，则DMA数据传送过程结束，否则回到上面。

目前使用的DMA方式实际上有如下三种：

1) 周期窃取方式：

- 在每一条指令执行结束时，CPU测试有没有DMA服务申请。
- 借用CPU完成DMA工作流程。包括数据和主存地址的传送，交换个数计数器减1，主存地址的增值及一些测试判断等。
- 周期窃取方式的优点是硬件结构简单，比较容易实现。
- 缺点是在数据输入或输出过程种实际上占用了CPU的时间。

2) 直接存取方式：

- DMA控制器的数据传送申请直接发往主存储器
- 整个工作流程全部用硬件完成。
- 优点与缺点正好与周期窃取方式相反。

3) 数据块传送方式：

- 在设备控制器中设置一个比较大的数据缓冲存储器。设备控制器与主存储器之间的数据交换以数据块为单位，并采用程序中断方式进行。
- 采用数据块传送方式的外围设备有软盘驱动器、行式打印机、激光打印机、卡片阅读机、绘图仪等。

4.2 中断系统

4.2.1 中断源的组织

4.2.2 中断系统的软硬件分配

4.2.3 中断源的识别方法

4.2.4 中断现场的保存和恢复

4.2.5 中断屏蔽

4.2.1 中断源的组织

- 中断系统需要硬件和软件共同来实现。
- 引起中断的各种事件称为中断源。
- 中断系统的复杂性实际上主要是由中断源的多样性引起的。
- 中断源可以来自系统外部，也可以来自机器内部，甚至处理机本身。
- 中断可以是硬件引起的，也可以是软件引起的。
- 把各种各样的中断源分类、分级组织好，是中断系统的关键之一。

1. 中断源的种类

- 1) 由外围设备引起的中断。低速外围设备每传送一个字节申请一次中断；高速外围设备的前、后处理。
- 2) 由处理机本身产生的中断。如算术溢出，除数为零，数据校验错等。
- 3) 由存储器产生的中断。如地址越界、页面失效、访问存储器超时等。
- 4) 由控制器产生的中断。如非法指令、堆栈溢出、时间片到、切换到特权态。
- 5) 由总线产生的中断。输入输出总线出错, 存储总线出错等。
- 6) 实时过程控制产生的中断。
- 7) 实时钟的定时中断。
- 8) 多处理机系统中，从其它处理机发送来的中断。
- 9) 程序调试过程中，由断点产生的中断。
- 10) 硬件故障中断。
- 11) 电源故障中断。

2. 中断源的分类组织

- 中断源分类组织的目的：在响应中断后能尽快找到中断入口。
- 根据中断事件的紧迫程度，中断源工作速度、性质等进行分类
- 为每一类中断源分配一个硬件的中断入口，在进入这个入口之后，再通过软件找到具体的中断源。
- 可屏蔽中断与不可屏蔽中断，或称一般中断和异常中断。

IBM公司的机器，把中断源分为 7 类：

- (1) 机器检验出错中断。由硬件或软件故障时产生。
- (2) 程序性错误引起的中断。
- (3) 访问管理程序中断。当用户程序执行访管指令引起的中断。
- (4) 可以抑制的机器检验错误引起的中断。
- (5) 外部事件中断。
- (6) 输入输出中断。
- (7) 重新启动中断。处理机不能禁止这类中断

3. 中断优先级

- 安排中断优先顺序主要由下列因素来决定：
 - 中断源的急迫性。
 - 设备的工作速度。
 - 数据恢复的难易程度。
 - 要求处理机提供的服务量。
- 中断优先级与中断服务顺序
 - 要求：响应速度快，灵活性好。
 - 做法：由硬件排队器决定中断优先级，通过软件设置中断屏蔽码改变中断服务顺序。

例如：在IBM 370系列机中，把7类中断分为5个中断优先级，从高到低分别是：

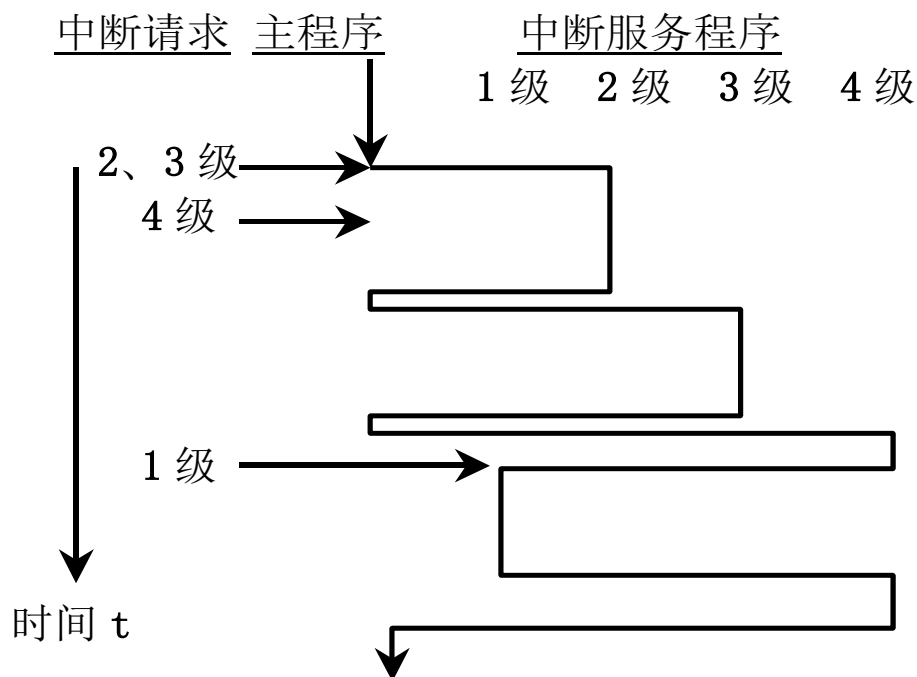
- (1) 紧急的机器检验错误引起的中断
- (2) 调用管理程序，程序性错误，可以抑制的机器检验错误引起的中断。
- (3) 外部事件引起的中断
- (4) 外围设备的中断
- (5) 重新启动引起的中断

例如：DEC公司的机器，其优先级从高到低分别是：

- (1) 总线错误引起的中断
- (2) 主存刷新中断
- (3) 指令错误引起的中断
- (4) 程序跟踪中断
- (5) 电源掉电中断
- (6) 在线停机中断
- (7) 在线事件中断(如实时钟等)
- (8) 外围设备中断
- (9) 用户程序中断

例：某处理机共有4个中断源，中断优先级从高到低分别是：1级、2级、3级和4级。当处理机在执行主程序时，同时有3级和2级两个中断源向处理机发出中断服务请求。当处理机为2级中断源服务时又有4级中断源发出中断服务请求。当处理机为4级中断源服务时又有1级中断源发出中断服务请求。

解：处理机响应各中断源的中断请求和执行中断服务程序的过程如下：



按照中断优先级响应中断请求的例子

4.2.2 中断系统的软硬件分配

- 有些功能必须用硬件实现，有的功能必须用软件实现，而大部分功能既可以用硬件实现，也可以用软件实现。
- 恰当分配中断系统的软硬件功能，是中断系统最关键问题

1. 主要考虑的两个因素：

- 中断响应时间：中断响应时间是一个非常重要的指标。
- 灵活性：硬件实现速度快，灵活性差；软件实现正好相反

2. 中断处理过程（表示一般用硬件实现，表示一般用软件实现，表示可以用硬件实现，也可以用软件实现）

- 现行指令结束，且没有更紧急的服务请求
- 关CPU中断
- 保存断点，主要保存PC中的内容
- 撤消中断源的中断请求
- 保存硬件现场，主要是PSW及SP等
- 识别中断源
- 改变设备的屏蔽状态
- 进入中断服务程序入口

- 保存软件现场，在中断服务程序中使用的通用寄存器等
- 开CPU中断，可以响应更高级别的中断请求
- 中断服务，执行中断服务程序
- 关CPU中断
- 恢复软件现场
- 恢复屏蔽状态
- 恢复硬件现场
- 开CPU中断
- 返回到中断点

- **必须用硬件实现的有：**保存中断点和进入中断服务程序入口。
 - 这两个功能相当于执行一条转子程序指令，因为中断发生在现行程序的什么地方是不确定的，不能由程序员来安排。
- **必须用软件实现的有：**中断服务和返回到中断点。
 - 返回到中断点，通过执行一条中断返回指令来实现，
 - 中断服务必须用软件实现，因为是“程序中断方式”。

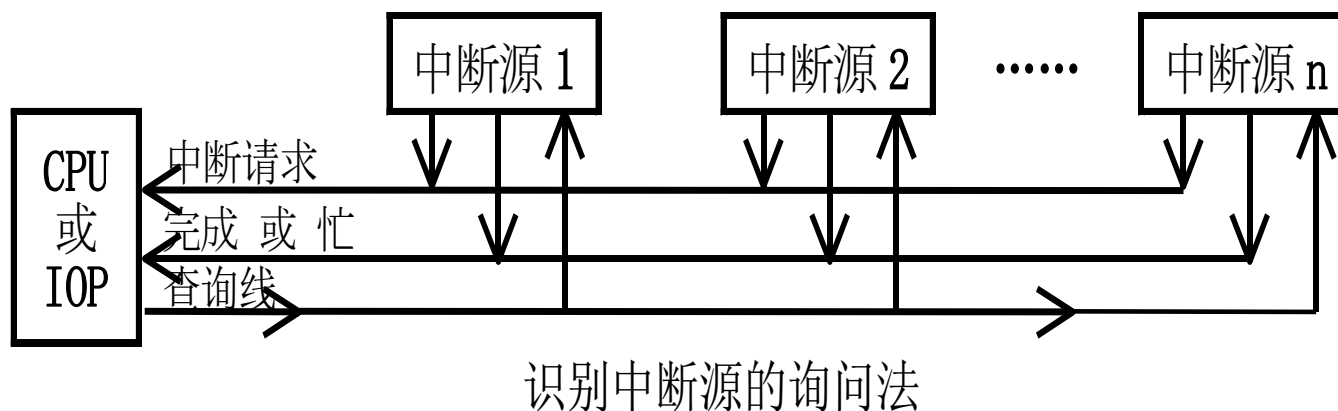
3. 中断响应时间

- 定义：从中断源向处理机发出中断服务请求开始，到处理机开始执行这个中断源的中断服务程序时为止，这一段时间称为中断响应时间。
- 影响中断响应时间的因素主要有4个：（前2个属于处理机设计，后2个属于中断系统）
 - 1) 最长指令执行时间。有些指令的执行时间很长，甚至无法预测。
 - 2) 处理其它更紧急的任务所用时间。如处理DMA请求等。
 - 3) 从第一次关CPU中断到第一次开CPU中断所经历的时间。中断系统的软件与硬件功能分配，主要就是要考虑这一段内要所的事情用软件来实现，还是用硬件来实现。
 - 4) 通过软件找到中断服务程序入口所用时间。主要是第1和第3两部分。其中，第1部分是指令系统设计时考虑的问题，在中断系统的设计中，主要考虑第3部分。

4.2.3 中断源的识别方法

1. 识别中断源的查询法

- 所有中断源共用一条中断请求线
- 处理机响应中断后都进入同一个程序入口
- 用软件找出申请中断的中断源
- 主要优点：灵活性好；主要缺点：速度慢。



例如： 打印机、键盘和显示器三个中断源寻找中断入口的过程：

```
INTR: SKIP DZ, PRN    ; 如果打印机DONE=0, 跳过下一条指令
      JMP  PRINT      ; 转入打印机的中断服务程序入口PRINT
      SKIP DZ, KEY    ; 测试键盘的DONE=0?
      JMP  KEYBORD    ; 转入键盘的中断服务程序入口KEYBORD
      SKIP DN, MNT    ; 测试显示器的DONE=1?
      JMP  ERROR      ; 所有中断源均没有请求中断服务
      JMP  MONITOR    ; 转显示器的中断服务程序入口MONITOR
PRINT: .....        ; 打印机中断服务程序
      RNTI            ; 返回到中断点
KEYBORD: .....      ; 键盘输入的中断服务程序
      RNTI            ; 返回到中断点
MONITOR: .....      ; 显示器输出的中断服务程序
      RNTI            ; 返回到中断点
ERROR: .....        ; 出错处理程序
      RNTI            ; 返回到中断点
```

2. 软件排队链法

- 设置一个中断请求寄存器，每个中断源在其中中占据一位，并且按照中断的优先级从高位到低的顺序排列。
- 所有中断源使用同一条公共的中断请求线，进入公共中断源服务程序入口，其过程与查询法相同。
- 在公共中断服务程序入口，用一条特殊指令读出中断请求寄存器中的内容，并根据读出的内容直接进入中断服务程序。
- 节省了用软件逐个寻找中断源的时间。

一个简单的程序例子:

INTA R1 ;中断请求寄存器中的内容读入R1

SBT R1, R2 ;找到发出请求的最高级中断源

JMP @TAB(R2) ;转向中断服务程序入口

TAB: DEV1 ;最高级中断服务程序入口地址

DEV2 ;第二级中断服务程序入口地址

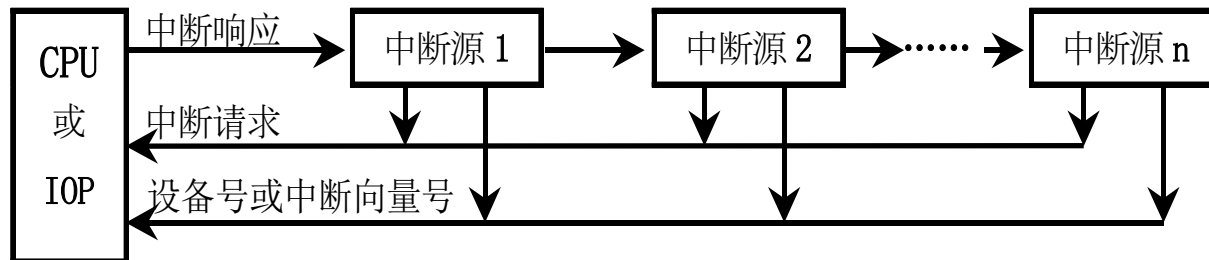
.....

DENn ;最低级中断服务程序入口地址

3. 串行排队链法

- 用硬件排队器和编码器，在所有请求中断服务的中断源中，找出具有最高优先级的中断源。
- 设置一个中断请求寄存器，每个中断源在其中中占据一位。
- 所有中断源使用同一条公共的中断请求线，进入公共中断源服务程序入口。
- 转入公共的中断服务程序后，用一条特殊指令直接读到所有请求中断服务的中断源中，具有最高优先级的中断源编号
- 特点：识别中断源的速度更快

• 硬件排队器和编码器



识别中断源的串行排队链法

一个简单的程序例子:

INTA R1 ;发出请求的最高级中断源送R1

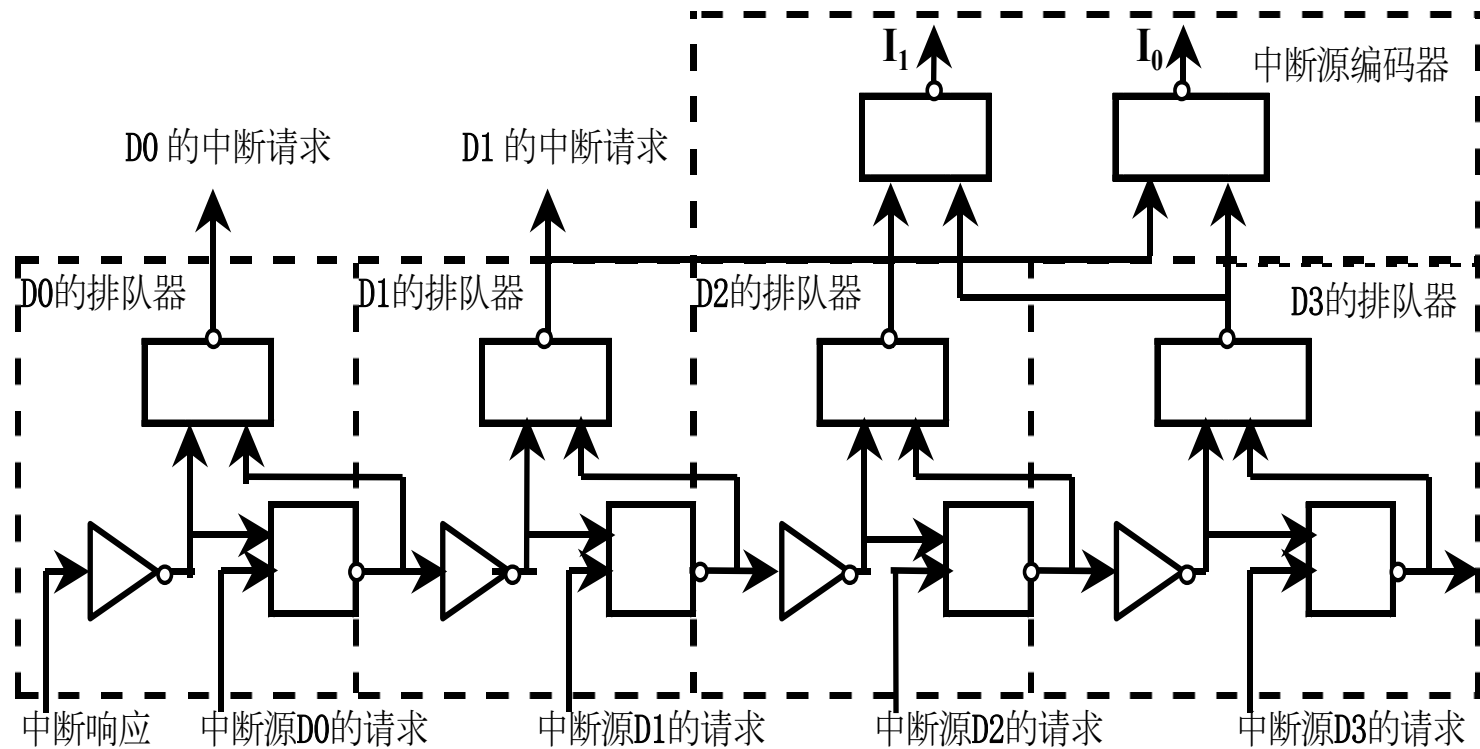
JMP @VTAB(R1);转向中断源的中断服务程序入口

VTAB: DEV1 ;最高级中断服务程序入口地址

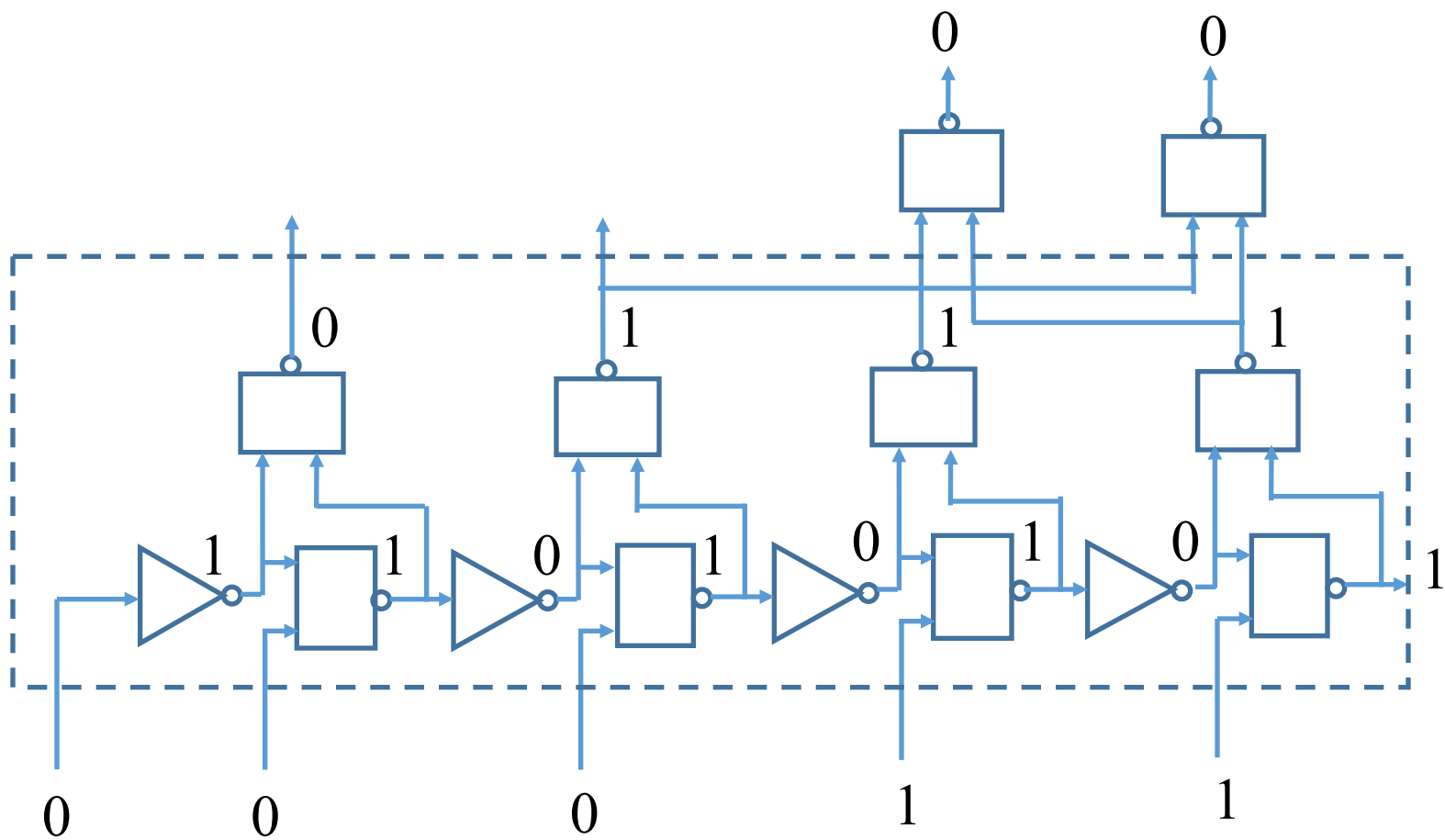
DEV2 ;第二级中断服务程序入口地址

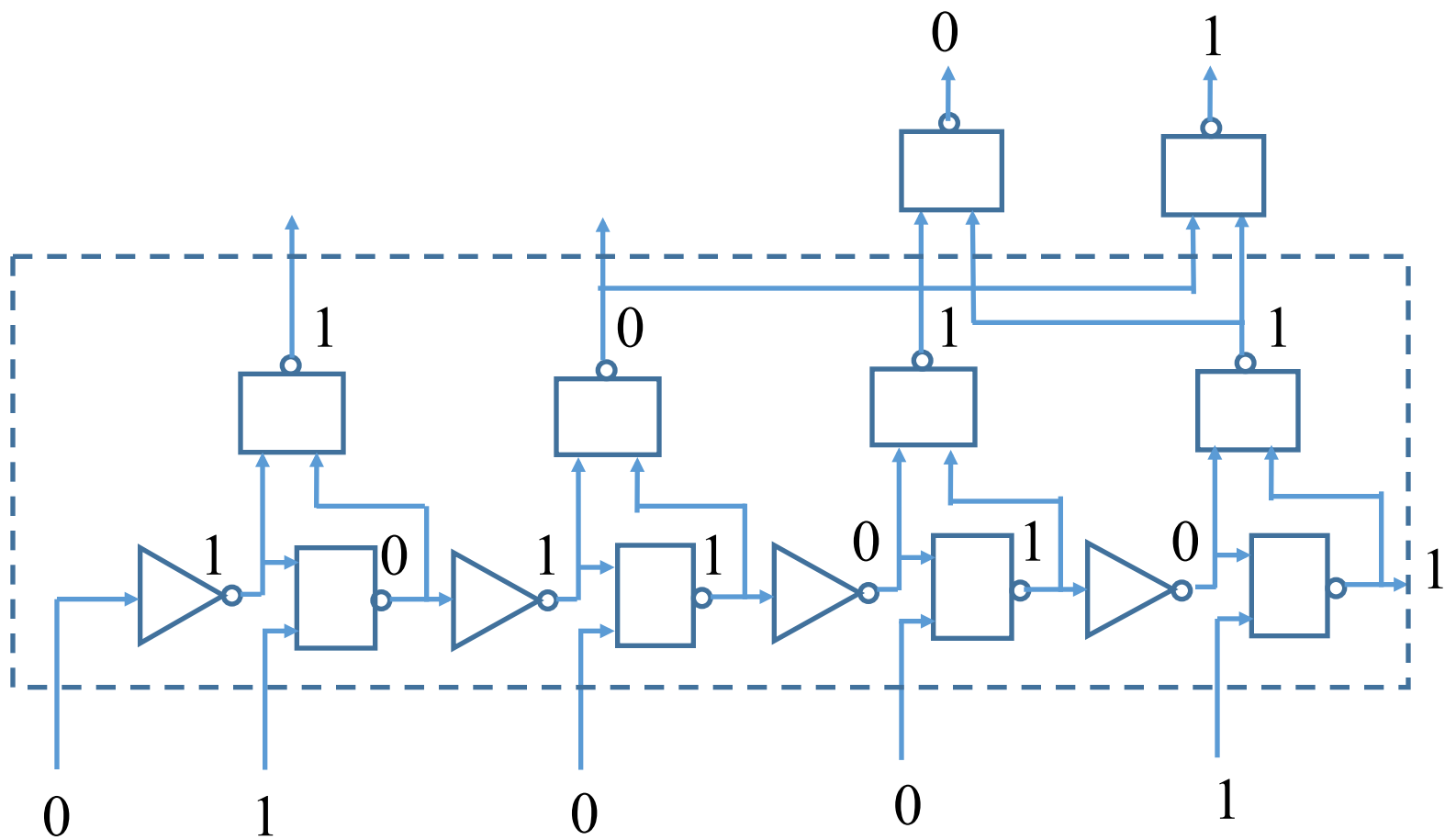
.....

DENn ;最低级中断服务程序入口地址



四个中断源的中断排队器和编码器





4. 中断向量法

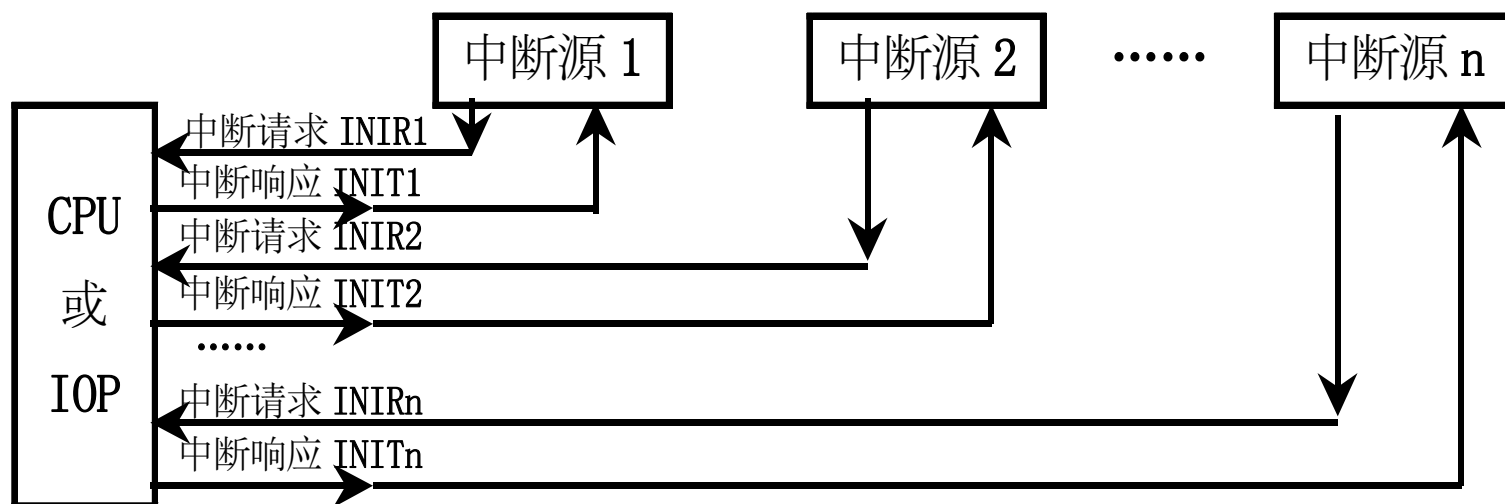
- 在主存储器的固定区域中开辟出一个专用的**中断向量区**。
- 用硬件排队器和编码器在所有请求中断服务的中断源中，产生具有最高优先级的中断源编号。
- 隐含执行上面方法中的两条识别中断源的指令，直接通过硬件转向这个中断源的中断服务程序入口。
- 不需要进入公共的中断服务程序，从而能够实现向中断服务程序入口地址的快速转移

- 上面的2、3、4三种识别中断源的方法都属于串行排队链法。
- 串行排队链法的优点：
 - 识别中断源的速度比较快，特别是中断向量法。
 - 实现比较简单，中断源与处理机的连线很少
- 串行排队链法的缺点：
 - 灵活性比较差，中断优先级是由硬件固定。
 - 可靠性比较差，由于排队链串行分布在各个中断源中。

5. 独立请求法

- 各个中断源使用自己独立的中断请求线。
- 如果同时有多个中断源请求中断服务，通过仲裁线路立即选择具有最高优先级的中断源，并向它发出中断响应信号INIT，处理机就可以立即转入这个中断源的中断服务程序。
- 独立请求法实际上是把分布在各个中断源内的串行排队器都集中到处理机中，从而克服了串行排队链法可靠性差的缺点，但灵活性差的缺点仍然存在。

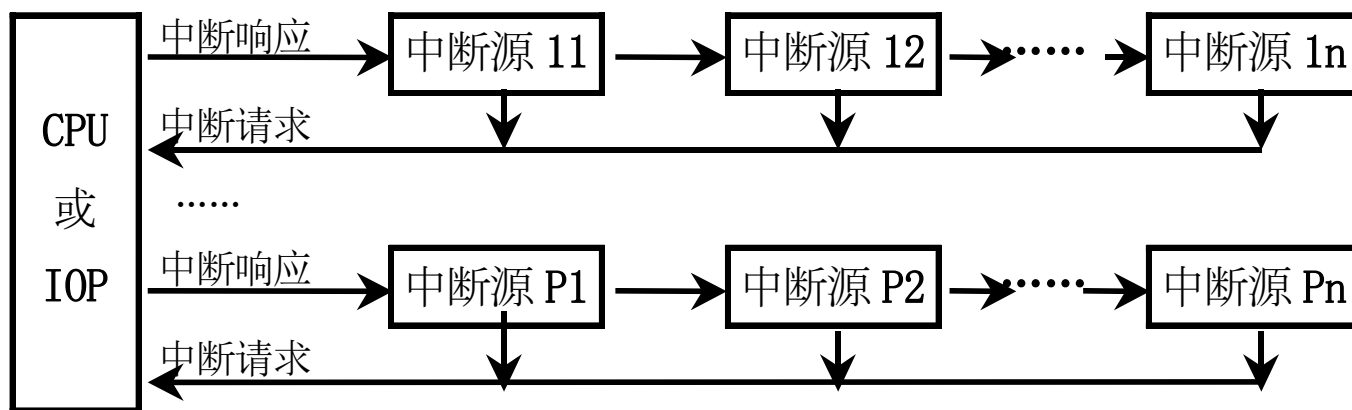
识别中断源的独立请求法



识别中断源的独立请求法

识别中断源的分组独立请求法

- 把独立请求法与串行排队链法结合起来。
- 中断源分组：组内采用串行排队链法，组间采用独立请求法。



识别中断源的分组独立请求法

4.2.4 中断现场的保存和恢复

1) 程序计数器PC，必须由硬件来完成保存

2) 处理机状态字、堆栈指针、基址寄存器、中断屏蔽码等

- 保存与恢复方法有：主存固定区域，压入系统堆栈、交换处理机状态字。也可以采用软件在中断服务程序中保存和恢复。

3) 软件现场：指在中断服务程序中被破坏的通用寄存器。

一般采用软件来保存和恢复现场，指令系统给予适当支持。也有些处理机采用硬件来保存软件现场，如Sparc处理机。

4.2.5 中断屏蔽

- 设置中断屏蔽有三个用处：

- 1) 在中断优先级由硬件确定了的情况下，改变中断源的中断服务顺序。
- 2) 决定设备是否采用中断方式工作。
- 3) 在多处理机系统中，把外围设备的服务工作分配到不同的处理机中。

- 中断屏蔽的实现方法主要有两种：

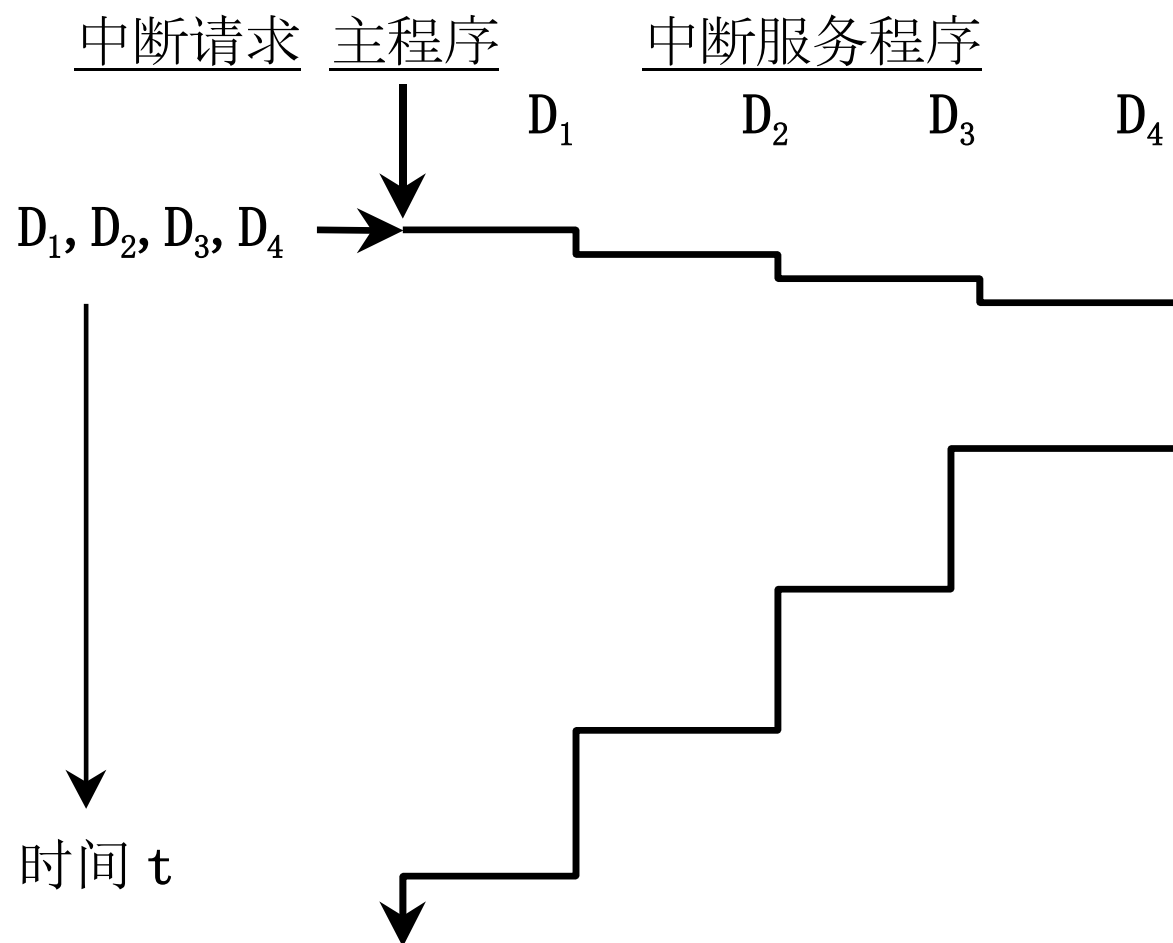
- 1) 每级中断源设置一个中断屏蔽位。
- 2) 改变处理机优先级

例：有四个中断源D₁、D₂、D₃和D₄，它们的中断优先级从高到低分别是1级、2级、3级和4级。这些中断源的正常中断屏蔽码和改变后的中断屏蔽码见下表。每个中断源一位，共4位屏蔽码。

中断源名称	中断优先级	正常中断屏蔽码 D ₁ D ₂ D ₃ D ₄	改变后的中断屏蔽码 D ₁ D ₂ D ₃ D ₄
D1	1	1 1 1 1	1 0 0 0
D2	2	0 1 1 1	1 1 0 0
D3	3	0 0 1 1	1 1 1 0
D4	4	0 0 0 1	1 1 1 1

解:

- 如果4个中断源都使用正常的中断屏蔽码，处理机的中断服务顺序将严格按照中断源的中断优先级进行。
- 如果改变中断屏蔽码，当 D_1 、 D_2 、 D_3 和 D_4 这4个中断源同时请求中断服务时，处理机实际为各个中断源服务的先后次序就会改变。
- 处理机响应的顺序是 D_1 、 D_2 、 D_3 、 D_4
- 实际服务的顺序是 D_4 、 D_3 、 D_2 、 D_1



方法二：改变处理机优先级

例：某处理机共有4个中断源 D_1 、 D_2 、 D_3 和 D_4 ，它们的硬件中断优先级从低到高分别为1级、2级、3级和4级。处理机本身的优先级最低，为0级。在中断源 D_1 、 D_2 、 D_3 、 D_4 的中断向量中，程序员为它们设置的优先级分别为4级、3级、2级、1级。

解：在处理机状态字中设置3个中断屏蔽位。

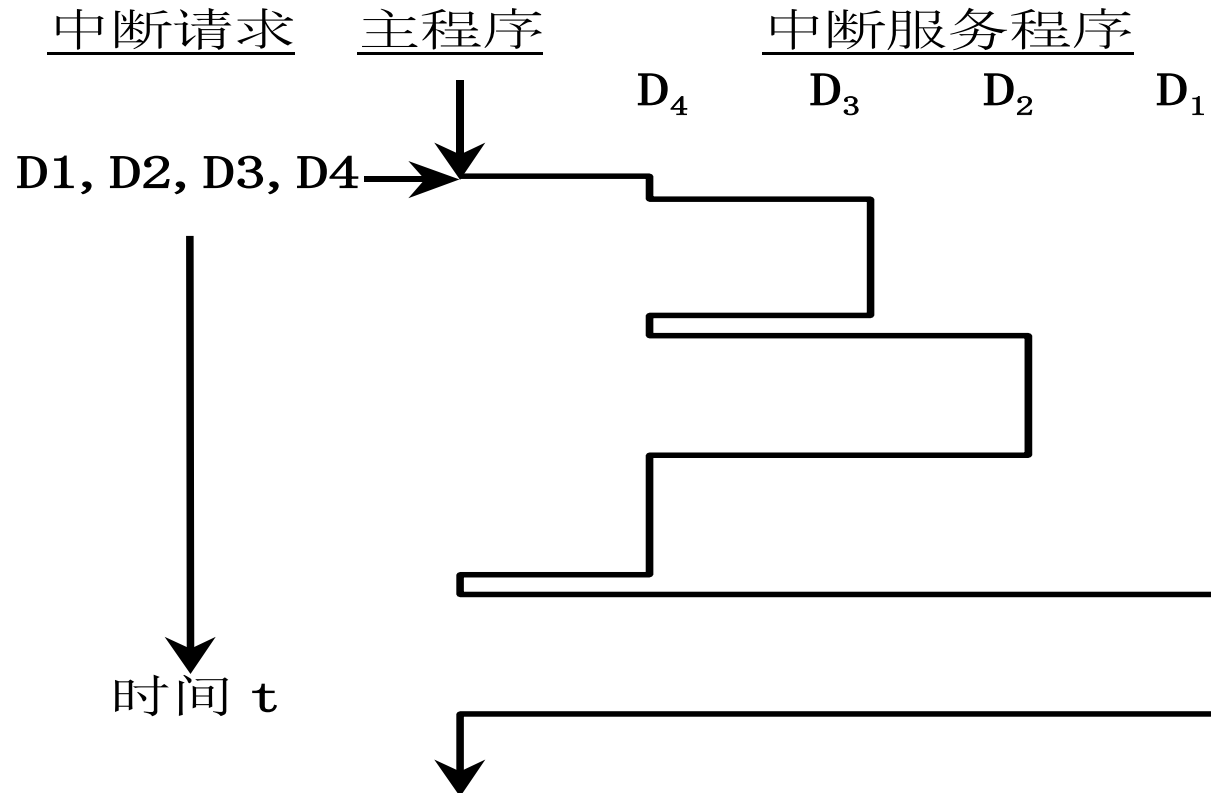
000为处理机本身的优先级，

001～100分别表示4个中断源的中断优先级。

当4个中断源同时请求中断服务时，

解：处理机实际完成中断服务的过程

是 D_3 、 D_2 、 D_4 、 D_1



两种方法的差别有：

(1) 两者使用的概念不同。

前者使用中断屏蔽，

后者使用中断优先级。

(2) 需要屏蔽码的位数不同。

前者所需要的屏蔽位数比较多，

$n: \log_2 n$

(3) 可屏蔽的中断源数量和种类不同。

前者可以任意屏蔽掉一个或几个中断源，

后者只能屏蔽掉比某一个优先级低的中断源。

4.3 通道处理机

把外围设备的管理工作从**CPU**中分离出来
主要用于**IBM**公司研制的机器中

4.3.1 通道的作用和功能

4.3.2 通道的工作过程

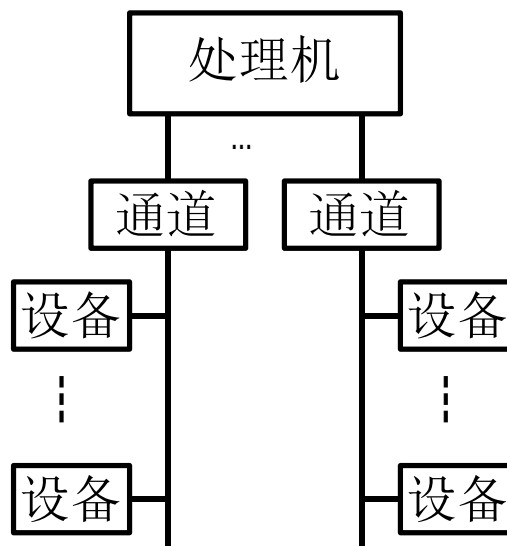
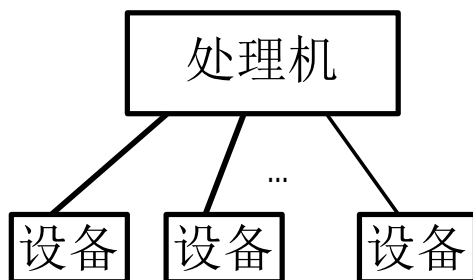
4.3.3 通道的种类

4.3.4 通道中的数据传送过程

4.3.5 通道流量分析

处理机与外部设备的连接方式

- (1) 直接连接
- (2) 通道处理机
- (3) 输入输出处理机



4.3.1 通道的作用和功能

1. 三种基本输入输出方式存在的问题：

- CPU的输入输出负担很重，不能专心用于用户程序的计算工作。
- 低速外围设备，每传送每个字符都由CPU执行一段程序来完成。
- 高速外围设备的初始化、前处理和后处理等工作需要CPU完成。
- 大型机中的外围设备台数很多，但一般并不同时工作。让DMA控制器能被多台设备共享，提高硬件的利用率。

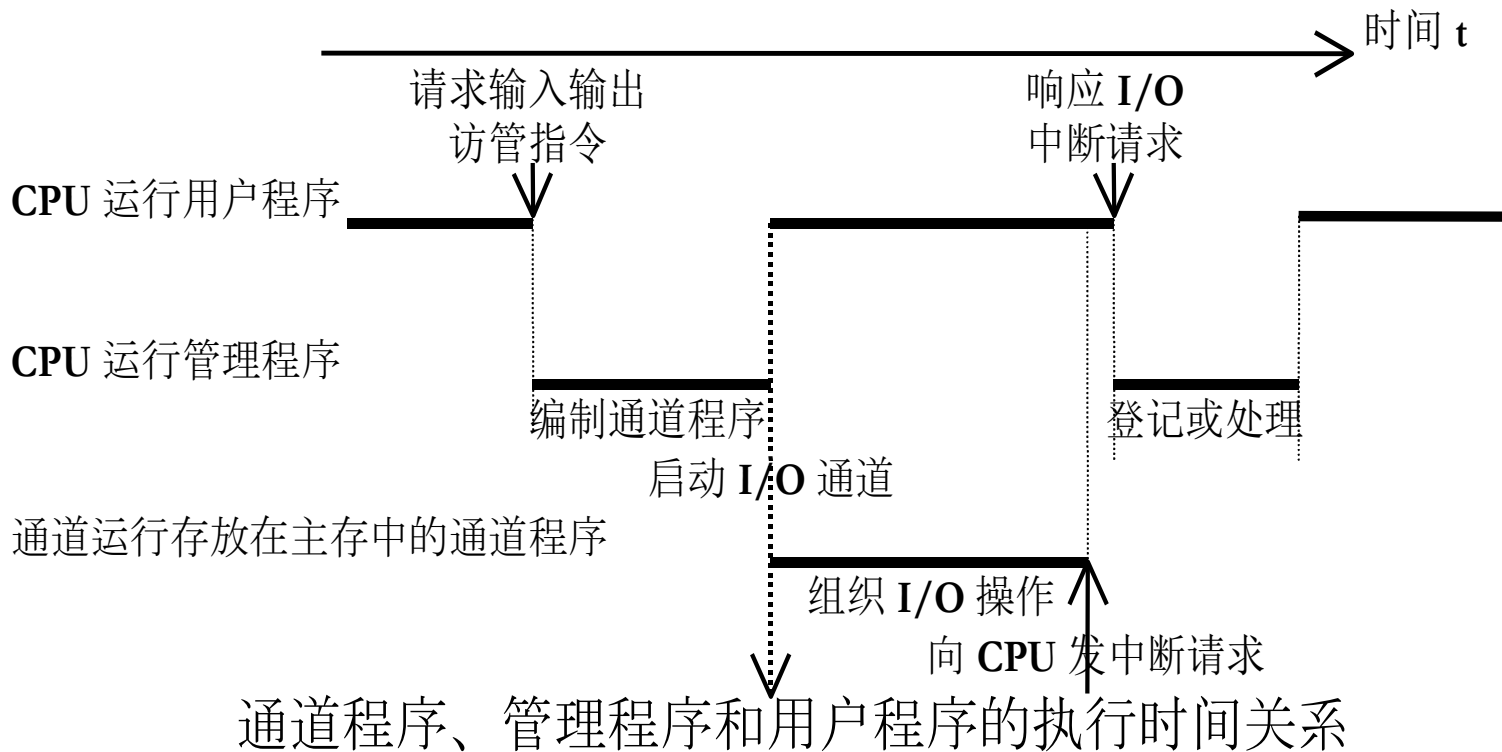
2. 通道的主要功能:

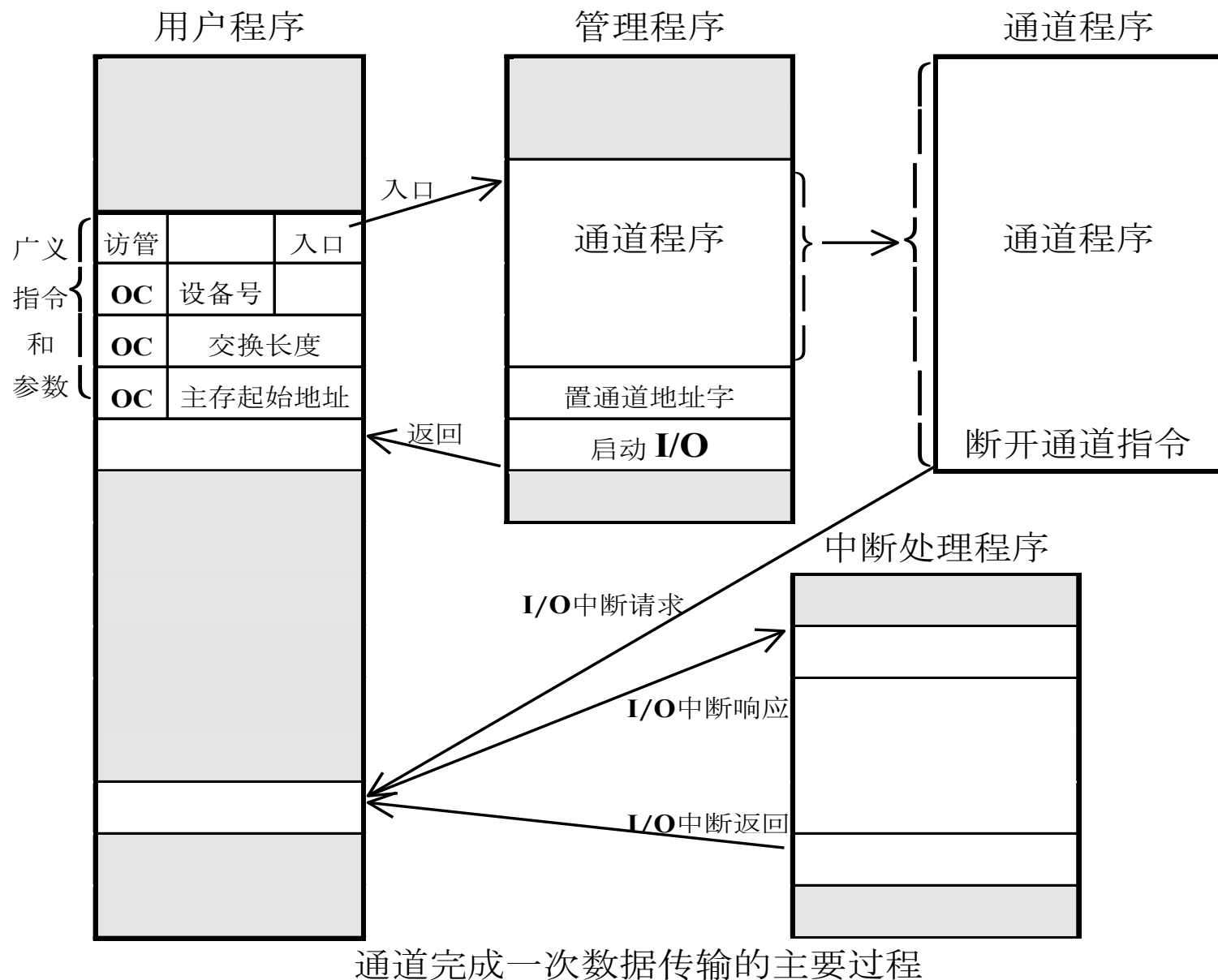
- 接受CPU发来的指令，选择一台指定的外围设备与通道相连接。
- 执行CPU为通道组织的通道程序。
- 管理外围设备的有关地址。
- 管理主存缓冲区的地址。
- 控制外围设备与主存缓冲区之间数据交换的个数。
- 指定传送工作结束时要进行的操作。
- 检查外围设备的工作状态，是正常或故障。
- 在数据传输过程中完成必要的格式变换。

4.3.2 通道的工作过程

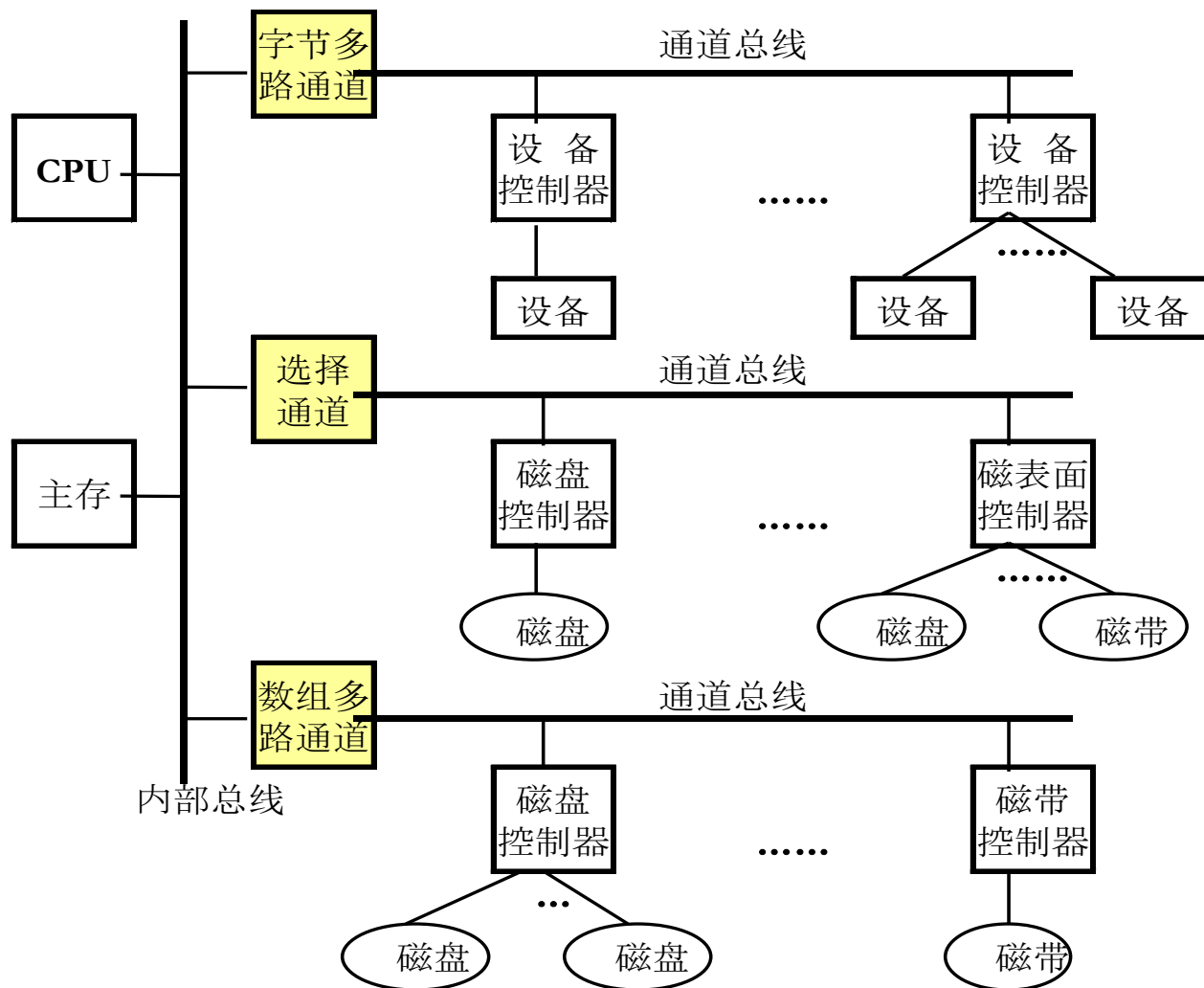
- 通道完成一次数据输入输出的过程分为三步：
 - 在用户程序中使用**访管指令**进入管理程序，由CPU通过管理程序组织一个**通道程序**，并启动通道。
 - 通道处理机执行通道程序，完成指定的数据输入输出工作。
 - 通道程序结束后再次调用管理程序进行处理。
- 每完成一次输入输出工作，CPU只需要两次调用管理程序。

通道的工作过程





4.3.3 通道种类

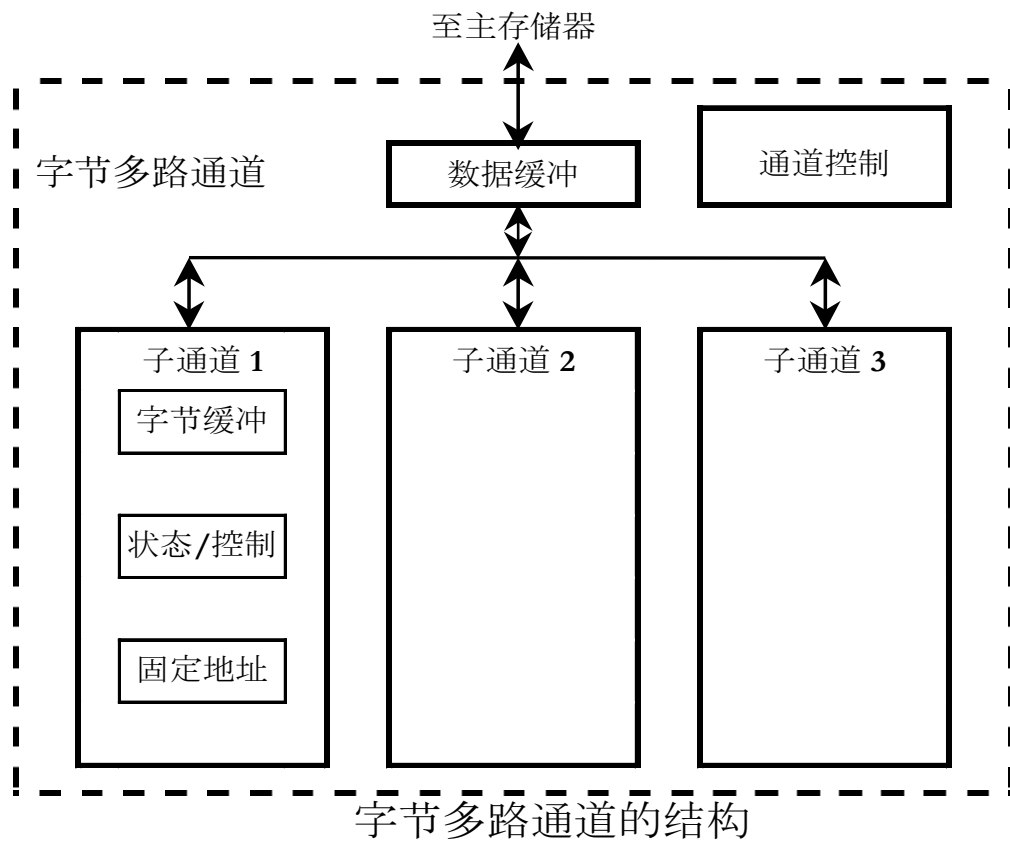


三种类型的通道与 CPU、设备控制器和外围设备的连接关系

1. 字节多路通道

为多台低中速的外围设备服务

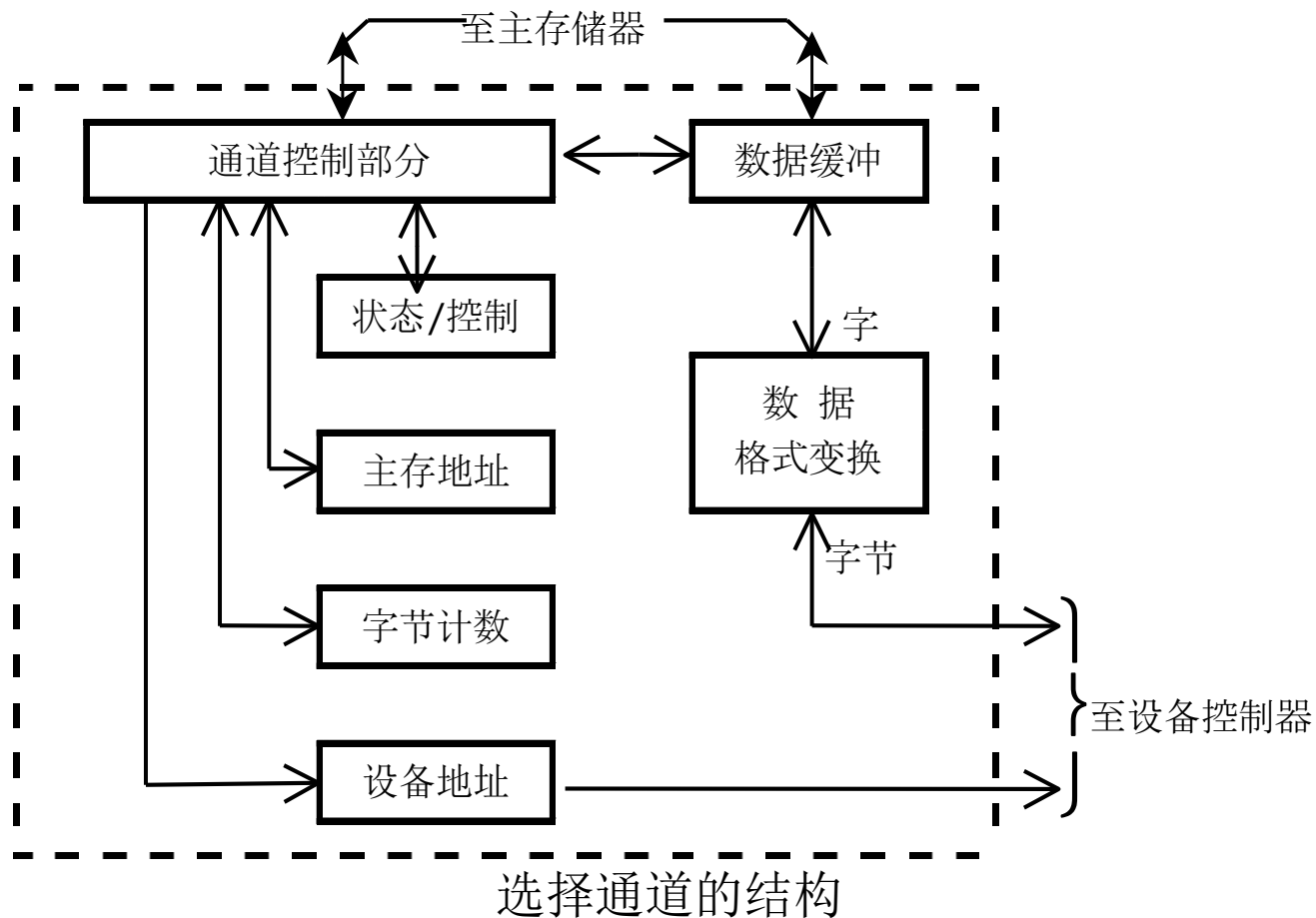
有多个子通道，每个子通道连接一个控制器



2. 选择通道

为高速外围设备服务

只有一个以成组方式工作的子通道



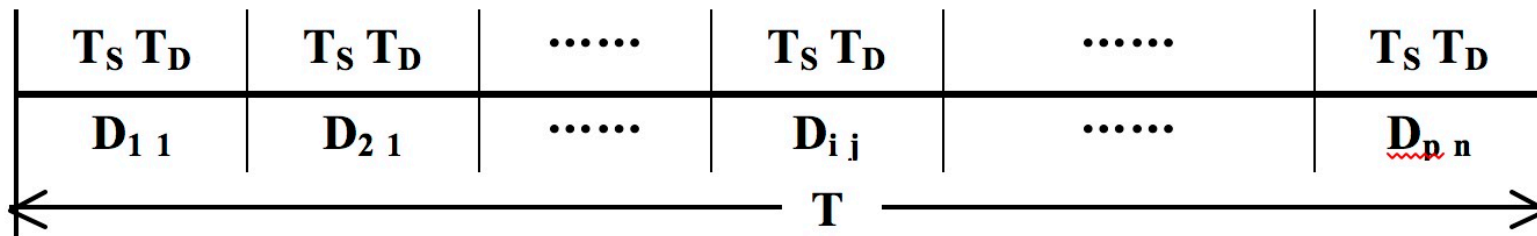
3. 数组多路通道

- 字节多路通道和选择通道的结合。
- 每次为一台高速设备传送一个数据块，并轮流为多台外围设备服务。
- 从磁盘存储器读出文件的的过程分为三步：定位、找扇区、读出数据
- 数组多路通道的实际工作方式是：在为一台高速设备传送数据的同时，有多台高速设备可以在定位或者在找扇区。
- 与选择通道相比，数组多路通道的数据传输率和通道的硬件利用都很高，控制硬件的复杂度也高。

4.3.4 通道中的数据传送过程

字节多路通道的数据传送过程

一个字节多路通道连接 P 台设备，每台设备都传送 n 个字节



T_S : 设备选择时间

T_D : 传送一个字节所用的时间

P : 在一个通道上连接的设备的台数

n : 每一个设备传送的的字节个数

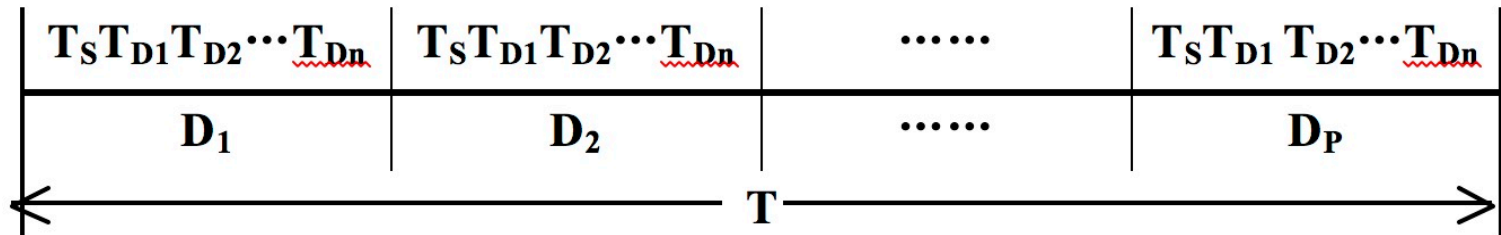
D_{ij} : 连接在通道上的第 i 台设备传送的第 j 个数据

T : 通道完成全部数据传送工作所需要的时间

所需要的总时间: $T_{BYTE} = (T_S + T_D) \cdot P \cdot n$

选择通道的数据传送过程

选择通道连接 P 台设备，每台设备都传送 n 个字节



T_S : 设备选择时间

T_{D_i} : 通道传送第 i 个数据所需要用的时间

T_D : 传送一个字节所用的时间

P : 在一个通道上连接的设备的台数

n : 每一个设备传送的的字节个数

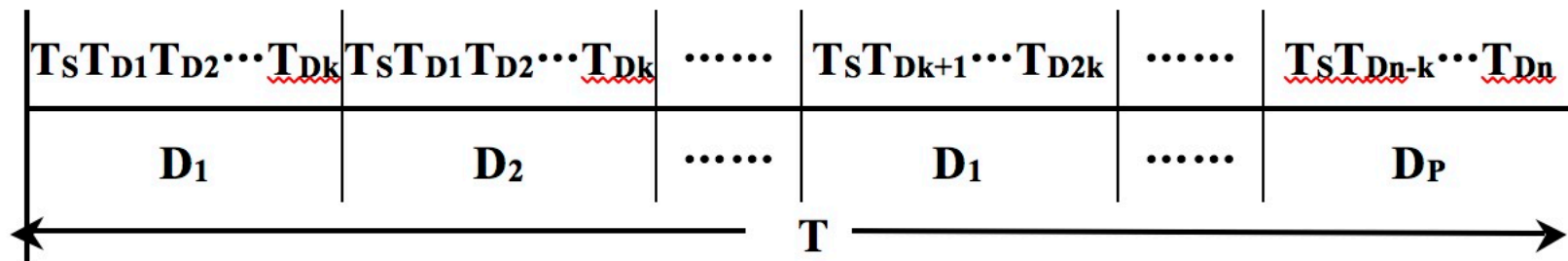
D_i : 通道正在为第 i 台设备服务，其中 $i = 1, 2, \cdots, n$

T : 通道完成全部数据传送工作所需要的时间

所需要的总时间: $T_{\text{SELECT}} = \left(\frac{T_S}{n} + T_D \right) \cdot P \cdot n$

- 数组多路通道的数据传送过程

数组多路通道连接 P 台设备，每台设备都传送 n 个字节



T_S : 设备选择时间

k : 一个数据块中的字节个数 ($k < n$)

T_{D_i} : 通道传送第 i 个数据所需要用的时间

T_D : 传送一个字节所用的时间

P : 在一个通道上连接的设备的台数

n : 每一个设备传送的的字节个数

D_i : 通道正在为第 i 台设备服务，其中 $i = 1, 2, \dots, n$

T : 通道完成全部数据传送工作所需要的时间

所需要的总时间: $T_{BLOCK} = \left(\frac{T_S}{k} + T_D \right) \cdot P \cdot n$

4.3.5 通道流量分析

- **通道流量**：单位时间内能够传送的最大数据量。又称**通道吞吐率**，**通道数据传输率**等。
- **通道最大流量**：通道在满负荷工作状态下的流量。
- 通道流量与连接在通道上的设备的数据传输率的关系如下：

$$f_{BYTE} = \sum_{i=1}^p f_i$$

$$f_{SELECT} = \max_i \{f_i \mid i = 1, \dots, p\}$$

$$f_{BLOCK} = \max_i \{f_i \mid i = 1, \dots, p\}$$

三种通道的最大流量计算公式:

$$f_{MAX.BYTE} = \frac{p \cdot n}{(T_S + T_D) \cdot p \cdot n} = \frac{1}{T_S + T_D} \text{ 字节/秒}$$

$$f_{MAX.SELECTE} = \frac{p \cdot n}{(T_S / n + T_D) \cdot p \cdot n} = \frac{1}{T_S / n + T_D} \text{ 字节/秒}$$

$$f_{MAX.BLOCK} = \frac{p \cdot n}{(T_S / k + T_D) \cdot p \cdot n} = \frac{1}{T_S / k + T_D} \text{ 字节/秒}$$

- 为保证通道不丢失数据, 通道的实际流量应不大于通道最大流量

$$f_{BYTE} \leq f_{MAX.BYTE}$$

$$f_{SELECTE} \leq f_{MAX.SELECTE}$$

$$f_{BLOCK} \leq f_{MAX.BLOCK}$$

例：一个字节多路通道连接 D_1 、 D_2 、 D_3 、 D_4 、 D_5 共5台设备，这些设备分别每10us、30us、30us、50us和75us发出一次数据传送请求。

(1)计算这个通道的实际流量和工作周期。

(2)如果这个字节多路通道的**最大流量正好等于通道实际流量**，并假设数据传输率高的设备，其优先级也高。5台设备在0时刻同时向通道发出第一次传送数据的请求，并在以后的时间里按照各自的数据传输率连续工作。画出通道分时为各台设备服务的时间图，并计算处理完各设备的第一次请求的时刻。

(3)从时间图中发现什么问题？如何解决？

解:

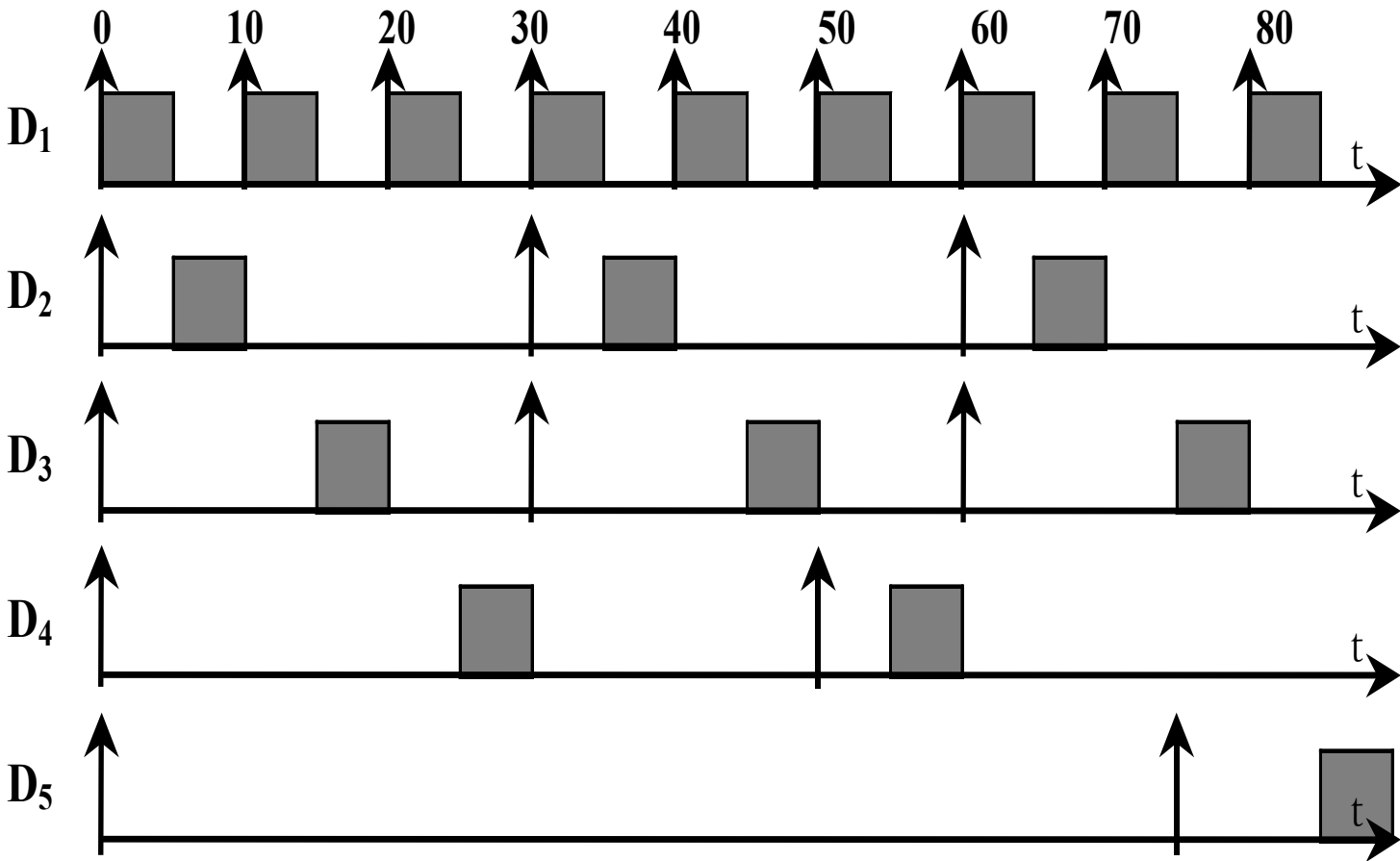
(1) 通道的实际流量为:

$$f_{BYTE} = \left(\frac{1}{10} + \frac{1}{30} + \frac{1}{30} + \frac{1}{50} + \frac{1}{75} \right) MB/S = 0.2 MB/S$$

通道的工作周期为:

$$t = \frac{1}{f_{BYTE}} = 5 \text{ 微秒/字节}$$

通道处理完各设备这个第一次请求的时间:



(2)处理完各设备这个第一次请求的时间：

D_1 : 5us;

D_2 : 10us;

D_3 : 20us;

D_4 : 30us。

D_5 的第一次请求没有响应，数据丢失。

(3) D_5 的第一次请求没有得到响应的原因分析：对所有设备的请求时间间隔取最小公倍数，在这一段时间内通道的流量是平衡的，但在任意一台设备的任意两次时间传送请求之间并不能保证都能得到通道的响应

可以采取下列方法：

方法一：增加通道的最大工作流量。例如，把通道的工作流量增加到0.25MB/S（工作周期为4 μ s）。

方法二：动态改变设备的优先级。例如，在30 μ s至70 μ s之间临时提高设备D₅的优先级。

方法三：增加缓冲存储器。例如，只要为设备D₅增加一个数据缓冲寄存器，它的第一次请求可以在第85 μ s处得到响应，第二次请求可以在第145 μ s处得到响应。

4.4 输入输出处理机

能够独立承担输入输出工作的专用处理机

4.4.1 输入输出处理机的作用

4.4.2 输入输出处理机的种类

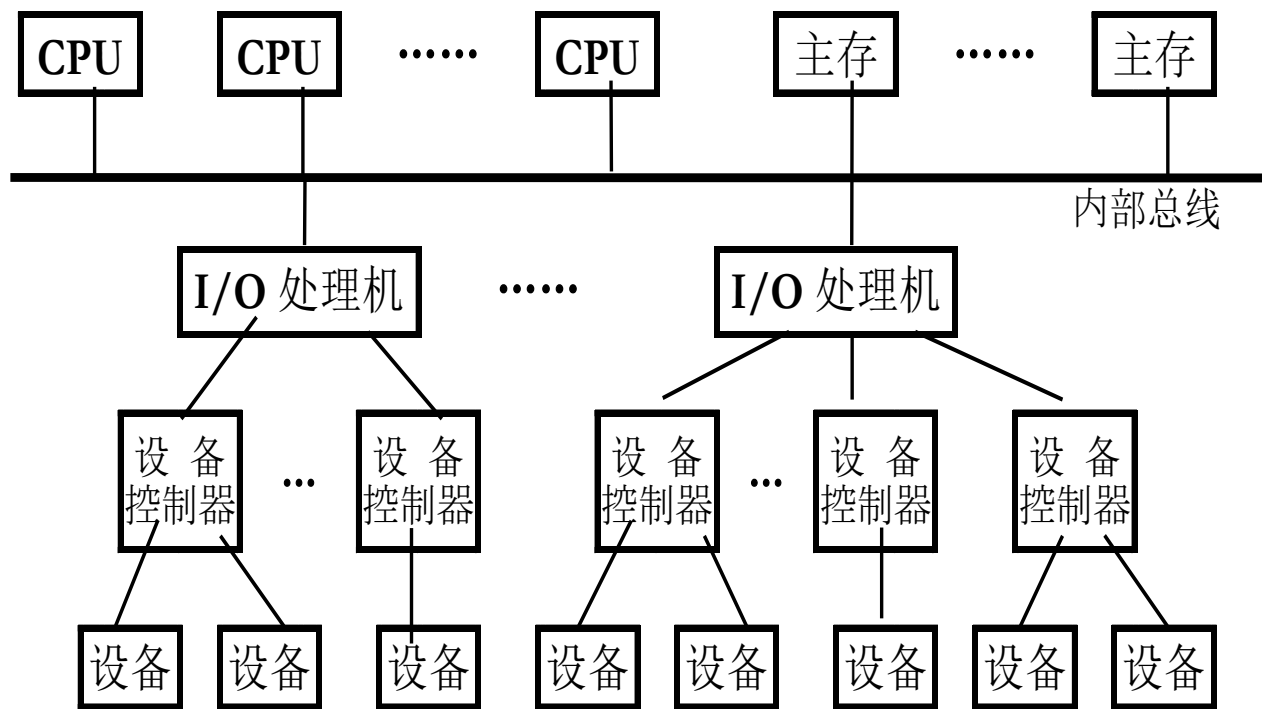
4.4.3 输入输出处理机实例

4.4.1 输入输出处理机的作用

通道处理机存在的问题:

- (1) 每完成一次输入输出操作要两次中断CPU的现行程序。
- (2) 通道处理机不能处理自身及输入输出设备的故障。
- (3) 数据格式转换、码制转换、数据块检验等工作要CPU完成。
- (4) 文件管理、设备管理等工作，通道处理机本身无能为力。

典型的输入输出处理机



输入输出处理机除了能够完成通道处理机的全部功能之外，还具有如下功能：

- (1) 码制转换。
- (2) 数据校验和校正。
- (3) 故障处理。
- (4) 文件管理。
- (5) 诊断和显示系统状态。
- (6) 处理人机对话。
- (7) 连接网络或远程终端。

- 输入输出处理机还可以根据需要完成分配给它的其它任务，如数据库管理等。
- 除了具有数据的输入输出功能之外，还具有运算功能和程序控制等功能。
- 不仅能够执行输入输出指令，还能够执行算术逻辑指令和程序控制指令等，就象一般的处理机那样。

4.4.2 输入输出处理机的种类

根据是否共享主存储器分为：

(1)共享主存储器的输入输出处理机。

CDC公司的CYBER，Texas公司的ASC，

(2)不共享主存储器的输入输出处理机。

STAT-100巨型机

根据运算部件和指令控制部件是否共享分为：

(1)合用同一个运算部件和指令控制部件。

造价低，控制复杂。如CDC-CYBER和ASC

(2)独立运算部件和指令控制部件。

独立运算部件和指令控制部件已经成为主流。

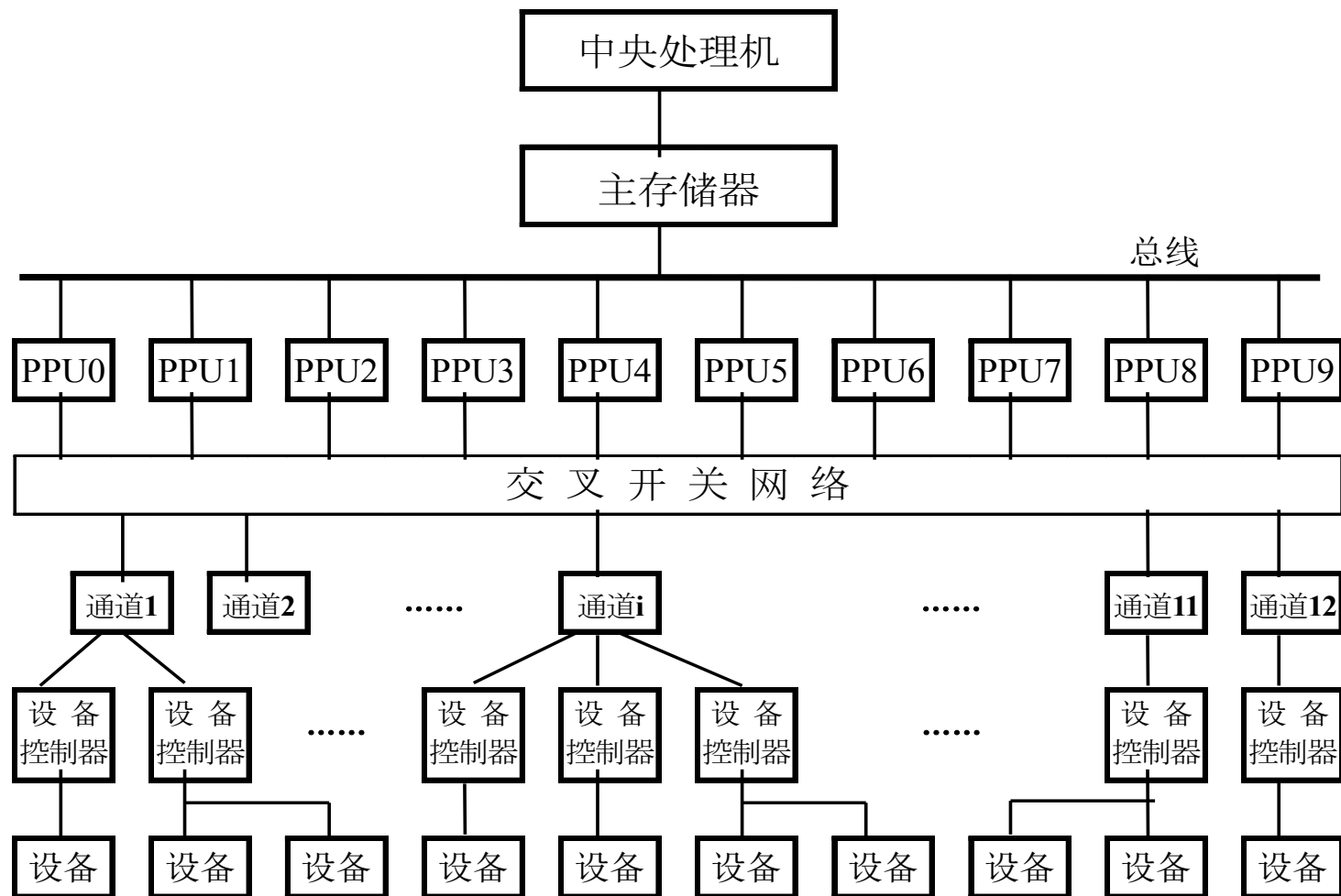
如B-6700大型机和STAT-100巨型机等。

输入输出处理机的多种组织方式:

- (1)多个输入输出处理机从功能上分工。
- (2)以输入输出处理机作为主处理机。
- (3)采用与主处理机相同型号的处理机作为输入输出处理机。
- (4)采用廉价的微处理机来专门承担输入输出任务。

4.4.3 输入输出处理机实例

- CYBER1700大型机的输入输出处理机
- PPU0~PPU9通过总线分时共享主存储器，并通过交叉开关网络共享12个输入输出通道
- 每个PPU有一个容量 $4K \times 13$ 位的局部存储器
- 每台PPU有66种指令，包括算逻指令、访存指令，输入输出指令及程序控制指令等。
- 当用户程序需要进行输入输出操作时，由中央处理机发出请求调用输入输出处理机，由输入输出处理机管理外围设备完成全部输入输出工作。



CYBER 1700 计算机系统的结构框图

本章重点：

1. 了解三种基本输入输出方式的原理及特点
2. 中断系统中的软硬件功能分配
3. 中断优先级和中断屏蔽的原理及方法
4. 通道中的数据传送过程与流量分析
5. 输入输出处理机的作用及种类