第4章　输入输出系统

肖梦白

xiaomb@sdu.edu.cn

https://xiaomengbai.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

4.5 总线

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。

4.1.1 输入输出系统的特点

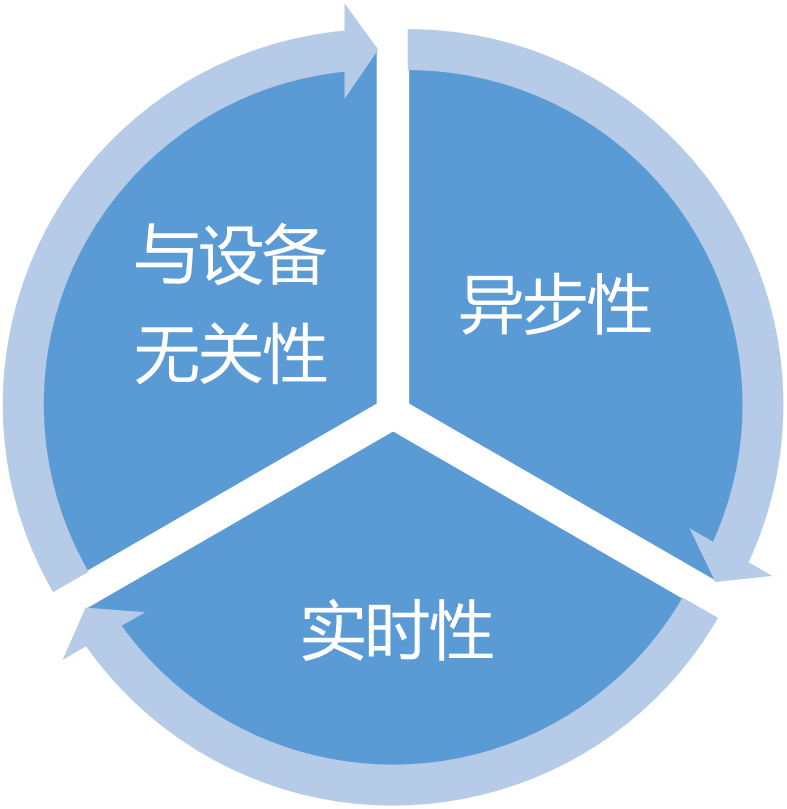
4.1.2 输入输出系统的组织方式

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。

- 输入输出系统的特点集中反映在异步性、实时性和与设备无关性三个基本要求上，并对输入输出系统的组织产生决定性影响



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

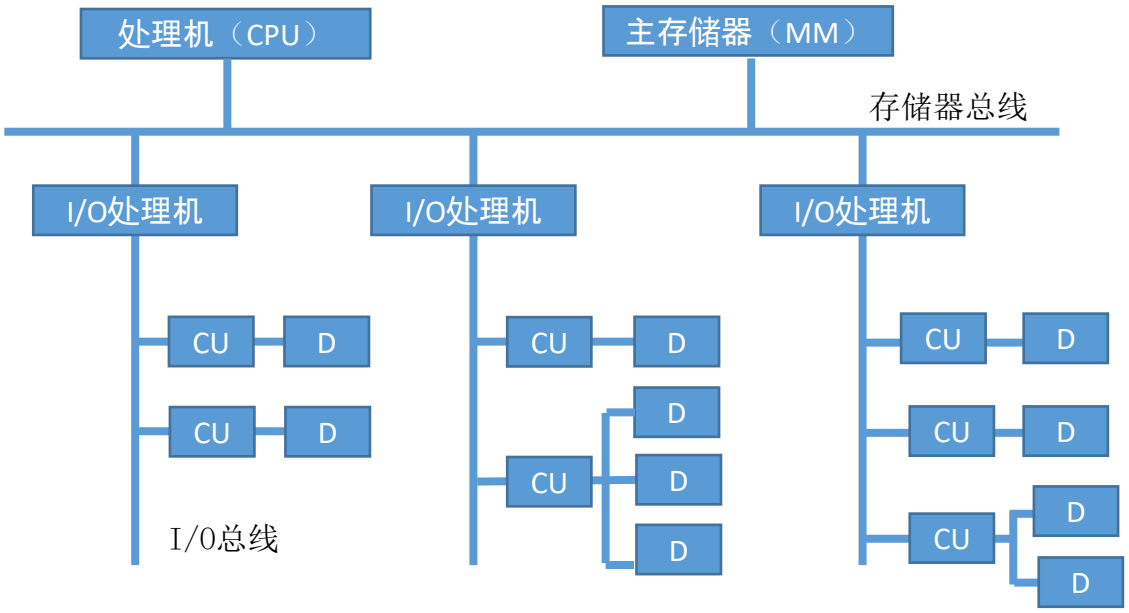
- 独立于具体设备的标准接口。例如，串行接口、并行接口、SCSI（Small Computer System Interface）接口等
- 计算机系统的使用者，在需要更换外围设备时，各种不同型号，不同生产厂家的设备都可以直接通过标准接口与计算机系统连接。
- 处理机采用统一的硬件和软件对品种繁多的设备进行管理。
- 某些计算机系统已经实现了即插即用技术。

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
- 外围设备与处理机之间的交互式随机的
- 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

4.1.2 输入输出系统的组织方式

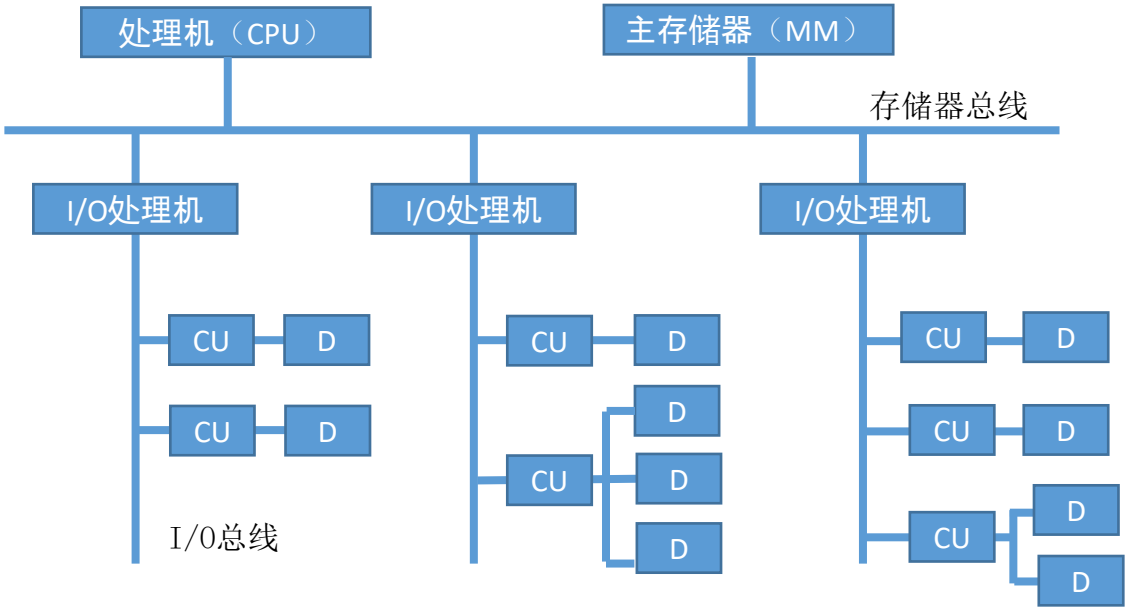
- 针对异步性，采用自治控制的方法
  - 输入输出系统是独立于CPU之外的自治系统
  - 处理机与外围设备之间要有恰当的分工



4.1.2 输入输出系统的组织方式

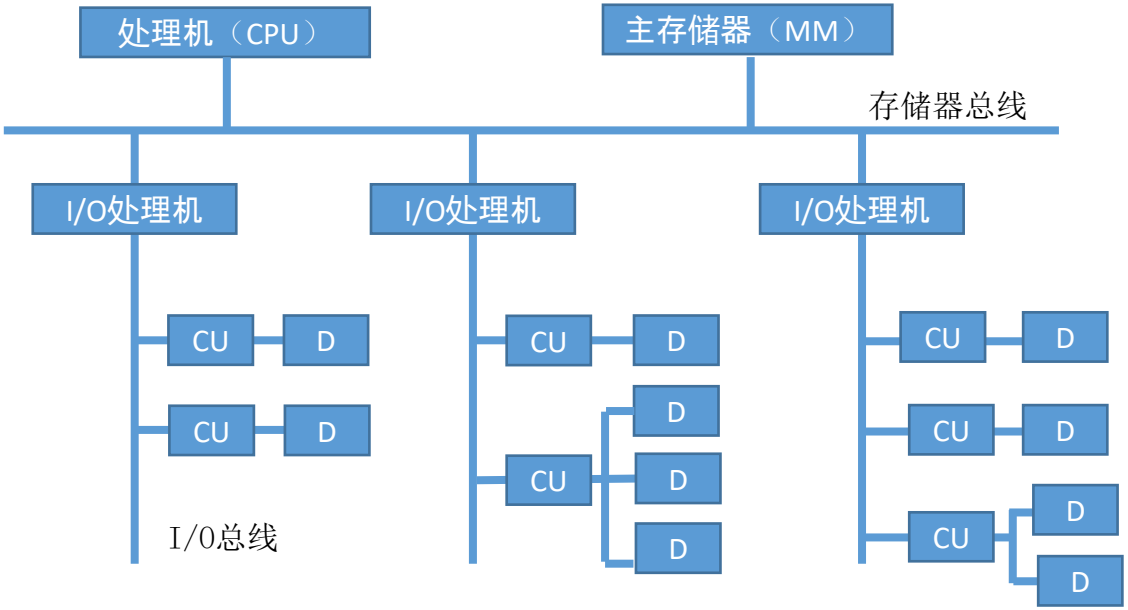
➢针对实时性，采用**层次结构**的方法

- 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



4.1.2 输入输出系统的组织方式

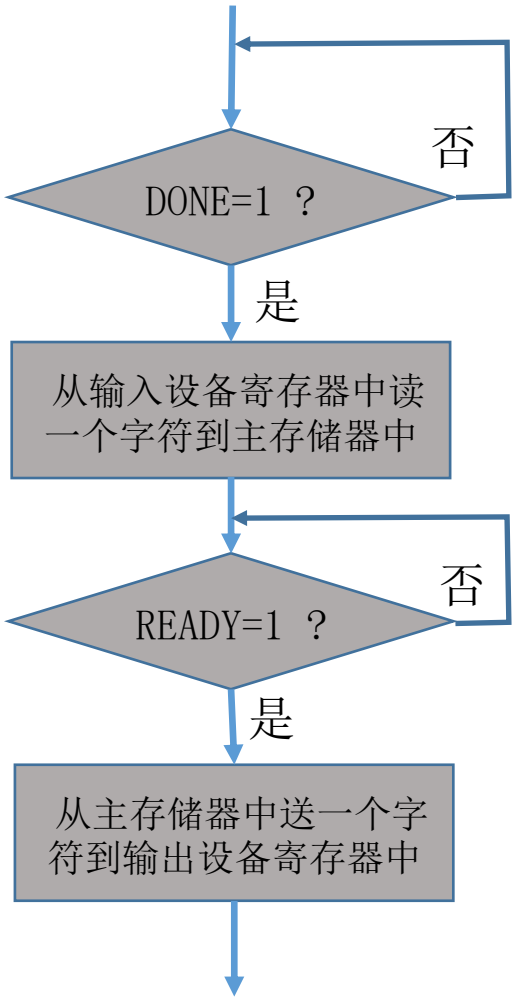
- 针对与设备无关性，采用**分类处理**的方法
  - 根据工作方式、工作速度和使用场合进行分类
  - 例如：按照工作速度分类：
    - 面向字符的设备，如字符终端、打字机等
    - 面向数据块的设备，如磁盘、磁带、光盘等。



4.1.3 基本输入输出方式

1. 程序控制输入输出方式（Polling）

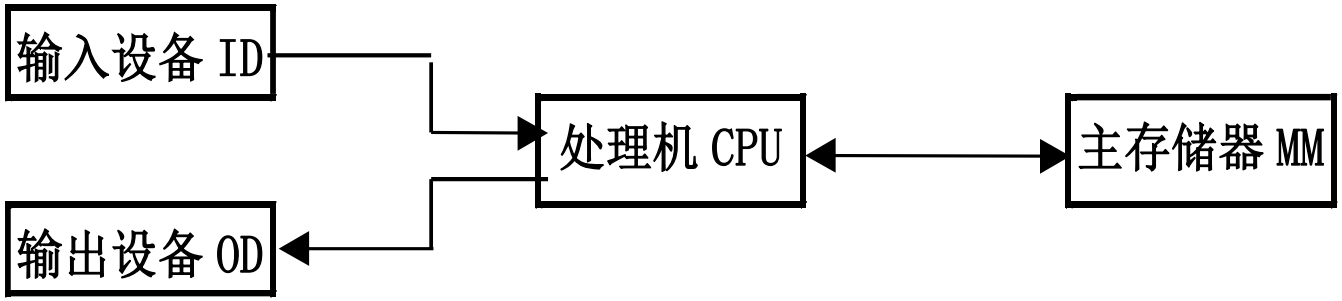
- 状态驱动输入输出方式、应答输入输出方式、查询输入输出方式、条件驱动输入输出方式
- 程序控制输入输出方式的4个特点：
  1) 何时对何设备进行输入输出操作受CPU控制
  2) CPU要通过指令对设备进行测试才能知道设备的工作状态。空闲、准备就绪、忙碌等
  3) 数据的输入和输出都要经过CPU
  4) 用于连接低速外围设备，如终端、打印机等



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。

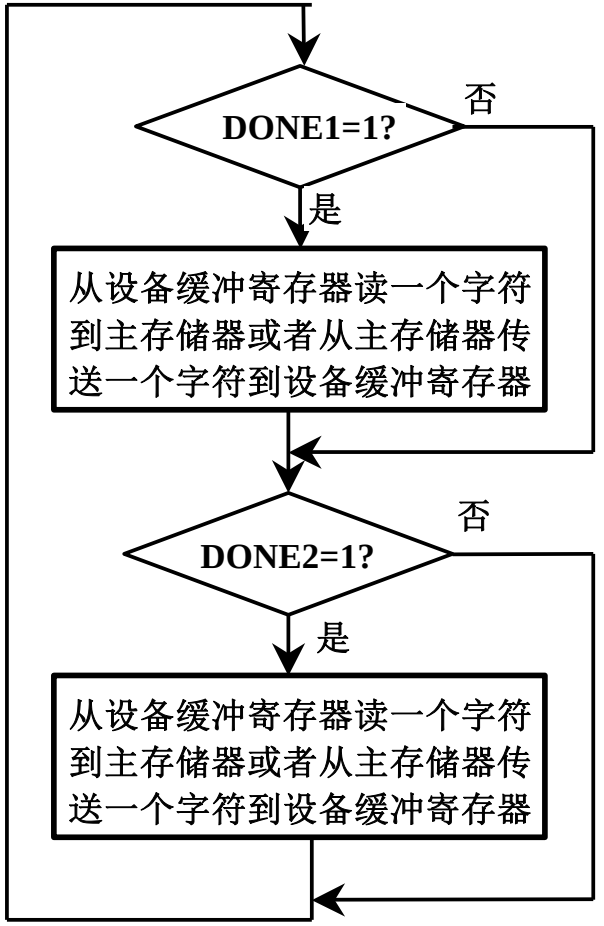
解：处理机用一条指令就能向打印机传送4个字符。因此，处理机的实际利用率只有即4千万分之一。

$$100/(10^9\times4)=0.25\times10^{-7}$$



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

4.1.3 基本输入输出方式

2.中断输入输出方式（Interrupts）

- **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。

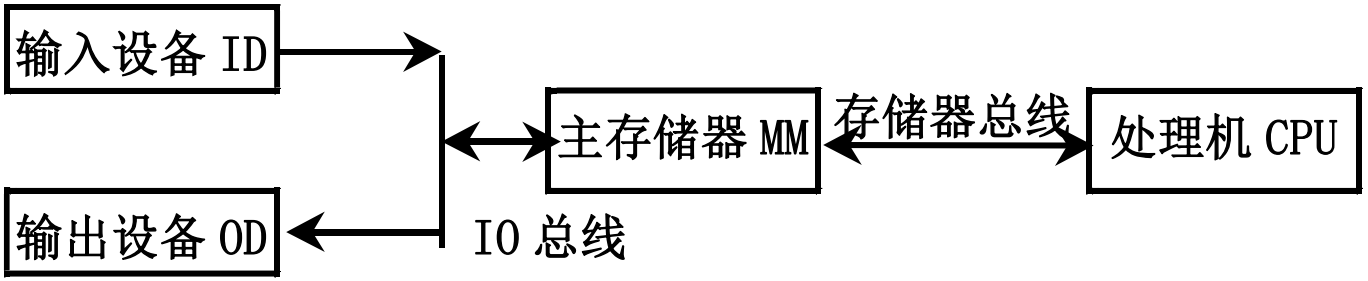
  3）数据的输入和输出都要经过CPU。

  4）用于连接低速外围设备。

4.1.3 基本输入输出方式

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 方式的数据传送过程

4.1.3 基本输入输出方式

3.直接存储器访问方式

➢ DMA方式具有如下特点：

1） 主存储器既可以被CPU访问，也可以被外围设备访问

2） 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，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，其数据交换过程全部在硬件控制下完成

4） 在DMA方式开始和结束时，需要处理机进行管理

5） CPU与外围设备并行工作，整个数据传送过程不需要CPU的干预

4.1.3 基本输入输出方式

3.直接存储器访问方式

- DMA输入设备的工作流程如下：

1） 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）

2） 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；

3） 若一个字已装配满，则将BD中的数据送主存数据寄存器。

4） 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

5） 把DMA控制器内的数据交换个数计数器减 1

6） 若交换个数为0，则DMA数据传送过程结束，否则回到上面

4.1.3 基本输入输出方式

3.直接存储器访问方式

- DMA输出设备的工作流程如下：

1） 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

2） 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

3） 把BD中的数据写到输出介质上（可能要逐个字符输出）

4） 把DMA控制器内的数据交换个数计数器中的内容减 1

5） 若交换个数为0，则DMA数据传送过程结束，否则回到上面

4.1.3 基本输入输出方式

3.直接存储器访问方式

➢目前使用的DMA方式实际上有如下三种：

1） 周期窃取方式：

2） 直接存取方式：

3） 数据块传送方式：

4.1.3 基本输入输出方式

3.直接存储器访问方式

➢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：

3） 数据块传送方式：

4.1.3 基本输入输出方式

3.直接存储器访问方式

➢目前使用的DMA方式实际上有如下三种：

1） 周期窃取方式：

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：

4.1.3 基本输入输出方式

3.直接存储器访问方式

➢目前使用的DMA方式实际上有如下三种：

1） 周期窃取方式：

2） 直接存取方式：

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。

RDMA (Remote Direct Memory Access)

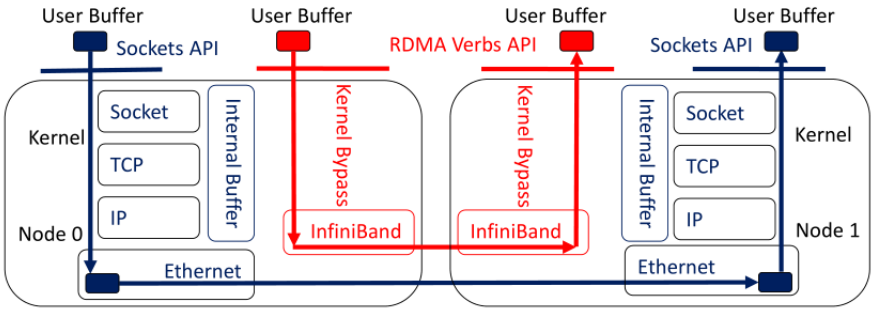


Fig. 4: TCP/IP *vs.* RDMA. RDMA can directly send data in a user buffer of the local node to a receive buffer at the remote node (or vice versa). The remote CPU and both side OS kernels are bypassed. While, TCP/IP needs OS kernels to process data packets, involving several internal memory copies. CPUs on both sides are also required.

Polling vs. Event-based

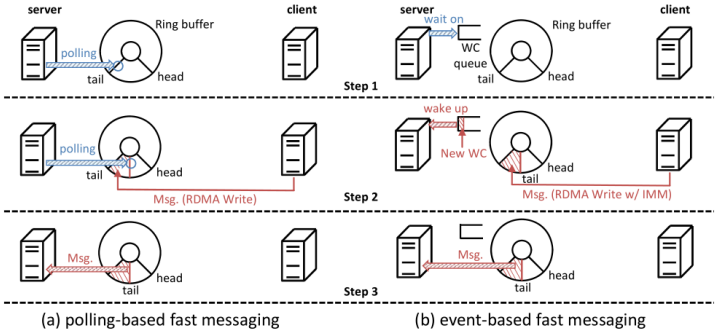


Fig. 6: Polling- *vs.* event-based fast messaging. (a) For polling-based fast messaging, the server keeps polling the tail of the ring buffer for detecting message arrival (**step 1**). As a message is starting to be delivered to the ring buffer via RDMA Write, the server changes the polling position to know when delivery is completed (**step 2**). Eventually, the server can retrieve the complete message from the ring buffer (**step 3**). (b) For event-based fast messaging, the server waits on a Work Completion (WC) queue and yields the CPU (**step 1**). If a message has arrived via RDMA Write with Immediate Data (RDMA Write w/ IMM), the RDMA NIC will also generate a notification in the WC queue and wake up the thread waiting on it (**step 2**), which then retrieves the message (**step 3**).

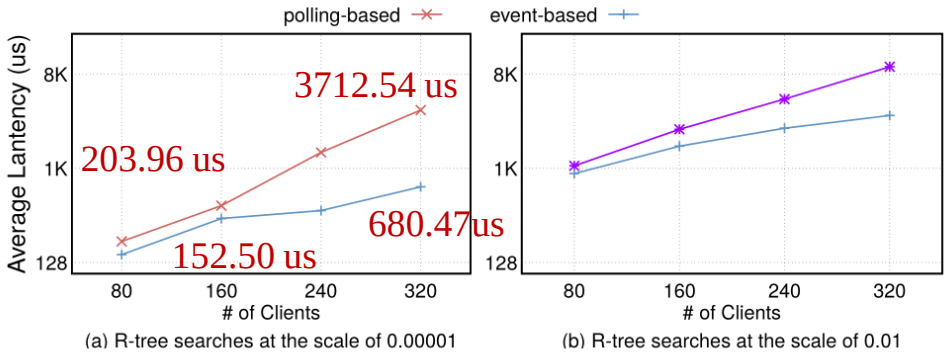


Fig. 7: Performance comparisons of polling- *vs.* event-based fast messaging on 100 Gbps InfiniBand.

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。

2. 中断源的分类组织

- 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

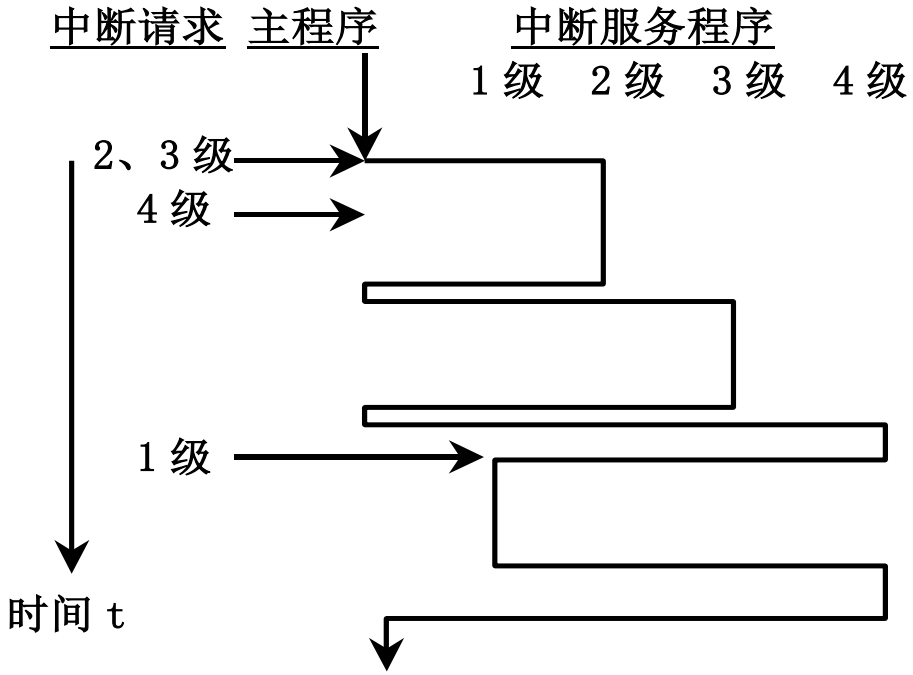
(7)在线事件中断(如实时钟等)

(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

4.2.2 中断系统的软硬件分配

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

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

4.2.2 中断系统的软硬件分配

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

9) 保存软件现场，在中断服务程序中使用的通用寄存器等

10) 开CPU中断，可以响应更高级别的中断请求

11) 中断服务，执行中断服务程序

12) 关CPU中断

13) 恢复软件现场

14) 恢复屏蔽状态

15) 恢复硬件现场

16) 开CPU中断

17) 返回到中断点

4.2.2 中断系统的软硬件分配

2. 中断处理过程

➢**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”。

4.2.2 中断系统的软硬件分配

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3个：（前2个属于处理机设计，后1个属于中断系统）

  1） 最长指令执行时间。

  2） 处理其它更紧急的任务所用时间。

  3） 从第一次关CPU中断到第一次开CPU中断所经历的时间。

  - 撤销中断服务请求、保存软硬件现场、识别中断源、改变设备屏蔽状态

4.2.2 中断系统的软硬件分配

3. 中断响应时间

- 影响中断响应时间的因素主要有3个：

  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。

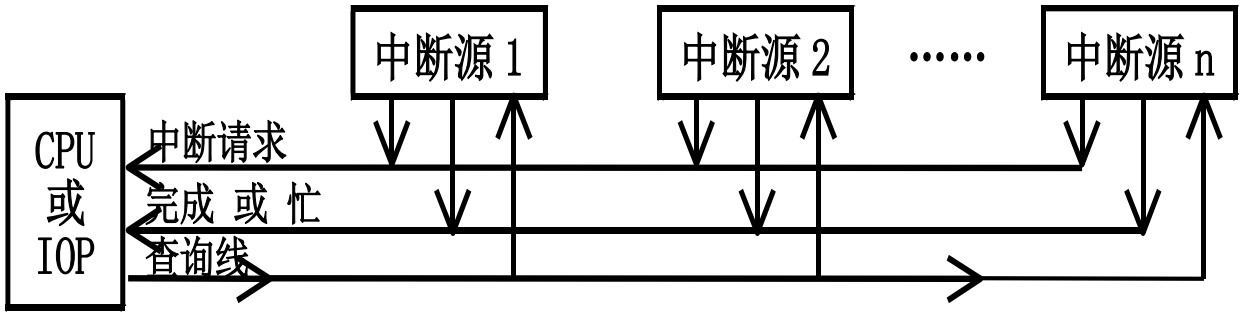
  2）处理其它更紧急的任务所用时间。如处理DMA请求等。

  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。

4.2.3 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。



识别中断源的询问法

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
```

4.2.3 中断源的识别方法

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，用一条特殊指令将中断请求寄存器中的内容读到通用寄存器，然后用一条按位扫描指令找到第一位为“1”的位号（该位号实际上是所有请求中断服务的中断源中，具有最好优先级的中断源的相对编号）。
- 通过变址转移指令，直接转移到这个中断源的中断服务程序入口
- 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：

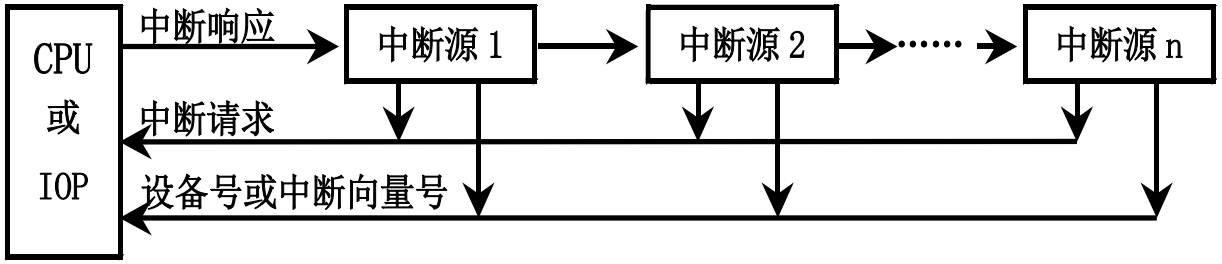
```
        INTA  R1          ;中断请求寄存器中的内容读入R1
        SBT   R1，R2      ;找到发出请求的最高级中断源
        JMP  @TAB(R2)     ;转向中断服务程序入口
TAB:  DEV1                ;最高级中断服务程序入口地址
        DEV2              ;第二级中断服务程序入口地址
        ……
        DENn              ;最低级中断服务程序入口地址
```

4.2.3 中断源的识别方法

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

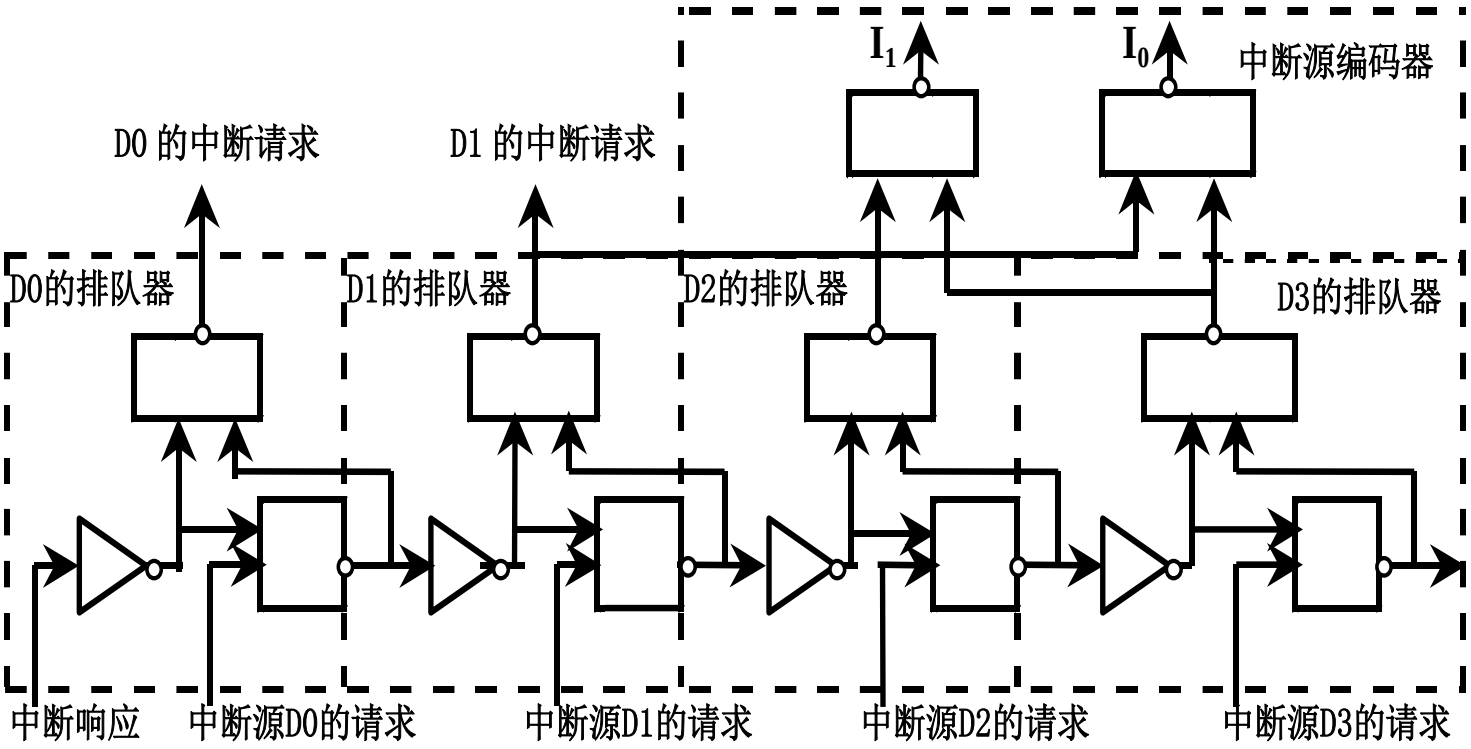


识别中断源的串行排队链法

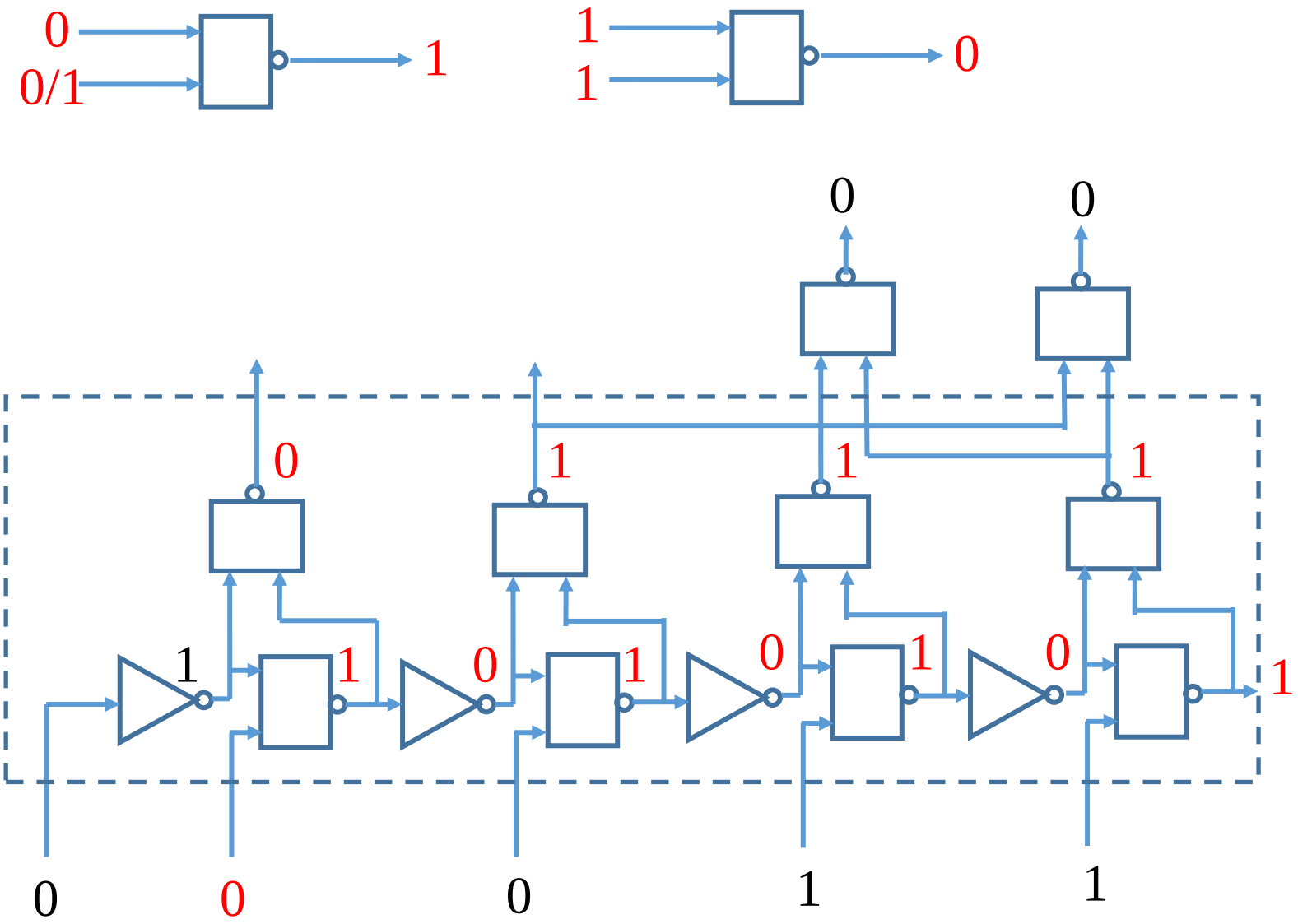
一个简单的程序例子：

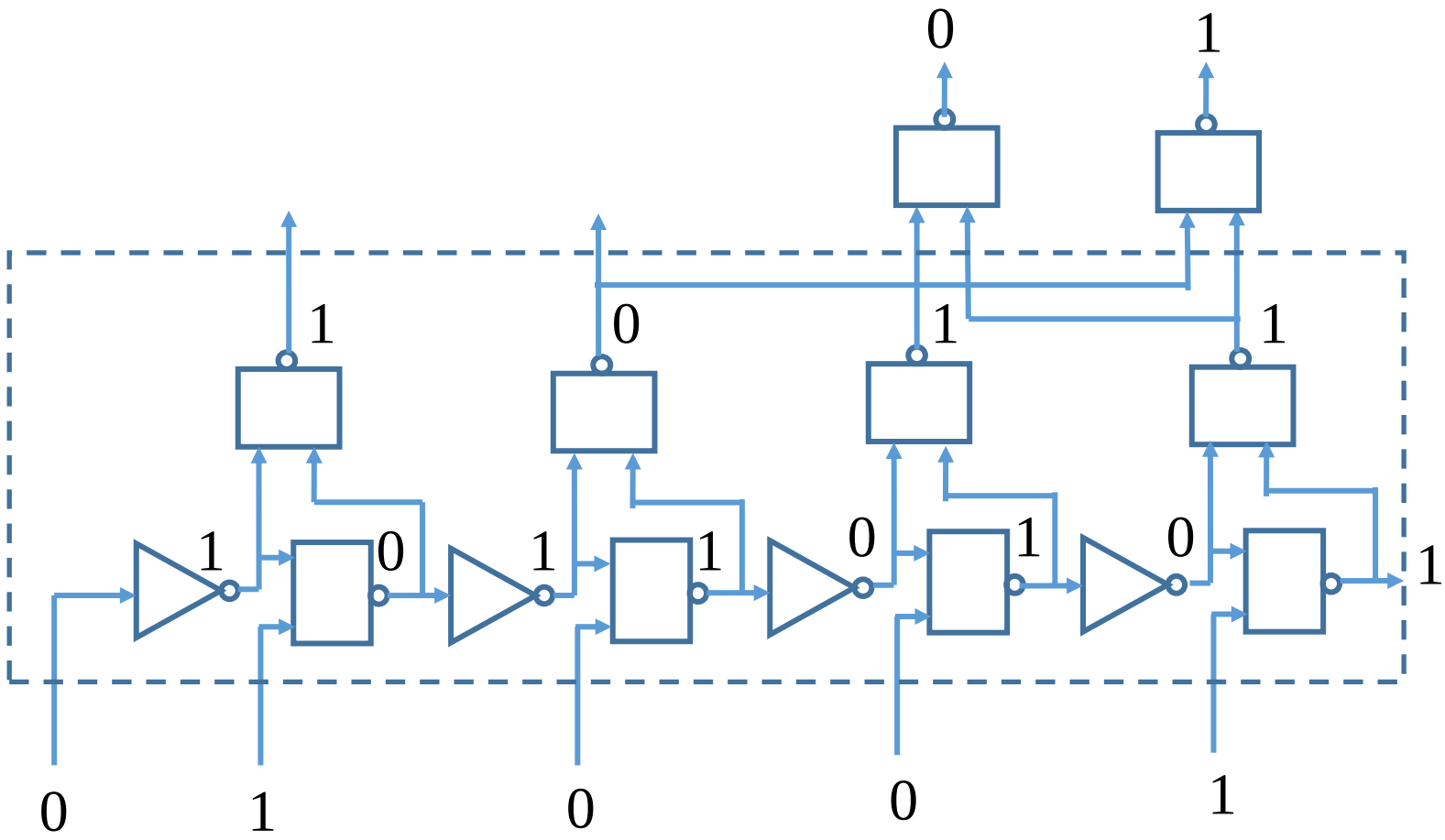
```
        INTA  R1          ;发出请求的最高级中断源送R1
        JMP  @VTAB(R1);转向中断源的中断服务程序入口
VTAB:  DEV1               ;最高级中断服务程序入口地址
        DEV2              ;第二级中断服务程序入口地址
          ……
        DENn              ;最低级中断服务程序入口地址
```

中断响应：低电平（0）
中断请求：低电平（0）



四个中断源的中断排队器和编码器


0
0/1
1
1
1
0
0
0
0
1
1
1
1
1
0
1
0
1
0
1
0
1
0
0
0
1
1


0
1
1
0
1
1
1
0
1
0
1
1
0
1
0
1
0
1
0
1

4.2.3 中断源的识别方法

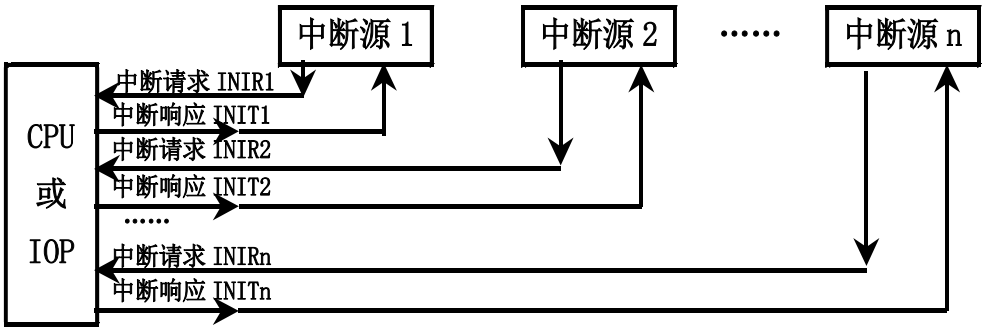
4. 中断向量法

- 将以上两种方法全部用硬件实现
  - 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

4.2.3 中断源的识别方法

5.独立请求法（Independent Request Method）

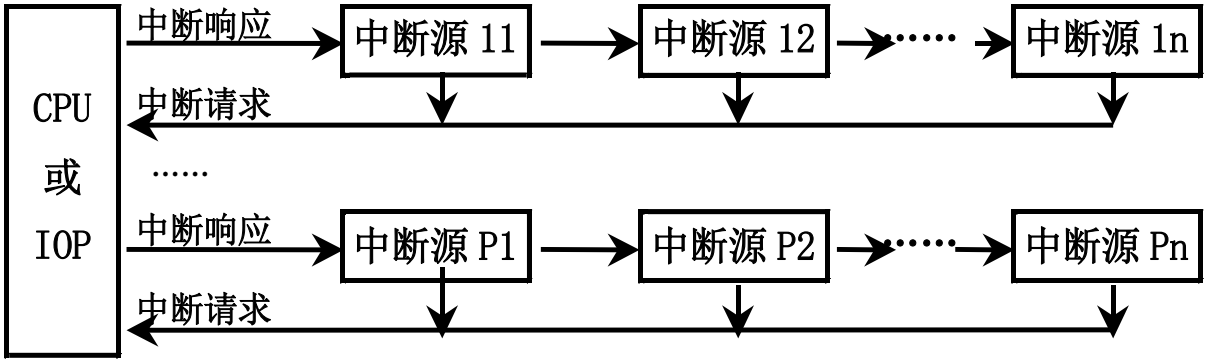
- 各个中断源使用自己独立的中断请求线。
- 如果同时有多个中断源请求中断服务，通过仲裁线路立即选择具有最高优先级的中断源，并向它发出中断响应信号INIT，处理机就可以立即转入这个中断源的中断服务程序。
- 独立请求法实际上是把分布在各个中断源内的串行排队器都集中到处理机中，从而克服了串行排队链法可靠性差的缺点，但灵活性差的缺点仍然存在。



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。

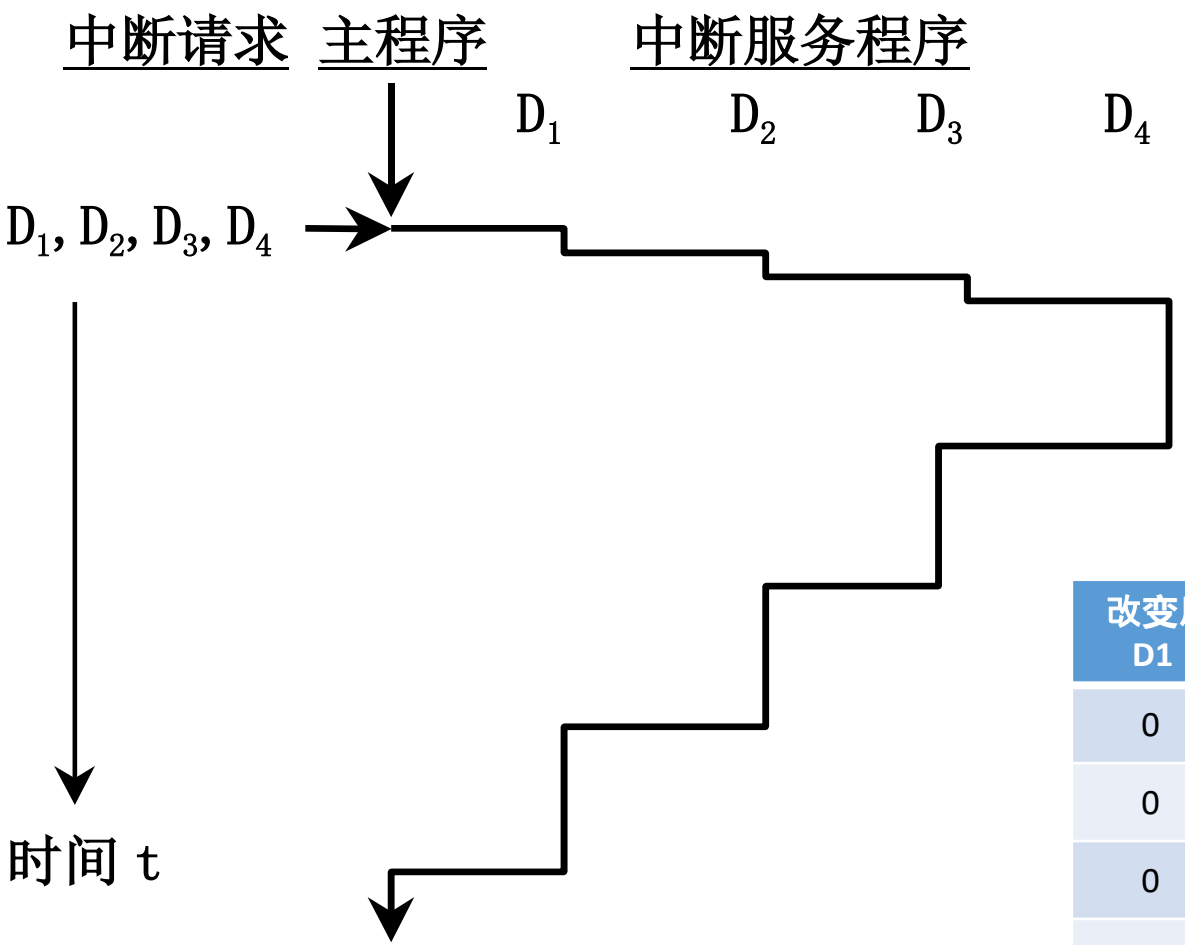
2） 改变处理机优先级

例：有四个中断源D_1、D_2、D_3和D_4，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。(1:屏蔽；0：开放)

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

解：

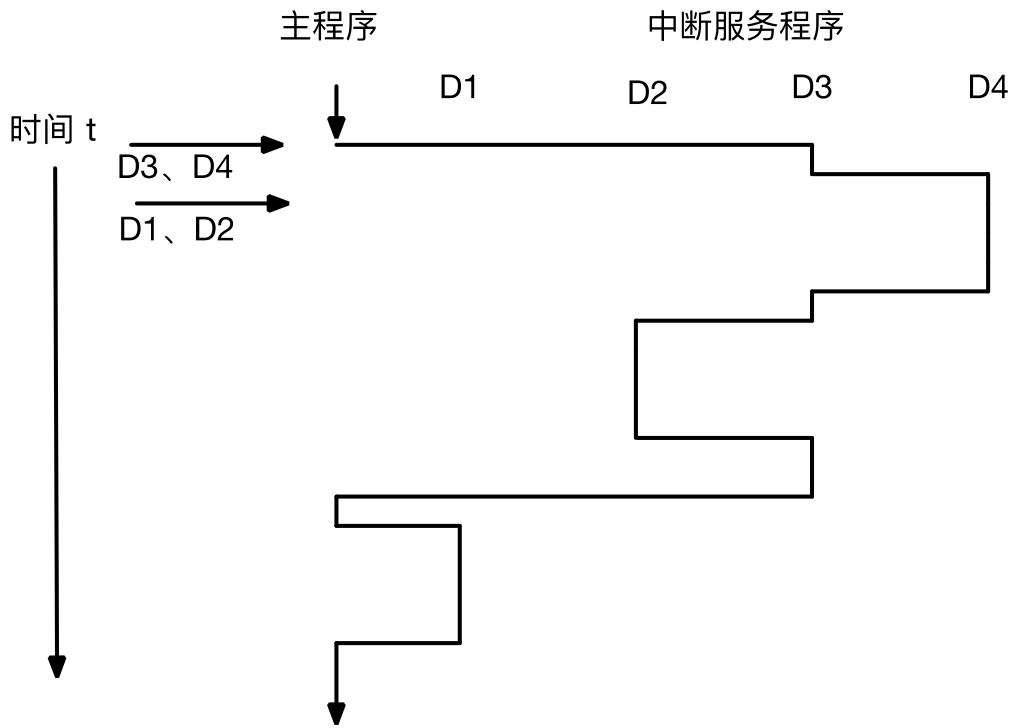
- 如果4个中断源都使用正常的中断屏蔽码，处理机的中断服务顺序将严格按照中断源的中断优先级进行。
- 如果改变中断屏蔽码，当D_1、D_2、D_3和D_4这4个中断源同时请求中断服务时，处理机实际为各个中断源服务的先后次序就会改变。
- 处理机响应的顺序是D_1、D_2、D_3、D_4
- **实际服务的顺序是D_4、D_3、D_2、D_1**



改变后中断屏蔽码			
D1	D2	D3	D4
0	1	1	1
0	0	1	1
0	0	0	1
0	0	0	0

- 若要将实际中断服务先后顺序改为 4-2-3-1，计算新的中断屏蔽码
- 假设从处理机响应中断源的中断服务请求开始，到运行中断服务程序第一次开中断所用的时间为1个时间单位，处理机运行中断服务程序的其他部分所用的时间为3个时间单位。当处理机在执行主程序时，中断源D3和D4同时发出中断服务请求，过两个时间单位后，D1和D2同时发出中断服务请求。采用改变后的中断屏蔽码，画出处理机响应个中断源的中断服务请求和实际运行中断服务程序过程的示意图。

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



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。

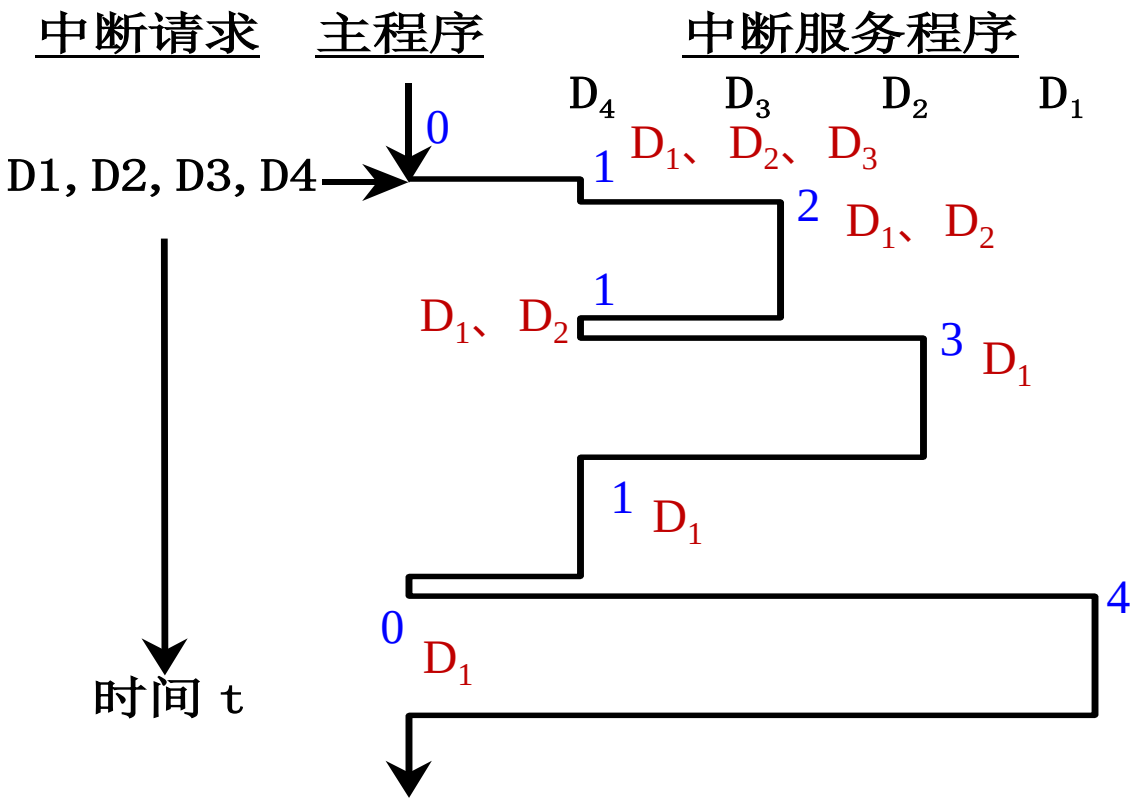
当4个中断源同时请求中断服务时，

	D_1	D_2	D_3	D_4
中断优先级	1	2	3	4
处理器优先级	4	3	2	1

解：处理机实际完成中断服务的过程

是 D_3、D_2、D_4、D_1

	D_1	D_2	D_3	D_4
中断优先级	1	2	3	4
处理器优先级	4	3	2	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

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

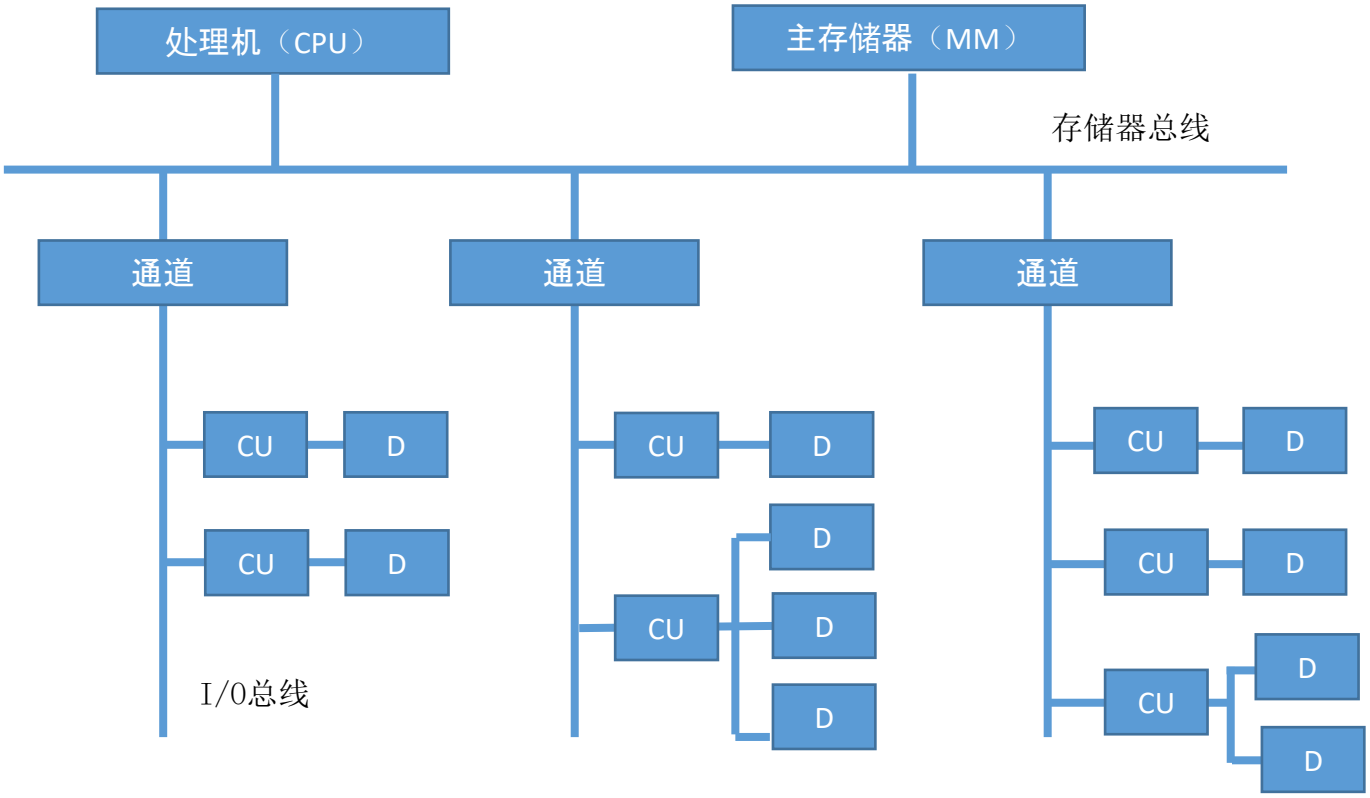
4.3.4 通道中的数据传送过程

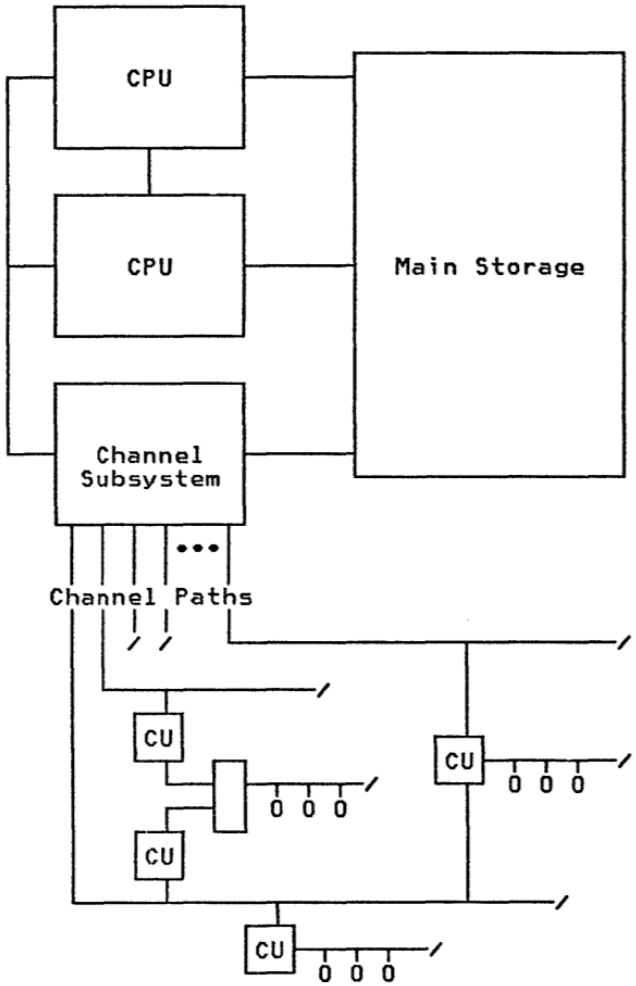
4.3.5 通道流量分析

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。


处理机（CPU）
主存储器（MM）
存储器总线
通道
通道
通道
CU
D
CU
D
CU
D
D
CU
D
D
CU
D
CU
D
D
CU
D
I/O总线



Logical Structure of a 370-XA System
with Two CPUs

4.3.1 通道的作用和功能

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。

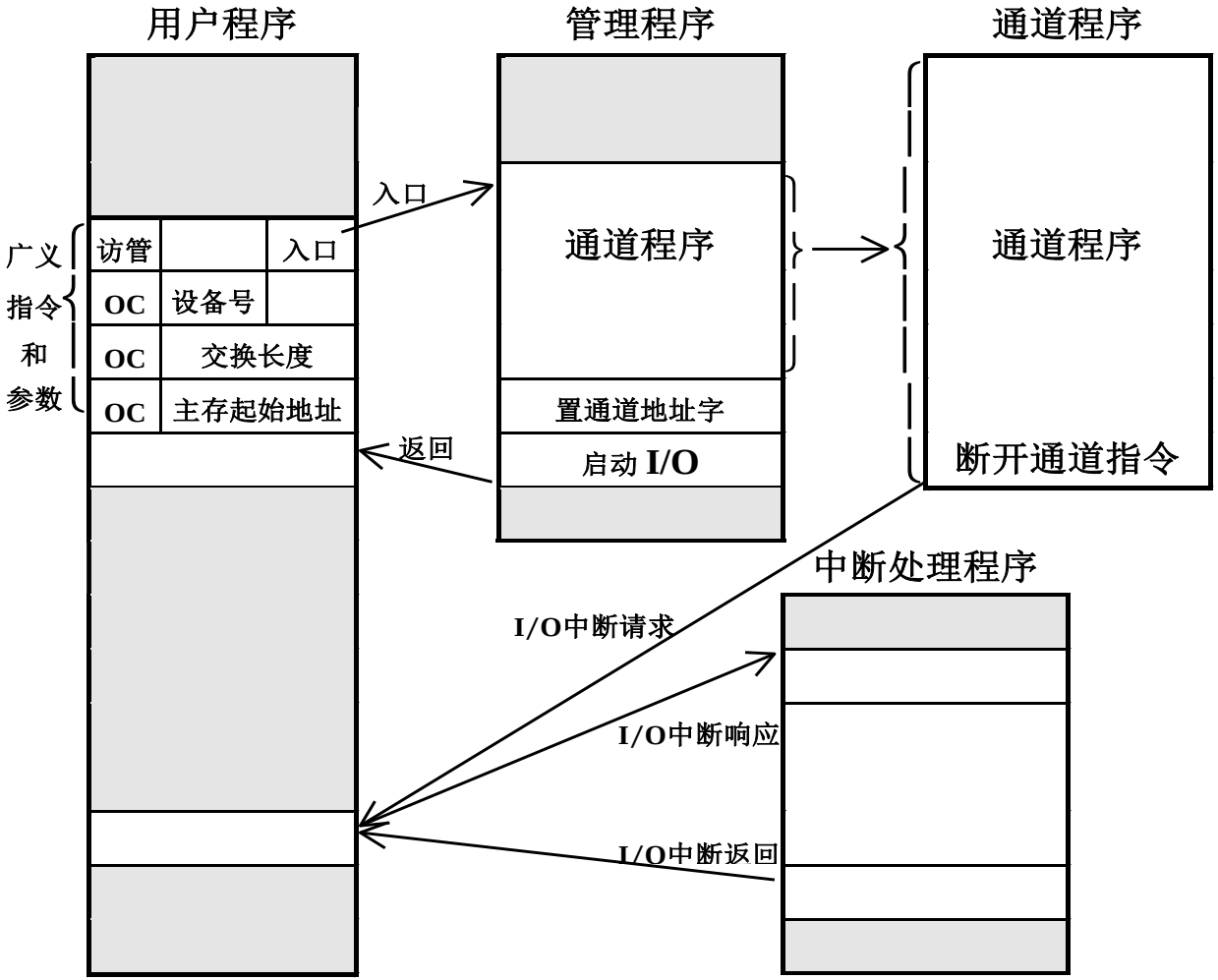
4.3.2 通道的工作过程

- 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。

```
SEEK              〈cylinder/head number〉
SEARCH KEY EQUAL 〈key value〉
TIC               *-8 Back to search if not equal
READ DATA         〈buffer〉
```

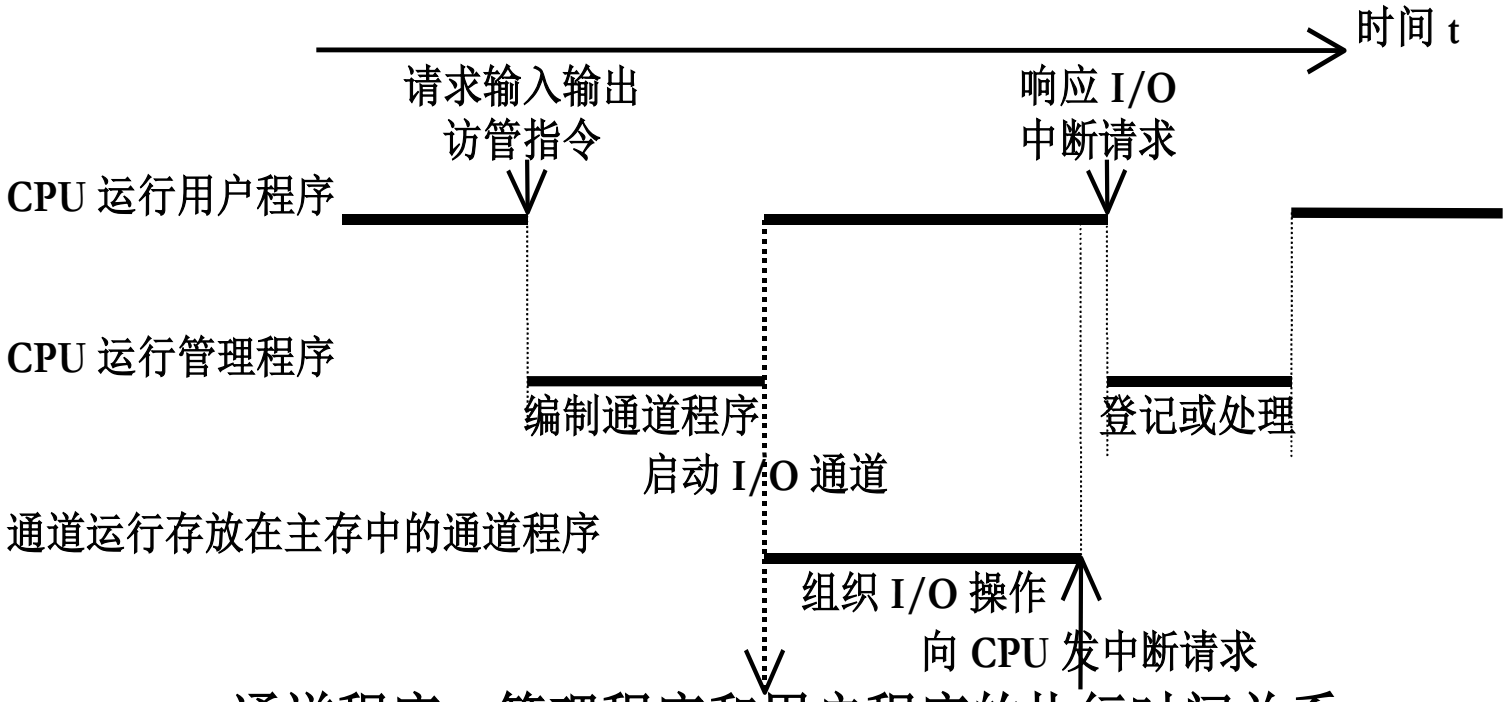
  - 通道程序结束后再次调用管理程序进行处理。
- 每完成一次输入输出工作，CPU只需要两次调用管理程序。

4.3.2 通道的工作过程



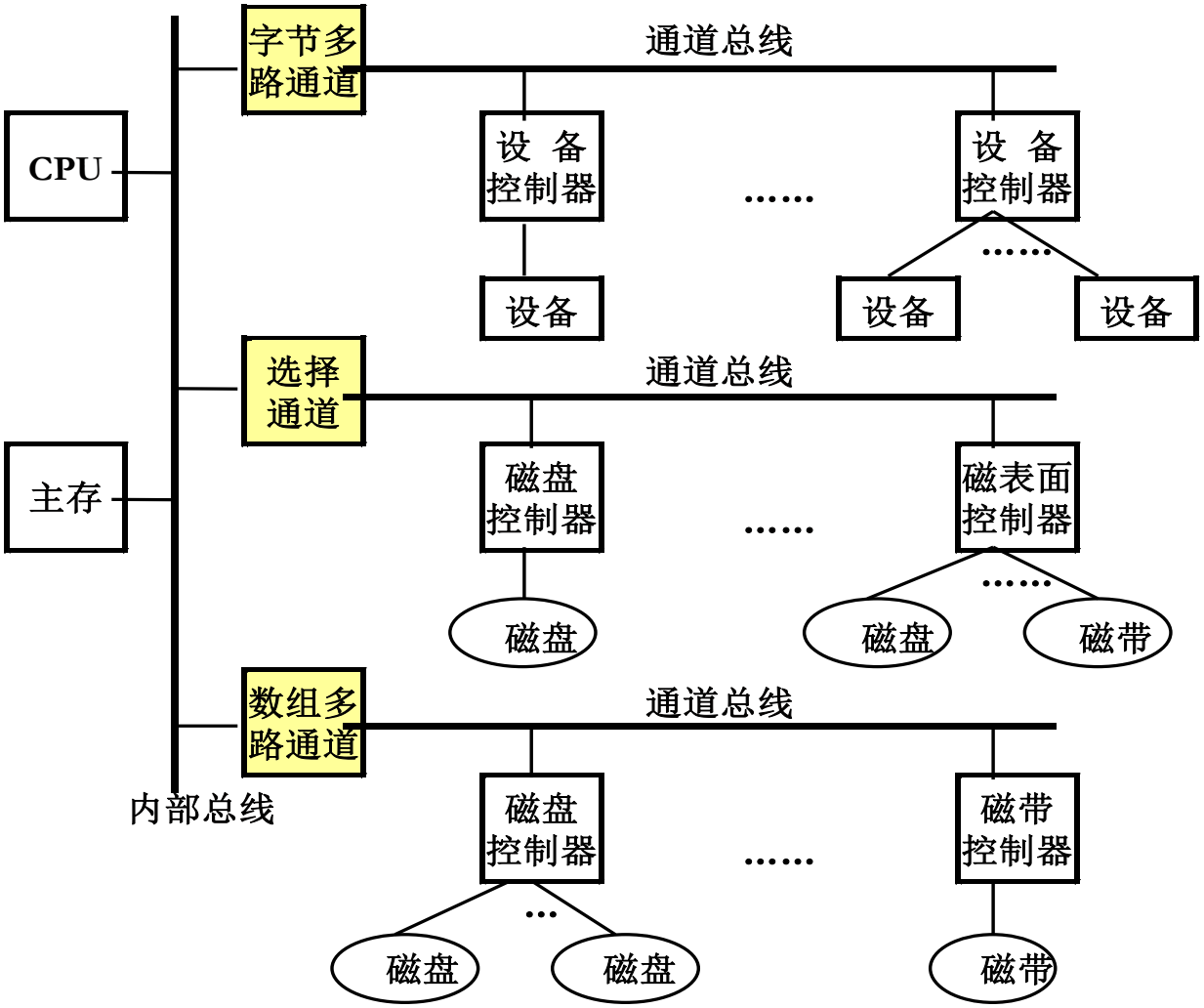
通道完成一次数据传输的主要过程

4.3.2 通道的工作过程



通道程序、管理程序和用户程序的执行时间关系

4.3.3 通道种类

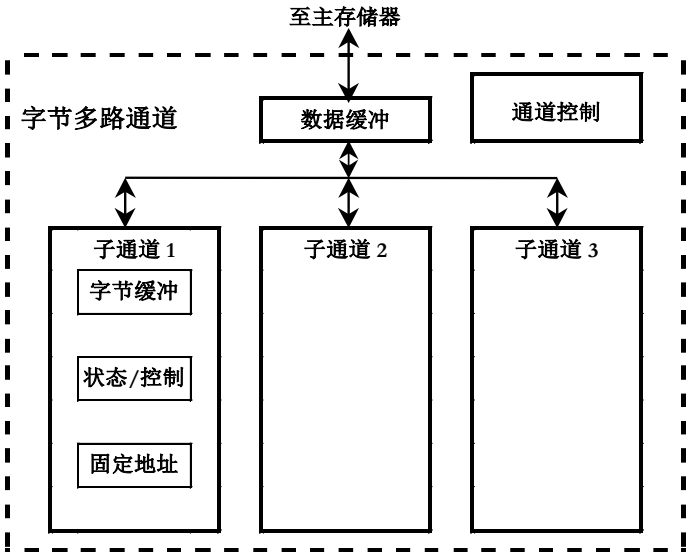


三种类型的通道与 CPU、设备控制器和外围设备的连接关系

4.3.3 通道种类

1. 字节多路通道（Byte Multiplexor Channel）

- 采用分时方式工作，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
- 子通道共享通道控制模块，拥有独立寄存器
- 可以在主存中开辟固定区域充当寄存器

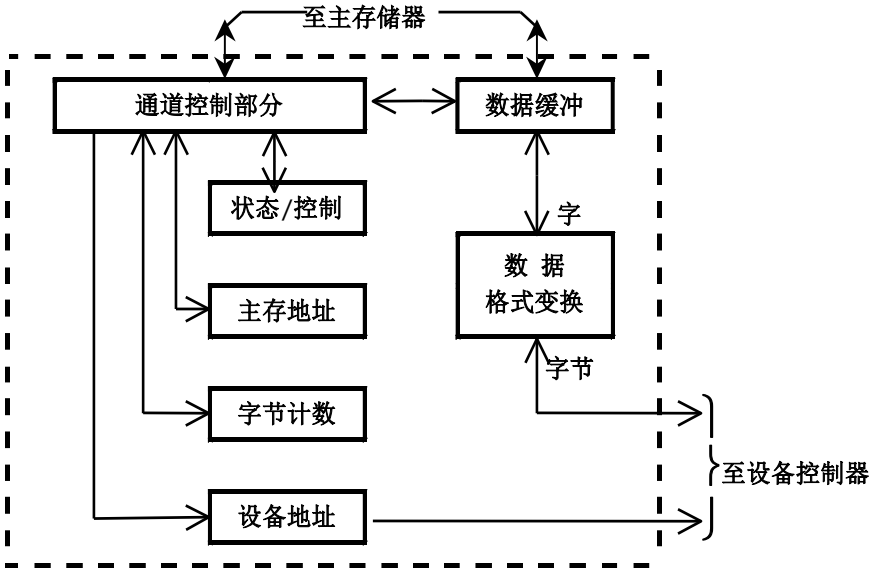


字节多路通道的结构

4.3.3 通道种类

2. 选择通道（Selector Channel）

- 为高速外围设备服务
- 一旦选中某一设备，通道就进入“忙”状态，直到该设备的数据传输工作全部完成为止
- 只有一个以成组方式工作的子通道



选择通道的结构

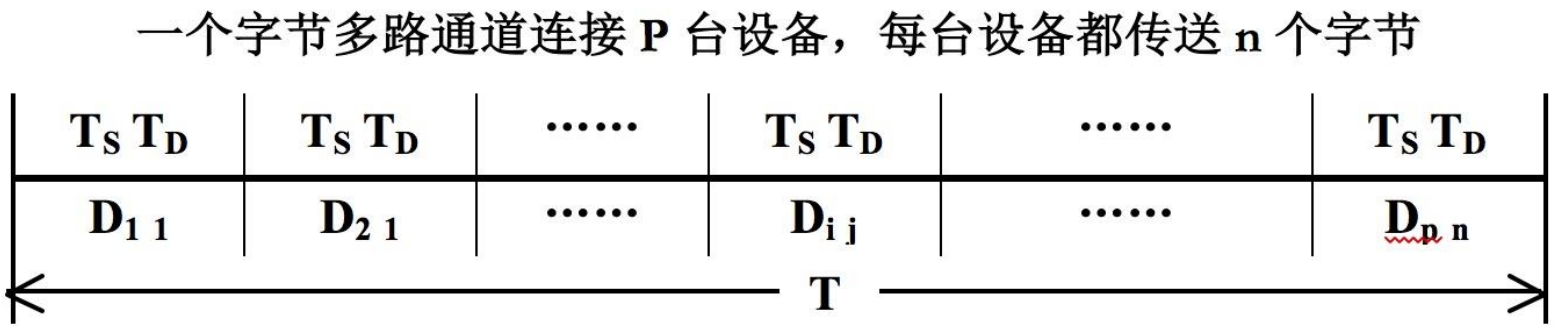
4.3.3 通道种类

3. 数组多路通道（Block Multiplexor Channel）

- 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
- 一次输入输出过程中要多次与同一台高速外围设备连接和断开，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



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

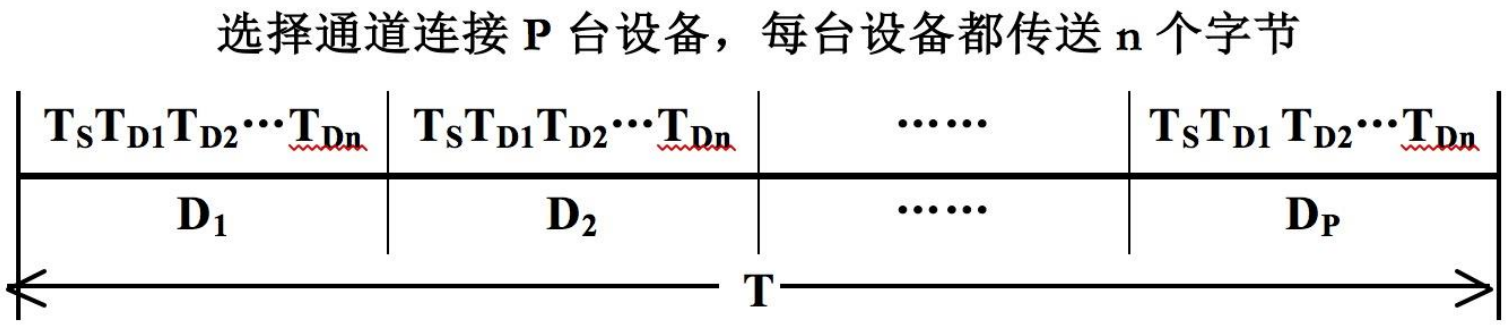
n：每一个设备传送的的字节个数

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



T_S：设备选择时间

T_{D_i}：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

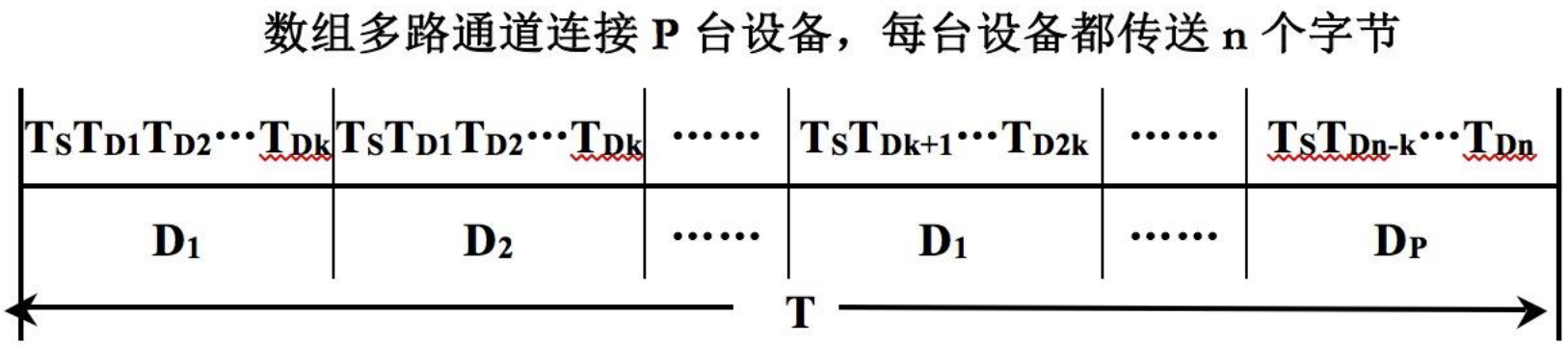
n：每一个设备传送的的字节个数

D_i：通道正在为第i台设备服务（$i = 1,2,\cdots,n$）

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：$T_{SELECT} = \left(\frac{T_S}{n} + T_D\right) \cdot P \cdot n$

数组多路通道的数据传送过程



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

P：在一个通道上连接的设备的台数

D_i：通道正在为第i台设备服务（$i = 1,2,\cdots,n$）

T：通道完成全部数据传送工作所需要的时间

k：一个数据块中的字节个数（$k < n$）

T_D：传送一个字节所用的时间

n：每一个设备传送的的字节个数

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

- 通道实际流量：单位时间内传送的数据量。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：

字节多路通道：

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

选择通道：

$$f_{SELECT} = \max_i\{f_i \,|i = 1, \ldots, p\}$$

数组多路通道：

$$f_{BLOCK} = \max_i\{f_i \,|i = 1, \ldots, p\}$$

- 三种通道的最大流量计算公式：

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

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

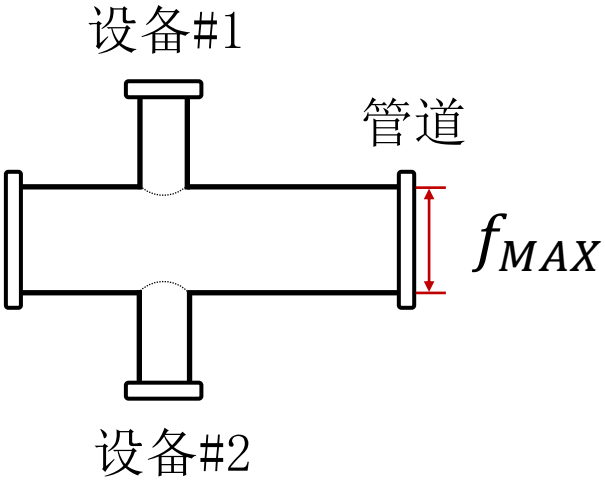
$$f_{MAX.BLOCK} = \frac{p \cdot n}{(T_S / k + T_D) \cdot p \cdot n} = \frac{1}{T_S / k + T_D} \text{字节/秒}$$

- 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

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

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

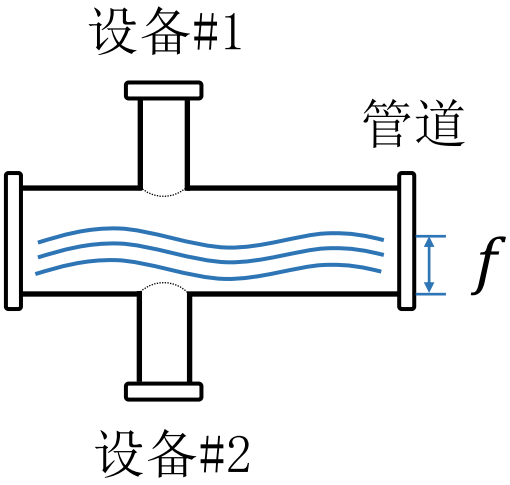
$$f_{BLOCK} \leq f_{MAX.BLOCK}$$



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

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

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



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

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

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

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：

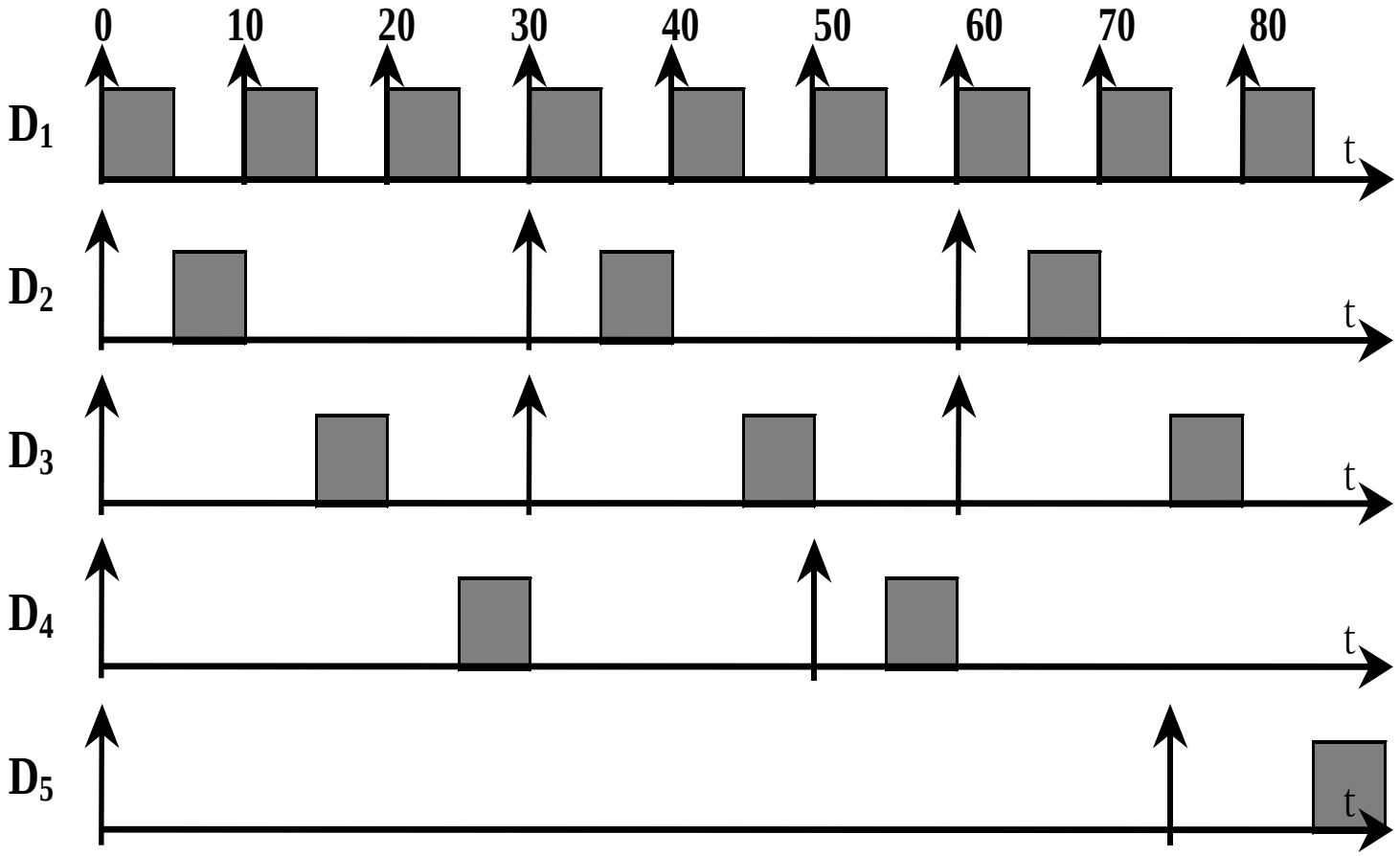
1MB/S =1Byte/uS

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

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；

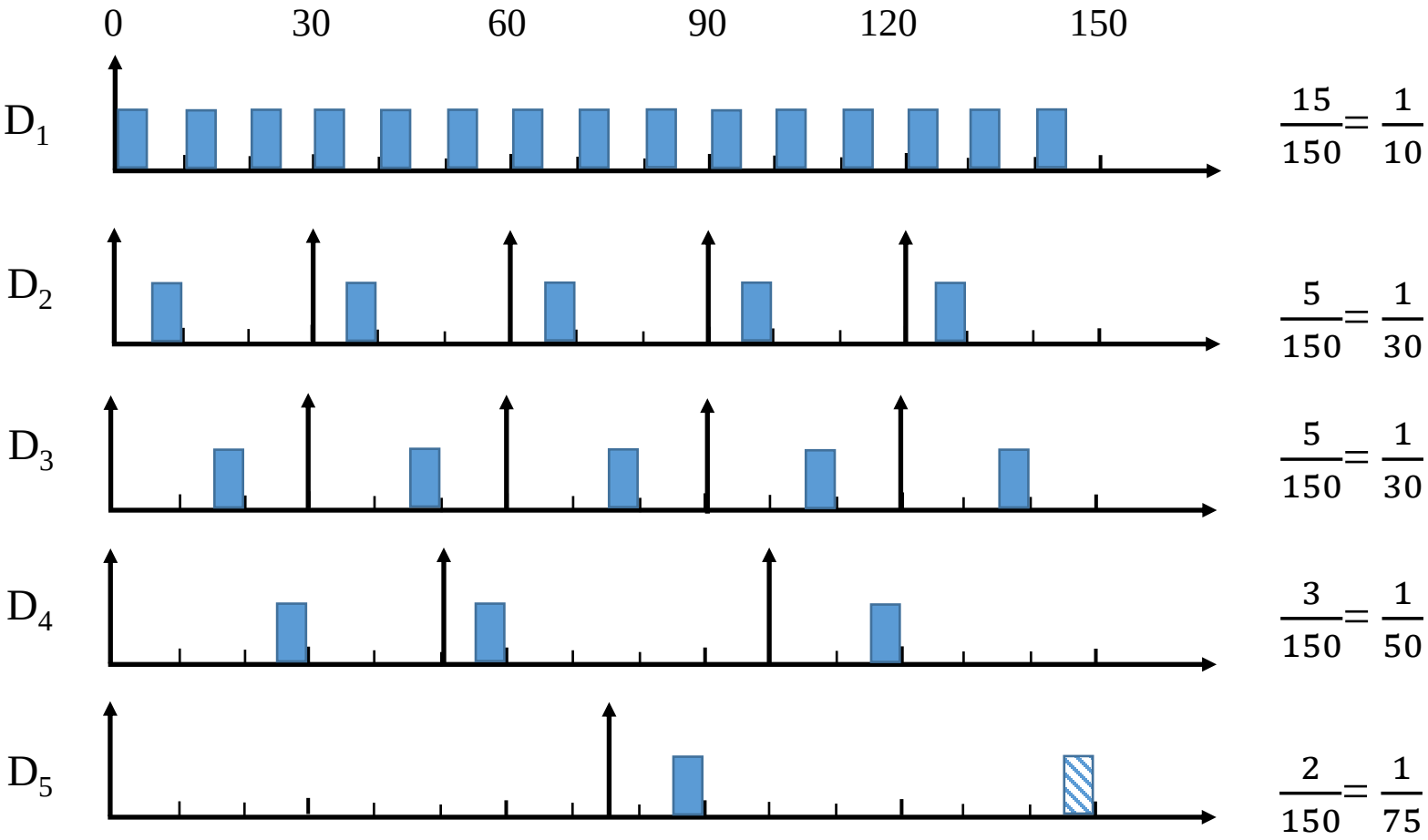
D_2：10us；

D_3：20us；

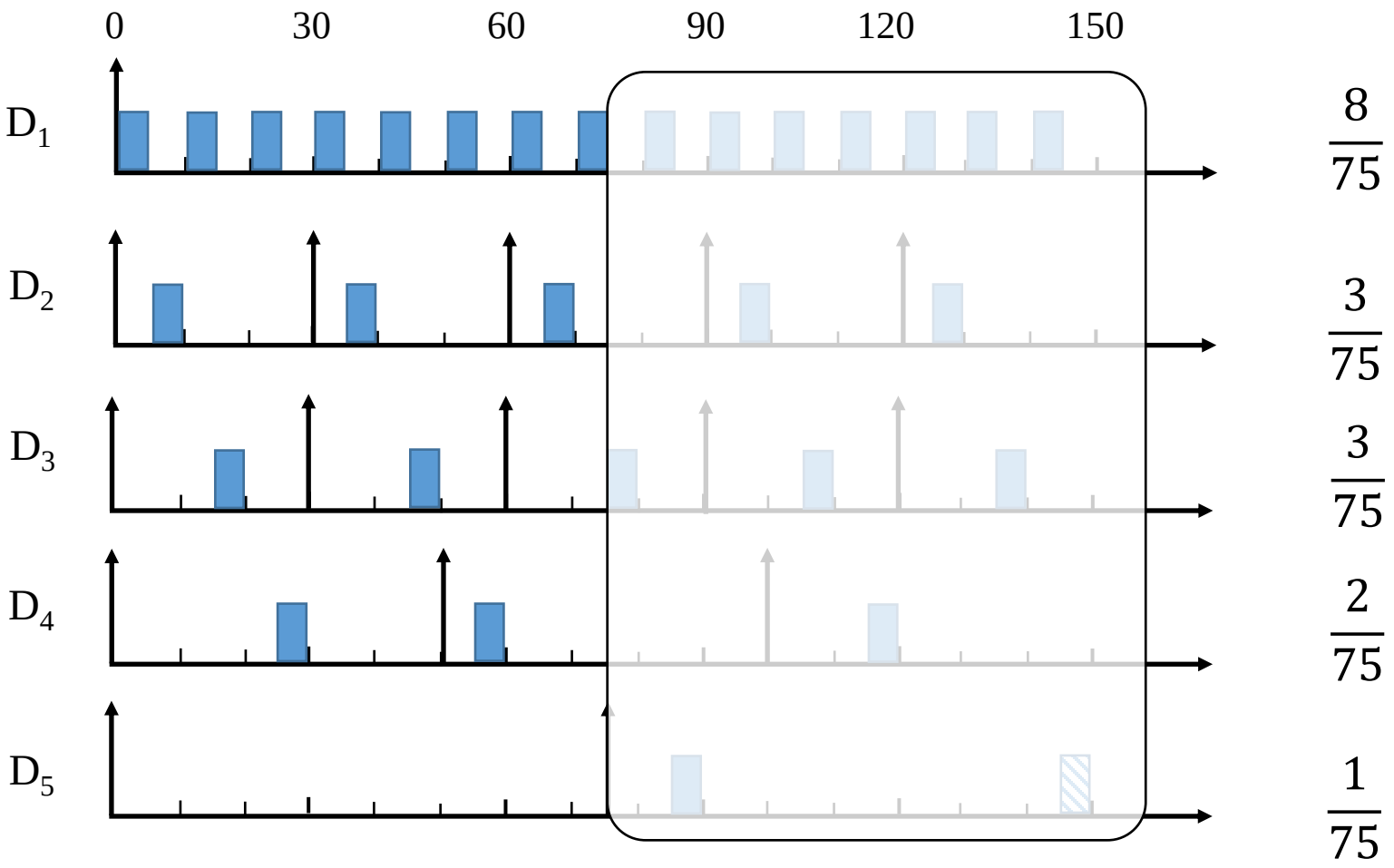
D_4：30us。

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



$$\frac{1}{10} + \frac{1}{30} + \frac{1}{30} + \frac{1}{50} + \frac{1}{75} = 0.2 \text{ MB/s}$$



$$\frac{8}{75}+\frac{3}{75}+\frac{3}{75}+\frac{2}{75}+\frac{1}{75}=\frac{17}{75}=0.2267\ \text{MB/s}$$

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4us）。
- **方法二：动态改变设备的优先级。**例如，在30*us*至70*us*之间临时提高设备D_5的优先级。
- **方法三：增加缓冲存储器。**例如，只要为设备D_5增加一个数据缓冲寄存器，它的第一次请求可以在第85us处得到响应，第二次请求可以在第145us处得到响应。

- 如果通道在数据传送期间，选择设备需要9.8us，传送一个字节数据需要0.2us。某低速设备每隔500us发送一个字节请求，那么至多可连接几台这种低速设备？
- 对于如下A~F这6种高速设备，一次通信传送的字节数不少于1024个字节，若采用选择通道，则哪些设备可挂，哪些设备不能挂？其中，A~F设备每发一个字节数据传送请求的时间间隔如下表所示

设备	A	B	C	D	E	F
发送请求间隔时间（us）	0.2	0.25	0.5	0.19	0.4	0.21

挂载低速设备的通道使用字节多路通道方式工作，该通道的最大流量为

$$f_{MAX.BYTE} = \frac{1}{T_S + T_D}$$

在各设备均被启动后，满负荷的最坏情况，要想在宏观上不丢失设备信息，通道的最大流量应大于或等于设备对通道的流量要求，即

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

假设设备数量为m，设备速率为f_i，则

$$\frac{1}{T_S + T_D} \geq mf_i$$

可以得到

$$m \leq \frac{1}{(T_S + T_D)f_i} = \frac{500us}{(9.8 + 0.2)us} = 50台$$

A~F属于高速设备，一次通信传送的字节数n，则通道的最大流量为

$$f_{MAX.BYTE} = \frac{n}{T_S + nT_D} = \frac{1}{\frac{T_s}{n} + T_D} = \frac{1}{\frac{9.8\mu s}{nB} + 0.2\mu s/B}$$

所以限制通道上所挂的设备速率

$$f_i \leq \frac{1}{\frac{9.8}{n} + 0.2} B/\mu s$$

$$\frac{1}{0.2096}$$

其中$n \geq 1024$。根据

设备	A	B	C	D	E	F
设备速率f_i（$B/\mu s$）	1/0.2	1/0.25	1/0.5	1/0.19	1/0.4	1/0.21

可知：只能挂载B、C、E、F四台设备

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

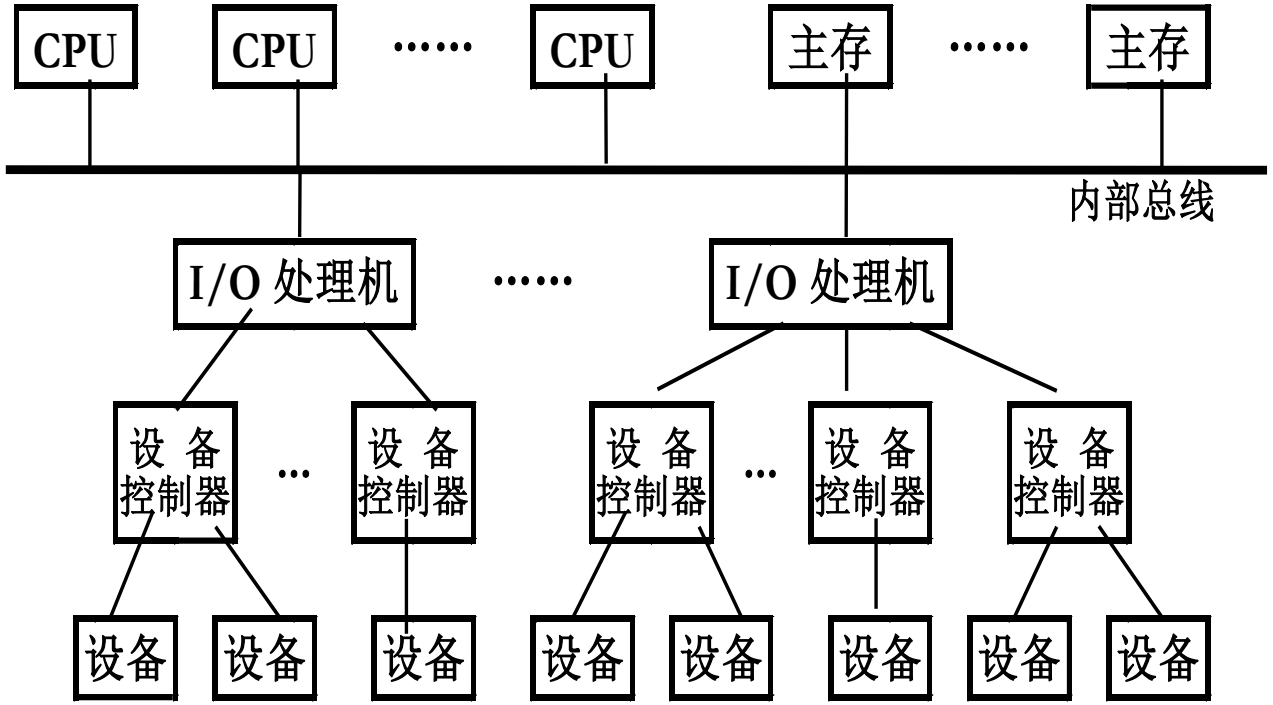
➢**通道处理机存在的问题：**

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。


CPU
CPU
……
CPU
主存
……
主存
内部总线
I/O 处理机
……
I/O 处理机
设 备
控制器
…
设 备
控制器
设 备
控制器
…
设 备
控制器
设 备
控制器
设备
设备
设备
设备
设备
设备
设备
设备

典型的输入输出处理机

4.4.1 输入输出处理机的作用

- **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。

4.4.1 输入输出处理机的作用

- 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，如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，如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。

4.4.2 输入输出处理机的种类

➢ **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。

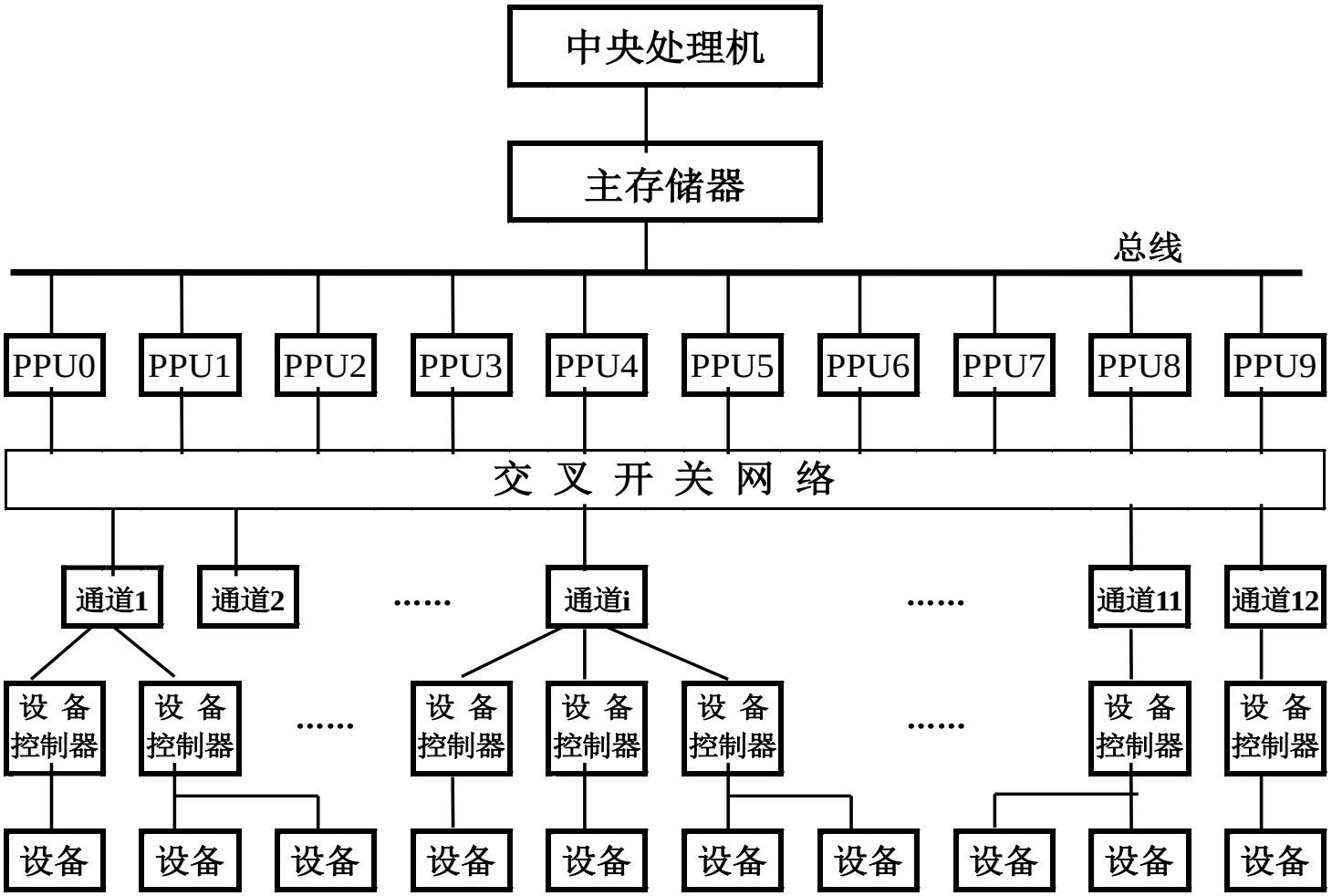
2） 以输入输出处理机作为主处理机。

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

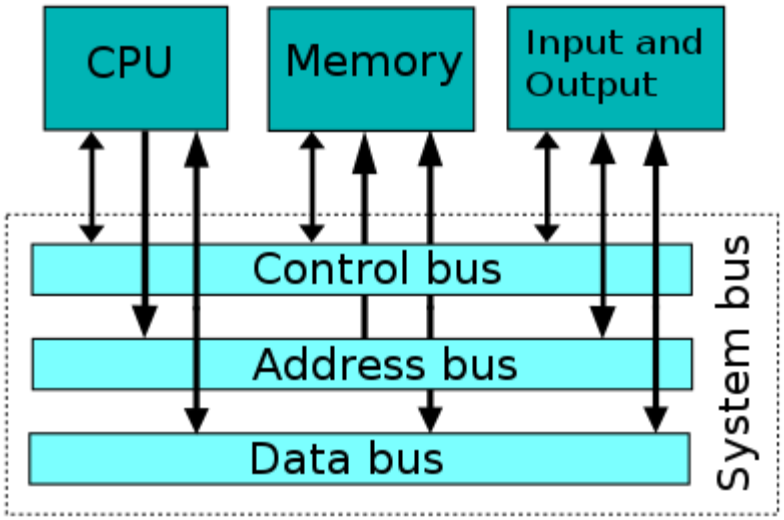
- CYBER1700大型机的输入输出处理机
- PPU0～PPU9通过总线分时共享主存储器，并通过交叉开关网络共享12个输入输出通道
- 每个PPU有一个容量4K×13位的局部存储器
- 每台PPU有66种指令，包括算逻指令、访存指令，输入输出指令及程序控制指令等。
- 当用户程序需要进行输入输出操作时，由中央处理机发出请求调用输入输出处理机，由输入输出处理机管理外围设备完成全部输入输出工作。



CYBER 1700 计算机系统的结构框图

4.5 总线系统

- 总线是用于互连计算机、CPU、存储器、I/O端口及外部设备、远程通信设备间信息传送通路的集合
- 总线与其相配合的附属控制电路统称为总线系统
- 按照信息传送功能、性能的不同，总线系统包括数据线、地址线、时序和中断信号等控制 / 状态线、电源线、地线以及备用线等
  - 数据线的根数决定同时传送的数据位数，即数据通路宽度
  - 地址线的根数决定直接寻址的范围
  - 控制 / 状态线决定总线的功能和使用能力
  - 备用线用于系统功能的扩充



4.5.1 总线的分类

➢按系统中的位置：

- **芯片级**：CPU芯片内的总线
- **板级**：连接插件版内的各组件，局部总线、内部总线
- **系统级**：系统间或主机与I/O接口或设备之间的总线

➢按信息传送方向：

- **单向传输**：只能沿一个方向传输
- **双向传输**：**半双向**可沿相反方向传送，但同时只能向一个方向传送；**全双向**允许同时向两个方向传送，速度快、造价高、结构复杂

➢按使用方法：

- 专用总线：
- 非专用总线

4.5.1 总线的分类

➢按使用方法：

- **专用总线**：只连接一对物理部件的总线
  - **优点**：多个部件可以同时收发信息，不争用总线，系统流量高；通信时不用指明源和目的，控制简单；任何总线的失效只会使连接该总线的两个部件不能直接通信，但它们仍可以通过其他部件间接通信，因而系统可靠。
  - **缺点**：总线数多，总线数与部件数成平方倍关系增加$\frac{n(n-1)}{2}$；难以小型化、集成电路化；总线较长时，成本相当高；利用率低；不利于系统模块化。
  - 只适用于实现某个设备（部件）仅与另一个设备（部件）连接。
- **非专用总线**

4.5.1 总线的分类

➢按使用方法：

- 专用总线
- 非专用总线：可以被多种功能或多个部件所分时共享，同一时间只有一对部件可使用总线进行通信
  - **单总线**：低性能微、小型机，既是主存总线又是I/O总线；**双总线或多总线**：大型系统；**远距离通信总线**：多个远程终端共享主机系统；**纵横交叉开关**：多处理机系统互联。
  - **优点**：总线数少、造价低；总线接口标准化，模块性强；可扩充能力强；部件的增加不会使电缆、接口和驱动电路激增；易用多重总线提高总线的带宽和可靠性，使故障弱化
  - **缺点**：系统流量小，经常会出现争用总线的情况，降低效率

4.5.2 总线的控制方式

- **集中式控制**：总线控制机构基本集中在一起，不论是在连接到总线的一个部件中，还是在单独的硬件中
- **分布式控制**：总线控制逻辑分散在连到总线的各个部件间

4.5.2 总线的控制方式

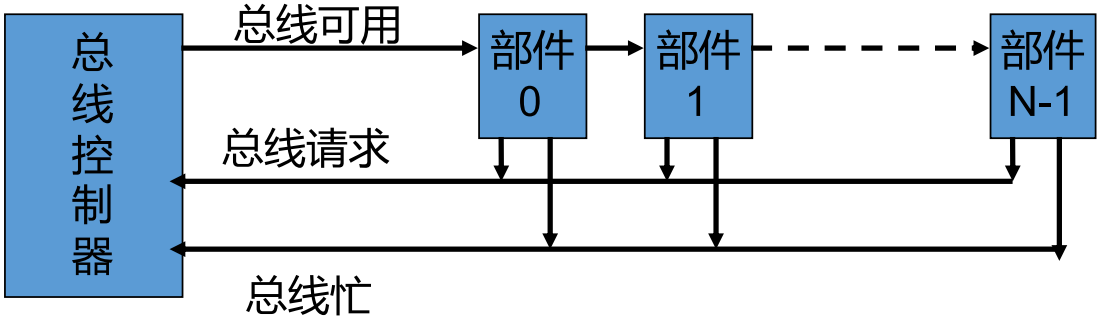
➢集中式的优先次序确定

- 串行连接
- 定时查询
- 独立请求
- 采用何种方式取决于控制线数目、总线分配速度、灵活性、可靠性等因素的综合权衡

4.5.2 总线的控制方式

➢集中式的串行链接

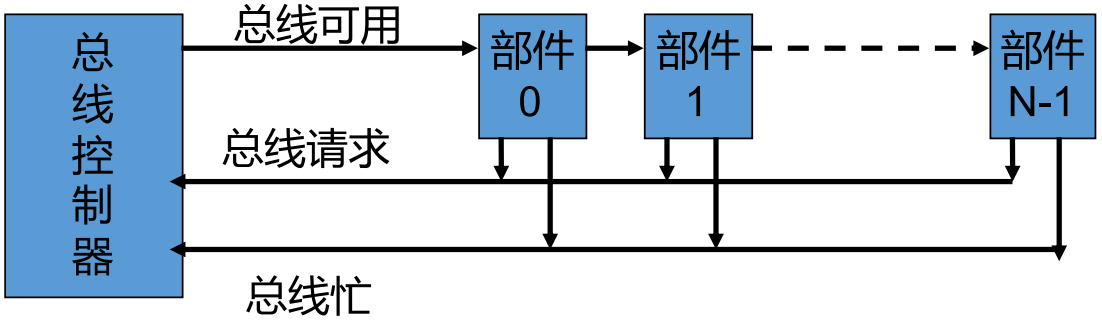
- 所有部件都经公共的“总线请求”线向总线控制器发出请求
- 当“总线忙”信号未建立，总线空闲，请求响应，送出“总线可用”信号，串行通过每个部件
- 如某个部件接收到“总线可用”信号但未发送请求，则送往下一个部件，如该部件发出过请求，则停止信号
- 建立“总线忙”，清除请求信号，准备数据传送
- 数据传送期间，“总线忙”维持“总线可用”的建立
- 传送完，去除“总线忙”和“总线可用”信号



4.5.2 总线的控制方式

➢集中式的串行链接

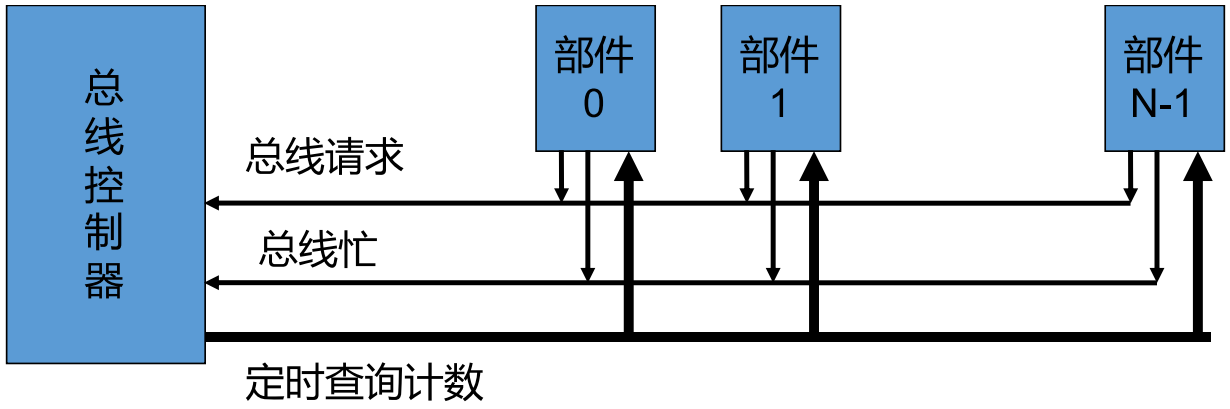
- 优先次序是由“总线可用”线所接部件的物理位置决定的
- 优点：选择算法简单，控制总线少（3根）；部件增减容易，可扩充性好；逻辑简单容易通过重复设置提高可靠性
- 缺点：对“总线可用”线及其有关电路的失效很敏感；灵活性差增加；限制总线的分配速度；受总线长度的限制，增减或移动设备受到限制



4.5.2 总线的控制方式

➢集中式定时查询

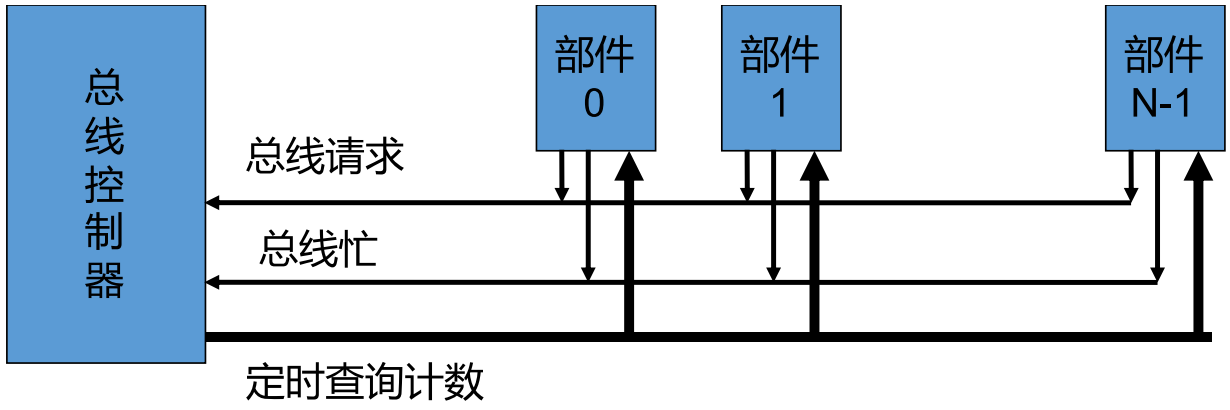
- 通过计数器来查询发出请求的部件
- 如果总线分配前计数器清0，同串行链接
- 如果总线分配前不清0，则是循环优先级
- 如果总线分配前计数器设置某个初值，则指定这个部件为最高优先级



4.5.2 总线的控制方式

➢集中式定时查询

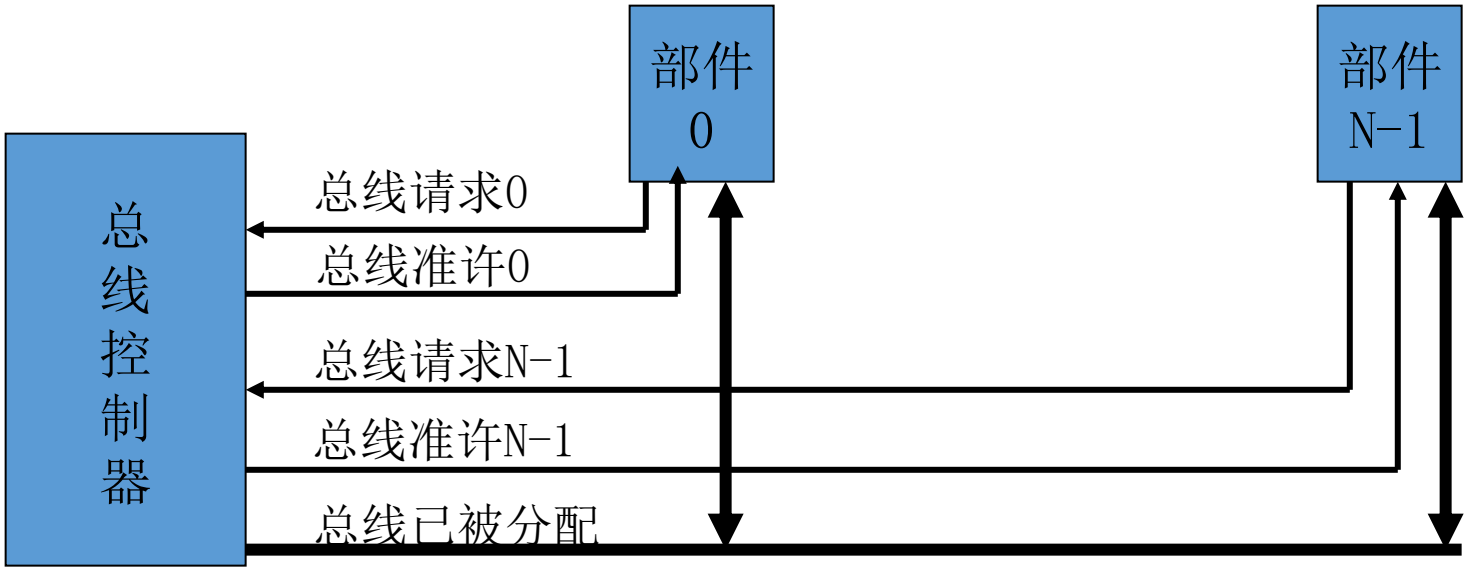
- 优点：；因计数器初值，部件号均可由程序设定，优先次序可由程序控制，灵活性强；可靠性高
- 缺点：控制线数较多，需$2+\log_2 N$，扩展性稍差；控制较为复杂；速度取决与计数信号的频率和部件数，不能很高



4.5.2 总线的控制方式

➢集中式独立请求

- 共享总线的每个部件各自都有一对“总线请求”和“总线准许”线
- 总线控制器根据某种算法对同时送来的多个请求进行仲裁

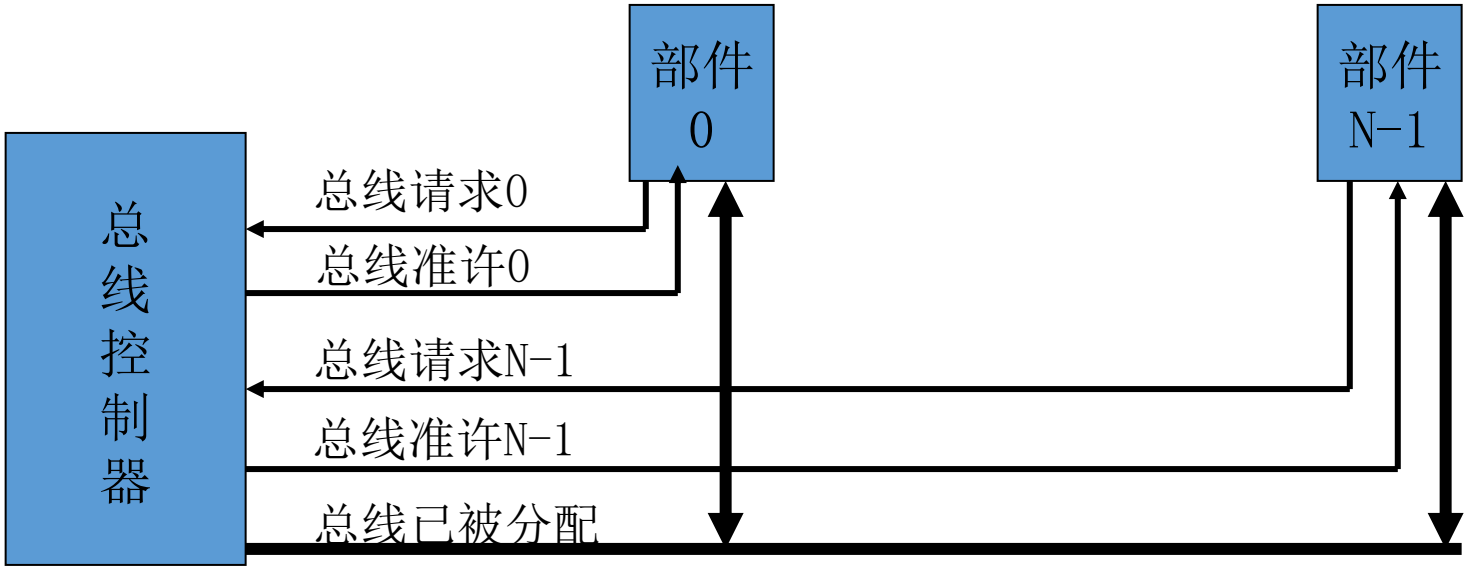


4.5.2 总线的控制方式

➢集中式独立请求

- 优点：总线分配速度快，不用查询；可以使用程序可控的预订方式、自适应方式、循环方式或它们的混和方式灵活确定下一个使用总线的部件；能方便地隔离失效部件的请求

- 缺点：控制线数量多，N个设备必须有2N+1根控制线；总线控制器复杂



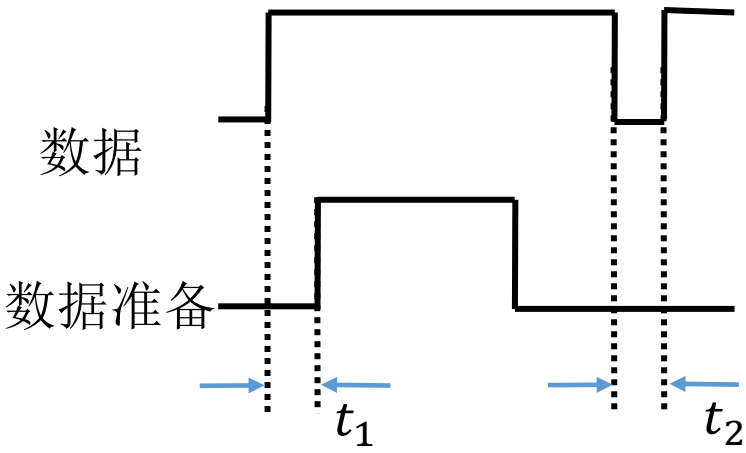
4.5.3 总线的通讯方式

- 同步通讯：两个部件的信息是通过定宽、定距的系统时标进行同步的。
  - 传送率高，受总线长度影响小
  - 出现同步误差，受干扰
- 异步通讯
  - 单向控制：通讯过程只由目的或源部件中的一个控制。
    - 源控制
    - 目的控制
  - 双向控制：由源和目的双方控制。

4.5.3 总线的通讯方式

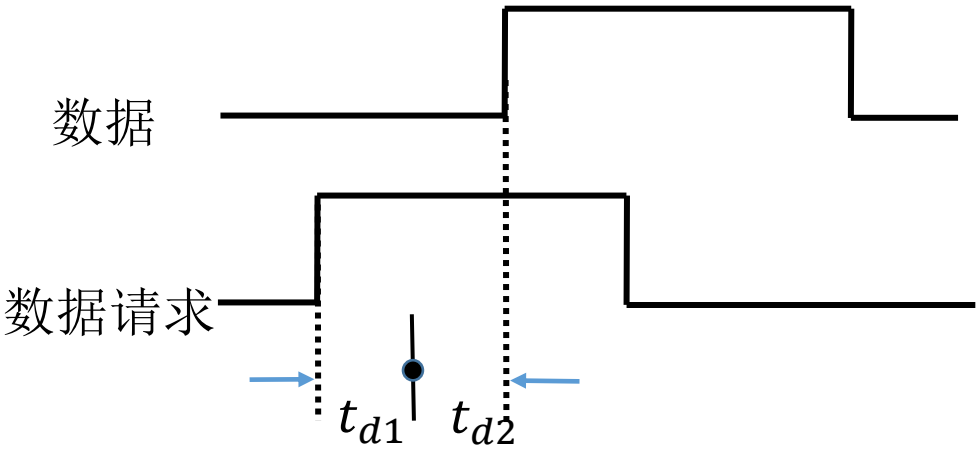
➢异步单向源控式通信

- 优点：简单、高速
- 缺点：没有来自目的部件的信息指明数据传送是否有效；不同速度之间的部件之间通信比较复杂；部件内需设置缓冲器以缓冲来不及处理的数据；效率低，高速部件难以发挥效能；要求“数据准备”信号干扰要小，否则易误认成有效信号



4.5.3 总线的通讯方式

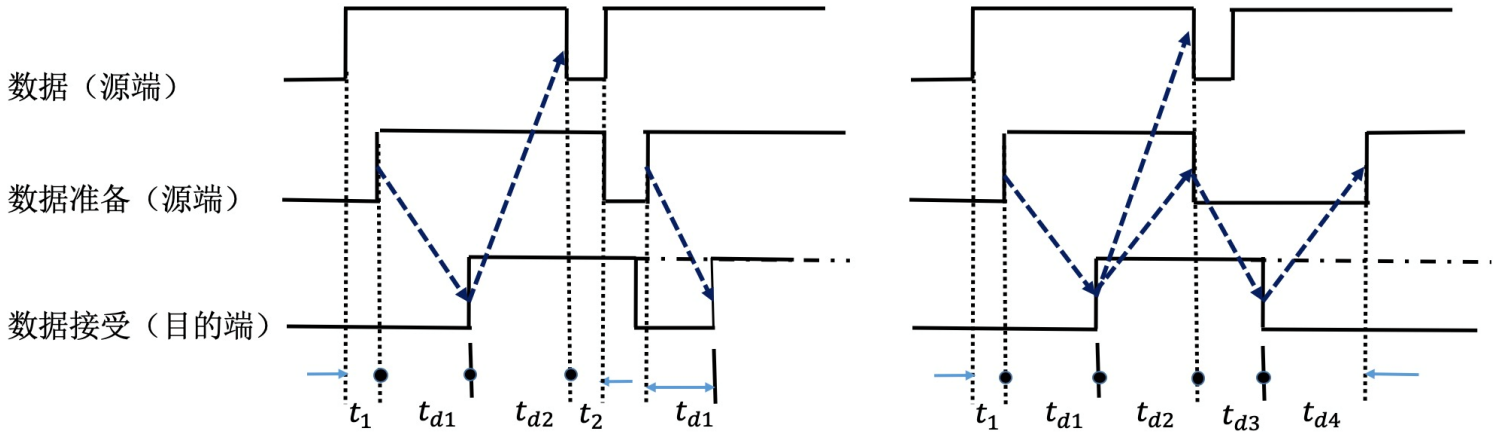
➢异步单向目控式通信：解决传送有效性检验的问题



4.5.3 总线的通讯方式

➢异步请求 / 回答式双向控制

- 单向控制的缺点是不能保证下一数据传送之前让所有数据线和控制线的电平信号恢复成初始状态，从而可能造成错误状态
- 为解决上述问题，可以采用异步请求 / 回答式双向控制

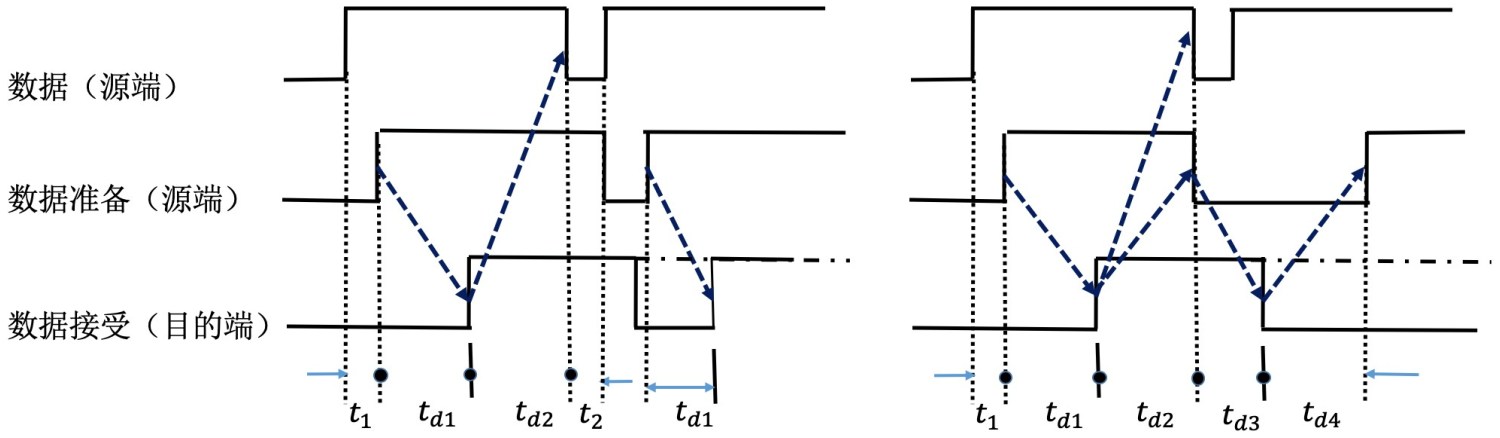


非互锁方式　　互锁方式

4.5.3 总线的通讯方式

➢异步请求 / 回答式双向控制

- 异步双向互锁方式虽然增加了信号沿总线来回传送的次数，使控制硬件变得复杂，但它能适应各种不同速度的I/O设备，保证数据传送的正确性，且具有较高的数据传送速率，因为它总是以所接源和目的部件中相对较低的速率来通信，比同步方式总是以所有部件中最低速率来通信的效率要高，所以I/O总线中最广泛使用的还是**异步双向互锁通信方式**。



非互锁方式　　互锁方式

4.5.4 数据宽度

- 数据宽度是I/O设备取得I/O总线后所传送数据的总量，而数据通路宽度是数据总线的物理宽度（即一个时钟周期所传送的数据量）
- 一个数据宽度可能需要多个数据周期才能完成
- 数据宽度的类型
  - 单字（单字节）宽度适合输入机、打印机等低速设备
  - 定长块宽度适合于磁盘等高速设备
  - 可变长块宽度适合于高优先级的中高速磁带、磁盘等设备

4.5.5 总线线数

- 总线线数越多，成本越高，干扰越大，可靠性越低，占用空间也越大，但传送速度和流量也越高；总线长度越长，成本越高，干扰越大，波形畸变越严重，可靠性越低。因此，总线增长，线数就应该减少。
- 在满足性能的前提下应尽量减少线数。总线的线数可通过用线的组合、编码及并/串-串/并转换来减少，但一般会降低总线的流量
- 总线的类型、控制方式、通信技术、数据宽度和总线的线数确定后，总线的申请、使用方式及相应的规范也就确定了。
- 总线的流量取决于该总线所接的外设的数量、种类以及传输信息的方式和速率要求。

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
6. 总线系统