

# 输入输出 系统



## 第3章 输入输出系统

### 3.1 输入输出系统概述

### 3.2 磁盘阵列

### 3.3 总线设计

### 3.4 通道处理机

### 3.5 中断系统



## 3.1 输入输出系统概述

输入输出系统包括输入输出设备、设备控制器及与输入输出操作有关的软硬件。

输入输出系统的主要功能是对指定的外设进行输入、输出操作，同时也完成许多其他的管理和控制。有的输入输出系统还能对要传送的信息进行格式变换，形成和产生有关输入输出操作是否完成或在执行过程中是否有错的状态控制信息，经中断系统传送给操作系统去分析和处理。



## 3.1 输入输出系统概述

### 输入输出系统的特点

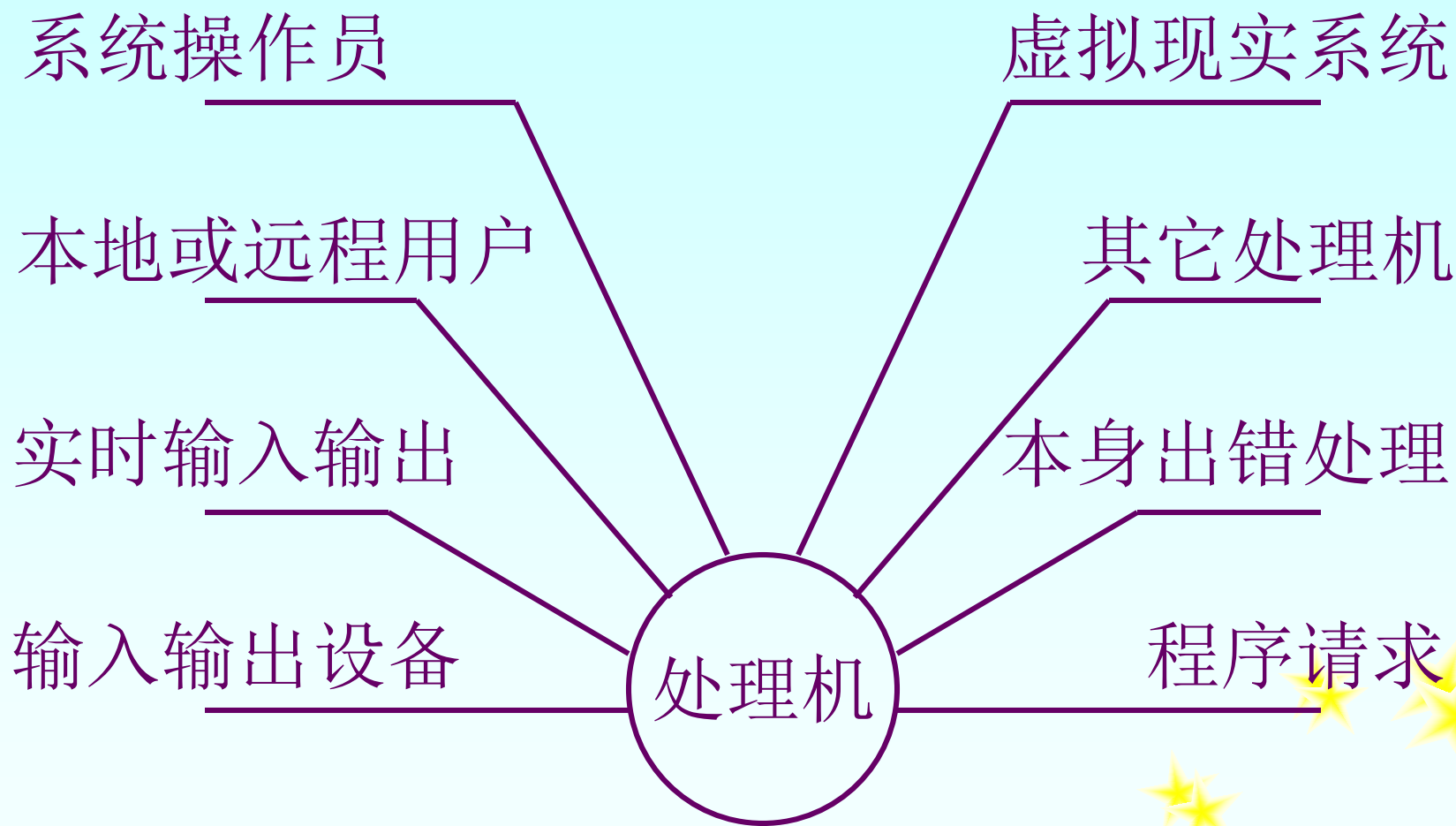
输入输出系统涉及到机、光、电、磁、声、自动控制等多种学科。

用户无需了解输入输出系统和输入输出设备的具体细节就能使用输入输出设备。

处理机的外部世界包括：本地和远程用户、系统操作员、操作控制台、输入输出设备、辅助存储器、其它处理机、各种通信设备和虚拟现实系统等。



## 3.1 输入输出系统概述



## 3.1 输入输出系统概述

### 1、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够并行工作。

### 2、实时性

对于一般外部设备：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。



## 3.1 输入输出系统概述

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。

### 3、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。



## 3.1 输入输出系统概述

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。





## 3.1 输入输出系统概述

### 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。

针对实时性，采用层次结构的方法。

针对与设备无关性，采用分类处理方法。

#### 1、自治控制

输入输出系统是一个独立于处理机之外的自治系统。

处理机与外围设备之间要有恰当的分工。



## 3.1 输入输出系统概述

### 2、层次结构

最靠近处理机的是输入输出处理机、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相连接。

设备控制器控制外围设备工作。



### 3、分类处理

为面向字符的设备（**character-oriented device**）；指工作速度比较低的机电类设备。例如，字符终端、打字机等。

面向数据块的设备主要指工作速度比较高的外围设备；例如，磁盘、磁带、光盘的辅助存储器，行式打印机等。



## 3.1 输入输出系统概述

输入输出系统的发展经历了 3 个阶段，对应于 3 种方式，即程序控制输入输出(包括全软的、程序查询状态驱动的、中断驱动的几种)、直接存储器访问(DMA)和I/O处理机方式。这 3 种方式可以分别用在不同的计算机系统上，也可以用在同一个计算机系统上作为相互补充。



## 3.2 磁盘阵列

### 3.2.1 RAID 简介

RAID是Redundent Array of Inexpensive Disks的缩写，直译为“廉价冗余磁盘阵列”，也简称为“磁盘阵列”。后来RAID中的字母I被改作为Independent，RAID就成了“独立冗余磁盘阵列”，但这只是名称的变化，实质性的内容并没有改变。可以把RAID理解成一种使用磁盘驱动器的方法，它将一组磁盘驱动器用某种逻辑方式联系起来，作为逻辑上的一个磁盘驱动器来使用。一般情况下，组成的逻辑磁盘驱动器的容量要小于各个磁盘驱动器容量的总和。



## 3.2 磁盘阵列

RAID的优点如下：


- ① 成本低，功耗小，传输速率高。很多磁盘驱动器同时传输数据，而这些磁盘驱动器在逻辑上又是一个磁盘驱动器，所以使用**RAID**可以达到单个磁盘驱动器几倍、几十倍甚至上百倍的速率。
- ② 提供容错功能。这是使用**RAID**的第二个原因，因为如果不考虑磁盘上的循环冗余校验（**CRC**）码的话，普通磁盘驱动器无法提供容错功能。**RAID**的容错是建立在每个磁盘驱动器的硬件容错功能之上的，所以它提供更高的安全性。
- ③ **RAID**比起传统的大直径磁盘驱动器来，在同样的容量下，价格要低许多。



## 4.1 存储体系的概念和并行主存系统

### 3.2.2 RAID的分级

**RAID**可以分为**7**个级别，即**RAID0~RAID6**。在**RAID1~RAID5**的几种方案中，不论何时**有**磁盘损坏，都可以随时拔出损坏的磁盘再插入好的磁盘（需要硬件上的热插拔支持），数据不会受损，失效盘的内容可以很快地重建，重建的工作由**RAID**硬件或**RAID**软件来完成。但**RAID0**不提供错误校验功能，所以有人说它不能算作是**RAID**，其实这也是**RAID0**为什么被称为**0级RAID**的原因——**0**本身就代表“没有”。



## 3.2 磁盘阵列

RAID级别	名称	数据 磁盘数	可正常工作的 最多失效磁盘 数	检测 磁盘数
RAID0	无冗余无校验的磁盘阵列	8	0	0
RAID1	镜象磁盘阵列	8	1	8
RAID2	纠错海明码磁盘阵列	8	1	4
RAID3	位交叉奇偶校验的磁盘阵列	8	1	1
RAID4	块交叉奇偶校验的磁盘阵列	8	1	1
RAID5	无独立校验盘的奇偶校验磁盘阵列	8	1	1
RAID6	双维无独立校验盘的奇偶校验磁盘阵列	8	2	2





## 3.2 磁盘阵列

- ❖ **RAID0**为无冗余无检验的磁盘阵列。所有的磁盘都可以并行工作，各自读出相应的部分。不提供数据冗余，只要有一个磁盘出现故障，整个系统将无法正常工作。
- ❖ **RAID1**称为镜像磁盘阵列。**RAID1**在每次写入数据时，都会将数据复制到你镜像盘上。某个磁盘出现故障，就由其镜像盘提供数据，系统仍能继续工作。
- ❖ **RAID2**为纠错海明码磁盘阵列。每个数据盘存放数据字的一位，还需要**3**个磁盘来存放海明检验位。如果出现了**1**位错误，则可以立即加以纠正。



## 3.2 磁盘阵列

- ❏ **RAID3**是**RAID2**的一个简化版本，称为位交叉奇偶检验磁盘阵列。校验盘专门用于存放数据盘中相应数据的奇偶校验位。
- ❏ **RAID4**是块交叉奇偶检验磁盘阵列。以块为单位进行交叉存储和计算奇偶检验。
- ❏ **RAID5**是无独立校验盘的奇偶校验磁盘阵列。每一行数据块的检验块被依次错开、循环地存放不同的盘中，以达到均匀分布的目的。
- ❏ **RAID6**是在**RAID 5**基础上为了进一步加强数据保护而设计的一种**RAID**方式，实际上是一种扩展**RAID 5**等级。增加了第二个独立的奇偶校验信息块。



## 3.2 磁盘阵列

**RAID**级别的选择有三个主要因素：可用性（数据冗余）、性能和成本。如果不要求可用性，选择**RAID0**以获得最佳性能。如果可用性和性能是重要的而成本不是一个主要因素，则根据硬盘数量选择**RAID1**。如果可用性、成本和性能都同样重要，则根据一般的数据传输和硬盘的数量选择**RAID3**、**RAID5**。



## 3.3 总线设计

在大多数小型和微型计算机系统中，计算机的各子系统之间通过总线（Bus）实现连接。

### 3.3.1 总线特点

总线是一组能为多个部件分时共享的公共信息传送线路。共享是指总线上可以挂接多个部件，各个部件之间相互交换的信息都可以通过这组公共线路传送；分时是指同一时刻总线上只能传送一个部件发送的信息。总线的优点是成本低、简单；缺点是总线的带宽形成了信息交换的瓶颈，从而限制了系统中总的I/O吞吐量。



## 3.3 总线设计

### 1. 总线事务

通常把在总线上一对设备之间的一次信息交换过程称为一个“总线事务”，把发出总线事务请求的部件称为主设备，与主设备进行信息交换的对象称为从设备。例如CPU要求读取存储器中某单元的数据，则CPU是主设备，而存储器是从设备。总线事务类型通常根据它的操作性质来定义，典型的总线事务类型有“存储器读”、“存储器写”、“I/O读”、“I/O写”、“中断响应”等，一次总线事务简单来说包括两个阶段：地址阶段和数据阶段。



## 3.3 总线设计

### 2. 总线使用权

- 总线是由多个部件和设备所共享的，为了正确地实现它们之间的通信，必须有一个总线控制机构，对总线的使用进行合理的分配和管理。
- 主设备发出总线请求并获得总线使用权后，就立即开始向从设备进行一次信息传送。称为主从关系。主设备负责控制和支配总线，向从设备发出命令来指定数据传送方式与数据传送地址信息。只有获得总线使用权的设备才是主设备。
- 通常，将完成一次总线操作的时间称为总线周期。总线使用权的转让发生在总线进行一次数据传送的结束时刻。



## 3.3 总线设计

### 总线的类型

就允许信息传送的方向来说，总线可以有单向传输和双向传输两种。双向传输又有半双向和全双向的不同。前者虽可以沿相反的方向传送，但同一时刻只能向其中的一个方向传送。后者允许同时在两个方向传送。全双向的速度快，但造价高，结构复杂。

总线按其用法可以分成专用的和非专用的。



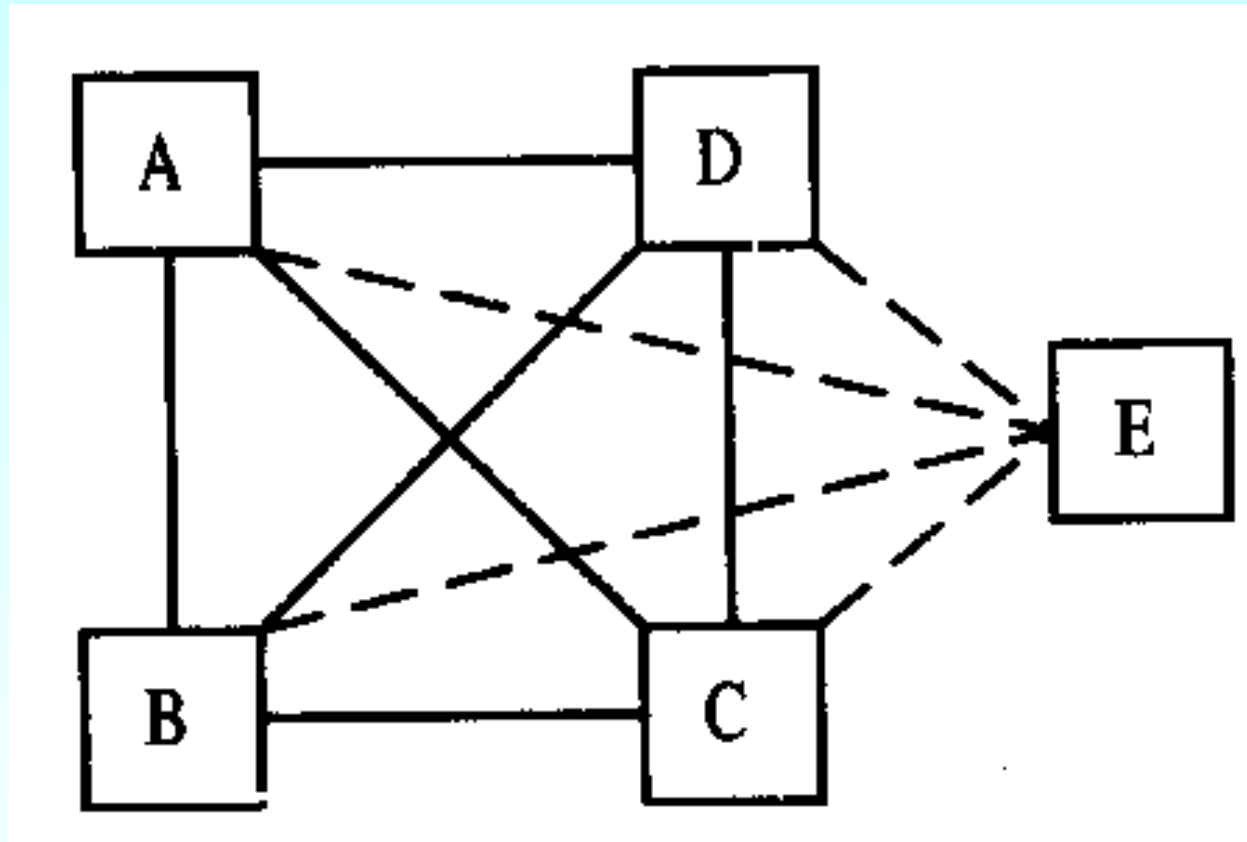


图 所有部件之间用专用总线互连





## 3.3 总线设计

### 3.3.2 总线的数据宽度与总线线数

#### 1. 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，它不同与前述的数据通路宽度。数据通路宽度是数据总线的物理宽度，也就是数据总线的线数。而两次分配总线期间所传送的数据宽度可能要经过多个时钟周期分次传送才能完成。数据宽度有单字（单字节）、定长块、变长块、单字加定长块和单字加变长块等。



### 3.3 总线设计

单字（单字节）宽度适合于低速设备。因为这些设备在每次传送一个字（字节）后的访问等待时间很长，在这段时间里让总线释放出来为别的设备服务，可大大提高总线利用率和系统效率。

定长块宽度适合于高速设备，可以充分利用总线带宽。定长块也不用指明传送信息的长度，简化了控制。但由于块的大小固定，当它要比实际传送的信息块小得多时，仍要多次分配总线。

变长块宽度适合于高优先级的中高速设备，灵活性好，可按设备的特点动态地改变传送块的大小，使之与部件的物理或逻辑信息块的大小一致。



### 3.3 总线设计

单字加定长块宽度适合于速度较低而优先级较高的设备。这样，定长块的大小就不必选择过大，信息块超过定长块的部分可用单字处理，从而减少总线带宽、部件的缓冲器空间，减少部件可用能力的浪费。不过，若传送的信息块小于定长块的大小，但字数又不少时，设备或总线的利用率会降低。

单字加变长块宽度是一种灵活有效但却是复杂、花钱的方法。当要求传送单字时比只能成块传送的方法节省了不少起始辅助操作；而当成块传送时，块的大小又能调整到与部件和应用的要求相适应，从而优化了总线的使用。



### 3.3 总线设计

#### 2. 总线的线数

总线需要有发送电路、接收电路、传输导线或电缆、转接插头和电源等，这部分比起逻辑线路的成本高得多，而且转接器往往占系统物理空间的相当部分，是降低系统可靠性的主要部分。总线的线数越多，成本越高、干扰越大、可靠性越低、占用的空间也越大，当然传送速度和流量也越高。此外，总线的长度越长，成本越高，干扰越大，波形畸变越严重，可靠性越低。为此，越是长的总线，其线数就应尽可能减少。数据总线的宽度有一位、一个字节或一个全字等等。



在满足性能要求以及所用通信类型和速率适配的情况下，应尽量减少总线的线数。通过采用线的组合、并/串—串/并转换和编码可以减少总线的线数，但这通常会降低总线的流量。



## 3.3 总线设计

### 3.3.3 总线的性能指标

#### (1) 总线宽度

总线宽度指的是总线的线数，它决定了总线所占的物理空间和成本。对总线宽度最直接的影响是地址线和数据线的数量。主存空间和I/O空间的扩充使地址线数量的增加，并行传输要求有足够的数据线。如64位数据线和64位地址线在高档微机中已较为普遍，在大型高性能计算机中数据线和地址线更多。

例1：使用ISA总线（20位地址线）允许寻址的主存空间有多大？使用PCI总线（32位地址线）允许寻址的主存空间又有多大？

解：ISA总线的主存空间= $2^{20}$ 个主存单元=1M个主存单元

PCI总线的主存空间= $2^{32}$ 个主存单元=4G个主存单元



## 3.2 总线设计

### (2) 总线带宽

总线带宽定义为总线的最大数据传输速率，即每秒传输的字节数。在同步通信中，总线的带宽与总线时钟密不可分，总线时钟频率的高低决定了总线带宽的大小。

总线带宽=总线宽度×总线频率

总线的实际带宽还会受到总线长度（总线延迟）、总线负载、总线收发器性能等多方面因素的影响。

例2：PCI总线的时钟频率为33MHz/66MHz，当该总线进行32/64位数据传送时，总线带宽各是多少？

解：假设一个总线时钟周期 $T$ 完成一个数据的传送，时钟频率为 $f$ ，数据位为 $n$ ，总线带宽用 $Dr$ 表示，则 $Dr = \frac{n}{8 \times T}$   
 $= \frac{n \times f}{8}$ 。

假设 $f=33\text{MHz}=33 \times 10^6/\text{s}$ ， $n=32$ 位，根据定义可得  
 $Dr=4 \times 33 \times 10^6/\text{s}=132\text{MB/s}$





## 3.2 总线设计

例3：假设某系统总线在一个总线周期中并行传输4字节信息，一个总线周期占用2个时钟周期，总线时钟频率为10MHz，求总线带宽。

解：因为一个总线周期占用2个时钟周期，完成一个32位数据的传送。总线时钟频率  $f=10\text{MHz}$ ，时钟周期  $T=1/f=0.1\mu\text{s}$ ，总线周期  $=2T=0.2\mu\text{s}$ 。一个总线周期中并行传输4字节信息，则总线带宽是  $4 \div 0.2=20\text{MB/s}$ 。





## 3.2 总线设计

### (3) 总线负载

总线负载是指连接在总线上的最大设备数量。大多数总线的负载能力是有限的。

### (4) 总线复用

总线分时复用是指在不同时段利用总线上同一个信号线传送不同信号，例如地址总线 and 数据总线共用一组信号线。采用这种方式的目的的是减少总线数量，提高总线的利用率。

### (5) 总线猝发传输

猝发式数据传输是一种总线传输方式，即在一个总线周期中可以传输存储地址连续的多个数据。



## 3.3 总线设计

### 3.3.4 总线定时控制

总线的定时控制方式一般分为同步方式和异步方式。

#### 1. 同步定时方式

系统采用一个统一的时钟信号来协调发送和接收双方的传送定时关系。时钟产生相等的时间间隔，每个间隔构成一个总线周期。在一个总线周期中，发送和接收双方可以进行一次数据传送。由于是在规定的时间段内进行I/O操作，所以，发送者不必等待接收者有什么响应，当这个时间段结束后，就自动进行下一个操作。

同步方式中的时钟频率必须能适应在总线上最长的延迟和最慢的接口的需要。



## 3.3 总线设计

### 2. 异步定时方式

异步定时方式也称为应答方式。在这种方式下，没有公用的时钟，也没有固定的时间间隔，完全依靠传送双方相互制约的“握手”信号来实现定时控制。

把交换信息的两个部件或设备分为主设备和从设备，主设备提出交换信息的“请求”信号，经接口传送到从设备；从设备接到主设备的申请后，通过接口向主设备发出“回答”信号，整个“握手”过程就是一问一答地进行的。

异步控制能保证两个工作速度相差很大的部件或设备间可靠地进行信息交换；但是控制较同步方式稍复杂一些，成本也会高一些。



### 3.3 总线设计

异步方式根据“请求”和“回答”信号的撤销是否互锁，有3种情况：

#### (1) 不互锁

“请求”和“回答”信号都有一定的时间宽度，“请求”信号的结束和“回答”信号的结束不互锁。

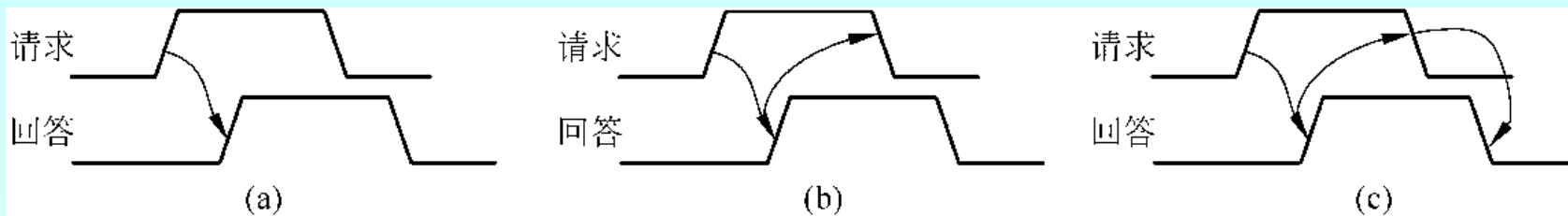
#### (2) 半互锁

“请求”信号的撤销取决于接收到“回答”信号，而“回答”的撤销由从设备自己决定。

#### (3) 全互锁

“请求”信号的撤销取决于“回答”信号的来到，而“请求”信号的撤销又导致“回答”信号的撤销。最高的灵活性和可靠性，付出了接口电路复杂性的代价





### 3.3 总线设计

为了提高可靠性，要求目的部件对数据是否已被接收以及是否正确均能给以回答。如果同步时间片的宽度宽到能为每个字的传送作出回答，则它必须按接到总线上的最低速的部件来考虑，这就会使同步通信的数据传送速率低于后面讲的异步通信。一种解决办法是在正常时，目的部件不作回答，源部件也不等待回答信号，但如果发生错误，则目的部件将在同步时间片过去之后，发回源部件一个出错信号，这样，就不会降低正常时总线的传送速率。但是这种办法中，源部件必须设置较大容量的缓冲器来保留已传送，但未经证实和回答过的所有数据，以备重发之用。



## 3.3 总线设计

### 总线的通信技术

#### 1. 同步通信

同步通信时，两个部件之间的信息传送是通过定宽、定距的系统时标进行同步的。这种方式的信息传送速率高，受总线的长度影响小，但会因时钟在总线上的时滞而造成同步误差，且时钟线上的干扰信号易引起误同步。



## 3.3 总线设计

### 2. 异步通信

由于I/O总线一般是为具有不同速度的许多I/O设备所共享，因此宜于采用异步通信。异步通信又可分为单向控制和双向(请求/回答)控制两种。

单向控制指的是通信过程只由目的或源部件中的一个控制。而双向控制是由源和目的双方共同控制。单向控制又有源控制和目的控制两种。





## 3.3 总线设计

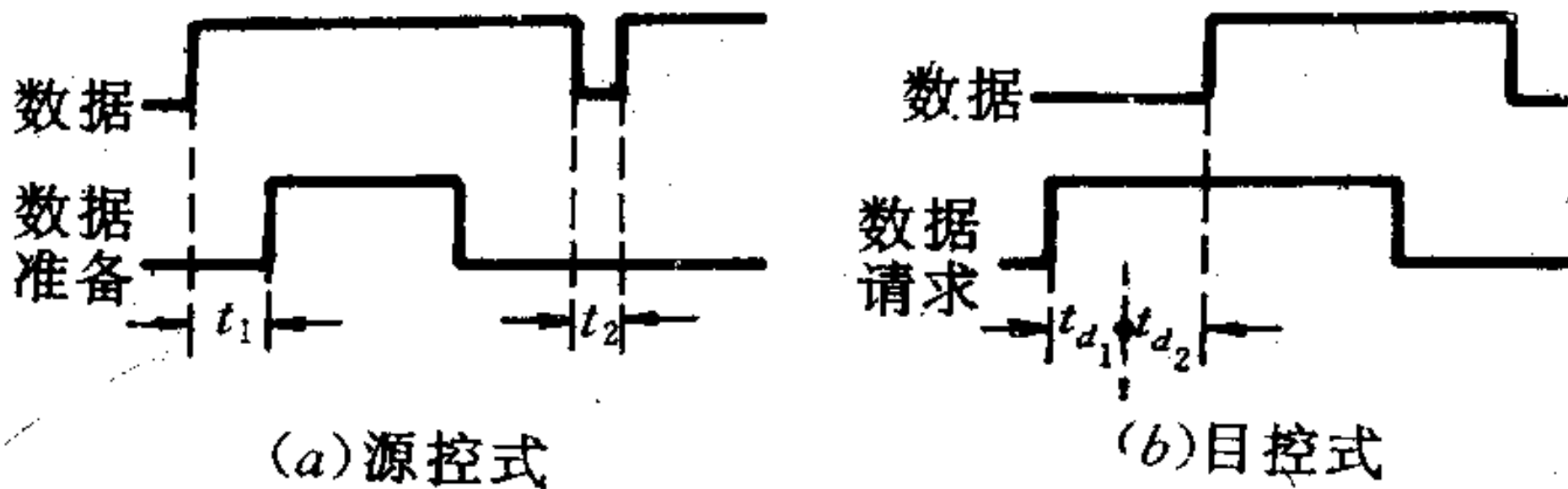


图 3.5 异步单向控制通信



## 3.3 总线设计

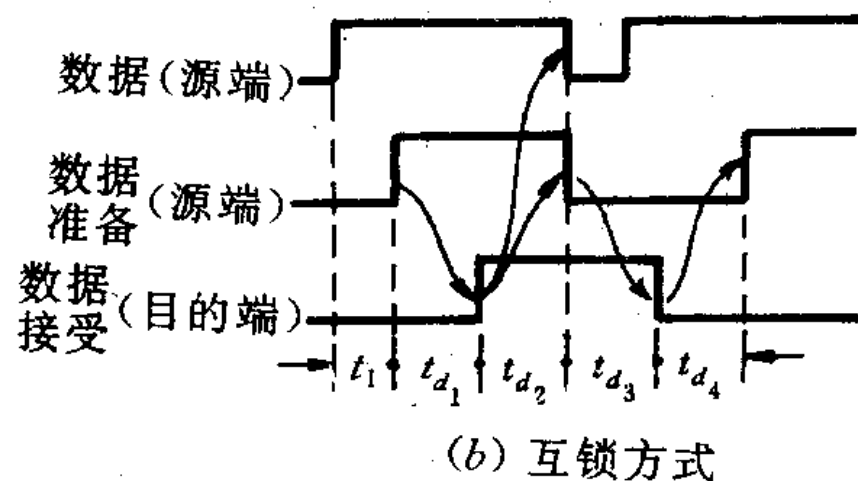
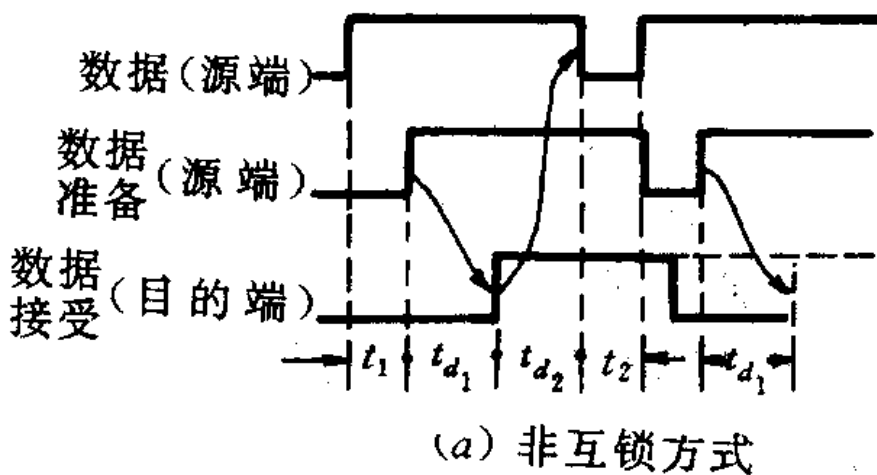


图 3.6 源控式异步双向控制通信



## 3.3 总线设计

### 3.3.5 总线的控制方式

当采用非专用总线时，由于可能发生多个设备或部件同时申请使用总线，就得有总线控制机构来按照某种优先次序裁决，保证在同一时间内只能有一个高优先级的申请者取得对总线的使用权。

如果总线控制逻辑基本上集中放在一起，不论是放在连接到总线的的一个部件中，还是放在单独的硬件中，都称为集中式控制。而当总线控制逻辑分散于连接到总线的各个部件中时，就称为分布式总线控制。这里先讲集中式总线控制。



## 3.3 总线设计

### 1. 总线的集中仲裁方式

为了保证同一时刻只有一个申请者使用总线，总线控制机构中设置有总线判优和仲裁控制逻辑，即按照一定的优先次序来决定哪个部件首先使用总线，只有获得总线使用权的部件，才能开始数据传送。总线控制逻辑集中在一处（如在CPU中）的，称为集中式控制，就集中式控制而言，有3种常见的优先权仲裁方式：



## 3.3 总线设计

### (1) 链式查询方式

链式查询方式的总线控制器使用三根控制线与所有部件和设备相连：

总线请求（**BR**）：该线有效，表示至少有一个部件或设备要求使用总线。

总线忙（**BS**）：该线有效，表示总线正在被某部件或设备使用。

总线批准（**BG**）：该线有效，表示总线控制器响应总线请求。



## 3.3 总线设计

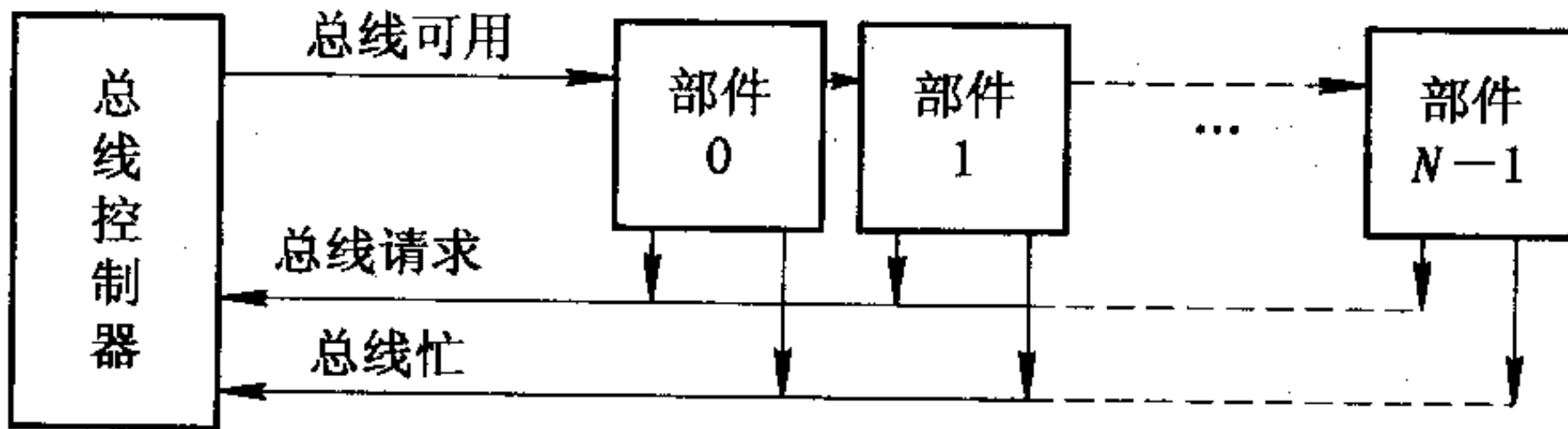


图 3.2 集中式串行链接



### 3.3 总线设计

链式查询的优点是只用**3**根线就能按一定的优先次序来实现总线控制，并很容易扩充。缺点是对查询链的故障很敏感，如果第 $i$ 个部件中的查询链电路有故障，那么第 $i$ 个以后的部件都不能工作。另外，因为查询的优先级是固定的，所以当优先级较高的部件出现频繁的总线请求时，优先级较低的部件就可能会难以得到响应。



## (2)计数器定时查询方式

计数定时查询方式的总线上的每个部件可以通过公共的**BR** 线发出请求，总线控制器收到请求之后，在**BS**为“**0**”的情况下，让计数器开始计数，定时地查询各个部件以确定是谁发出的请求。当查询线上的计数值与发出请求的部件号一致时，该部件就使**BS**线置“**1**”，获得了总线使用权，并中止计数查询，直至该部件完成数据传送之后，撤消**BS**信号。





## 3.3 总线设计

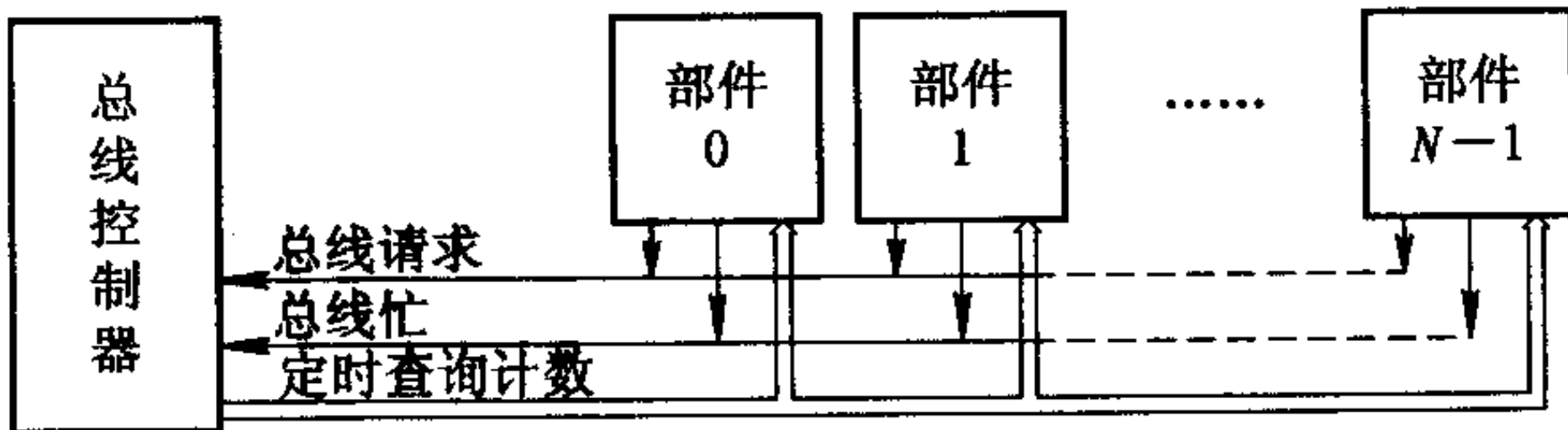


图 3.3 集中式定时查询



### 3.3 总线设计

这种计数可以从“0”开始，也可以从中止点开始。如果从“0”开始，各部件的优先次序和链式查询方式相同，优先级的次序是固定的。如果从中止点开始，即为循环优先级，各个部件使用总线的级别将相等。计数器的初始值还可以由程序来设置，这就可以方便地改变优先次序，增加系统的灵活性。定时查询方式的控制线数较多，对于 $n$ 个部件，共需 $2 + \lceil \log_2 n \rceil$ 根线。



### 3.3 总线设计

#### (3)独立请求方式

在独立请求方式中，每一个共享总线的部件均有一对控制线：总线请求**BRI**和总线批准**BGI**。当某个部件请求使用总线时，便发出**BRI**，总线控制器中有一排队电路，根据一定的优先次序决定首先响应哪个部件的请求**BRI**，然后给该部件送回批准信号**BGI**。

独立请求方式的优点是响应时间快，然而这是以增加控制线数和硬件电路为代价的。对于 $n$ 个部件，控制线的数目将达 $2n+1$ 根。此方式对优先次序的控制也是相当灵活的，它可以预先固定，也可以通过程序来改变优先次序。



## 3.3 总线设计

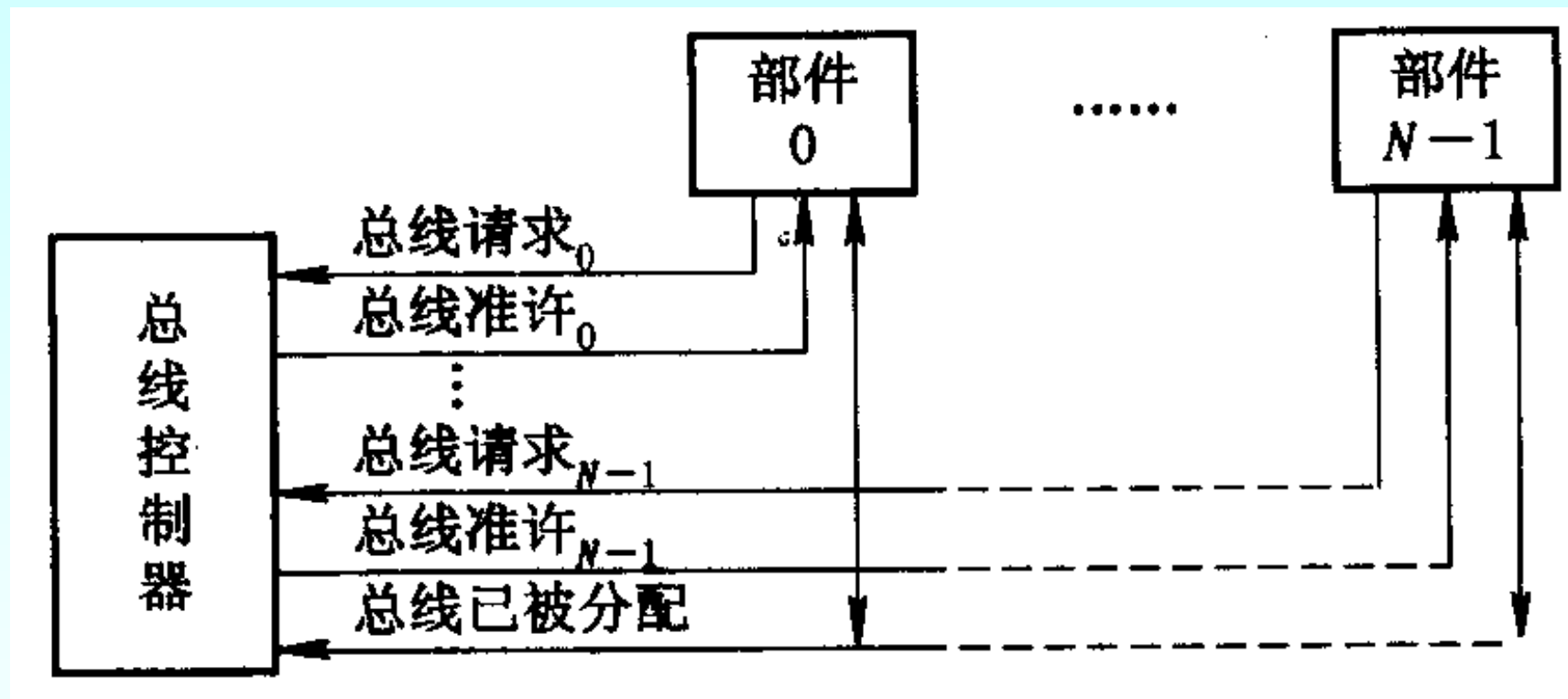


图 3.4 集中式独立请求



### 3.3 总线设计

## 2. 总线的分布仲裁方式

分布仲裁方式不需要中央仲裁器，即总线控制逻辑分散在连接于总线上的各个部件或设备中。连接到总线上的主方可以启动一个总线周期，而从方只能响应主方的请求。每次总线操作，只能有一个主方占用总线控制权，但同一时间里可以有一个或多个从方。对多个主设备提出的占用总线请求，一般采用优先级或公平策略进行仲裁。



## 3.3 总线设计

### 1. 自举分布式

- 每个设备的优先级固定，需要请求总线控制权的设备在各自对应的总线请求线上送出请求信号。在总线仲裁期间，每个设备通过取回的信息能够检测出其他比自己优先级高的设备是否发出了总线请求，如果没有，则立即使用总线，并通过总线忙信号阻止其他设备使用总线；如果一个设备在发出总线请求的同时，检测到其他优先级更高的设备也请求使用总线，则本设备不能马上使用总线。



## 3.3 总线设计

### 2. 冲突检测分布式

- 当某个设备要使用总线时，首先检查是否有其他设备正在使用总线，如果没有，则它就置总线忙，并直接使用总线。若两个设备同时检测到总线空闲，那它们可能都会立即使用总线，从而发生冲突。因此，每个设备在使用过程中，会侦听总线以检测是否发生冲突，当发生冲突，两个设备都会停止传输，延迟一个随机时间之后再重新使用总线，以避免冲突。一般用在网络通信总线上，**Ethernet**就是使用该方案。



## 3.3 总线设计

### 3. 并行竞争分布式

- 这是一种较复杂但有效的裁决方案。其基本思想是：总线上的每个设备都有一个唯一的仲裁号，需要使用总线的主控设备把自己的仲裁号发送到仲裁线上，这个仲裁号将用在并行竞争算法中。每个设备根据仲裁算法决定在一定时间段后占用总线还是撤销仲裁号。





### 工作原理

把对外围设备的管理工作从CPU分离出来。

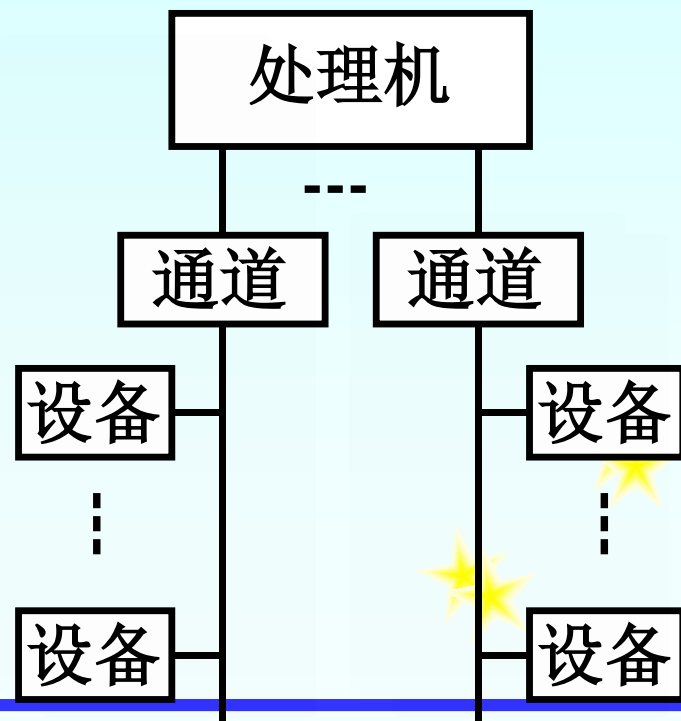
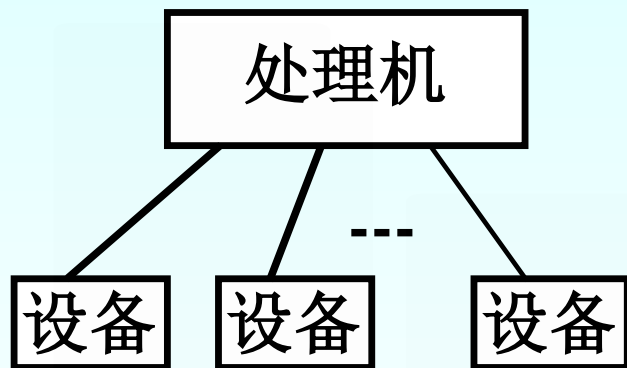
从IBM 360系列机开始，普遍采用通道处理机技术。



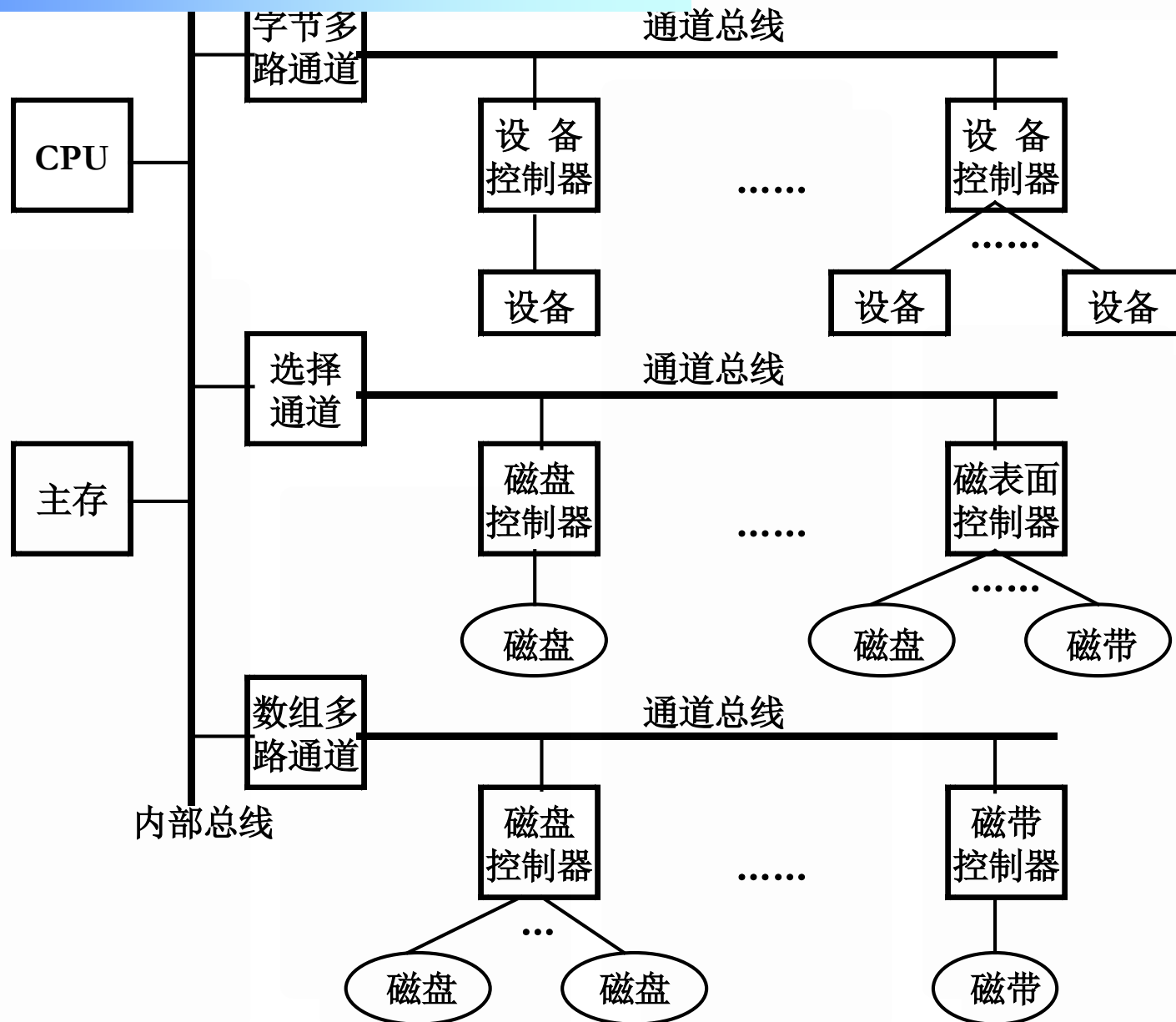
## 3.4 通道处理机

### 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



## 3.4 通道处理机



三种类型的通道与 CPU、设备控制器和外围设备的连接关系



## 3.4 通道处理机

### 3.4.1 通道的作用和功能

#### 1、三种基本输入输出方式存在的问题：

(1) **CPU**的输入输出负担很重，不能专心于用户程序的计算工作。

低速外部设备，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**来完成。

(2) 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。



## 3.4 通道处理机

### 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的变换。🔊

## 3.4 通道处理机

### 3.通道的硬件组成

- 通道的硬件包括相关的寄存器和控制逻辑。主要的寄存器有：数据缓冲寄存器、主存地址计数器、传输字节数计数器、通道命令字寄存器、通道状态字寄存器、通道地址字寄存器等，其中通道命令字寄存器（**CCWR**）用来存放通道命令字（**CCW**）。**CCW**是控制**I/O**操作的关键参数，一条条的通道命令字（通道指令）构成通道程序，放在主存中。通道地址字寄存器（**CAWR**）指出了通道程序在主存中的起始地址。通道状态字寄存器（**CSWR**）记录了通道程序执行后本通道和相应设备的各种状态信息，通道状态字（**CSW**）。可供**CPU**了解通道、设备状态和操作结束的原因。



## 3.4 通道处理机

### 3.4.2 通道工作过程

- 用户通过调用通道完成一次数据传输的过程如图所示，**CPU**执行用户程序和管理程序，通道执行通道程序的时间关系如图所示。



## 3.4 通道处理机

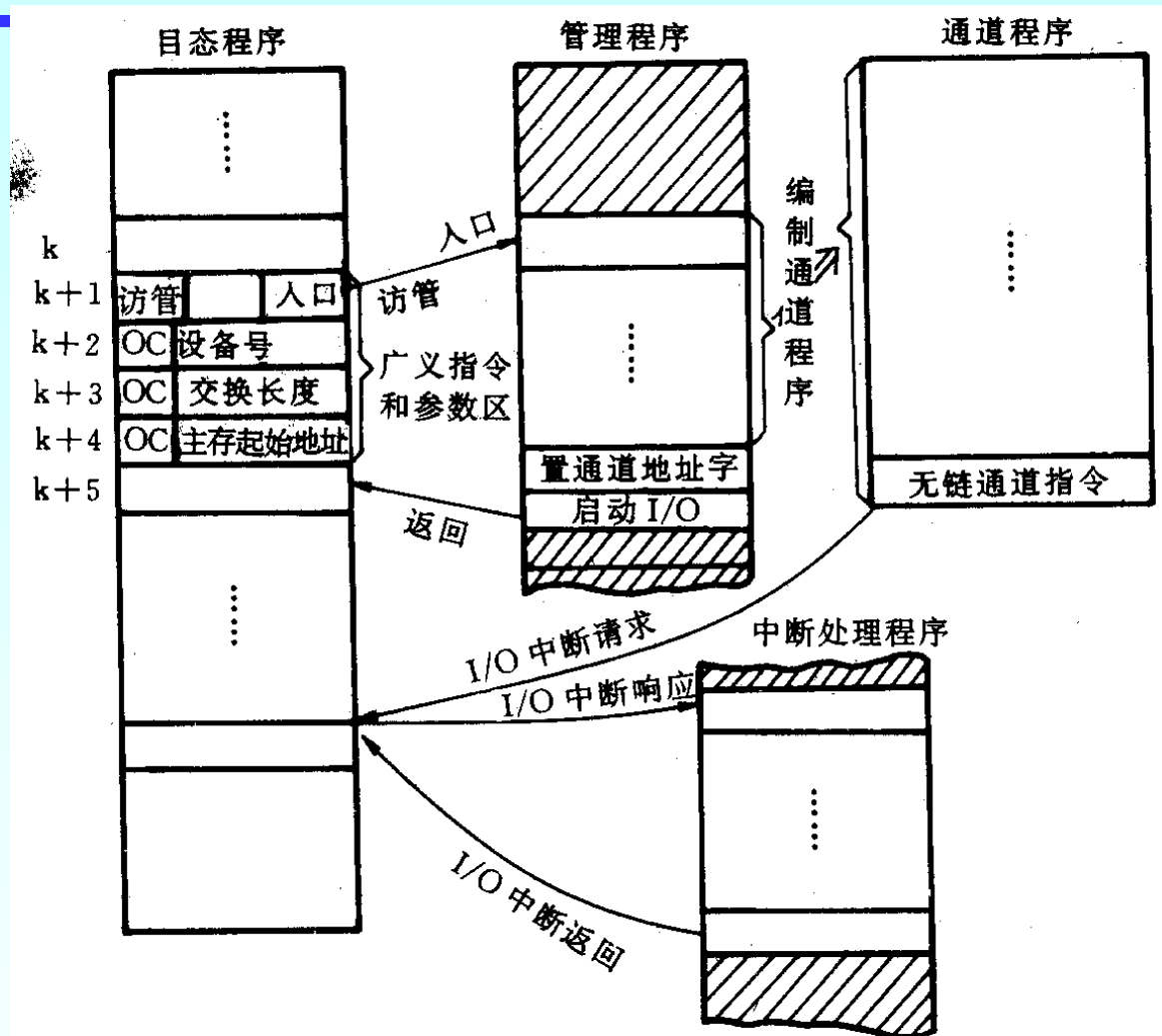


图 3.10 通道处理机输入输出的主要过程



## 3.4 通道处理机

通道完成一次数据输入输出的过程需三步：

- (1) 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。
- (2) 通道处理机执行通道程序，完成指定的数据输入输出工作。
- (3) 通道程序结束后第二次调用管理程序对输入输出请求进行处理。

每完成一次输入输出工作，**CPU**只需要两次调用管理程序，大大减少了对用户程序的打扰。



## 3.4 通道处理机

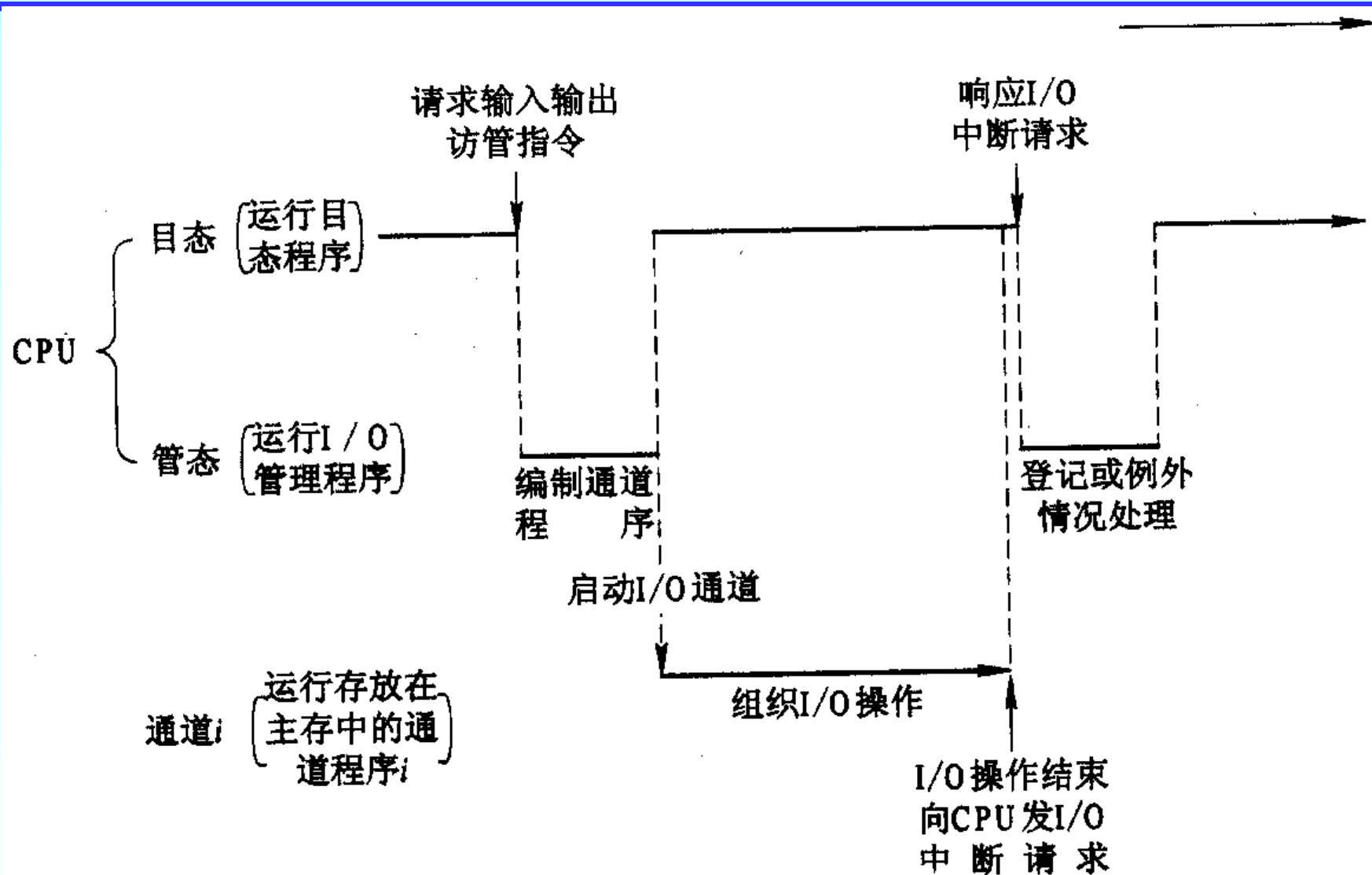
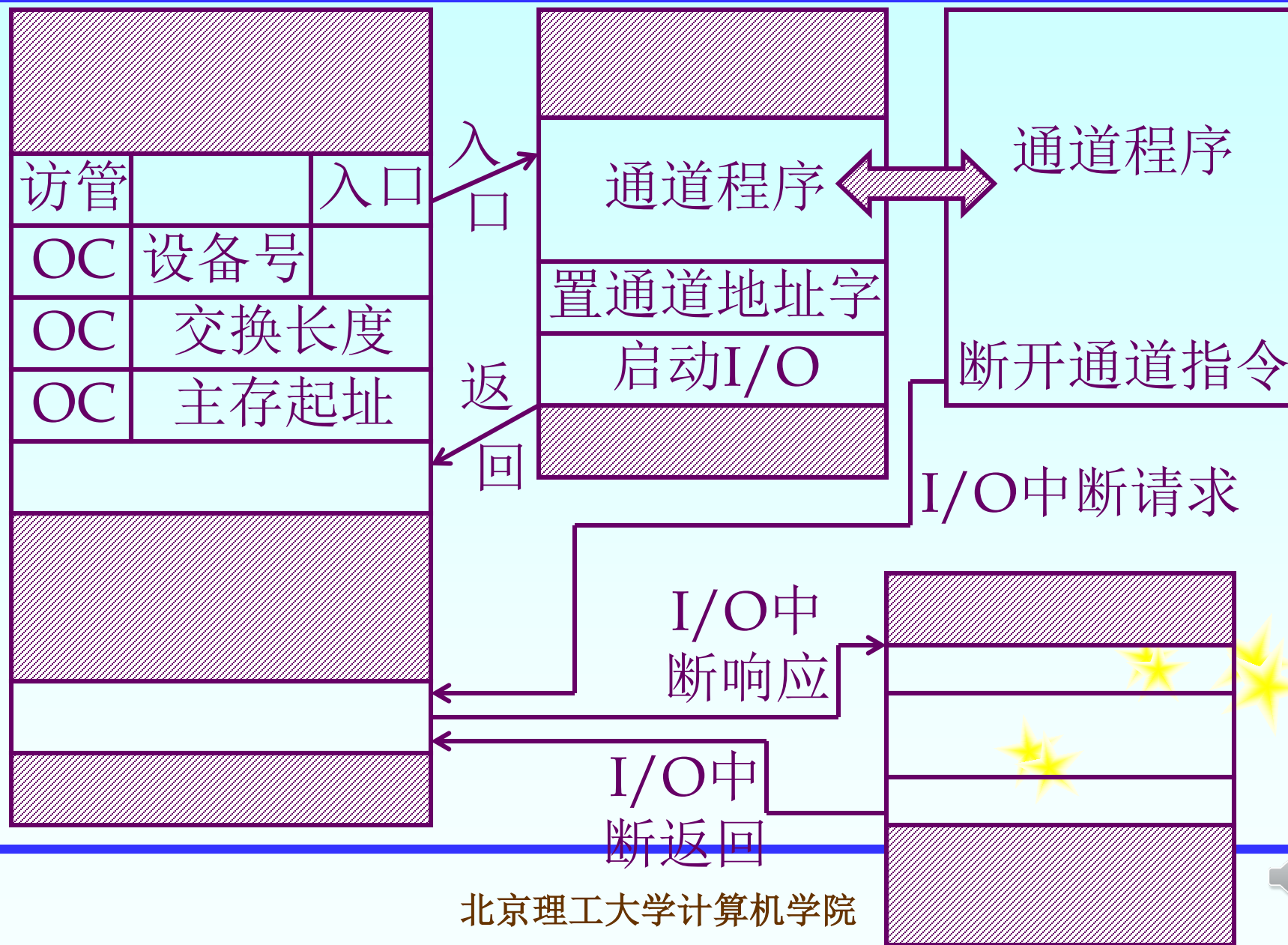


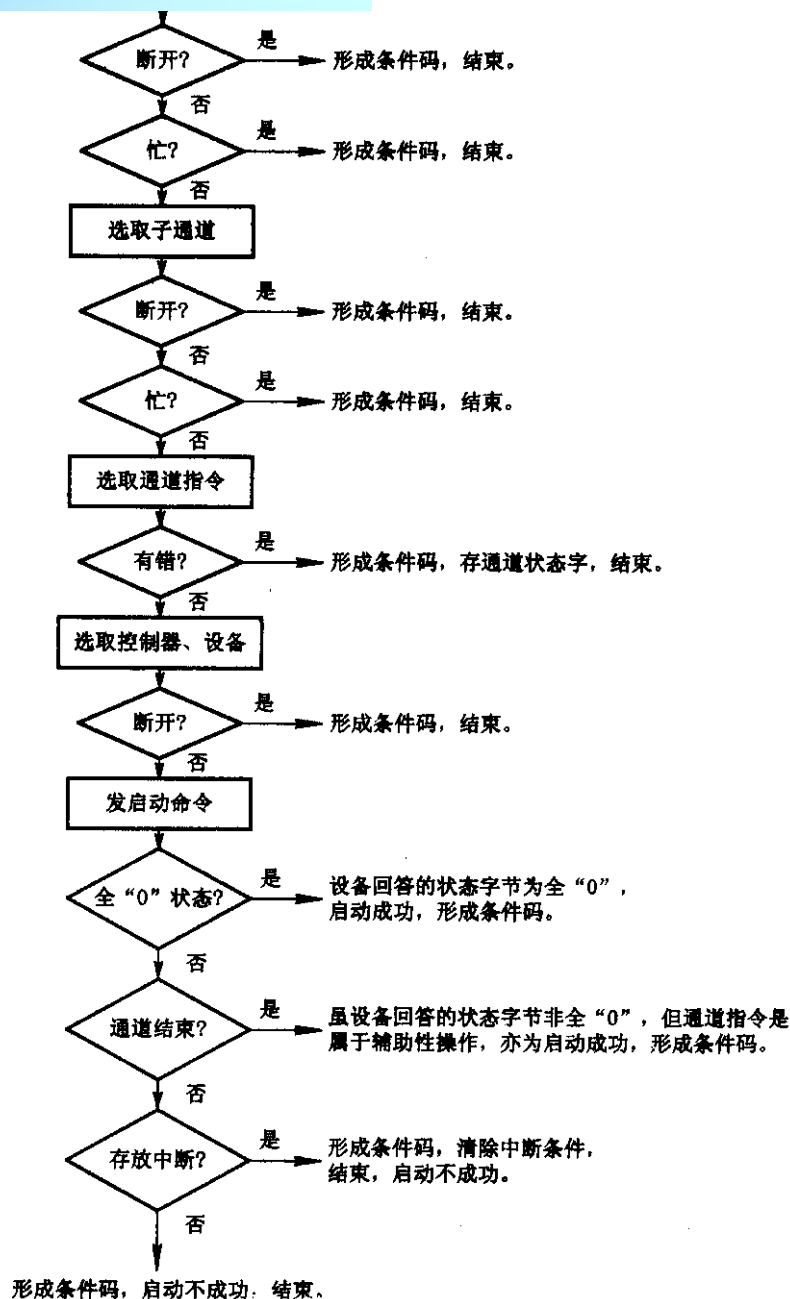
图 3.11 通道处理机输入输出主要过程的时间关系示意图

## 3.4 通道处理机



图

3.12 启动I/O指令流程(注: 这里的结束, 表示释放通道)



## 3.4 通道处理机

### 3.4.3通道的类型

按照输入/输出信息的传送方式，通道分为三种类型：字节多路通道、选择通道和数组多路通道。

#### (1)字节多路通道

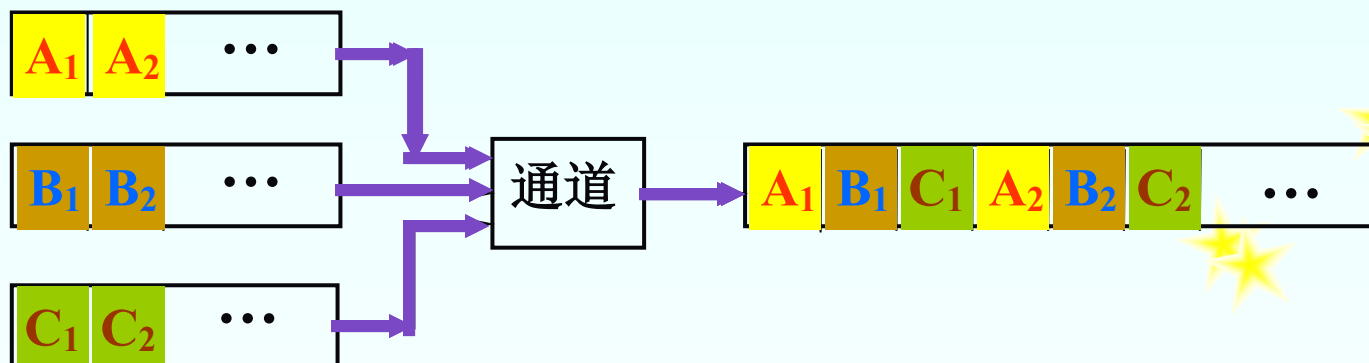
字节多路通道是一种简单的共享通道，用于连接与管理多台低速设备，以字节交叉方式传送信息。

字节多路通道包含有多个子通道，每个子通道连接一个设备控制器。

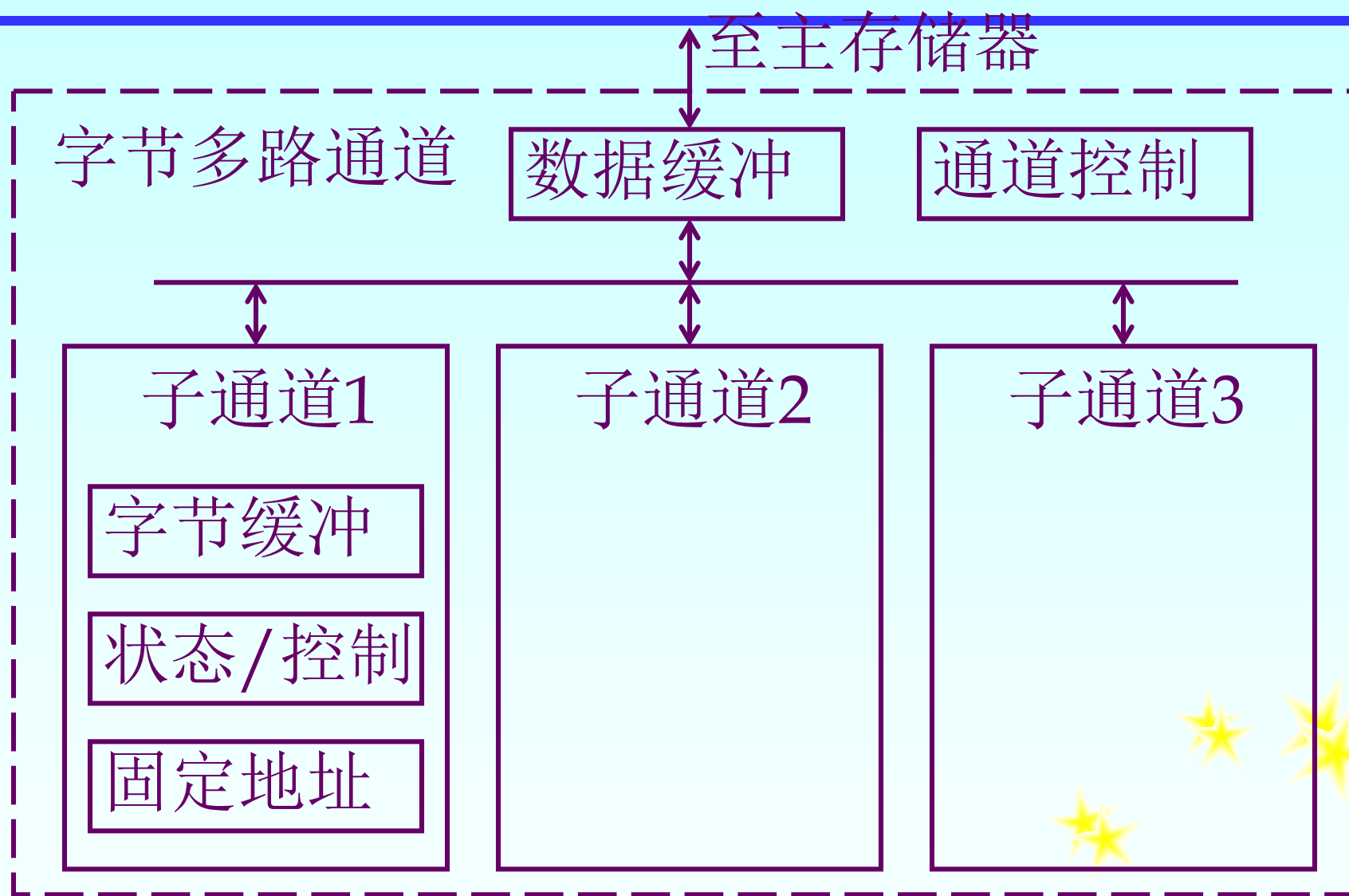


## 3.4 通道处理机

通道不间断地、轮流地启动每个设备控制器，当通道为一个设备传送完一个字节后，就转去为另一个设备服务。当通道为某一设备的传送时，其它设备可以并行地工作，准备需要传送的数据字节或处理收到的数据字节，以准备接收新的数据字节。这种轮流服务是建立在主机的速度比外设的速度高的基础之上，它可以提高系统的工作效率。



## 3.4 通道处理机



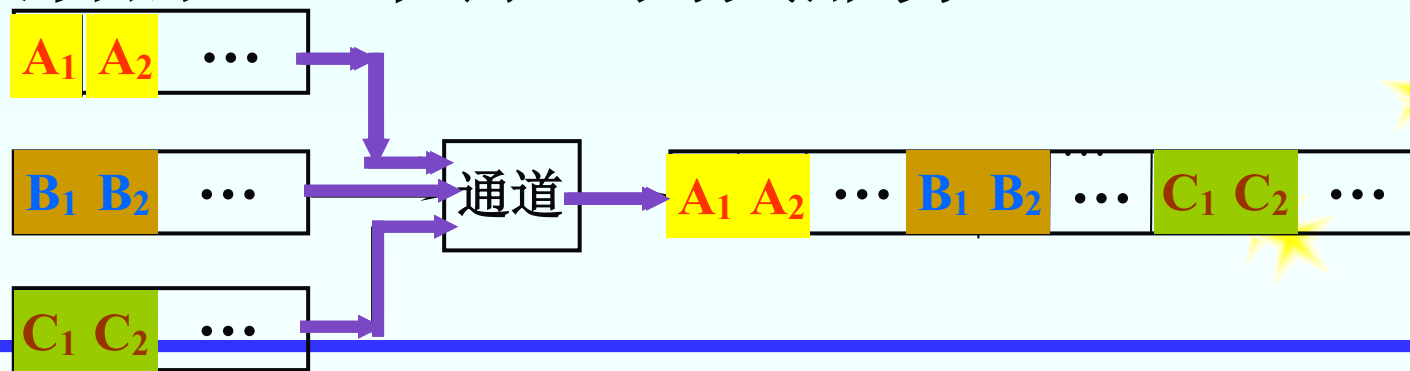
字节多路通道的结构



## 3.4 通道处理机

## (2) 选择通道

选择通道又称高速通道，在物理上它也可以连接多个设备，但这些设备不能同时工作，在一段时间内通道只能选择一台设备进行数据传送，此时该设备能独占整个通道。因此，选择通道一次只能执行一个通道程序，只有当它与主存交换完信息后，才能再选择另一台外部设备并执行该设备的通道程序，为其服务。

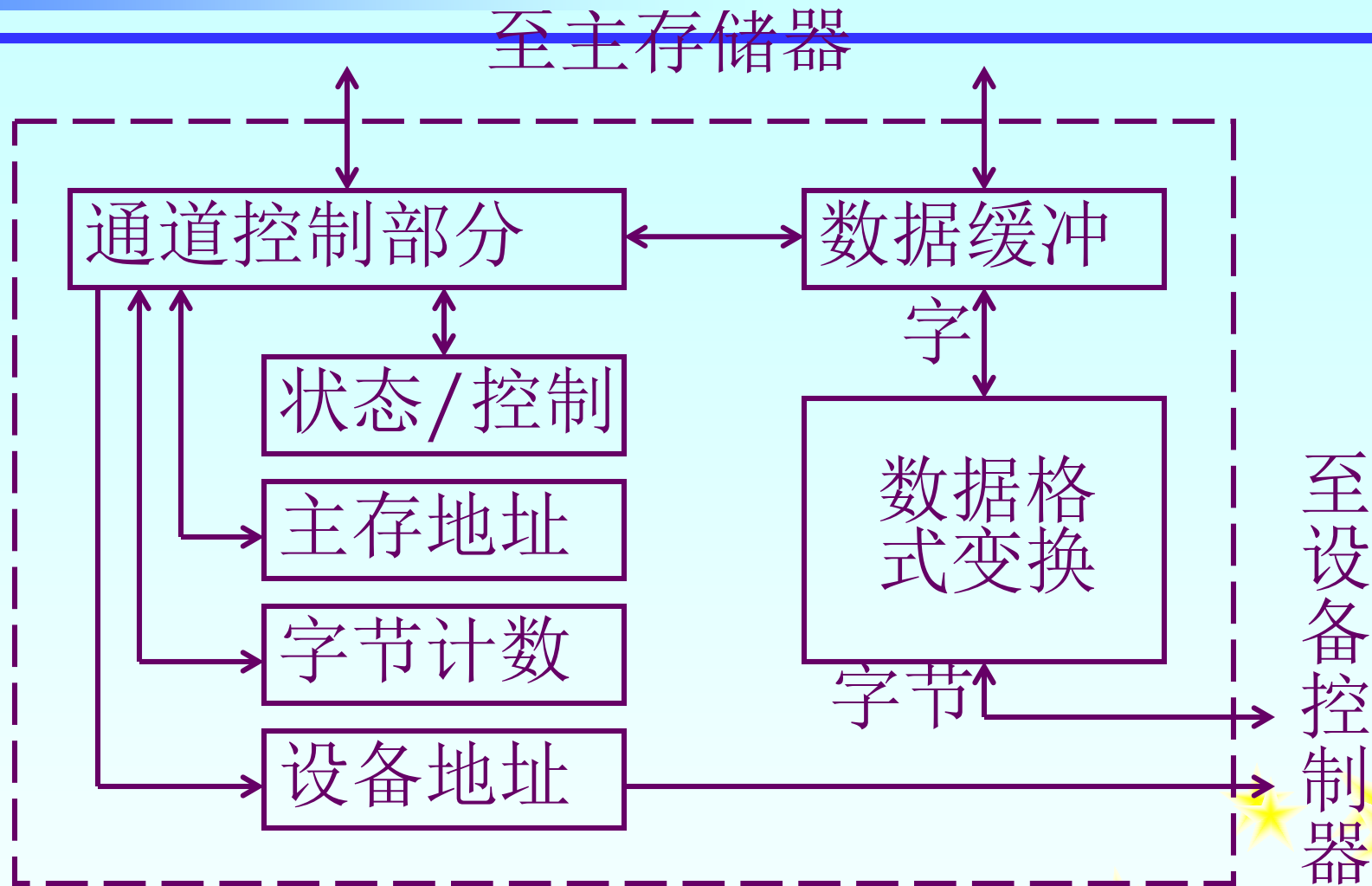




每个选择通道只有一个以成组方式工作的子通道，逐个为多台高速外围设备服务。



## 3.4 通道处理机



选择通道的结构



### (3) 数组多路通道

数组多路通道是把字节多路通道和选择通道的特点结合起来的一种通道结构。它的基本思想是：当某设备进行数据传送时，通道只为该设备服务；当设备在执行辅助操作（如磁头移动等）时，通道暂时断开与这个设备的连接，挂起该设备的通道程序，去为其他设备服务。

数组多路通道有多个子通道，既可以执行多路通道程序，像字节多路通道那样，所有子通道分时共享总通道；又可以用选择通道那样的方式成组传送数据。既具有多路并行操作的能力；又具有很高的数据传送速率，使通道的效率充分得到发挥。



### 3.4 通道处理机

从磁盘存储器读出一个文件的过程分为三步：定位、找扇区、读出数据，前两步为辅助操作。

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。



## 3.4 通道处理机

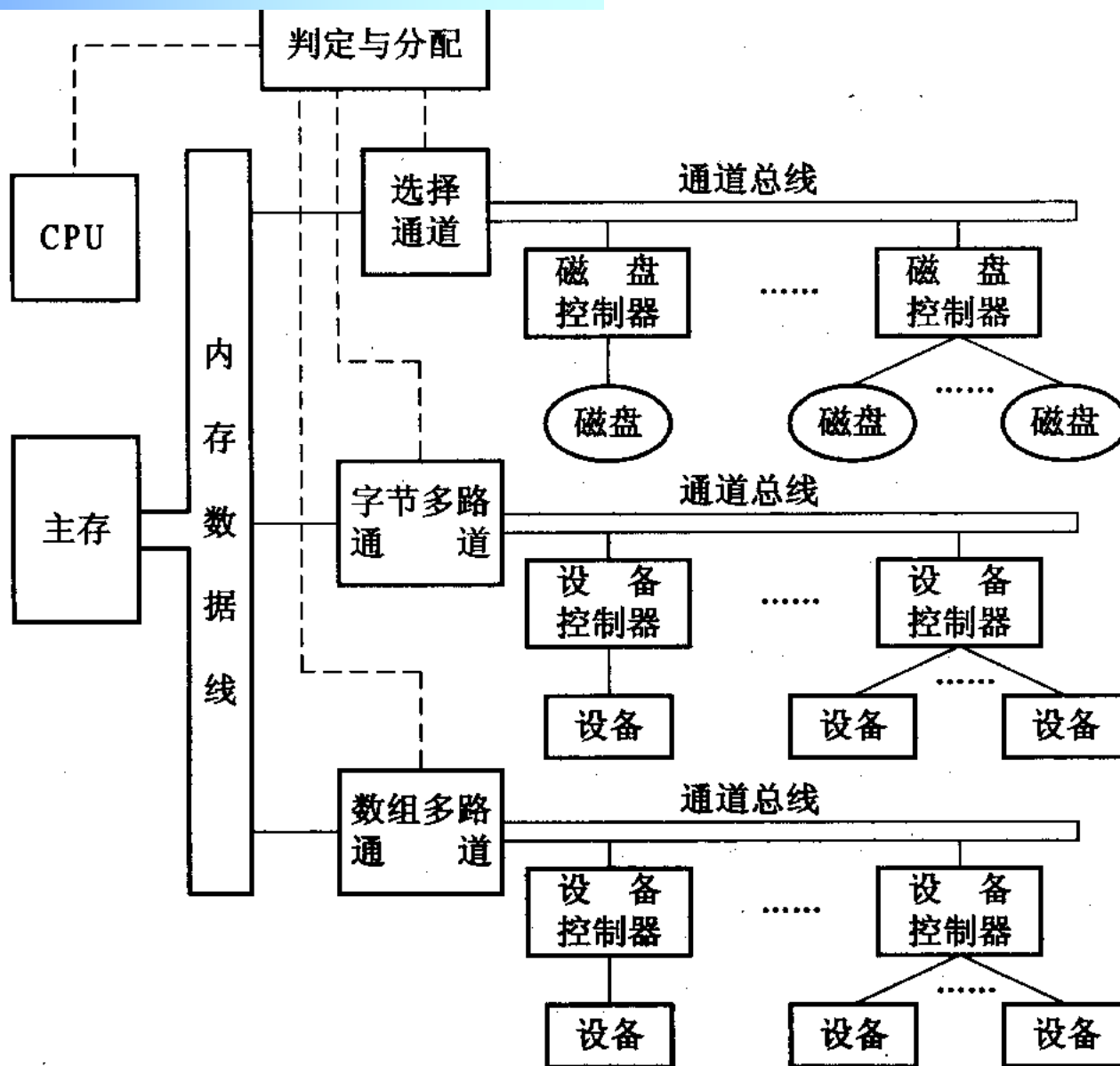


图 3.13 IBM 370 的 I/O 结构

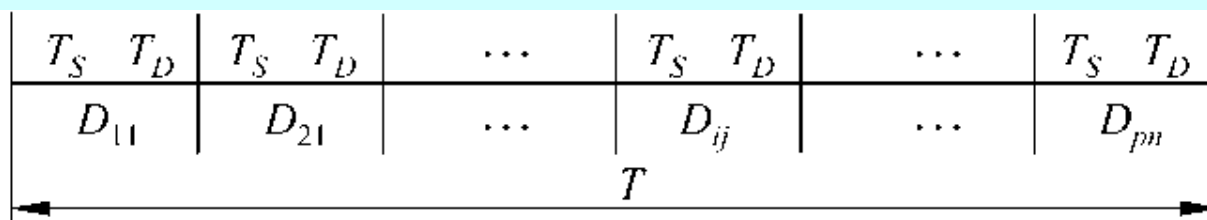
## 3.4 通道处理机

### 3.4.4 通道中的数据传送过程

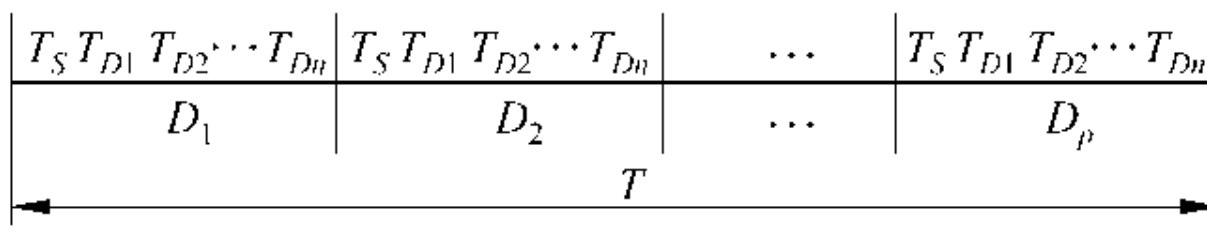
- 一个字节多路通路是分时为多台低速和中速外设服务的，在有  $P$  台设备同时连接到一个字节多路通道上时，它们的数据传送过程如图所示。



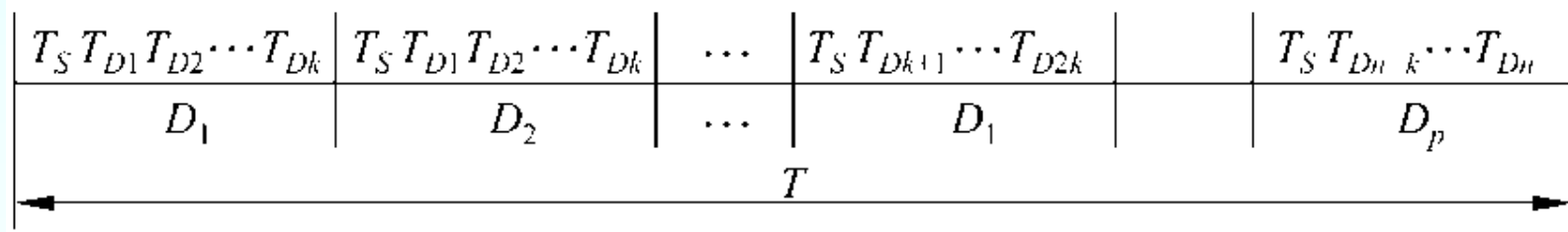
# 3.4 通道处理机



(a)



(b)



(c)



## 3.4 通道处理机

### 3.4.5 通道流量的分析

**通道流量：**单位时间内能够传送的最大大数据量。又称为通道吞吐率，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。





### 3.4 通道处理机

一个通道能达到的极限流量与其工作方式、数据传送期内选择一次设备的时间 $T_S$ 和传送一个字节的时间 $T_D$ 的长短有关。由于字节多路通道每选择一台设备只传送一个字节，故其通道极限流量

$$f_{\max \cdot \text{byte}} = 1/(T_S + T_D)。$$



### 3.4 通道处理机

选择通道每选择一台设备就把 $N$ 个字节全部传送完，其通道极限流量

$$f_{\max \cdot \text{select}} = N / (T_S + NT_D) = 1 / (T_S / N + T_D)。$$

数组多路通道每选择一台设备只传送 $K$ 个字节，如果要传送 $N$ 个字节，就得经  $[N/K]$  次传送才行，每次都要花去一个选择设备的时间 $T_S$ ，所以，其通道极限流量

$$f_{\max \cdot \text{block}} = K / (T_S + KT_D) = 1 / (T_S / K + T_D)。$$



### 3.4 通道处理机

显然，若通道的 $T_S$ 、 $T_D$ 一定，且 $N > K$ 时，字节多路方式时所能达到的极限流量最小，数组多路方式时的极限流量居中，选择方式时的极限流量最大。



## 3.4 通道处理机

三种通道的最大流量计算公式如下：

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

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

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



### 3.4 通道处理机

对于采用字节交叉方式工作的字节多路通道应为该通道所接各设备的字节传送速率之和，即

$$f_{byte \cdot j} = \sum_{i=1}^{p_j} f_{i \cdot j}$$

而对于其他两种类型的通道应为所接各设备的字节传送速率中之最大的那个，即

$$f_{block \cdot j} = \max_{i=1}^{p_j} f_{i \cdot j}$$

$$f_{select \cdot j} = \max_{i=1}^{p_j} f_{i \cdot j}$$



### 3.4 通道处理机

为了保证通道能够正常工作，不丢失数据，必须满足设备要求通道的实际最大流量不超过通道所能达到的极限流量这一流量设计的最基本原则，因此，对上述 3 种类型的通道应分别满足关系式：

$$f_{byte \cdot j} \leq f_{\max \cdot byte \cdot j}$$

$$f_{block \cdot j} \leq f_{\max \cdot block \cdot j}$$

$$f_{select \cdot j} \leq f_{\max \cdot select \cdot j}$$



## 3.4 通道处理机

如果I/O系统有 $m$ 个通道，其中1至 $m_1$ 为字节多路通道， $m_1+1$ 至 $m_2$ 为数组多路通道， $m_2+1$ 至 $m$ 为选择通道，则该I/O系统工作时的极限流量将为

$$f_{\max} = \sum_{j=1}^{m_1} f_{\max \cdot \text{byte} \cdot j} + \sum_{j=m_1+1}^{m_2} f_{\max \cdot \text{block} \cdot j} + \sum_{j=m_2+1}^m f_{\max \cdot \text{select} \cdot j}$$

必然会满足

$$f_{\max} \geq \sum_{j=1}^{m_1} \sum_{i=1}^{p_j} f_{i \cdot j} + \sum_{j=m_1+1}^{m_2} \max_{i=1}^{p_j} f_{i \cdot j} + \sum_{j=m_2+1}^m \max_{i=1}^{p_j} f_{i \cdot j}$$



### 3.4 通道处理机

设有一字节多路通道，它有 3 个子通道：“0”号、“1”号高速印字机各占一个子通道；“0”号打印机、“1”号打印机和“0”号光电输入机合用一个子通道。假定数据传送期内高速印字机每隔 25  $\mu\text{s}$  发一个字节请求，低速打印机每隔 150  $\mu\text{s}$  发一个字节请求，光电输入机每隔 800  $\mu\text{s}$  发一个字节请求，则这 5 台设备要求通道的流量为

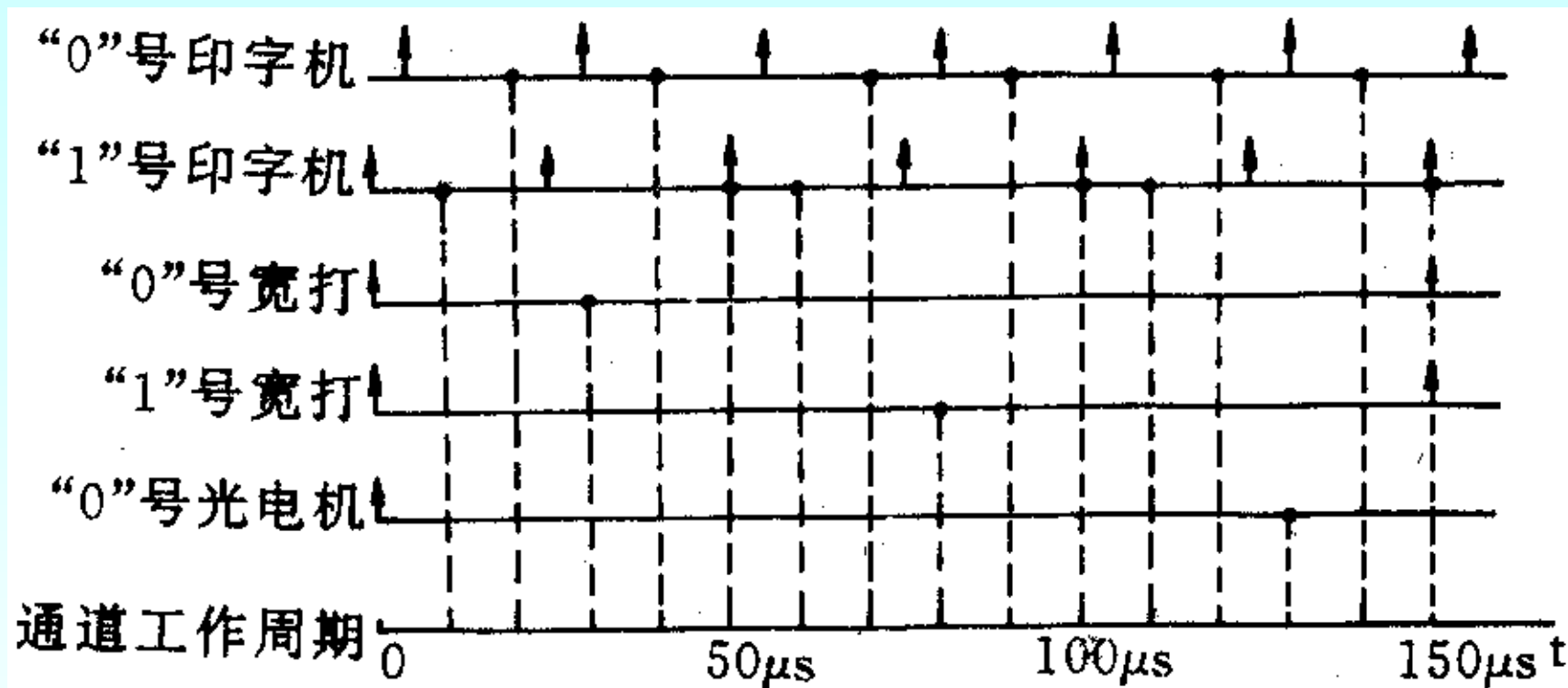
$$f_{\text{byte} \cdot j} = \sum_{i=1}^5 f_{i \cdot j} = \frac{1}{25} + \frac{1}{25} + \left( \frac{1}{150} + \frac{1}{150} + \frac{1}{800} \right)$$

$$\approx 0.095 \text{ MB/s}$$





## 3.4 通道处理机



“ $\uparrow$ ”表示设备提出申请的时刻

“ $\downarrow$ ”表示通道处理完设备申请的时刻

图 3.14 字节多路通道响应和处理各设备请求的时间示意图

### 例3-1

一个字节多路通道连接**D1**、**D2**、**D3**、**D4**、**D5**共**5**台设备，这些设备分别每**10 $\mu$ s**、**30  $\mu$ s**、**30  $\mu$ s**、**50  $\mu$ s**和**75  $\mu$ s**向通道发出一次数据传送的服务请求，请回答下列问题：

**(1)** 计算这个字节多路通道的实际流量和工作周期。



### 3.4 通道处理机

**(2)** 如果设计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，通道响应它的数据传送请求的优先级也高。**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关系图，并计算这个字节多路通道处理完各台设备的第一次数据传送请求的时刻。



## 3.4 通道处理机

**(3)** 从时间关系图上发现什么问题？如何解决这个问题？

解：

通道的实际流量为：

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

通道的工作周期为： **$t = 1/f_{BYTE} = 5\mu\text{s}/\text{byte}$**



## 3.4 通道处理机

通道处理完各设备这个第一次请求的时间如下：

**D1:  $5\mu\text{s}$ ;**

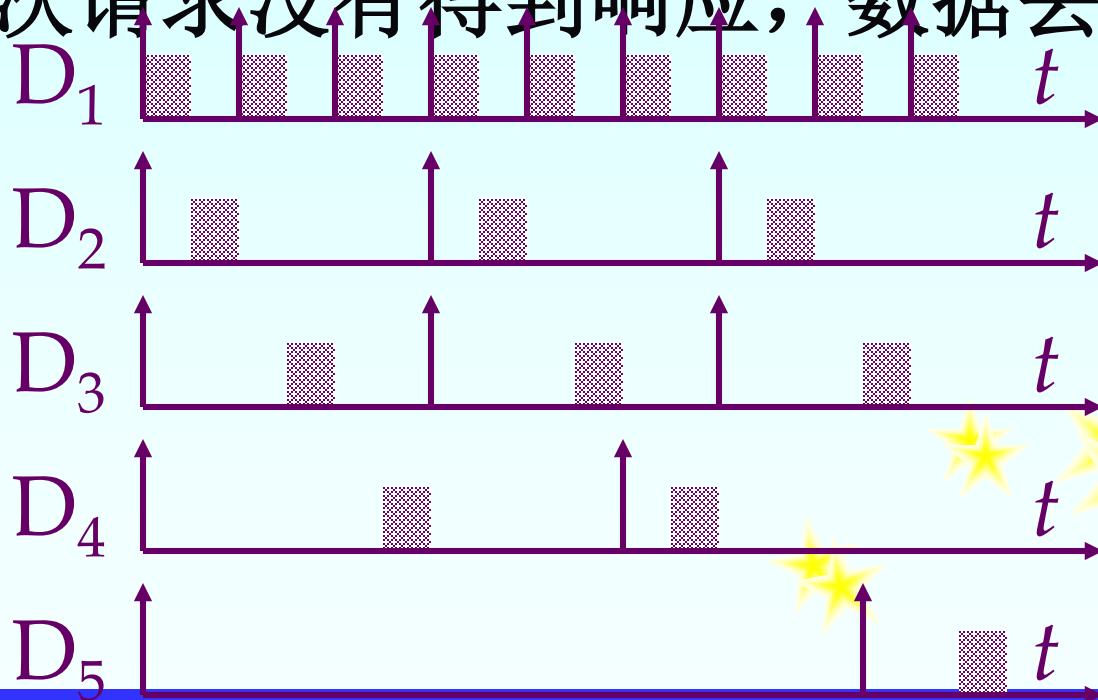
**D2:  $10\mu\text{s}$ ;**

**D3:  $20\mu\text{s}$ ;**

**D4:  $30\mu\text{s}$**

设备**D5**的第一次请求没有得到响应，数据丢失。

设备**D5**第一次请求未得到响应的  
原因分析：



### 3.4 通道处理机

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，但是，在任意设备的任意两次传送请求之间并不能保证都能得到通道的响应。

为了保证字节多路通道能够正常工作，可以采取下列几种方法：

**方法一：**增加通道的最大流量。保证连接在通道上的所有设备的数据传送请求能够及时得到通道的响应。

**方法二：**动态改变设备的优先级。例如，在 **$30\mu\text{s}$** 至 **$70\mu\text{s}$** 之间临时提高设备**D5**的优先级。



### 3.4 通道处理机

**方法三：**增加缓冲存储器。特别是对优先级比较低的设备。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 $\mu$ s**处得到响应，第二次请求可以在第**145 $\mu$ s**处得到响应。



## 3.5 中断系统

为处理一个中断请求，必须调出相应的中断处理程序。如果中断源比较少时，通过中断系统硬件就可以比较方便地对每个中断源直接形成相应的中断处理程序入口，进入相应的中断处理程序。但对中、大型多用途机器，中断源一般可多达数十至数百个。如果为每个中断源单独形成入口，不仅硬件难以实现，代价也很大，就是在中断处理上也没有这种必要。因为不少中断源的性质比较接近，可以将它们分别归成几类，对每一类给定一个中断处理程序入口，再由软件转入对相应的中断源进行处理，这样可以大大简化中断处理程序入口形成硬件。





## 3.5 中断系统

### 3.5.1 中断的分类和分级

引起中断的各种事件称为中断源。中断源向中断系统发出请求中断的申请，称为中断请求。同时可能有多个中断请求，这时中断系统需要按事先确定的中断响应优先次序对优先级高的中断请求予以响应。所谓中断响应就是允许其中断CPU现行政程序的运行，转去对该请求进行预处理，包括保存好断点现场，调出有关处理该中断的中断处理程序，准备运行。这部分工作在大多数机器上都是采用交换新旧程序状态字PSW的办法实现的。当然为了某种需要，中断系统也可以对中断请求进行屏蔽，使之暂时得不到响应。



## 3.5 中断系统

中断现场包括软件状态(如作业名称和级别, 上、下界值, 各种软件状态和标志等)和硬件状态(如现行指令地址, 条件码等状态信息, 各种控制寄存器及通用寄存器内容)。通常采取把分散于CPU各部分的硬件状态集成程序状态字, 然后由中断响应硬件通过将程序状态字(处理器状态字、换道区)存到主存指定单元或区域的方式来完成保存。再把新的程序或进程的程序状态字(处理器状态字、换道区)从主存另一指定单元或区域把内容传送到有关寄存器和计数器中, 建立起运行新的程序或进程的环境。硬件状态是全经中断响应硬件保存, 还是部分经它, 部分经中断处理程序保存, 具体机器的作不同的选择。



## 3.5 中断系统

### 3.5.2 中断系统的软硬件功能分配

中断系统的功能包括中断请求的保存和清除、优先级的确定、中断断点及现场的保存、对中断请求的分析和处理以及中断返回等，这些全是由中断响应硬件和中断处理程序共同完成的。因此，中断系统的软、硬件功能分配实质上就是中断处理程序软件和中断响应硬件的功能分配。



## 本章重点

输入输出系统

总线的控制方式

通道工作原理

通道类型

通道流量的分析



## 第三章作业

3-3

3-8

3-9

