南京理工大学

NANJING UNIVERSITY OF SCIENCE & TECHNOLOGY

第 6 章 总线与I/O系统组织

主讲：张功萱

第6.9节

I/O通道方式

1. 通道与DMA有哪些区别?
2. I/O通道有哪些类型?
3. 通道的组成结构是什么?
4. 什么是通道程序?

1. 概述

- I/O通道是一种能够执行有限I/O指令，并且能够被多台外围设备共享的小型DMA专用处理机。
- 在计算机系统中，通道作为一个独立的I/O控制部件，能执行有限的I/O通道指令，代替CPU管理和控制外设。通道使主机与I/O设备之间能够达到更高的并行程度。
- I/O通道的任务
- 控制、管理输入/输出操作，为I/O设备提供传送数据的通道。

I/O通道的特点

- (1) I/O通道有自己的指令系统，能够独立执行用通道命令编写的输入输出控制程序，产生相应的控制信号控制设备的工作。
- (2) I/O通道可根据需要控制多种不同的设备。
- (3) 每个I/O通道可以连接多个外部设备，每个外设对应一个子通道。
- (4) I/O通道通过系统中的数据通路与设备的控制器进行通信。

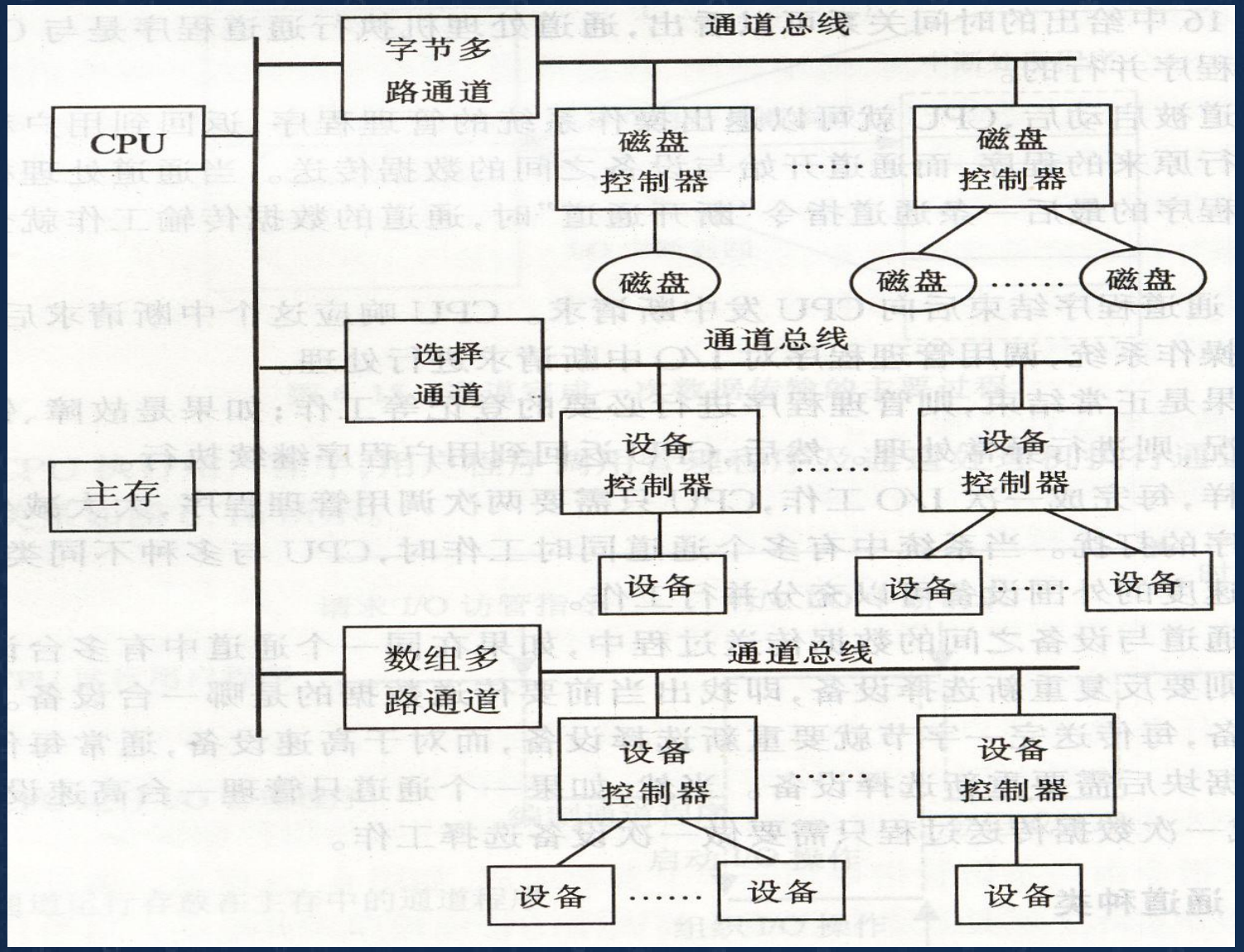
I/O通道与DMA方式的异同

- (1) 相同点
- I/O通道与DMA方式都是在主存与I/O设备之间建立数据通路，用控制器控制数据的直接传送。
- (2) 不同点
- DMA方式通过硬件控制主存与设备之间的信息传送。 I/O通道通过执行通道程序控制主存与设备之间的信息传送。
- DMA方式只能控制少量的同类设备，只能传送数据。 I/O通道可控制多种不同的设备，除传送数据外，还可以接口的初始化、故障诊断与处理等工作。

I/O通道与中断方式的异同

- ⑴ 相同点
- I/O通道与程序中断方式都是通过执行程序去管理I/O操作，因而灵活性较强，都可以通过扩展程序的功能来扩展处理能力。
- ⑵ 不同点
- 程序中断方式在数据传输时需要占用CPU的时间；
- I/O通道在被CPU启动后，基本可以取代CPU去管理I/O操作，使CPU可以从大部分I/O管理中解脱出来。

带有I/O通道的I/O系统结构



- 主机—通道—设备控制器—设备四级连接方式。

（1）CPU的任务

- 执行I/O指令。
- 启动/关闭通道与设备。
- 处理来自通道的中断，如数据传输中断、故障中断等。

- 通道的管理的任务由操作系统完成。

（2）I/O通道的任务

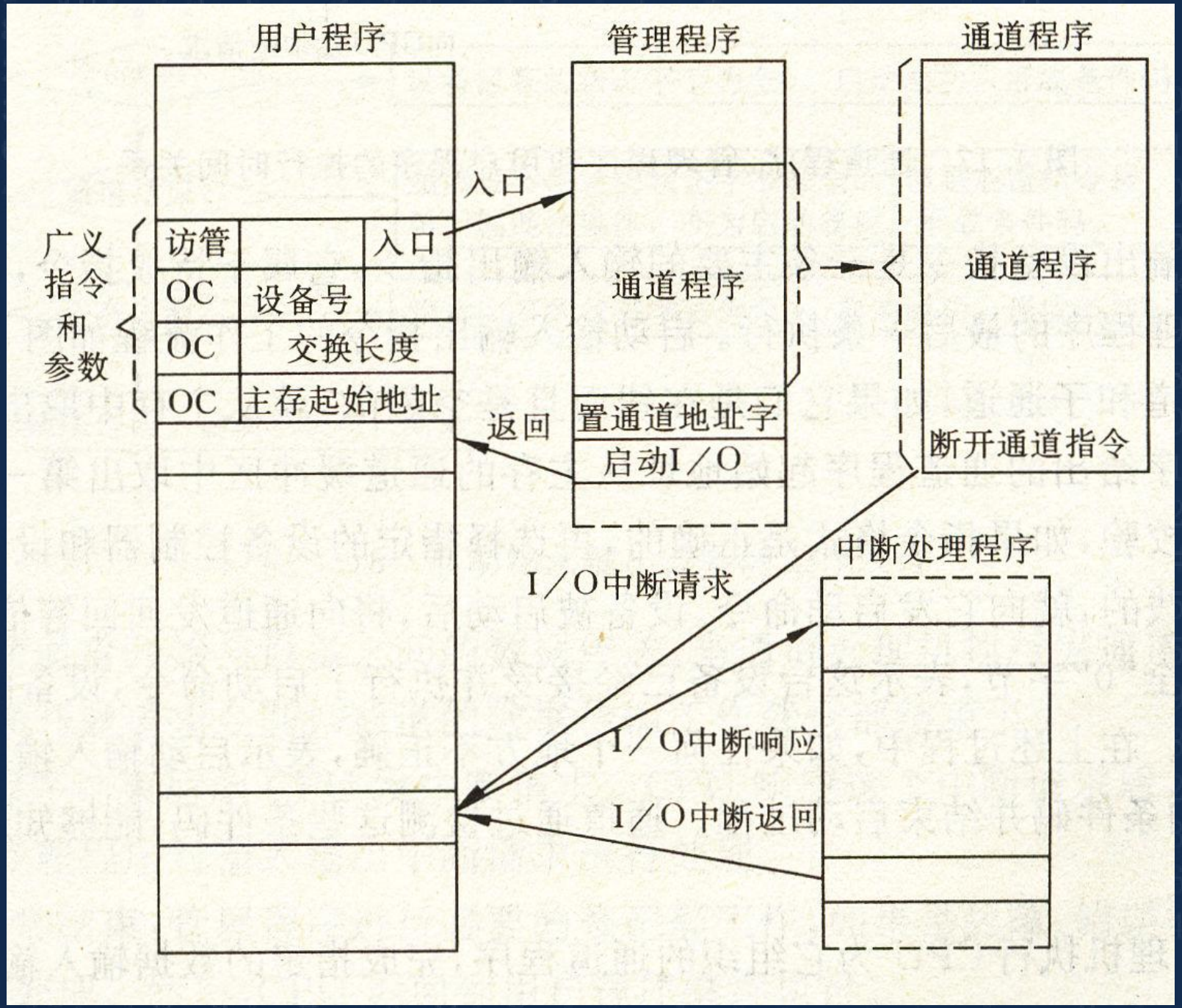
- ① 接受CPU发来的I/O指令，与指定的设备连接,访问指定的设备。
- ② 执行CPU为通道组织的通道程序。
- 从通道缓冲区中读取通道指令，经译码分析，向指定的设备控制器或设备发出各种操作控制命令。
- ③ 组织和控制数据在内存与外设之间的信息传送操作。
- 根据需要提供数据缓存空间以及数据存入内存或从内存中读取的地址；提供外设的有关地址；控制传送的数据量；指定传送工作结束时要进行的操作，根据对传送数据的计数判断数据传送工作是否结束。

（2）I/O通道的任务

- ④ 在数据传输过程中完成必要的格式变换，例如，把字拆卸为字节，或者把字节装配成字等。
- ⑤ 读取和接收外设的状态信息， 检查外围设备的工作状态是正常还是故障，形成通道状态信息，并根据需要将设备的状态信息送往主存指定单元保存。
- ⑥ 向**CPU**发出**I/O**中断请求。对来自外设及通道的中断请求按优先次序进行排队后报告**CPU**。
- 通道使用通道指令控制设备进行数据传送操作，并以通道状态字的形式接收设备控制器提供的外部设备的状态。

（3）设备控制器的任务

- ① 从通道接受通道指令，控制外部设备完成指定的操作。
- 如控制外设的启/停，向设备发出各种非标准的控制信号等。
- ② 向通道提供外部设备的状态。
- 如设备的忙、闲、出错信息等。
- ③ 将各种外部设备的不同信号转换成通道能够识别的标准信号。
- ④ 控制辅助操作。
- 如磁带的进带、倒带等操作。

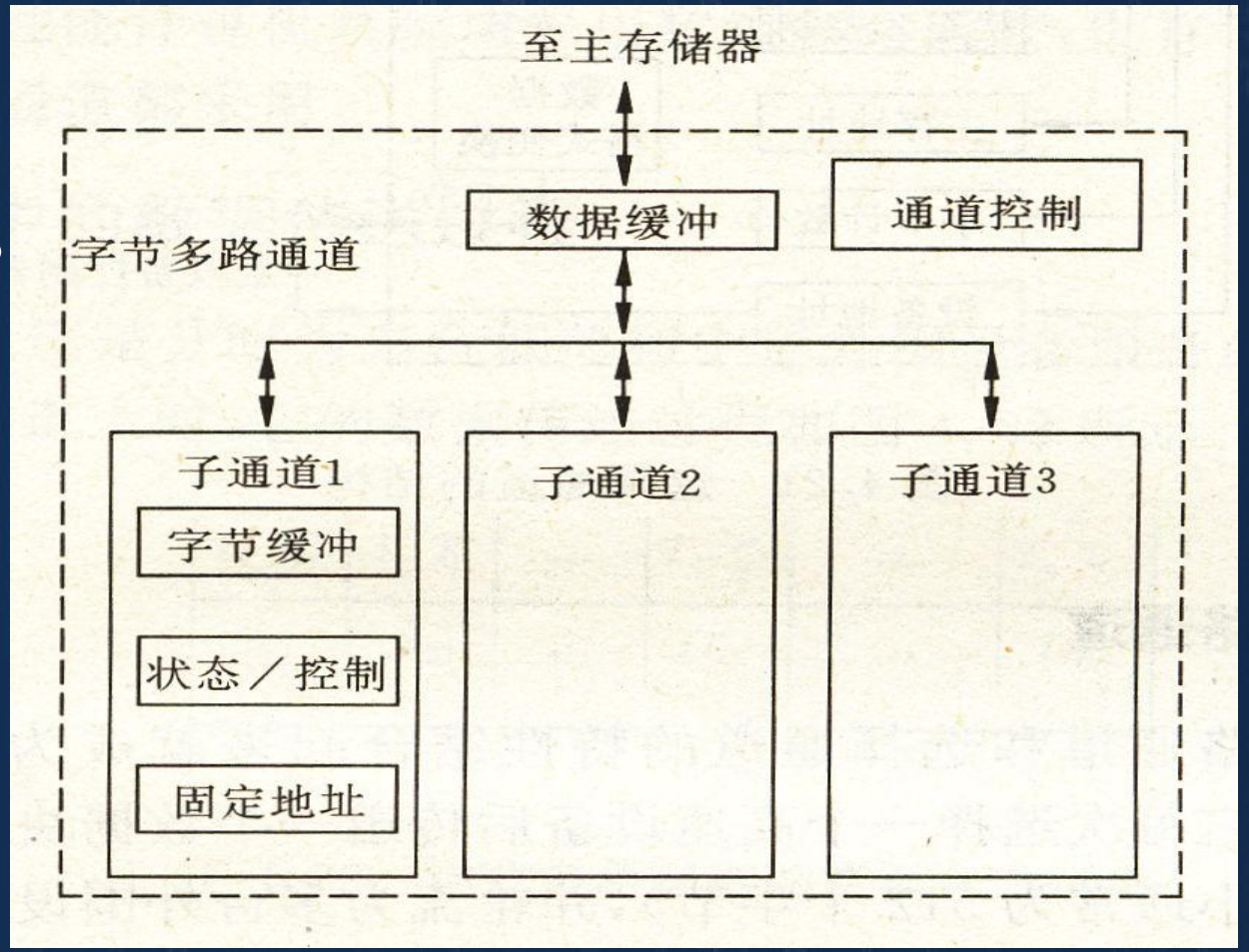

用户程序
管理程序
通道程序
广义指令和参数
访管
入口
OC
设备号
OC
交换长度
OC
主存起始地址
入口
通道程序
通道程序
返回
置通道地址字
启动I／O
断开通道指令
中断处理程序
I／O中断请求
I／O中断响应
I／O中断返回

2. 通道的类型

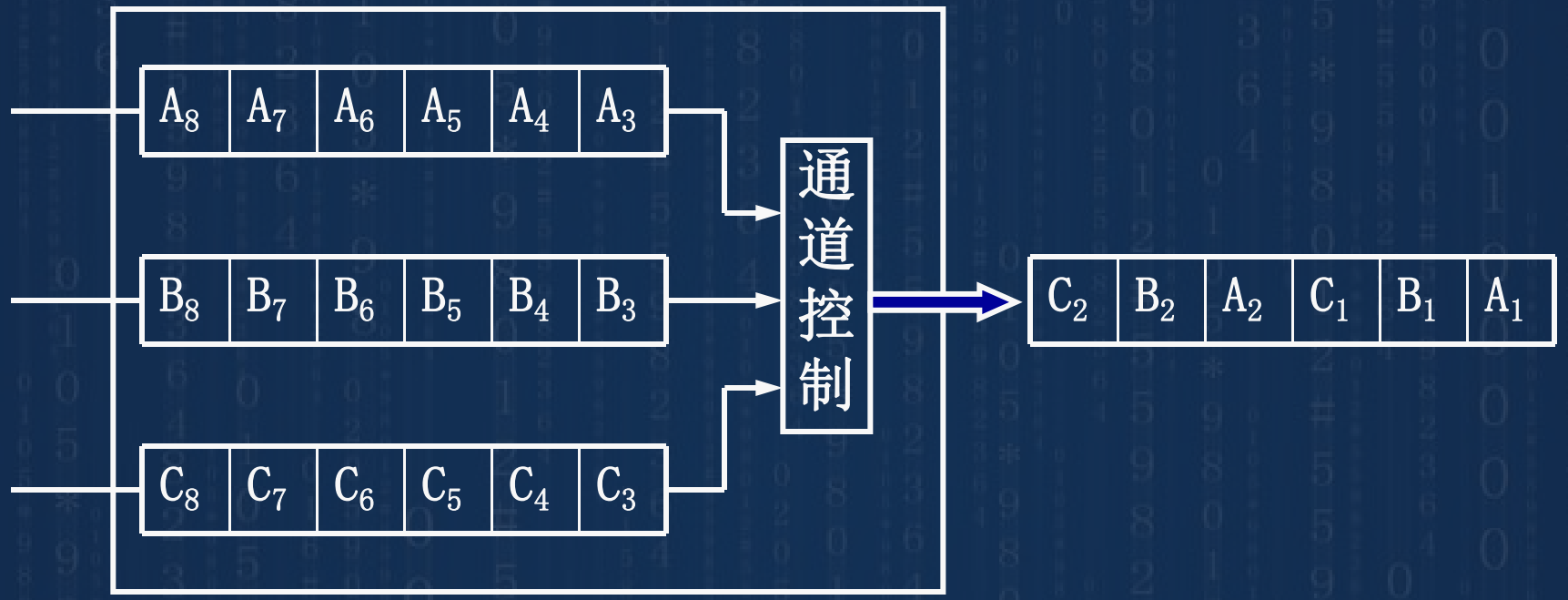
- 根据多台外设共享通道的不同情况，可将通道分为三种类型。
- 1）字节多路通道
- 字节多路通道是一种简单的共享通道，可以依靠通道与CPU之间的高速数据通路分时地为多台设备服务。
- 在字节多路通道中，一个通道含有多个子通道，使用公共的控制部分。每个子通道连接一个设备控制器，一个设备控制器可连接多台设备，设备可以采用字节交叉模式分时交替地使用通道进行数据传送。

- **字节交叉模式：**连接在通道上的各个设备轮流占用一个很短的时间片传输一个字节。

- 字节多路通道要求每种设备分时占用通道一个很短的时间段，不同的设备在各自分得的时间段内与通道建立传输连接，实现数据的传送。



字节多路通道的信息传送方式

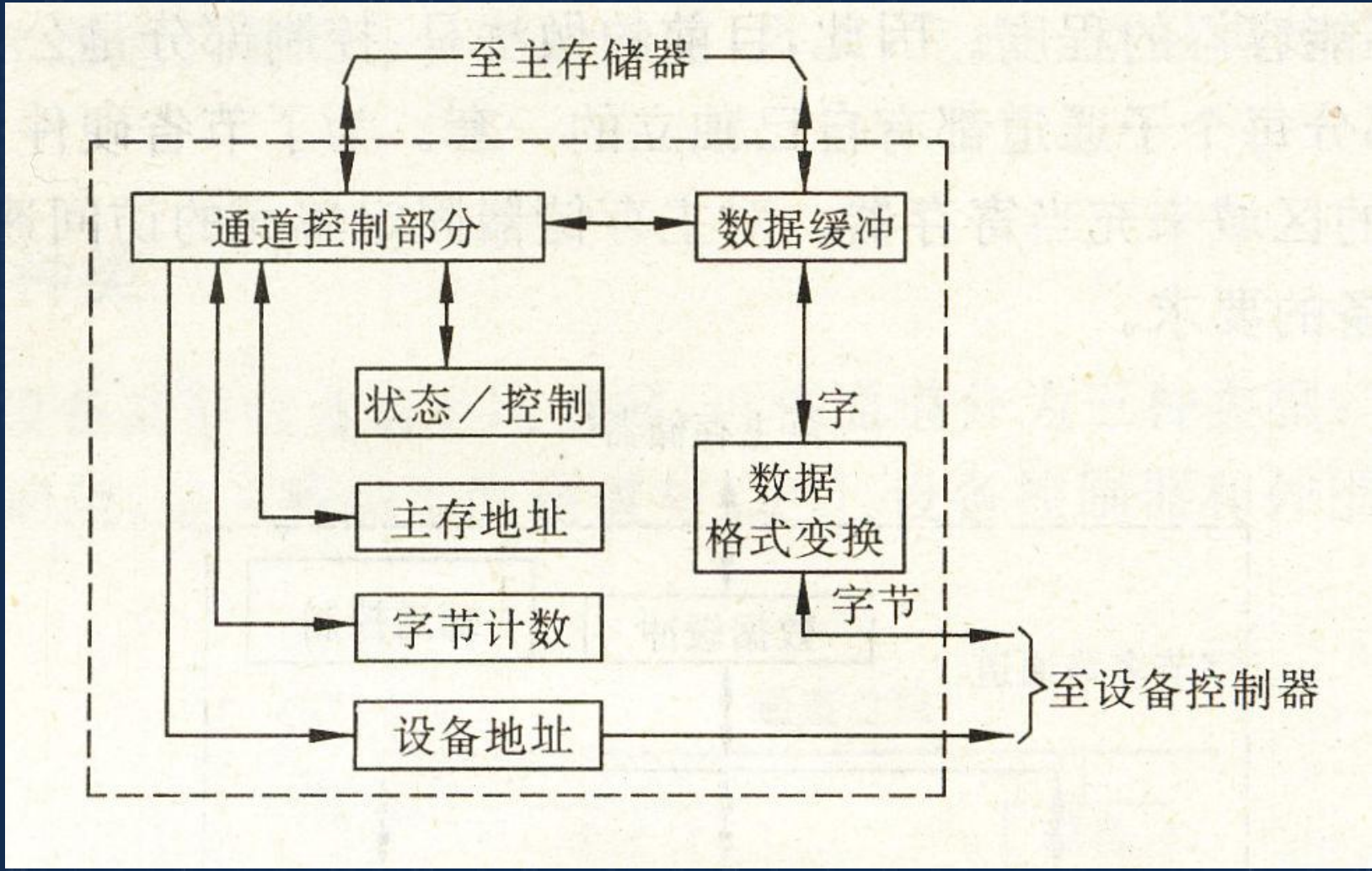


A_i、B_i、C_i 分别为传送的字节信息

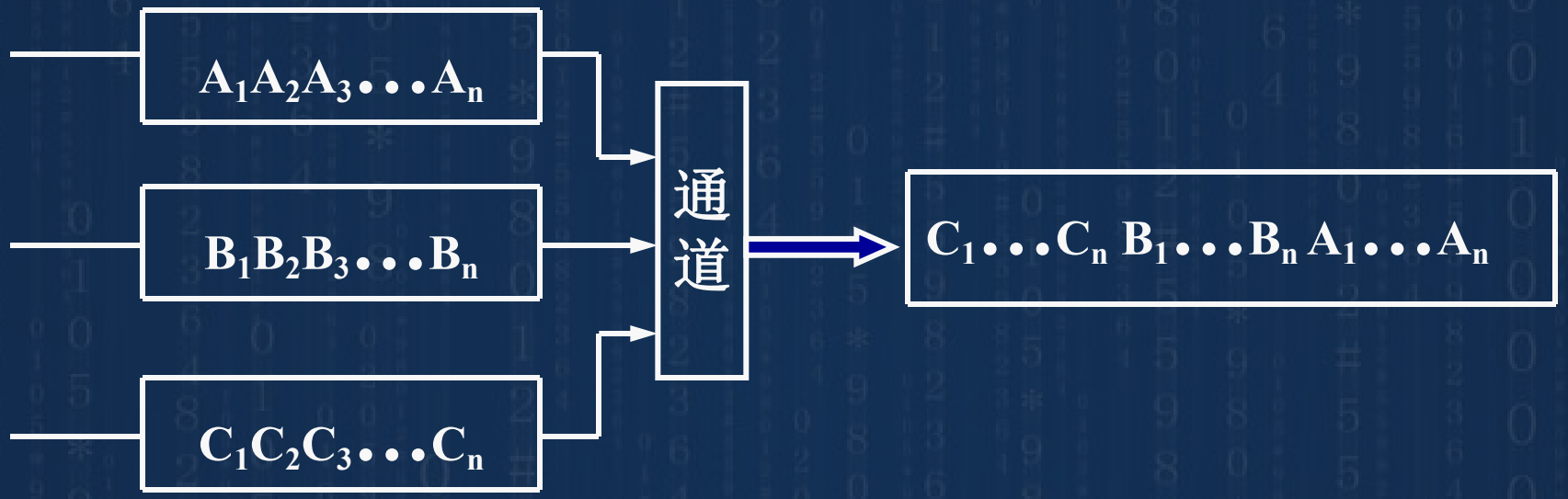
2）选择通道

- 选择通道只有一套完整的硬件，以独占的方式工作，逐个轮流地为物理上连接的几台高速外设服务。
- 选择通道在一段时间内单独为一台外设服务，但在不同的时间内可以选择不同的设备。
- 选择通道一旦选中某一设备，通道就进入“忙”状态直到该设备的数据传输工作全部结束为止。
- 选择通道传送的数据宽度是可变的，它为一台外设传送完数据后才转去处理其他外设。

选择通道的结构



选择通道的信息传送方式



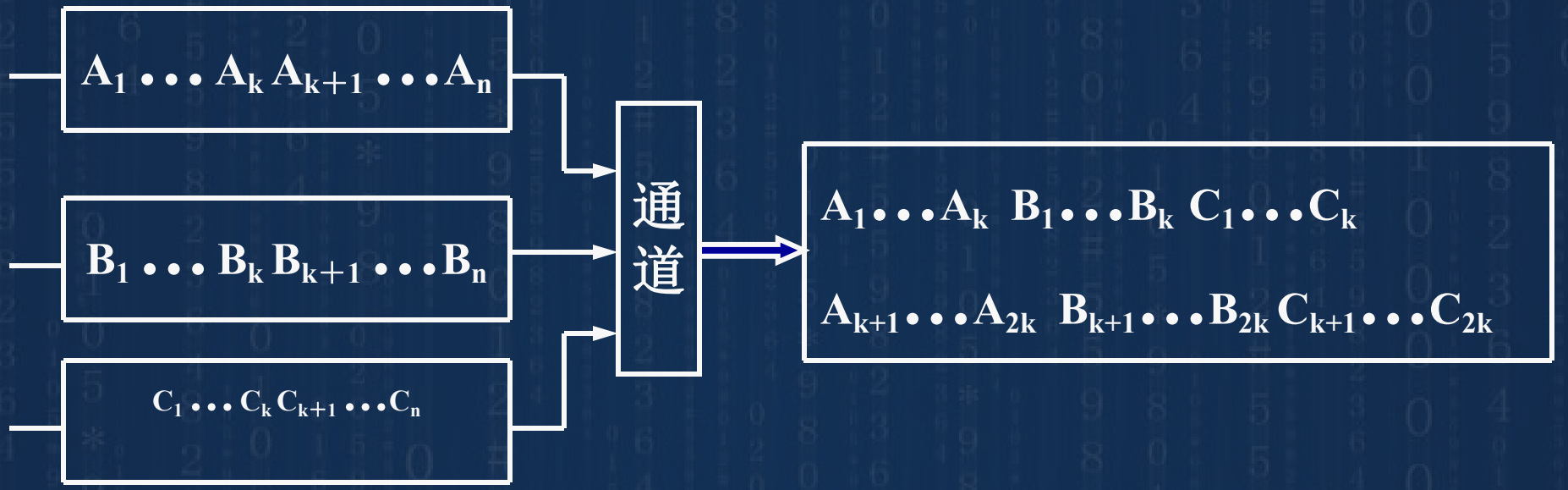
$A_1\cdots A_n$、$B_1\cdots B_n$、$C_1\cdots C_n$ 分别为成组数据

3）数组多路通道

- 数组多路通道将字节多路通道和选择通道的特性结合起来。一个通道可带有多个子通道，各子通道以成组交叉模式轮流使用通道。
- 成组交叉模式：利用通道传送完一组数据（数据块）后让出通道。
- 数组多路通道适用于以数组为单位的高速外设。
- 数组多路通道选择一个高速设备后，先向其发出一个寻找的命令，然后在这个设备寻找期间可以为其他设备服务。在设备寻找完成后再与其真正建立数据连接，并一直维持到一个数据块传输完毕。

- 数组多路通道的数据宽度是定长的，它既保留了选择通道高速传输的优点，又充分利用了控制型操作的时间间隔为其他设备服务，使通道的功能得到有效发挥，因此数组多路通道在实际系统中得到较为广泛的应用。

数组多路通道的信息传送方式



$A_1 \cdots A_k$、$B_1 \cdots B_k$、$C_1 \cdots C_k$ 分别为数据块

字节多路通道和数组多路通道的异同：

- 相同点：
- 都是多路通道，在一段时间内可以交替地执行多个设备的通道程序。
- 不同点：
- ① 数组多路通道允许多个设备同时工作，但只允许一个设备进行传输型操作，其他设备只能进行控制型操作。
- 字节多路通道不仅允许多个设备同时操作，而且允许它们同时进行传输型操作。

- ② 数组多路通道与设备之间的数据传送的基本单位是数据块，通道必须为一个设备传送完一个数据块以后才能为别的设备传送数据块。
- 字节多路通道与设备之间的数据传送基本单位是字节。通道为一个设备传送一个字节之后，又可以为另一个设备传送一个字节，因此各设备与通道之间的数据传送是以字节为单位交替进行的。

3. I/O指令、I/O通道指令与I/O通道程序

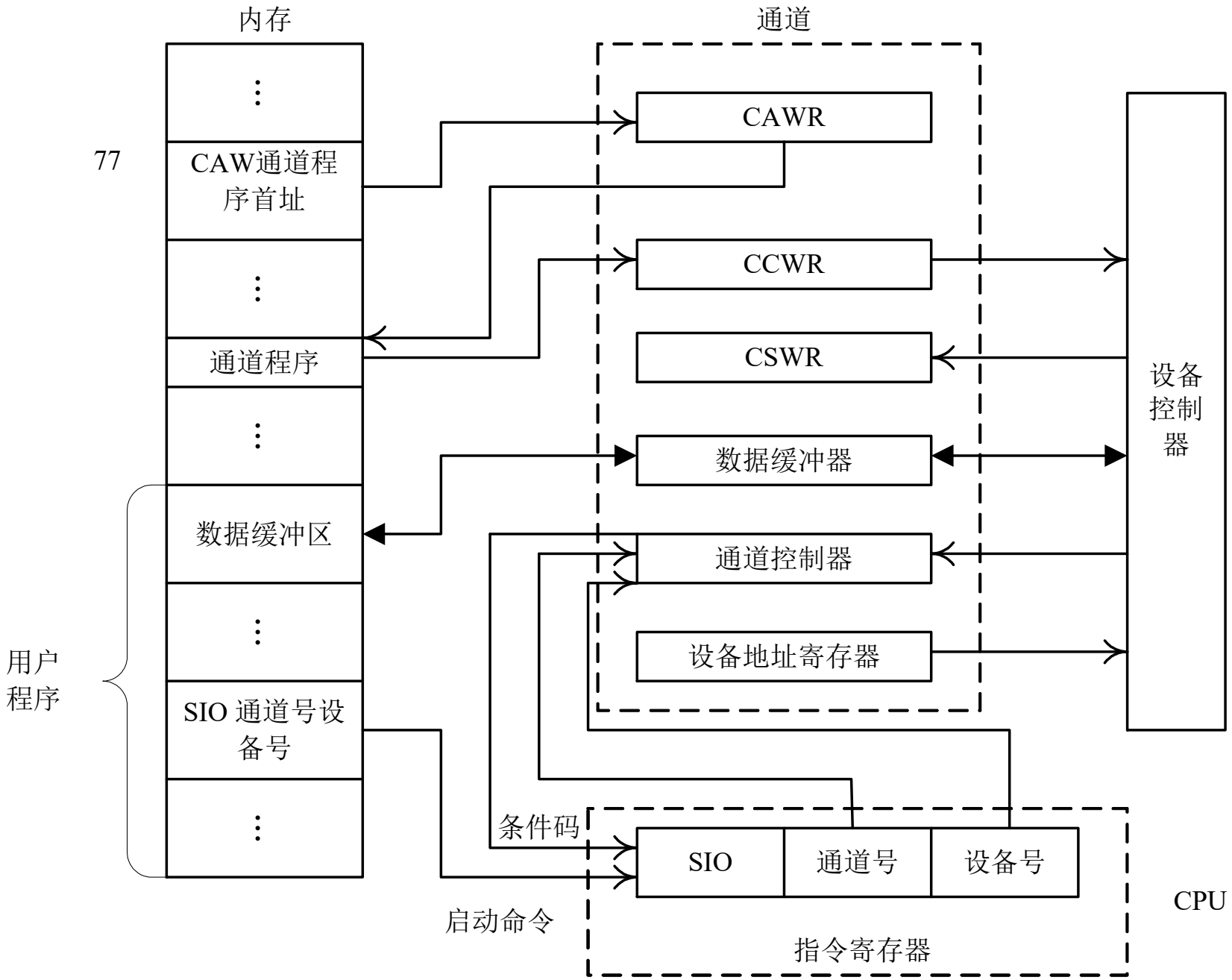
- **1）I/O指令**
- **I/O指令是计算机系统给用户使用的指令系统的一部分。由CPU负责解释执行。**
- **采用通道控制器后，I/O指令不再直接控制I/O数据的具体传送，一般只用于负责启、停I/O通道，查询通道及I/O设备状态，控制I/O通道进行某些操作等。**

- **2）I/O通道指令**
- **I/O通道指令又称为I/O通道控制字（CCW）。 CCW是用于编制I/O通道程序的指令，专供I/O通道解释执行，以实现I/O数据传输等I/O操作。**
- **3）I/O通道程序**
- **用CCW编制成有关的I/O通道程序，在CPU的命令下启动通道程序实现有关I/O操作，完成I/O通道对I/O设备的具体控制。**

- 在早期的I/O通道实现中，I/O通道程序存放在主机的主存中，即I/O通道与CPU共用主存。
- 目前，计算机系统中通常为I/O通道配置了局部存储器。这样可以减少CPU与I/O通道之间的冲突，进一步提高CPU与I/O通道工作的并行度。

4. 通道的组成结构

- 下图一种选择通道的简化组成模型，多路通道的组成基本与其相似。通道位于主机与设备控制器之间，图中示意性地画出了主要信息传送途径，不是实际的逻辑连接线。
  - 1）通道地址字寄存器（CAWR）
  - CAWR存放从主存中读出的通道地址字CAW，指明通道指令所在单元的地址码。对于IBM4300，字长32位，主存按字编址，而通道指令字长64位，每条通道指令占用2个存储单元。启动通道后，由主存固定单元读出CAW（通道程序首地址），每执行一条通道指令，CAWR＋2，指向下一条通道指令。通道中的CAWR类似于CPU中的PC。


内存
通道
77
CAW通道程序首址
通道程序
数据缓冲区
SIO 通道号设备号
用户程序
CAWR
CCWR
CSWR
数据缓冲器
通道控制器
设备地址寄存器
设备控制器
条件码
SIO
通道号
设备号
CPU
启动命令
指令寄存器

- **2）通道指令寄存器（CCWR）**
- 由主存读出的通道指令，存放在通道的CCWR中，据此向设备控制器发出控制命令。CCWR的作用类似于CPU中的指令寄存器IR。由于一条通道指令可以执行若干周期，以实现成组传送，所以每传送一次，需对CCWR中的数据地址与计数值进行±1修改。

- **3）数据缓冲寄存器**
- 当通道申请与主存传送数据时，由于访存冲突的存在，可能等待一段时间才会获得响应，所以通道设有足够大小的数据缓冲寄存器。通道与设备间可能按字节传送，而通道与主存之间则按字（多个字节）传送，因此通道的数据缓冲寄存器还应具有数据的组装与拆分的功能。

- **4）设备地址寄存器**
- **CPU启动通道的I/O指令中包含设备号，它被送入通道的设备地址寄存器。据此向I/O总线送出设备地址，经设备控制器译码产生设备选中信号。**
- **5）通道状态字寄存器（CSWR）**
- **CSWR存放本通道与设备的状态信息，供CPU的TCH指令查询。**

- 6）通道控制器（微命令发生器）
- 通道相当于一个专门执行通道指令的小型的CPU，为此也需要一个微命令发生器来控制通道的操作。它可以采用组合逻辑或微程序方式实现。
- 7）时序系统
- 负责I/O操作中有关的时序控制。

5. 通道的工作过程

- 在具有通道的计算机中，用户程序通常通过调用通道程序来完成数据输入输出的过程。其中CPU执行用户程序和管理程序，通道处理机执行通道程序。
- 在编制通道程序时，应根据I/O设备的需要在主存中开辟相应的输入/输出缓冲区，一般采取多缓冲区技术。

1）用广义指令进入管理程序，CPU组织通道程序，启动通道。

- 广义指令由一条访管指令和若干个参数组成。访管指令的地址码部分实际上就是这条访管指令要调用的管理程序入口地址。
- (1) 当用户程序执行到要求进行I/O操作的访管指令时，产生自愿访管中断请求。CPU响应这个中断请求后，转向管理程序入口。
- (2) 管理程序根据广义指令提供的参数包括设备号、交换长度和主存起始地址等信息来编制通道程序。编制好的通道程序放在主存中与这个通道相对应的通道程序缓冲区中。

- (3) 管理程序把通道程序的入口地址置入主存储器中的通道地址单元（例如，IBM4300约定为77号单元）。
- (4) 在管理程序的最后，用一条启动I/O设备指令SIO启动通道开始工作。

2） 通道进入设备选择阶段

- (1) 根据启动I/O指令中给出的通道和设备信息，选择通道和设备。

- (2) 如果指定的通道和子通道是在线且空闲的，就从主存中取出通道地址字CAW，按照CAW给出的通道程序起始地址从主存的通道缓冲区中取出第一条通道指令。

- (3) 选择指定的设备控制器和设备。判断被选择的设备是否在线（接通且空闲）。
- 如果判断被选择的设备接通且空闲，就向它发启动命令。设备被启动后，如果启动正常，就将向通道发回“启动正常”回答信号，表示这台设备已经接受并执行了启动命令，完成设备的启动过程。

- (4) 通道开始执行通道程序，进入信息传送阶段，同时管理程序返回用户程序继续执行。

3）信息传送阶段

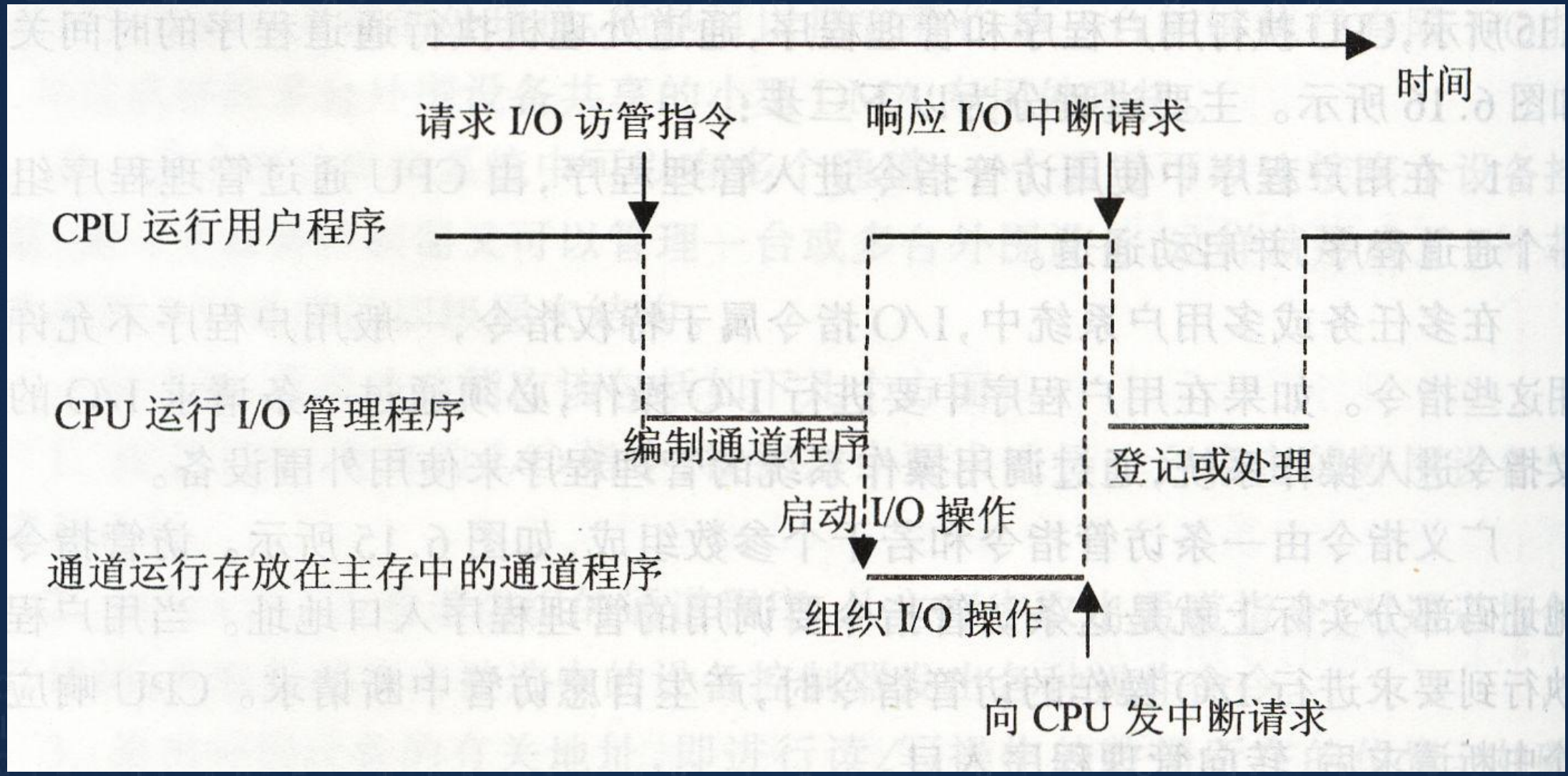
- (1) 通道执行通道程序，控制主存—通道—设备之间的信息传送，直至完成指定的I/O工作。
- (2) 当通道执行完通道程序的最后一条通道指令“断开通道”时，通道的数据传输工作就全部结束，进入通道信息传送结束阶段。
- (3) 通道程序执行完后，向用户程序发出中断请求，进行数据传送结束的处理。

4）信息传送结束阶段

- (1) 执行完 “断开通道”指令后，通道向CPU发中断请求。CPU响应中断请求后，再次进入操作系统，调用管理程序对I/O中断请求进行处理。
- 如果是正常结束，则管理程序进行必要的登记等工作后关闭通道，等待下一次通道传输。
- 如果是故障、错误等异常情况，则进行异常处理。
- (2) CPU返回用户程序继续执行。

4）信息传送结束阶段

- 如果执行数据传送指令，则每传送一次数据，相应修改通道指令字CCW中的数据地址与计数值。当计数值为0时，表明本次数据传送完毕。
- 如果所执行的通道指令中数据链标志CD与命令链标志CC均为0，表明这是本通道程序中的最后一条。在执行完这条通道指令后，结束通道程序。通道程序执行结束后，通道一方面向设备发出结束命令，一方面向CPU申请中断，并将通道状态字CSW写入主存某指定单元中，供中断处理程序分析，作结束处理。


时间
请求 I/O 访管指令
响应 I/O 中断请求
CPU 运行用户程序
CPU 运行 I/O 管理程序
编制通道程序
登记或处理
启动 I/O 操作
通道运行存放在主存中的通道程序
组织 I/O 操作
向 CPU 发中断请求

- 在通道与设备之间的数据传送过程中的设备选择工作
- ① 如果一个通道只管理一台高速设备，完成一次数据传送过程只需要做一次设备选择工作。
- ② 如果在同一个通道中有多台设备同时工作则要反复重新选择设备，即找出当前要传送数据的是哪一台设备。
- 对于低速设备，每传送完一字节就要重新选择设备；对于高速设备，通常每传送完一个数据块后需要重新选择设备。

- 采用通道方式后，每完成一次I/O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。同时当系统中有多个通道同时工作时，CPU可以与多种不同类型、不同工作速度的外围设备充分地并行工作。

*例子：IBM370为例

- （1）通道指令（通道控制字 CCW ）

- $CD_{32}=1$ （数据链）
- 两邻命令码同，主存区域不同的通道指令链接为通道程序
- $CC_{33}=1$ （控制链）
- 两邻命令码不同的通道指令链接为通道程序

- 命令码： 通道指令完成的操作，如读、写、转移、控制等
- 数据地址： 数据存放的主存首地址
- 计数值： 数据的传送量
- CC、CD链的组合：
- （2）通道程序
- 由CC或CD链连接起来
- 的通道指令的有序集合。

CC	CD	说　　明
0	0	表示为通道程序的最后一条指令
0	1	与后续命令同、数据不同 （数据链）
1	0	与后续命令不同（控制链）
1	1	禁止 （不存在）

（3）通道程序举例

- 例1. Read 400000H CD=0 CC=0 512
- (读设备）

- 例2. Read 400000H CD=1 CC=0 512
- Read 600000H CD=0 CC=0 1024

- 例3. Read 400000H CD=0 CC=1 512
- Search CD=0 CC=1
- Write 8A0000H CD=1 CC=0 256
- Write AC0000H CD=0 CC=0 512

6. 通道程序流程（大致有6个步骤）

- （1）取通道指令
- （通道进入数据传送期间）
- 从主存通道地址字单元（IBM370为72#）取出
  地址字（CAW）-→CAWR=→MAR
  =→取出CCW=→CCWR
- （2）置参数
- CAWR ←= CAWR+8 （准备取下一条指令）
  CAR ←= CCWR （数据地址段）
  WCNT ←= CCWR （计数值段）
  通道标志寄存器←= CC、CD等特征位

- （3）数据传送 （以 I/O➔RAM为例）
- ①读外设数据➔拆卸与装配➔数据缓冲➔MDR
- ②CAR（数据地址）➔MAR
- ③Write RAM
- ④修改参数： CAR+4 、WCNT-4 （一次传4B）
- （4）WCNT=0？否， 重复（3）
- （5）CD=0&&CC=0？
- 否，转（1），取下一条通道指令
- （6）结束处理：（CD=0&&CC=0, 触发中断）
- 结束中断➔中断寄存器，向CPU发中断请求，
- CPU转入执行中断服务程序。

本章小结

- 1. 总线技术及仲裁方法
- 2. 一些常用总线
- 3. I/O系统与CPU的交互方法
- 4. 中断系统及处理顺序
- 5. DMA传送方法
- 6. 通道方式