操作系统
第二十一讲

马佳曼

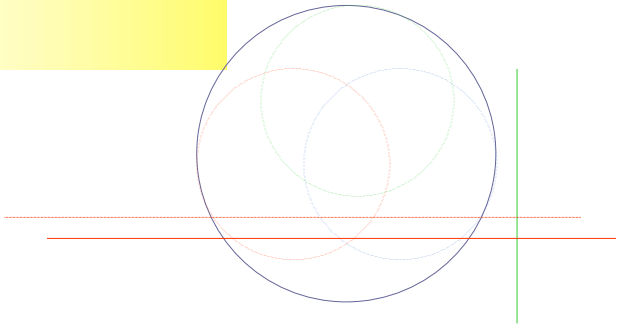
Jiaman.ma@nwpu.edu.cn

第六章 输入/输出系统

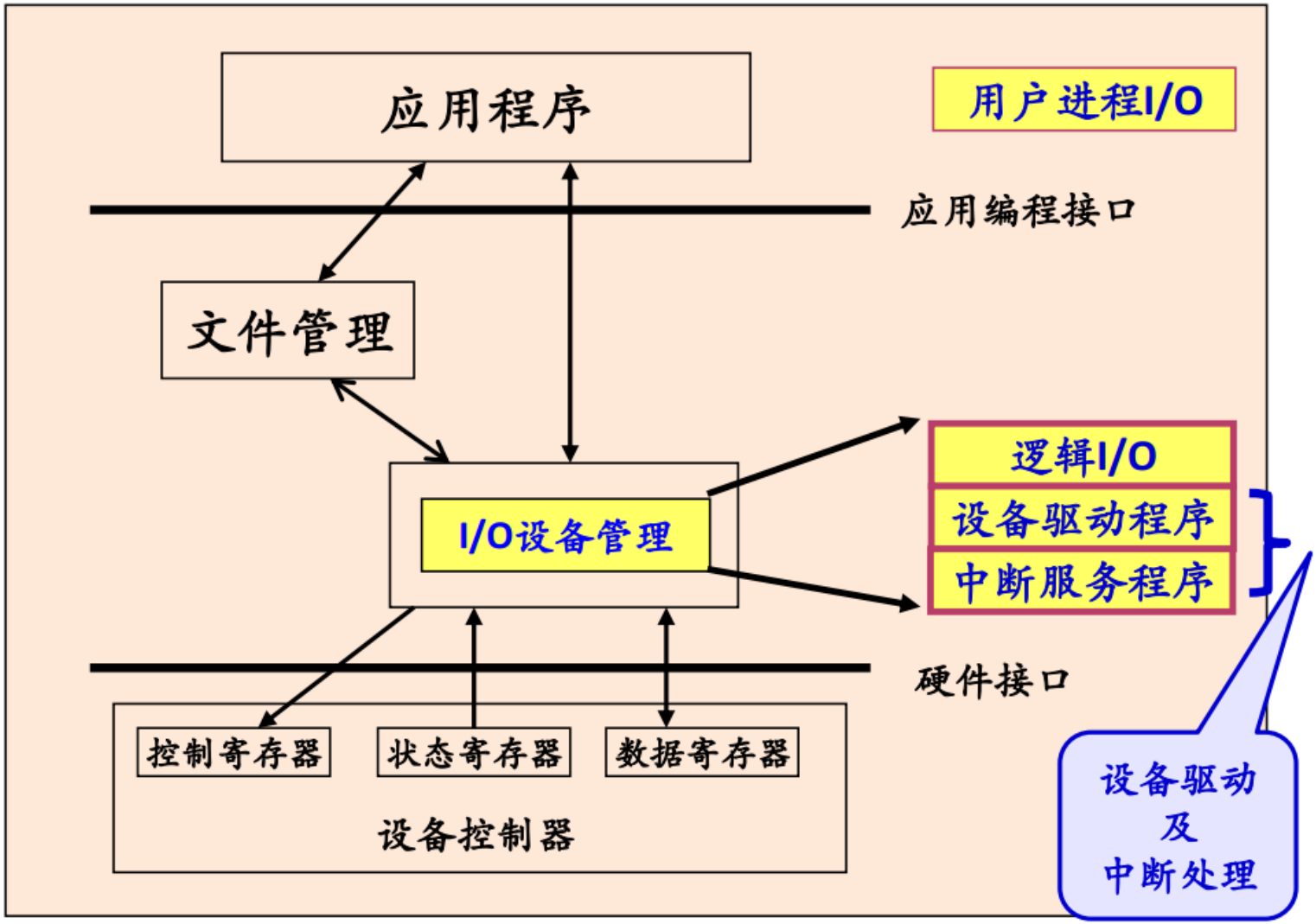
概述

I/O系统硬件特点

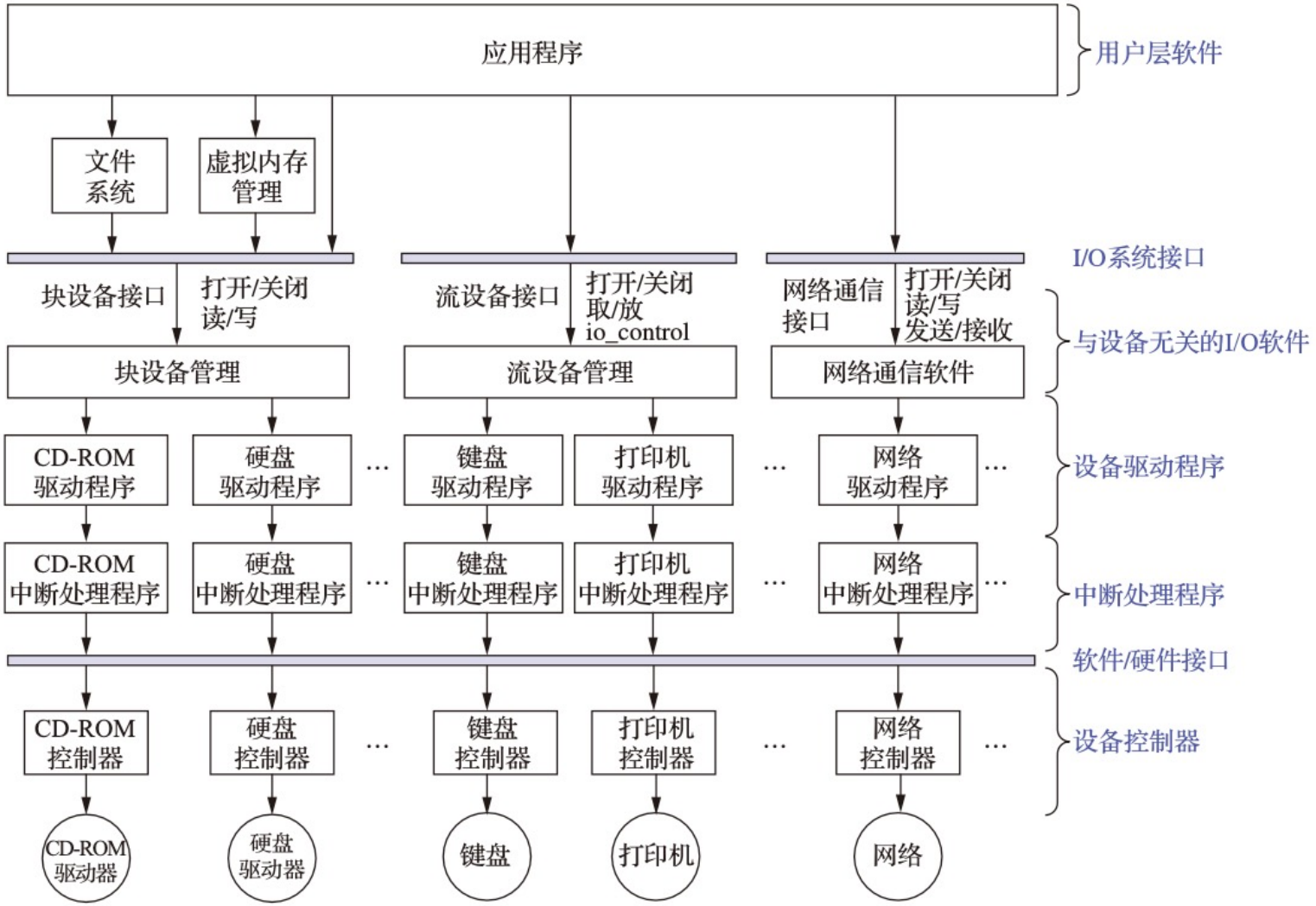
I/O控制方式



I/O管理示意图


应用程序
用户进程I/O
应用编程接口
文件管理
逻辑I/O
设备驱动程序
中断服务程序
I/O设备管理
硬件接口
控制寄存器
状态寄存器
数据寄存器
设备控制器
设备驱动
及
中断处理

I/O管理示意图


应用程序
用户层软件
文件系统
虚拟内存管理
I/O系统接口
块设备接口
打开/关闭
读/写
流设备接口
打开/关闭
取/放
io_control
网络通信接口
打开/关闭
读/写
发送/接收
与设备无关的I/O软件
块设备管理
流设备管理
网络通信软件
CD-ROM驱动程序
硬盘驱动程序
键盘驱动程序
打印机驱动程序
网络驱动程序
设备驱动程序
CD-ROM中断处理程序
硬盘中断处理程序
键盘中断处理程序
打印机中断处理程序
网络中断处理程序
中断处理程序
软件/硬件接口
CD-ROM控制器
硬盘控制器
键盘控制器
打印机控制器
网络控制器
设备控制器
CD-ROM驱动器
硬盘驱动器
键盘
打印机
网络

6.1 概述

- 设备的分类
- 设备管理的目标
- 设备管理的功能
- 设备管理数据结构

6.1.1 设备的分类

1、按传输速率分

- 低速设备
- 中速设备
- 高速设备

2、按信息交换的单位分类

- 字符设备：速率较低、中断驱动。以字符为单位存储、传输信息 传输速率低、不可寻址
- 块设备：速率高（几兆）、可随机访问任一块、DMA方式驱动。以数据块为单位存储、传输信息传输速率较高、可寻址（随机读写）

3、按资源管理方式分类

- 独占型设备。在一段时间内只能有一个进程使用的设备，一般为低速I/O设备（如打印机，磁带等）
- 共享型设备。在一段时间内可有多个进程共同使用的设备，多个进程以交叉的方式来使用设备，其资源利用率高（如硬盘）
- 虚拟设备。在一类设备上模拟另一类设备，常用独占设备模拟共享设备，用高速设备模拟低速设备，被模拟的设备称为虚设备

4、按外部设备的从属关系分

- 系统设备
- 用户设备

I/O的特点

- I/O性能经常成为系统性能的瓶颈
  - CPU性能不等于系统性能
  - CPU性能越高，与I/O差距越大
  - 进程切换多，系统开销大
- 操作系统庞大复杂：资源多、杂，并发，均来自I/O
- 与其他功能联系密切，特别是文件系统

6.1.2 设备管理的目标

1、实现设备独立性

2、提高设备利用率

3、设备的统一管理

- 速度
- 传递单位
- 操作方法和特性
- 出错条件

6.1.3 设备管理的功能

1. 能够隐藏 I/O 设备的细节
2. 能够保证设备无关性
3. 能够对 I/O 设备进行控制
4. 能够提高处理机和 I/O 设备的利用率
5. 能够确保对设备的正确共享
6. 能够处理错误

性能
CPU与I/O的速度差别大
→减少由于速度差异造成的整体性能开销
→尽量使两者交叠运行

6.1.4 设备管理数据结构

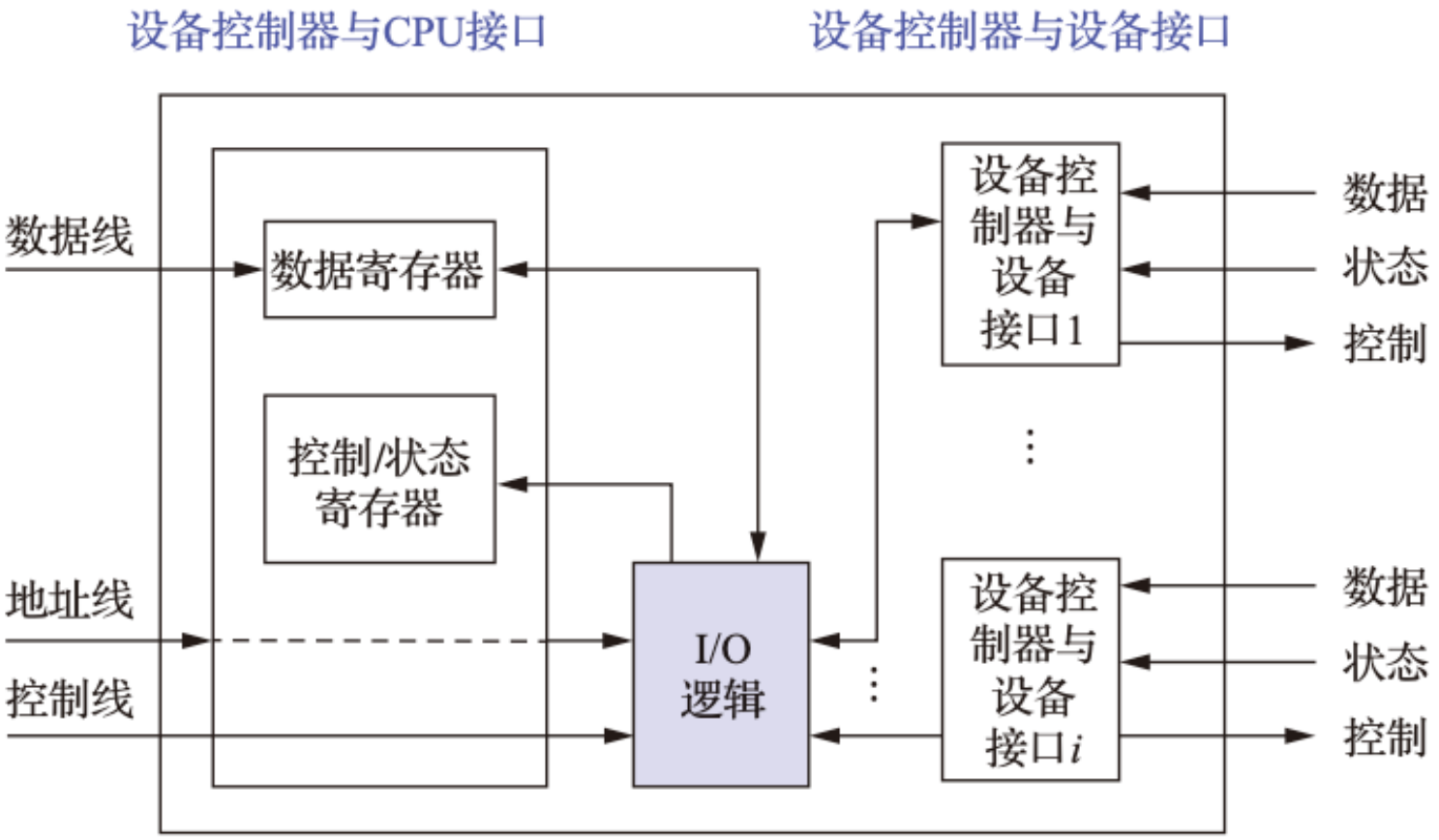
- 设备控制块(DCB)
  - 设备标识符
  - 设备属性
  - 设备I/O总线地址
  - 设备状态
  - 等待队列指针

DCB的结构

6.2 I/O系统硬件特点

- 设备组成
- （1）物理部分：物理设备机械部分，设备本身
- （2）电子部分：设备控制器
- 电子部分（设备控制器）完成的工作
  - 接收和识别命令
  - 数据交换
  - 标志和报告设备的状态
  - 地址识别
  - 数据缓冲区
  - 差错控制

6.2 设备控制器的组成

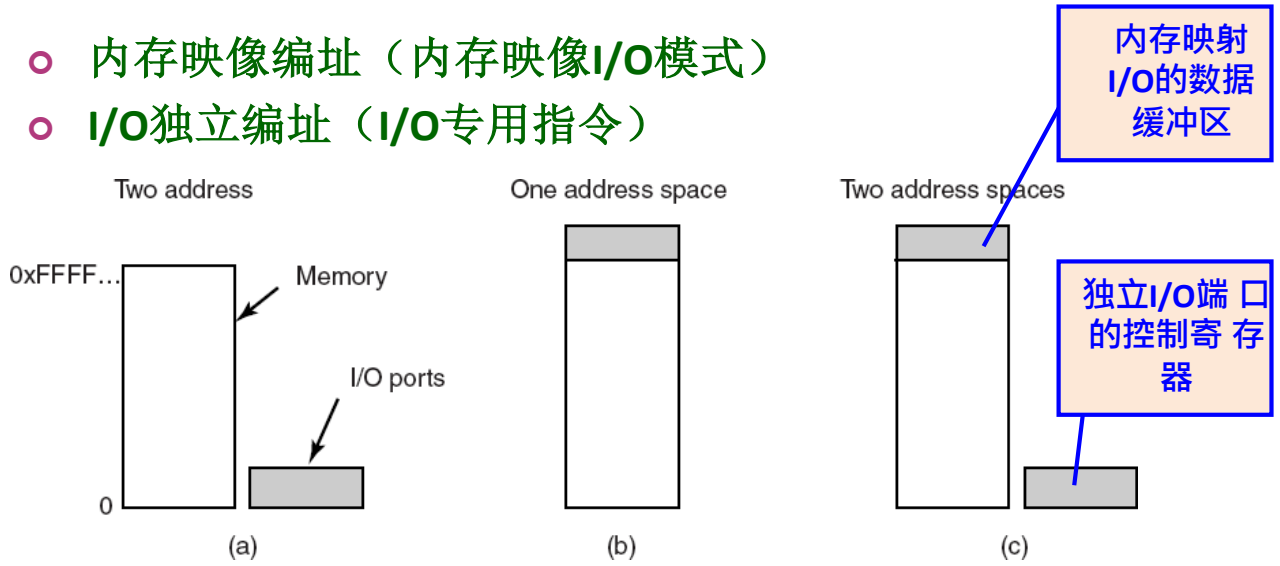

设备控制器与CPU接口
设备控制器与设备接口
数据线
数据寄存器
控制/状态
寄存器
地址线
控制线
I/O
逻辑
设备控
制器与
设备
接口1
设备控
制器与
设备
接口i
数据
状态
控制
数据
状态
控制

6.2.2 端口编址方法

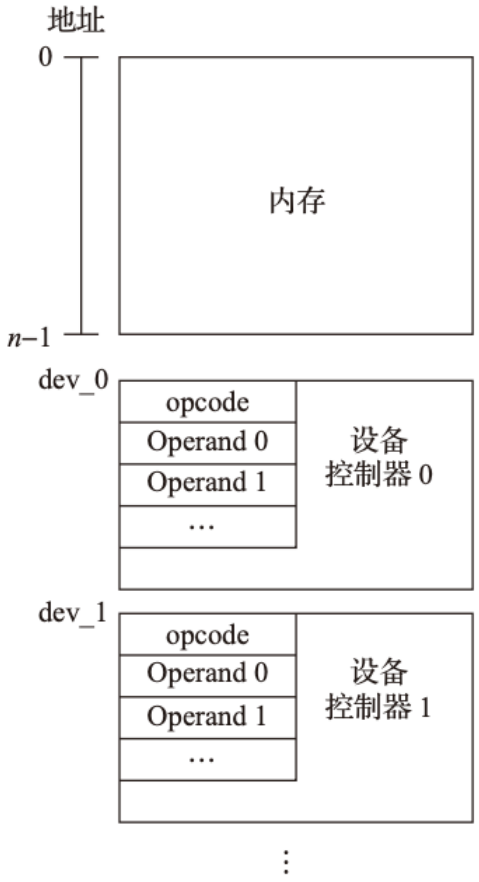
- I/O端口地址：接口电路中每个寄存器具有的、唯一的地址，是个整数
- 所有I/O端口地址形成I/O端口空间(受到保护)

I/O指令形式与I/O地址是相互关联的， 主要有两种形式：

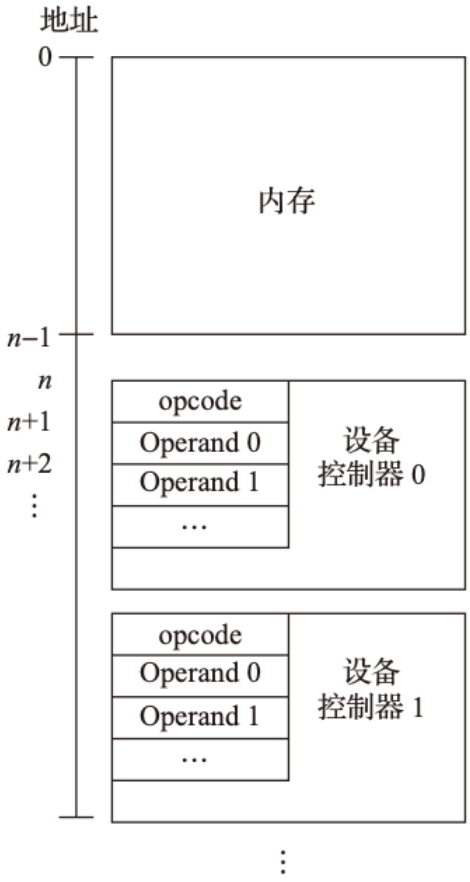
- 内存映像编址（内存映像I/O模式）
- I/O独立编址（I/O专用指令）



6.2.2 端口编址方法



（a）采用特定I/O指令形式



（b）采用内存映像I/O形式

6.2.2 端口编址方法

IO独立编址

- 分配给系统中所有端口的地址空间完全独立，与内存地址空间无关
- 使用专门的I/O指令对端口进行操作
- 优点
  - 外设不占用内存的地址空间
  - 编程时，易于区分是对内存操作还是对I/O端口操作
- 缺点：I/O端口操作的指令类型少，操作不灵活
- 例子：8086/8088，分配给I/O端口的地址空间64K，0000H~0FFFFH，只能用in和out指令进行读写操作

I/O地址范围	设备
000 – 00F	DMA控制器
020 – 021	中断控制器
040 – 043	定时器
060 – 063	键盘
200 – 20F	游戏控制器
2F8 – 2FF	辅助串口
320 – 32F	硬盘控制器
378 – 37F	并口
3D0 – 3DF	图像控制器
3F0 – 3F7	磁盘驱动控制器
3F8 – 3FF	主串口

6.2.2 端口编址方法

- 内存映像编址（内存映像I/O模式）：分配给系统中所有端口的地址空间与内存的地址空间统一编址
  - 优点
    - 凡是可对存储器操作的指令都可对I/O端口操作
    - 不需要专门的I/O指令
    - I/O端口可占有较大的地址空间
  - 缺点：占用内存空间
- I/O独立编址（I/O专用指令）：分配给系统中所有端口的地址空间完全独立，主机使用专门的I/O指令对端口进行操作
  - 优点
    - 外部设备不占用内存的地址空间
    - 程序设计时易于区分是对内存操作还是对I/O端口操作
  - 缺点：对I/O端口操作的指令类型少，操作不灵活

6.3 I/O控制方式

- （1）程序直接控制方式(循环查询I/O方式)

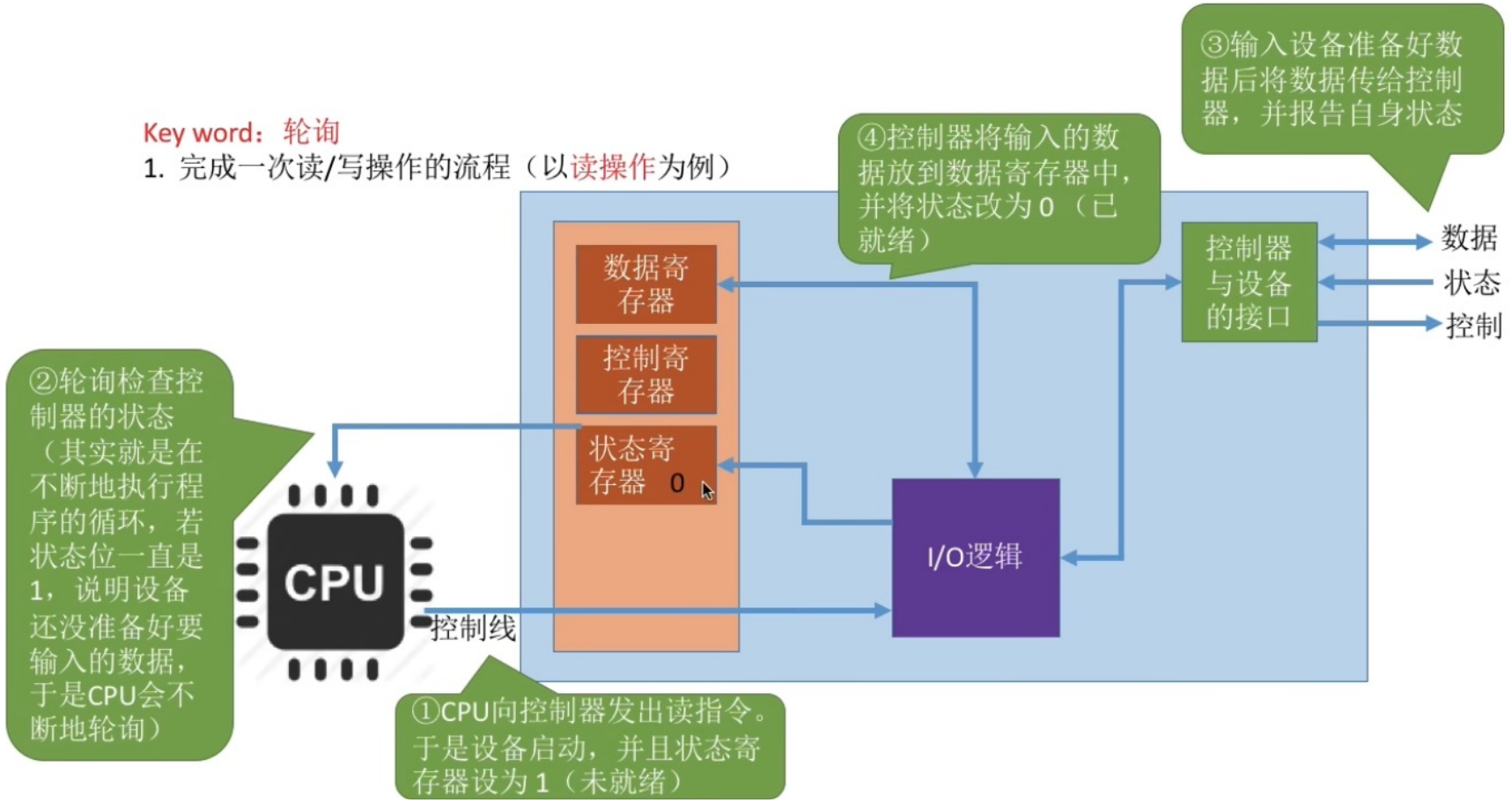
由CPU代表进程给I/O模块发I/O命令，进程进入忙等待，直到操作完成才继续执行

- （2）I/O中断方式

为了减少设备驱动程序不断地询问控制器状态寄存器的开销，I/O操作结束后，由设备控制器主动通知设备驱动程序

- （3）直接存储器存取方式(direct memory access，DMA)
- （4）I/O通道控制方式(I/O channel control)
- （5）外围处理机输入输出方式(peripheral processor unit)

6.3.1 循环查询I/O方式



浪费大量CPU时间

1. 完成一次读/写操作的流程（见右图，Key word：轮询）

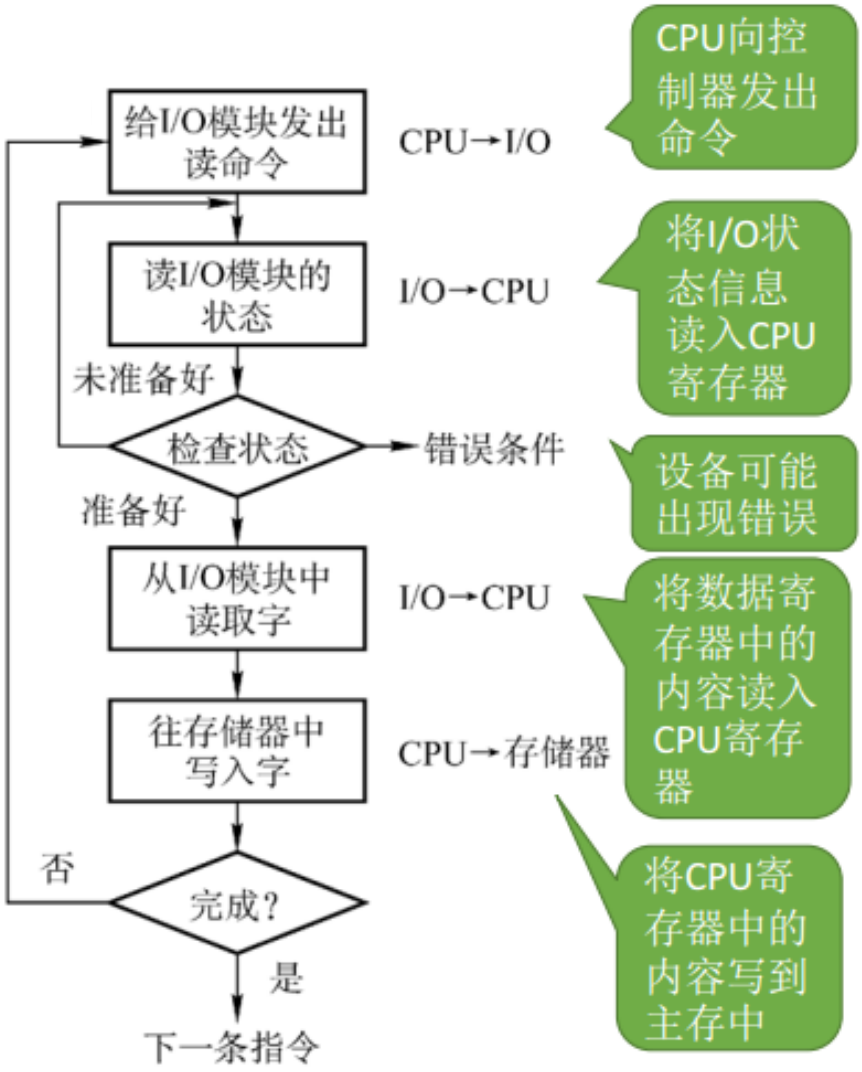
2. CPU干预的频率
很频繁，I/O操作开始之前、完成之后需要CPU介入，并且在等待I/O完成的过程中CPU需要不断地轮询检查。

3. 数据传送的单位
每次读/写一个字

指的是CPU的寄存器

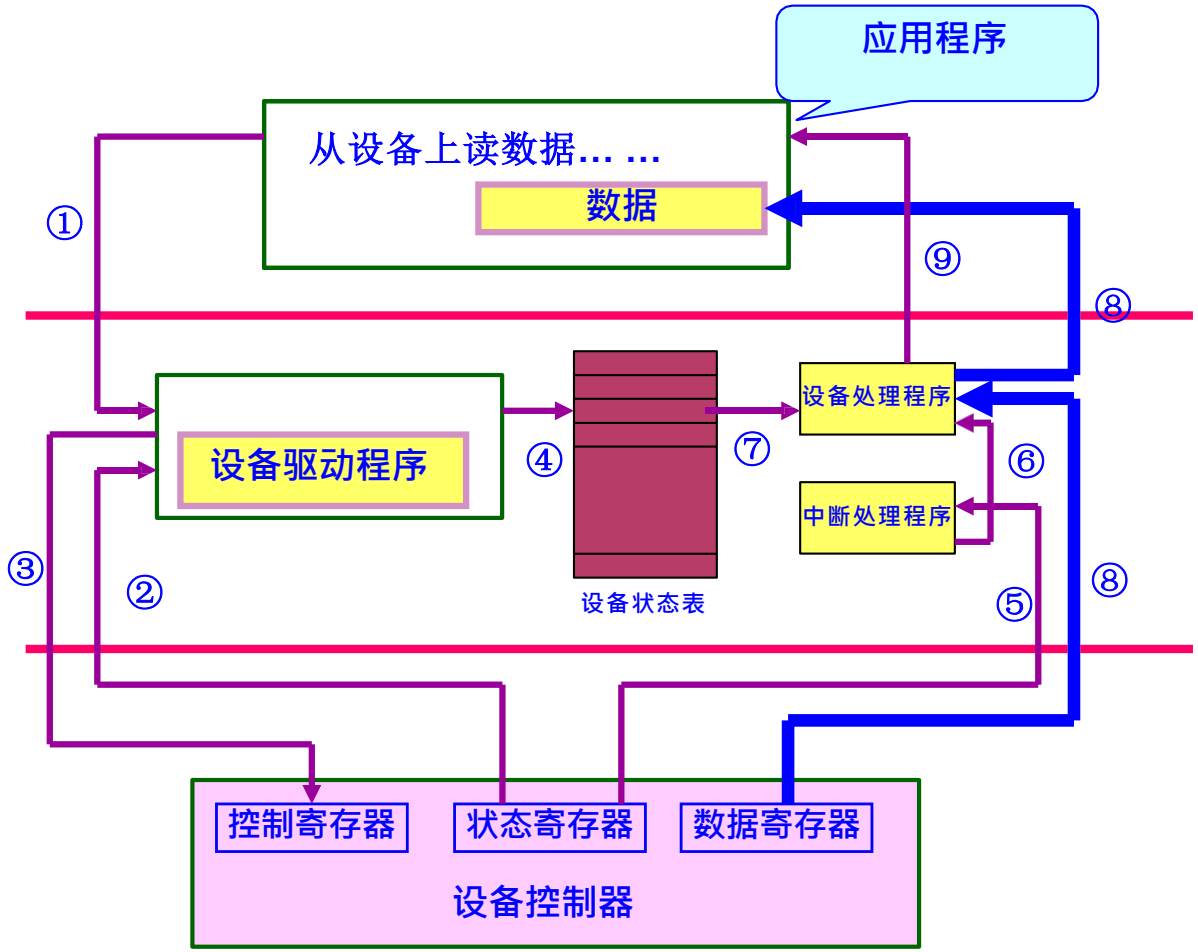
4. 数据的流向
读操作（数据输入）：I/O设备→CPU→内存
写操作（数据输出）：内存→CPU→I/O设备
每个字的读/写都需要CPU的帮助

5. 主要缺点和主要优点
优点：实现简单。在读/写指令之后，加上实现循环检查的一系列指令即可（因此才称为“程序直接控制方式”）
缺点：CPU和I/O设备只能串行工作，CPU需要一直轮询检查，长期处于“忙等”状态，CPU利用率低。



(a) 程序直接控制方式

6.3.2 I/O中断方式


应用程序
从设备上读数据... ...
数据
①
⑨
⑧
设备驱动程序
④
⑦
设备处理程序
⑥
中断处理程序
③
②
⑧
⑤
设备状态表
控制寄存器
状态寄存器
数据寄存器
设备控制器

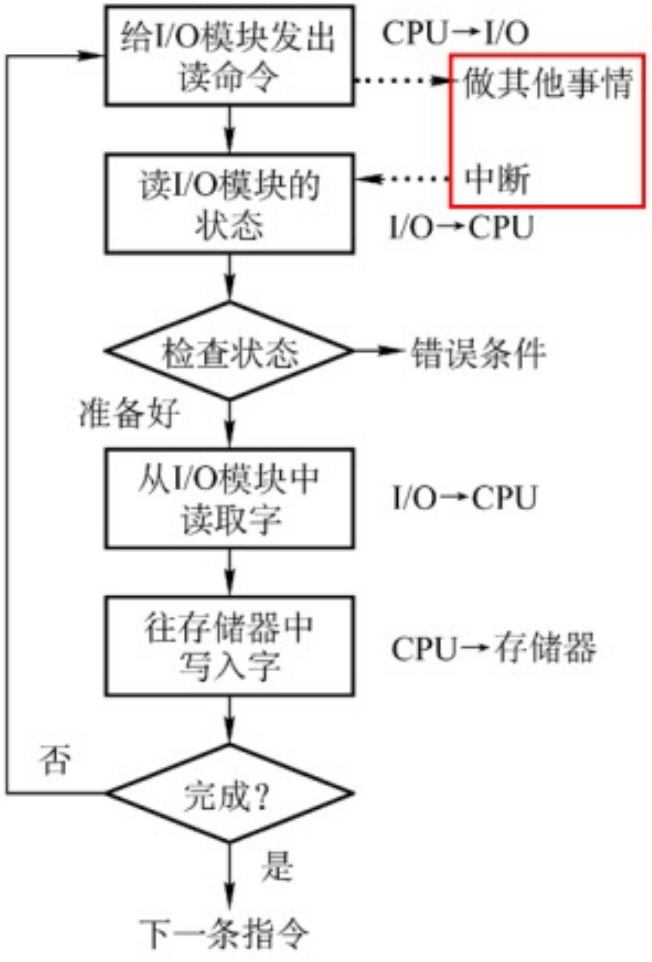
1. 完成一次读/写操作的流程（见右图，Key word：中断）

2. CPU干预的频率
每次I/O操作开始之前、完成之后需要CPU介入。
等待I/O完成的过程中CPU可以切换到别的进程执行。

3. 数据传送的单位
每次读/写一个字

4. 数据的流向
读操作（数据输入）：I/O设备→CPU→内存
写操作（数据输出）：内存→CPU→I/O设备

5. 主要缺点和主要优点
优点：与“程序直接控制方式”相比，在“中断驱动方式”中，I/O控制器会通过中断信号主动报告I/O已完成，CPU不再需要不停地轮询。
CPU和I/O设备可并行工作，CPU利用率得到明显提升。
缺点：每个字在I/O设备与内存之间的传输，都需要经过CPU。而频繁的中断处理会消耗较多的CPU时间。

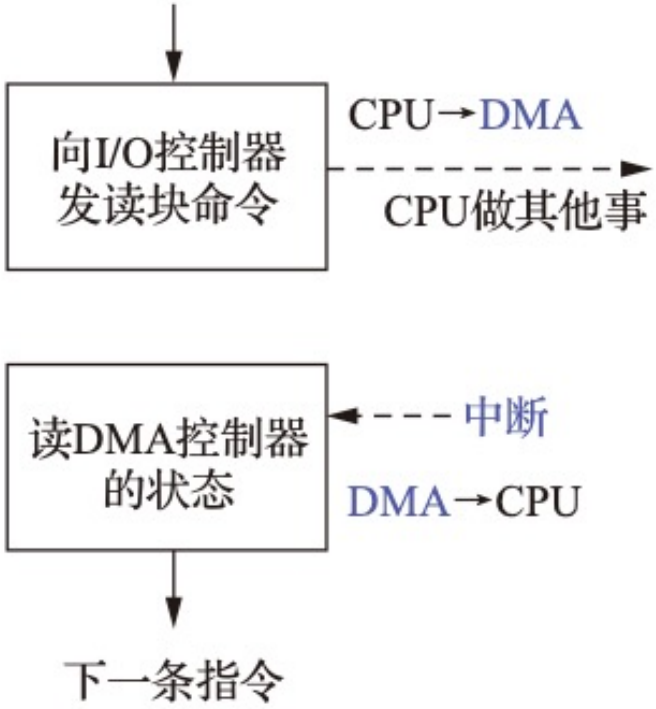


(b) 中断驱动方式

6.3.3 DMA方式

与中驱动方式对比，DMA 改进点：

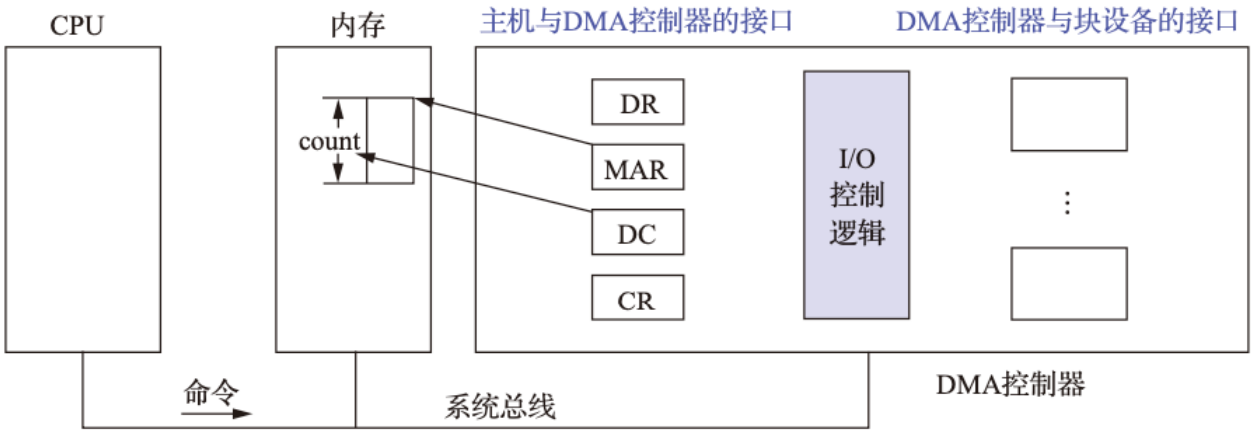
1. 数据的传送单位是块；
2. 数据的流向从设备直接放进内存，货从内存直接到设备，不需要 CPU 中转
3. 仅在传送一个货多个数据块的开始和结束是，才需要 CPU 干预



CPU指明此次要进行的操作（如：读操作），并说明要读入多少数据、数据要存放在内存的什么位置、数据在外部设备上的地址（如：在磁盘上的地址）

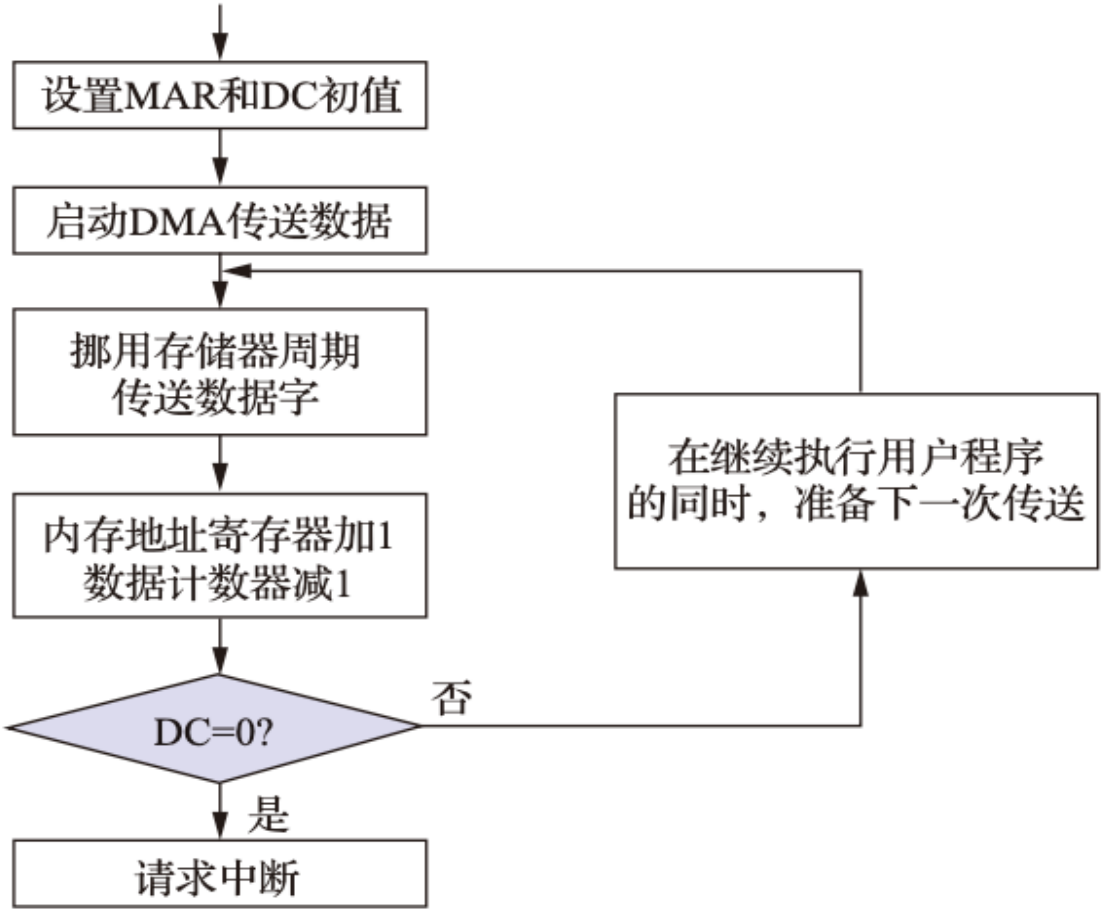
控制器会根据CPU提出的要求完成数据的读/写工作，整块数据的传输完成后，才向CPU发出中断信号

6.3.3 DMA方式

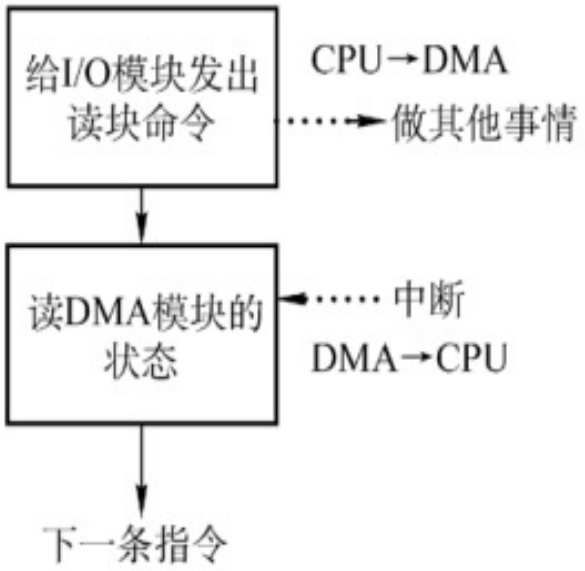


1. 命令寄存器(CR)：用于接收从CPU发来的I/O命令;
2. 内存地址寄存器(MAR)：输入时， 它存放把数据从设备传送到内存的起始目标地址，输出时，它存放由内存到设备的内存源地址;
3. 数据寄存器(DR)：用于暂存从设备到内存或从内存到设备的数据;
4. 数据计数器(DC)：用于存放本次CPU要读或写的字(节)数。

6.3.3 DMA方式


设置MAR和DC初值
启动DMA传送数据
挪用存储器周期
传送数据字
内存地址寄存器加1
数据计数器减1
DC=0?
否
是
请求中断
在继续执行用户程序
的同时，准备下一次传送

1. 完成一次读/写操作的流程（见右图）



2. CPU干预的频率
仅在传送一个或多个数据块的开始和结束时，才需要CPU干预。

3. 数据传送的单位
每次读/写一个或多个块（注意：每次读写的只能是连续的多个块，且这些块读入内存后在内存中也必须是连续的）

4. 数据的流向（不再需要经过CPU）
读操作（数据输入）：I/O设备→内存
写操作（数据输出）：内存→I/O设备

5. 主要缺点和主要优点
优点：数据传输以“块”为单位，CPU介入频率进一步降低。数据的传输不再需要先经过CPU再写入内存，数据传输效率进一步增加。CPU和I/O设备的并行性得到提升。
缺点：CPU每发出一条I/O指令，只能读/写一个或多个连续的数据块。
如果要读/写多个离散存储的数据块，或者要将数据分别写到不同的内存区域时，CPU要分别发出多条I/O指令，进行多次中断处理才能完成。

DMA方式与中断的主要区别

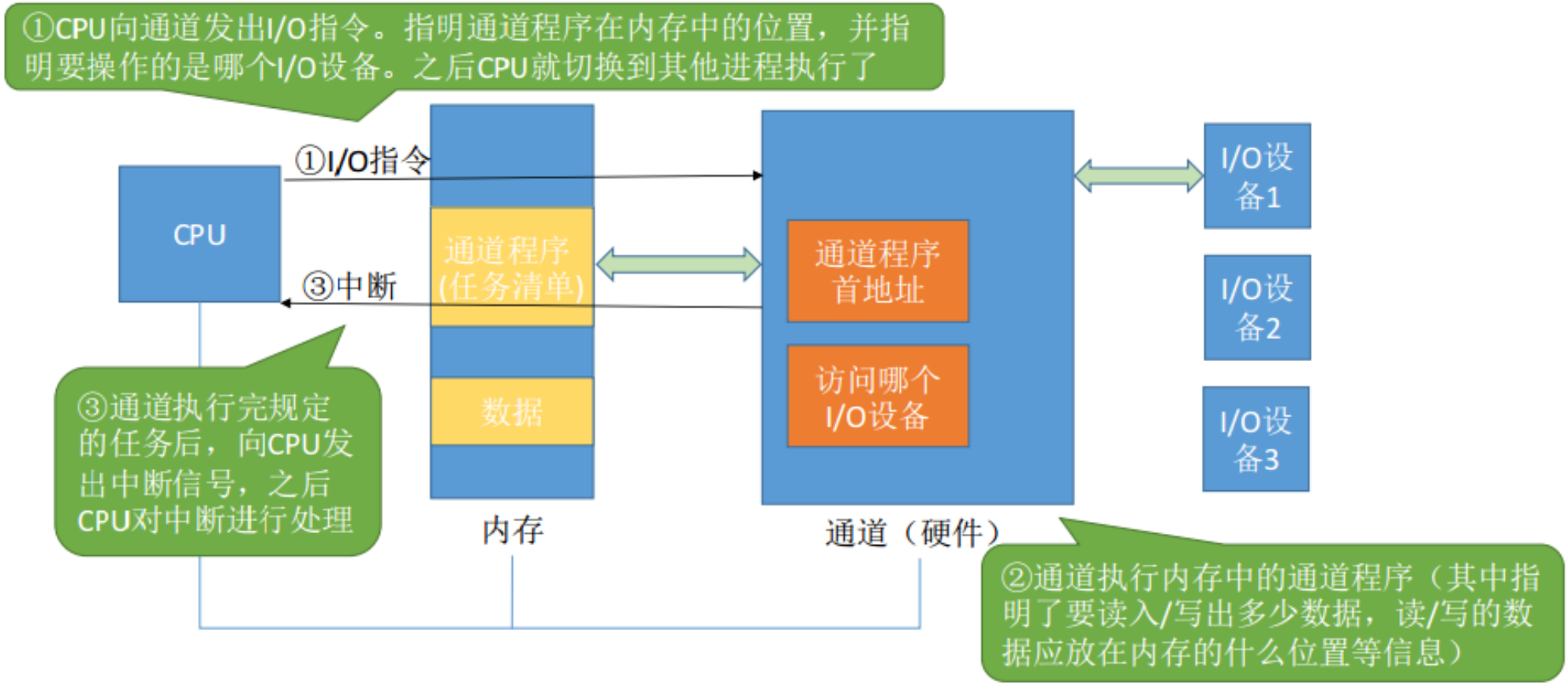
- 中断时机
  - 中断方式是在数据缓冲寄存器满后，发中断请求，CPU进行中断处理
  - DMA方式则是在所要求传送的数据块全部传送结束时要求CPU进行中断处理
- 数据传输
  - 中断方式的数据传送由CPU控制完成
  - DMA方式是在DMA控制器的控制下不经过CPU控制完成的

6.3.4 通道方式

- 通道：在CPU的控制下独立地执行通道程序，对外部设备的I/O操作进行控制，以实现内存与外设之间成批的数据交换。

- 通道程序是由通道指令组成，一个通道可以分时的方式执行几道程序。每道程序控制一台外部设备，因此每道通道程序称为子通道。

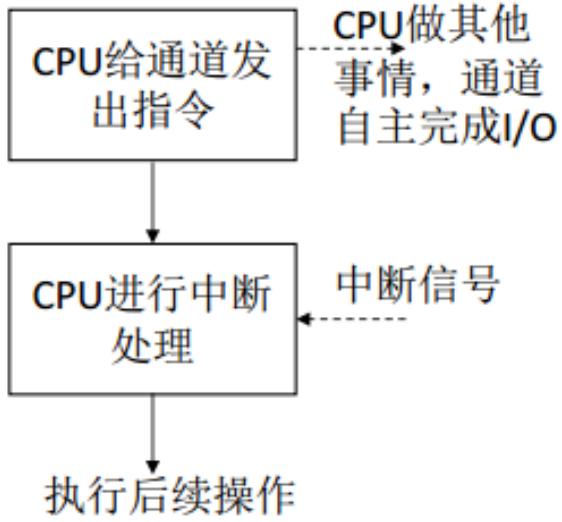
通道：一种硬件，可以理解为是“弱鸡版的CPU”。通道可以识别并执行一系列通道指令



通道：一种硬件，可以理解为是“弱鸡版的CPU”。通道可以识别并执行一系列通道指令

与CPU相比，通道可以执行的指令很单一，并且通道程序是放在主机内存中的，也就是说通道与CPU共享内存

1. 完成一次读/写操作的流程（见右图）



2. CPU干预的频率
极低，通道会根据CPU的指示执行相应的通道程序，只有完成一组数据块的读/写后才需要发出中断信号，请求CPU干预。

3. 数据传送的单位
每次读/写一组数据块

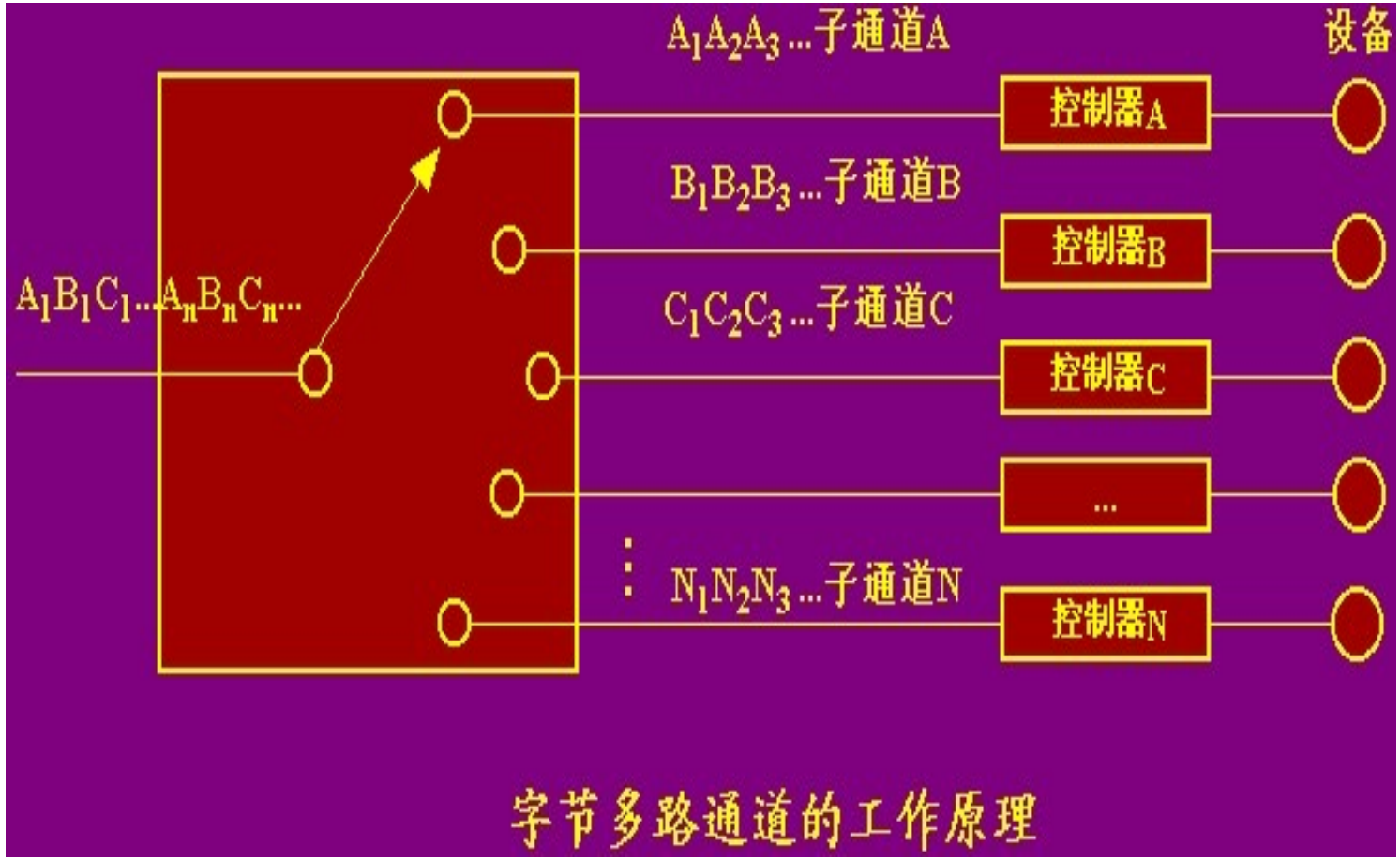
4. 数据的流向（在通道的控制下进行）
读操作（数据输入）：I/O设备→内存
写操作（数据输出）：内存→I/O设备

5. 主要缺点和主要优点
缺点：实现复杂，需要专门的通道硬件支持
优点：CPU、通道、I/O设备可并行工作，资源利用率很高。

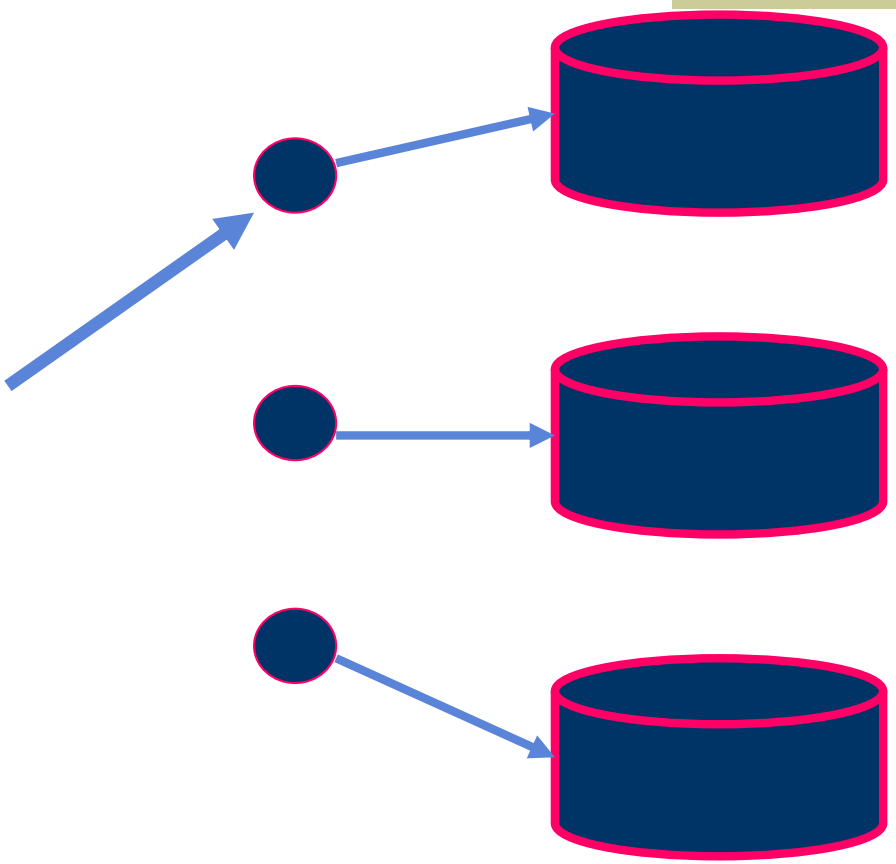
通道的种类

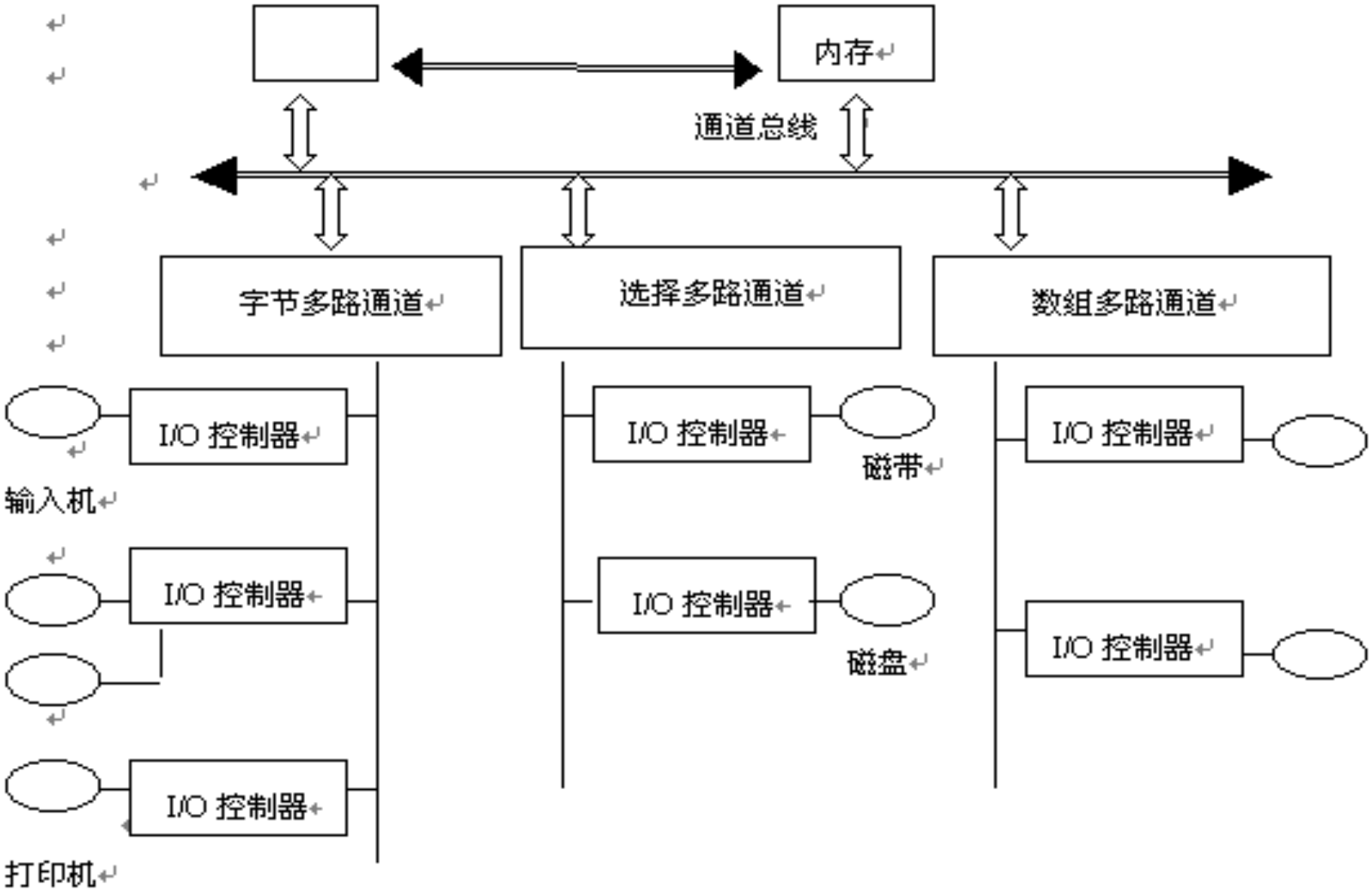
- 字节多路通道
- 数据选择通道
- 数据多路通道
- 在一大型系统中可以同时存在这三种类型的通道以便控制各种不同类型的设备。

字节多路通道


$A_1A_2A_3$ ...子通道A
设备
控制器A
$B_1B_2B_3$ ...子通道B
控制器B
$A_1B_1C_1...A_nB_nC_n...$
$C_1C_2C_3$ ...子通道C
控制器C
...
$N_1N_2N_3$ ...子通道N
控制器N
字节多路通道的工作原理

选择通道

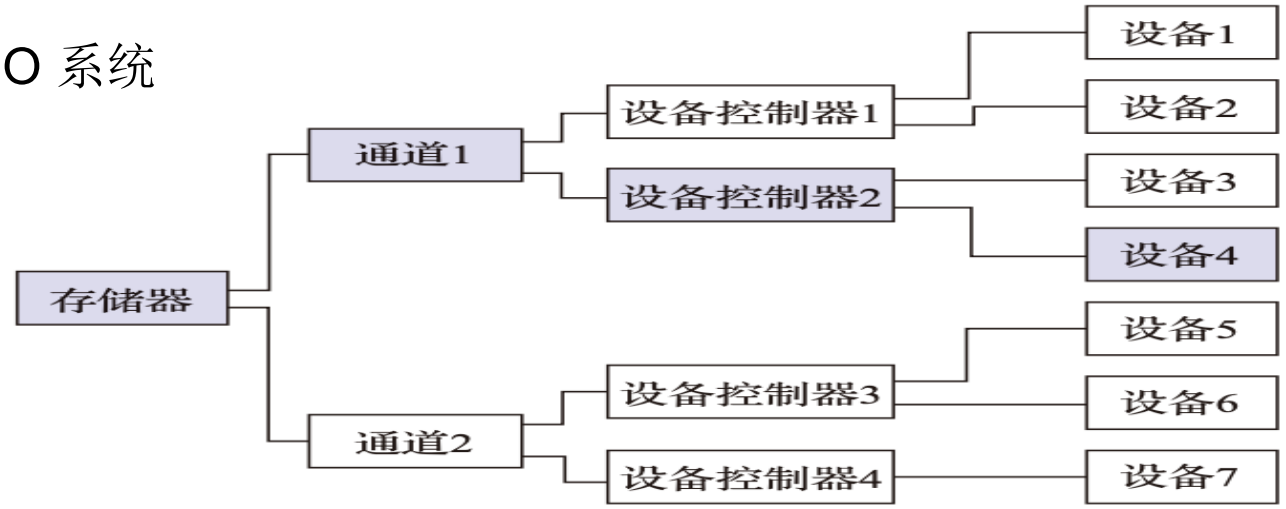



内存
通道总线
字节多路通道
选择多路通道
数组多路通道
I/O 控制器
I/O 控制器
I/O 控制器
I/O 控制器
I/O 控制器
I/O 控制器
输入机
磁带
磁盘
打印机

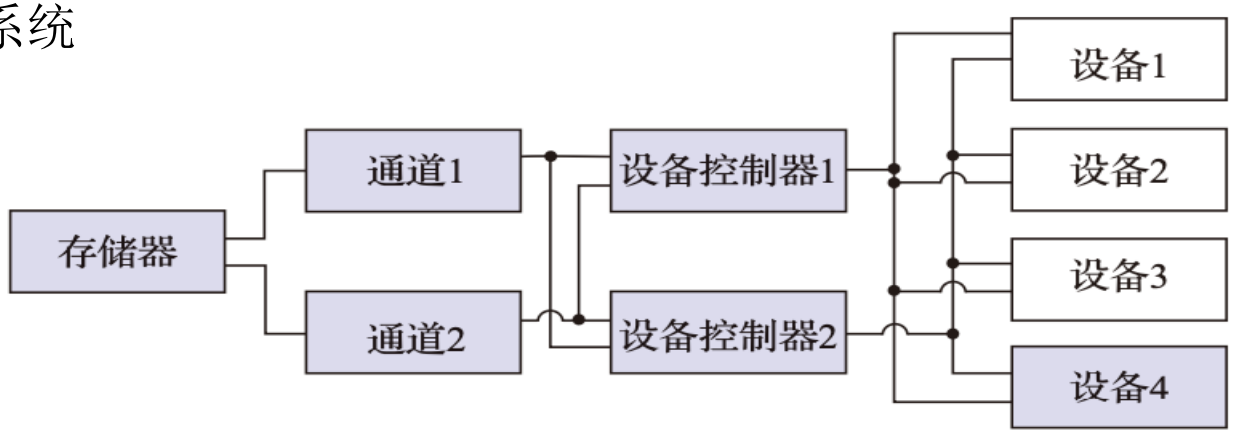
I/O系统结构

通道的瓶颈问题

单通路 IO 系统



多通路 IO 系统



	完成一次读/写的过程	CPU干预频率	每次I/O的数据传输单位	数据流向	优缺点
程序直接控制方式	CPU发出I/O命令后需要不断轮询	极高	字	设备→CPU→内存 内存→CPU→设备	每一个阶段的优点都是解决了上一阶段的最大缺点。 总体来说，整个发展过程就是要尽量减少CPU对I/O过程的干预，把CPU从繁杂的I/O控制事务中解脱出来，以便更多地去完成数据处理任务。
中断驱动方式	CPU发出I/O命令后可以做其他事，本次I/O完成后设备控制器发出中断信号	高	字	设备→CPU→内存 内存→CPU→设备	
DMA方式	CPU发出I/O命令后可以做其他事，本次I/O完成后DMA控制器发出中断信号	中	块	设备→内存 内存→设备	
通道控制方式	CPU发出I/O命令后可以做其他事。通道会执行通道程序以完成I/O，完成后通道向CPU发出中断信号	低	一组块	设备→内存 内存→设备	

难点理解：
通道=弱鸡版CPU
通道程序=任务清单