

# 输入输出系统

## I/O接口

### 功能

接受主机发送的I/O控制信号，实现主机和外部设备之间的信息交换

1. 选址功能【进行地址译码和设备选择】
2. 传送命令功能【I/O操作的控制与定时】
3. 传送数据功能【与主机和外设通信】
4. 反映I/O设备工作状态的功能【I/O过程中错误和状态检测】
5. 数据格式的转换

### I/O接口的基本结构

- 端口(Port): 接口电路中可以进行读写的寄存器
- 接口(Interface): 若干端口加上相应的控制逻辑
- I/O指令: 对数据缓冲寄存器、状态/控制寄存器的进行访问操作的指令
- 只能在OS内核的底层I/O软件中使用
- I/O指令实现的数据传送通常发生在通用寄存器和I/O端口之间
- 是一种特权指令

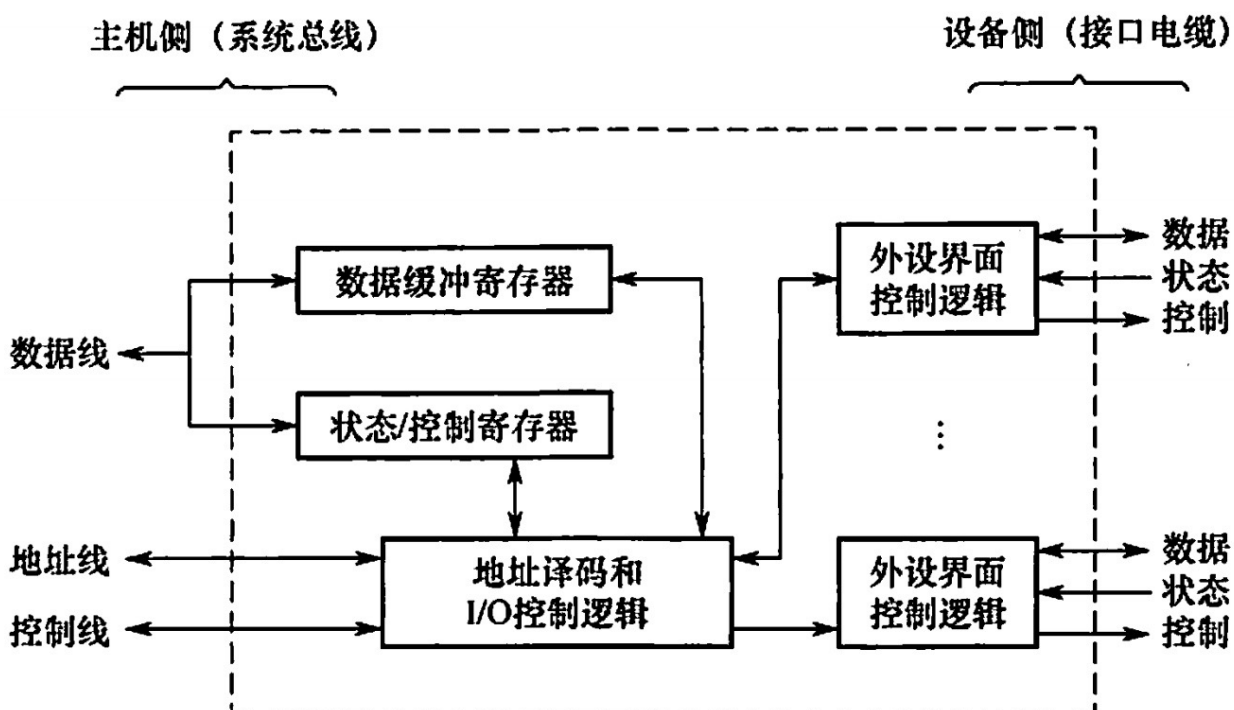


图 7.1 I/O 接口的基本结构

# I/O接口的类型

## 按数据传输方式分

- 并行接口：一个字节或一个字的所有位同时传送
- 串行接口：一位一位地传送

■ 接口要完成数据格式的转换

■ 这里的数据传送方式是指外设和接口一侧的传送方式

## 按主机访问I/O设备的控制方式分

- 程序查询接口
- 中断接口
- DMA接口

## 按功能选择的灵活性分

- 可编程接口
- 不可编程接口

# I/O端口及其编址

## I/O端口

- I/O端口：接口电路中可以被CPU直接访问的寄存器，即接口中用于暂存信息的寄存器
- 在执行一条指令时，CPU使用地址总线选择所请求的I/O端口，使用数据总线在CPU寄存器和端口之间传输数据
- I/O端口想要能被CPU访问，则必须对各个端口进行编号，每个端口对应一个端口地址
- 主要的I/O端口及其功能

数据端口	状态端口	控制端口
CPU对数据端口执行读写操作	对状态端口只能执行读操作	对控制端口只能执行写操作

## I/O编址

	统一编址（存储器映射方式）	独立编址（I/O映射方式）
定义	- 把I/O端口当做存储器的单元进行地址分配，CPU不需要设置专门的I/O指令，用统一的访存指令就可以访问I/O端口	- I/O端口的地址空间与主存地址空间无法从地址码的形式上区分  需要设置专门的I/O指令来访问I/O端口
特点	- 依靠地址码的不同区分存储单元和I/O设备	- 通过专门的I/O指令来区分存储单元和I/O设备
优点	- 不需要专门的I/O指令 - 可以使CPU访问I/O的操作更灵活、更方便 - 还可以使端口有较大的编址空间	- 输入/输出指令与存储器指令有明显区别 - 程序编制清晰，便于理解
缺点	- 端口占用存储器地址，使内存容量变小 - I/O设备进行数据输入/输出操作时，执行速度较慢	- 输入/输出指令少，一般只能对端口进行传送操作 - 尤其需要CPU提供存储器读/写、I/O设备读/写两组控制信号增加了控制的复杂性

## I/O方式

- I/O方式：输入/输出系统实现主机与I/O设备之间数据传送的控制方式

## 程序查询方式

### 定义

- 信息交换的控制完全由CPU执行程序实现
- 接口中设置一个数据缓冲寄存器（数据端口）和一个设备状态寄存器（状态端口）
- 主机进行I/O操作时，先发出询问信号，读取设备状态决定下一步操作到底是进行数据传送还是等待
- CPU一旦启动I/O，就必须停止现程序的运行，并在现行程序中插入一段程序

### 主要特点

- CPU有“踏步”等待现象
- CPU与I/O串行工作

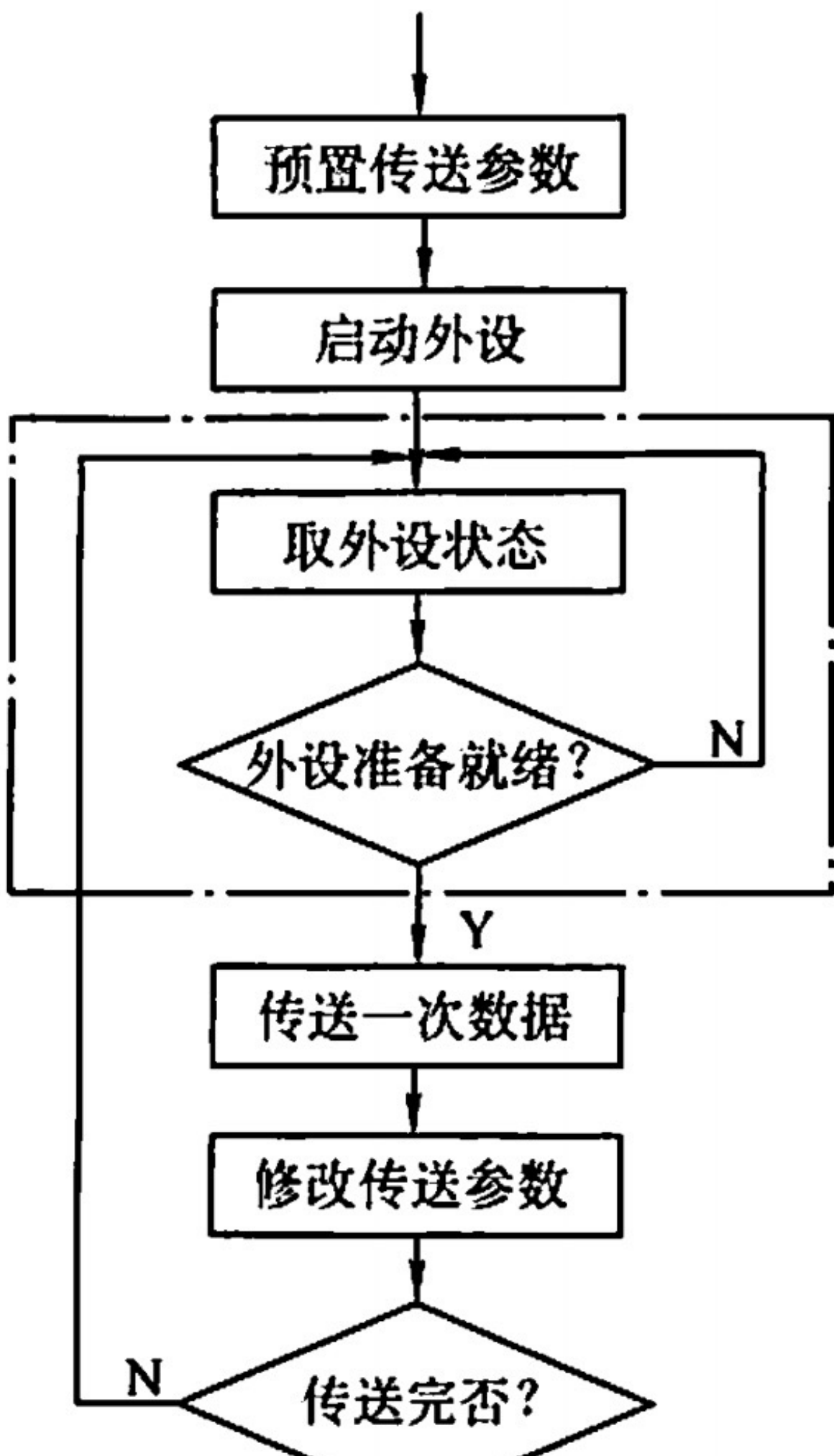
## 优点

- 端口设计简单，设备量少

## 缺点

- CPU在信息传送过程中要花费很多时间来查询和等待

- 而且在一段时间内只能和一台外设交换信息，效率大大降低



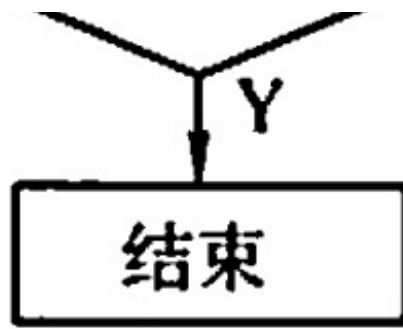


图 7.2 程序查询方式流程图

## 程序中断方式

### 基本概念

- 在计算机执行现程序的过程中，出现某些急需处理的异常情况或特殊请求CPU暂停中止现程序  
而转去对这些异常情况或特殊请求进行处理，处理完毕后再返回到现程序的断点处，继续执行原程序
- 早期的中断技术就是为了处理数据传送
- 中断响应阶段CPU进行的操作：关中断，保护断点和程序状态，识别中断源
- 多重中断系统在保护被中断进程现场时关中断，执行中断程序时开中断
- CPU一般在一条指令执行结束的阶段采样中断请求信号，查看是否存在中断请求，然后决定是否响应中断
- 中断隐指令的工作：关中断+保存断点+引出中断服务程序
- 通用计算器的保护由中断服务程序完成
- 中断优先级由屏蔽字决定，而不是根据请求的先后次序
- 有中断请求时，如果是关中断的姿态，或新中断请求的优先级较低，则不能响应新的中断请求

### 中断技术的主要功能

- 实现CPU和I/O设备的并行工作
- 处理硬件故障和软件错误
- 实现人机交互，用户干预机器需要用到中断系统
- 实现多道程序、分时操作，多道程序的切换需要借助于中断系统

- 实时处理需要借助中断系统来实现快速相应
- 实现应用程序和操作系统（管态程序）的切换，称为“软中断”
- 多处理器系统中各处理器之间的信息交流和任务切换

## 程序中中断的基本流程

1. 中断请求
2. 中断响应判优（通过硬件排队器实现）
  - ①不可屏蔽中断>内部异常>可屏蔽中断
  - ②内部异常中，硬件故障>软件中断
  - ③DMA中断请求优先于I/O设备传送的中断请求
  - ④在I/O传送类中断请求中，高速设备优先于低速设备
  - ⑤输入设备优先于输出设备，实时设备优先于普通设备
3. CPU响应中断的条件
  - ①中断源有中断请求
  - ②CPU允许中断与开中断（异常和不可屏蔽中断不受此影响）
  - ③一条指令执行完毕（异常不受此限制），且没有更紧迫的任务
4. 中断响应过程
  - ①关中断
  - ②保存断点
  - ③引出中断服务程序
5. 中断向量：分为向量中断（硬件）和非向量（软件）中断  
硬件识别方式：CPU响应中断后，通过识别中断源获得中断类型号然后据此计算出对应中断向量的地址；再根据该地址从中断向量表中取出中断服务程序的入口地址，并送入PC，以转而执行中断服务程序  
软件识别：CPU设置一个异常状态寄存器，用于记录异常原因，OS用一个统一的异常或中断查询程序，按优先等级顺序查询异常状态寄存器，以检测异常和中断的类型，先查到的先被处理，然后转到内核中相应的处理程序。
6. 中断处理过程

## 多重中断和中断屏蔽技术

### 单重中断

- 若CPU在执行中断服务程序的过程中，又出现了新的更高优先级的中断请求，而CPU对新的中断请求不予响应

### 多重中断（中断嵌套）

- 若CPU在执行中断服务程序的过程中，又出现了新的更高优先级的中断请求，CPU暂停现行的中断服务程序，转去处理新的中断请求

### 多重中断的条件



- 在中断服务程序中提前设置开中断指令
- 优先级别高的中断源有权中断优先级别低的中断指令

中断处理优先级可通过中断屏蔽技术动态调整

- 每个中断源有一个屏蔽触发器
- 1表示屏蔽该中断源的请求，0表示可以正常申请
- 所有屏蔽触发器组合在一起便构成一个屏蔽字寄存器
- 屏蔽字寄存器的内容称为屏蔽字

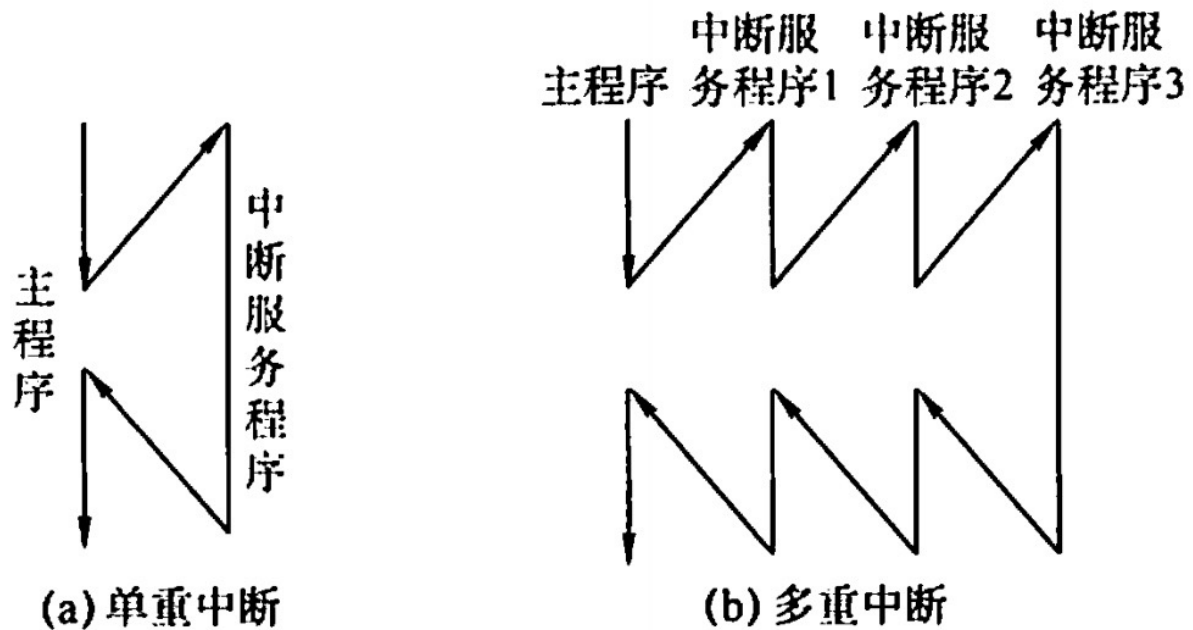


图 7.5 单重中断和多重中断示意图

### DMA (Direct Memory Access直接存储器存取方式) 方式

- 一种完全由硬件进行成组信息传送的控制方式
- 在数据准备阶段，CPU与外设并行工作，在外设与内存之间开辟一条“直接数据通路”，信息传送不再经过CPU（也就不需要保护、恢复CPU现场等操作），降低了CPU在传送数据时的开销

### DMA方式的特点

- 主存既可以被CPU访问，也可被外设访问
- 在数据块传送时，主存地址的确定、传送数据等都由硬件电路直接实现
- 主存中要开辟专用缓冲区，及时供给和接收外设的数据

- DMA传送速度快，CPU和外设并行工作，提高了系统效率
- DMA在传送开始前要通过程序进行预处理，结束后要通过中断方式进行后处理

## 什么是DMAC

- DMAC又叫DMA控制器，是对数据传送过程进行控制的硬件
- 在DMA过程中，DMAC将接管CPU的地址总线、数据总线和控制总线，CPU的主存控制信号被禁用

## DMAC的组成

主存地址计数器	存放要交换数据的主存地址
传送长度计数器	记录传送数据的长度，技术溢出时，数据传输即完毕，自动发送中断请求信号
数据缓冲寄存器	暂存每次传送的数据
DMA请求触发器	每当I/O设备准备好数据后，给出一个控制信号，使DMA请求触发器置位
“控制/状态”逻辑	由控制和时序电路及状态标志组成，用于指定传送方向，修改传送参数，并对DMA请求信号、CPU响应信号进行协调和同步。
中断机构	当一个数据块传送完毕后触发中断机构，向CPU提出中断请求

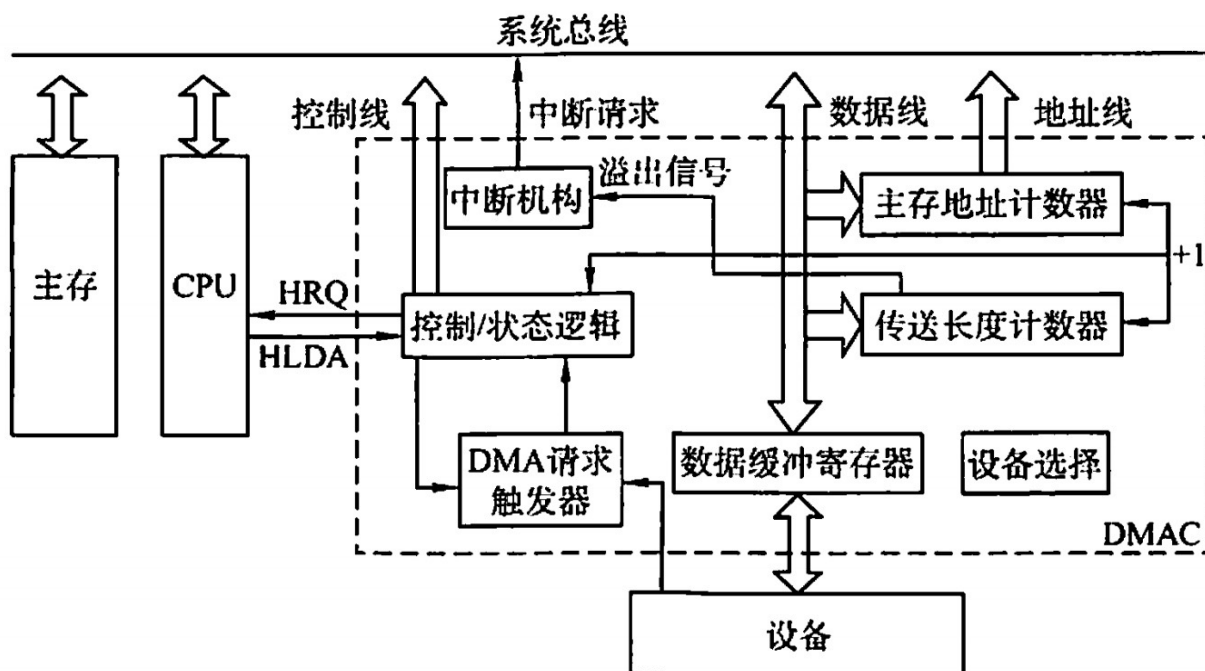


图 7.8 简单的 DMA 控制器

DMA的传送方式

当I/O设备和CPU同时访问主存时，可能发生冲突，为了有效使用主存，DMAC控制器和CPU通常采用3种方式使用主存

1. 停止CPU访问
1. 周期挪用：I/O访存优先级高于CPU访存，因为I/O不立即访存就可能会丢失数据，此时由I/O设备挪用几个存取周期
1. DMA与CPU交替访问：适用于CPU工作周期比主存存取周期长的情况。

DMA的传送过程

1. 预处理	
• 由CPU完成一些必要的准备工作	
2. 数据传送	
• 可以以单字节（或字）为基本单位	
• 也可以以数据块为基本单位（此时通过循环实现）	
3. 后处理	
• DMAC向CPU发出中断请求	
• CPU执行中断服务程序做DMA结束处理	
• 后处理包括	
○ 校验送入主存的数据是否正确	
○ 测试传送过程中是否出错（错误则转诊断程序）	
○ 决定是否继续使用DMA传送其他数据等	

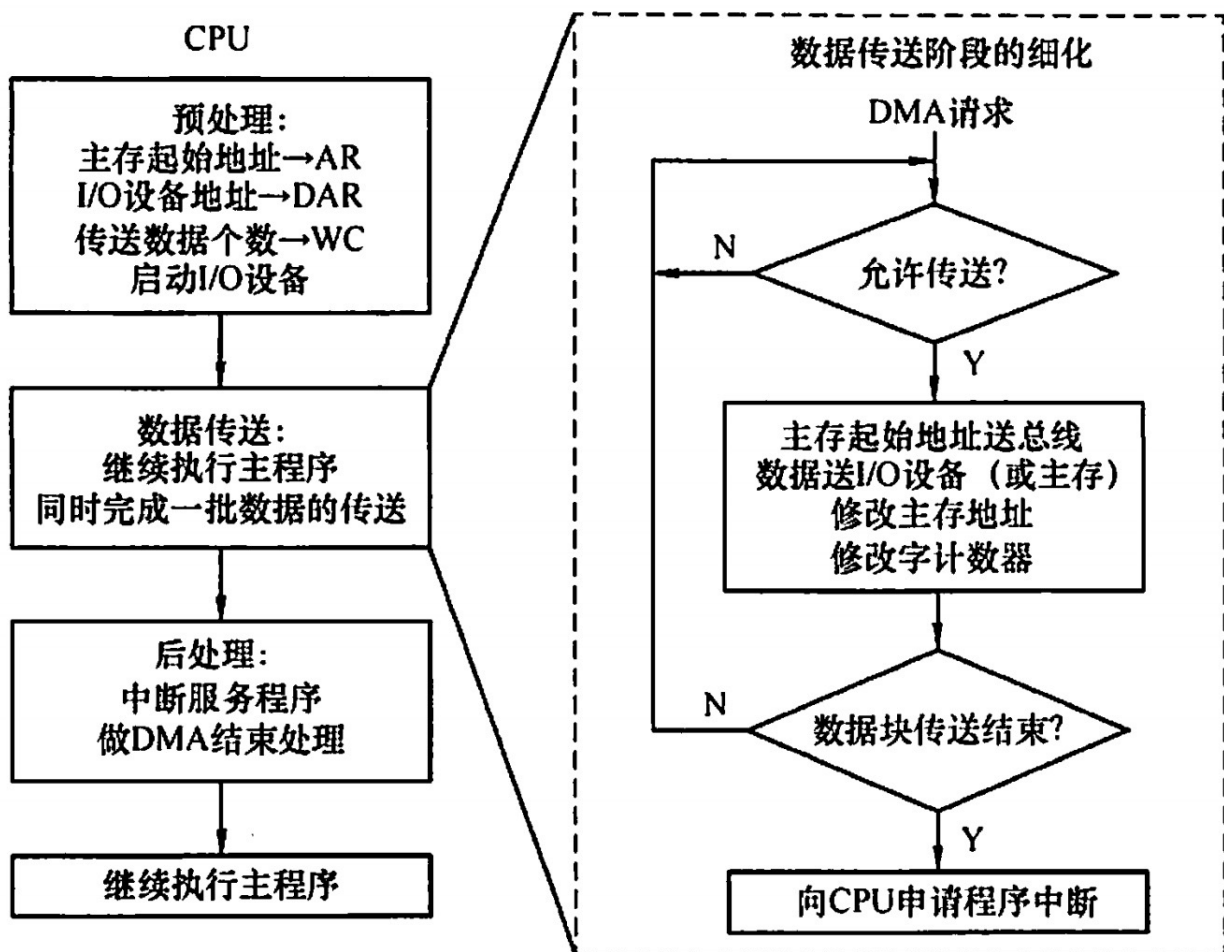


图 7.9 DMA 的传送流程

## DMA方式和中断方式的区别

DMA	中断方式
- DMA方式除了预处理和后处理，其他时候不占用CPU资源	- 中断方式是程序的切换，需要保护和恢复现场
- 对DMA请求的响应可以发生在每个机器周期结束时	- 对中断请求的响应只能发生在每条指令执行完毕时
- 在取指周期、间址周期、执行周期后都可以	- 即指令的执行周期后
- 只要CPU不占用总线就可以被响应	- 中断传送过程需要CPU的干预
- DMA传送过程不需要CPU的干预	
- 因此数据传输率非常高	
- 适合于高速外设的成组数据传送	
- DMA请求优先级高于中断请求	- 中断方式具有对异常事件的处理能力
- DMA方式仅局限于传送数据块的I/O操作	- 中断方式靠程序传送
- DMA方式靠硬件传送	

I/O控制方式	过程	CPU干预频率	每次传输数据的单位	数据流向	优缺点
程序直接控制方式	CPU发出指令后需要不断轮询。	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上一个阶段的 最大缺点。总体来说就是尽量 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 成其他任务。
中断驱动方式	CPU发出I/O指令后可以做其他事，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指令后可以做其他事。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	