

## 输入输出控制方式 (2)

兴  
大  
通  
交



## 本节内容

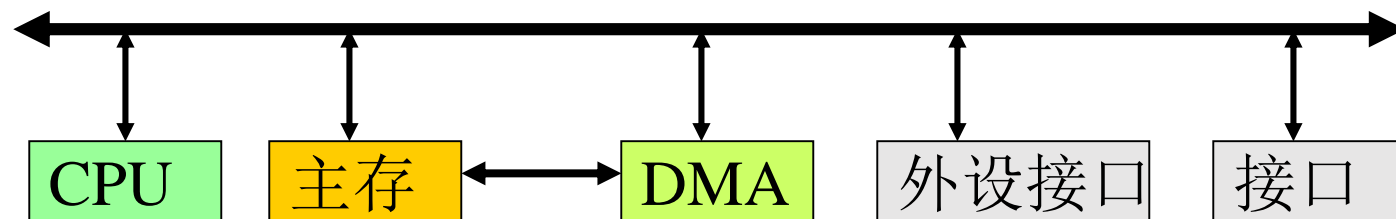


- 输入输出的控制方式
  - 程序查询方式
  - 中断方式
  - 直接存储器访问（**DMA**）控制方式
  - DMA: direct memory access

# DMA基本概念



- 中断方式每传送一个字或一个字节的数据就要执行一次中断服务程序，数据传送时仍然要占用CPU的时间，不适合于高速传输的系统。
- DMA在外设与主存之间建立一个由硬件管理的数据通路，使CPU不介入传送时的操作，数据也不经过CPU。
- 减少了CPU的开销，适用于成组交换数据的场合。例如：磁盘与内存之间

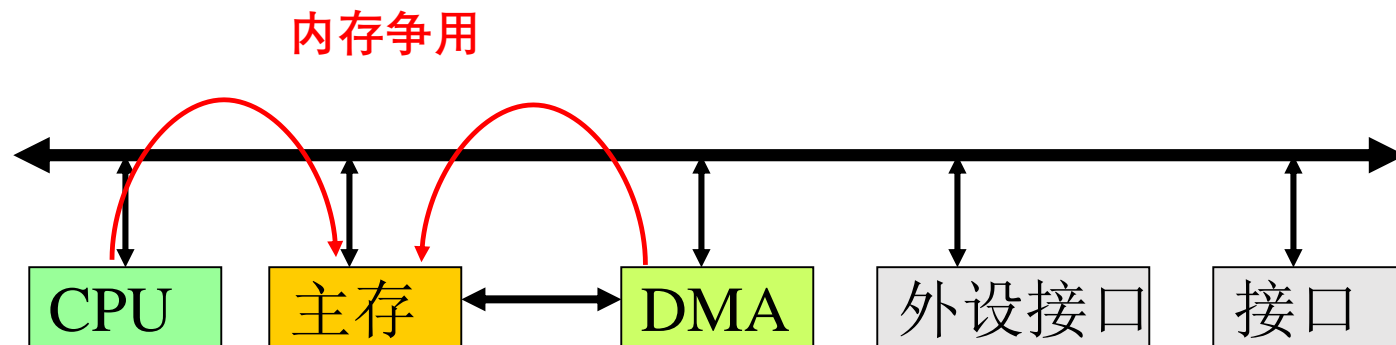




# 内存争用



- DMA方式进行数据传送时，CPU仍执行主程序，此时DMA控制逻辑与CPU可能同时要访问主存，引起主存使用权的冲突。
- 如何处理这种冲突呢？



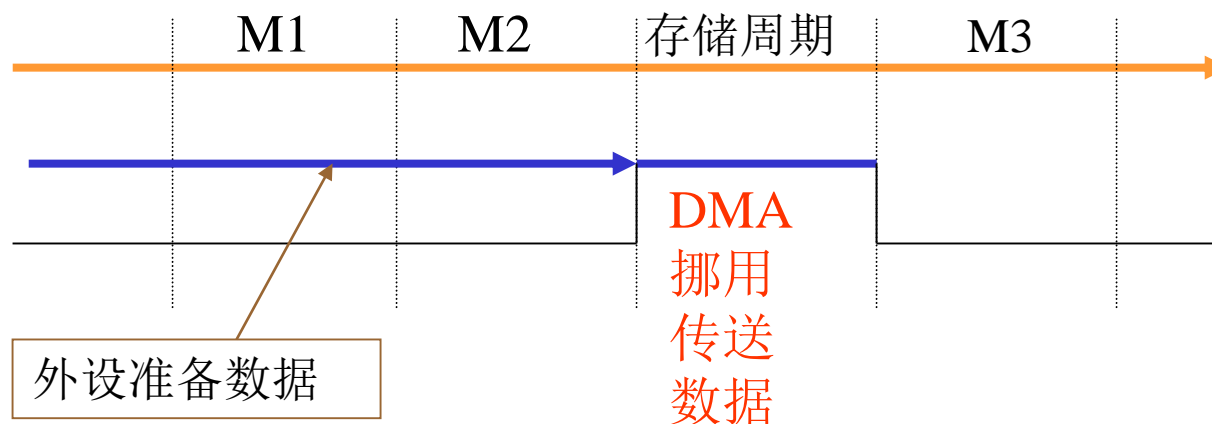
# 内存争用解决方案1：停止CPU使用主存

- 当DMA传送数据时，CPU停止工作，把主存使用权交给DMA控制逻辑
- 在DMA传送数据过程中，CPU处于等待状态
- 在这批数据传送结束后，DMA再交还主存使用权
- 方法简单、效率低

## 内存争用解决方案2：周期挪用法



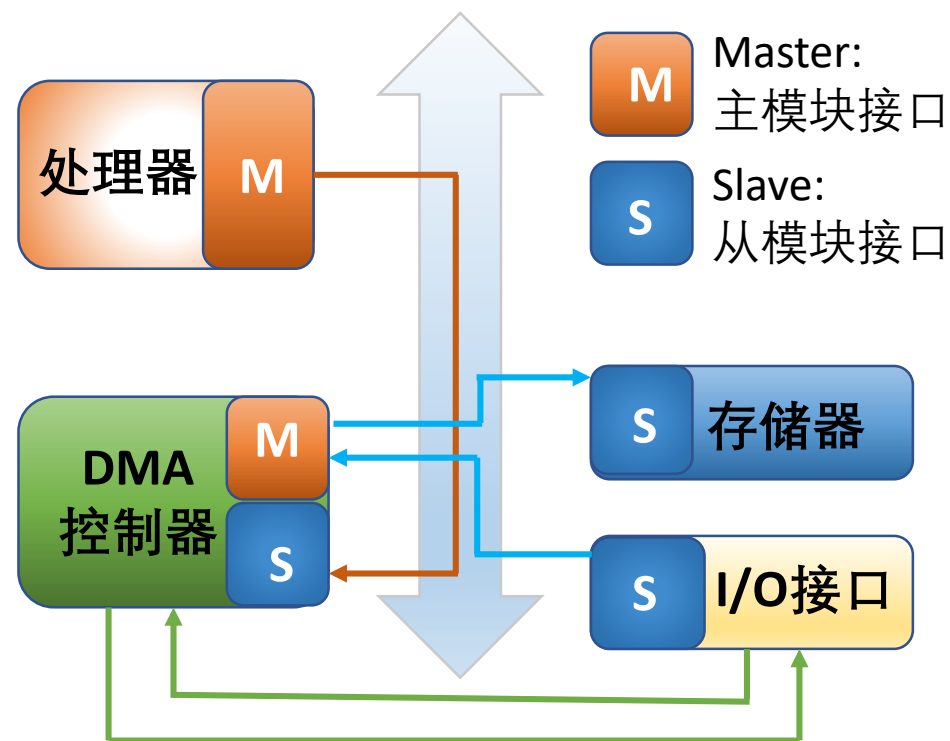
- DMA要求访问主存时，CPU暂停**一个或多个存储周期**。一个数据传送结束后，CPU继续运行。
- CPU现场并没有变动，只是延缓了对指令的执行，这种技术称为**周期挪用**，或称周期窃取。
- 如发生访存冲突，则DMA优先访问。



# DMA主要操作过程

以独立的DMA控制器（DMAC）为例：

1. CPU设置DMAC内部配置寄存器
2. DMAC处于空闲状态
3. 当外设准备好数据，I/O接口向DMAC发出DMA传送申请
4. DMAC响应I/O接口的申请
5. DMAC向I/O接口发起总线读传输
6. DMAC向存储器发起总线写传输
7. 重复5~6直到本次DMA传送完成



# DMA操作过程（准备阶段）

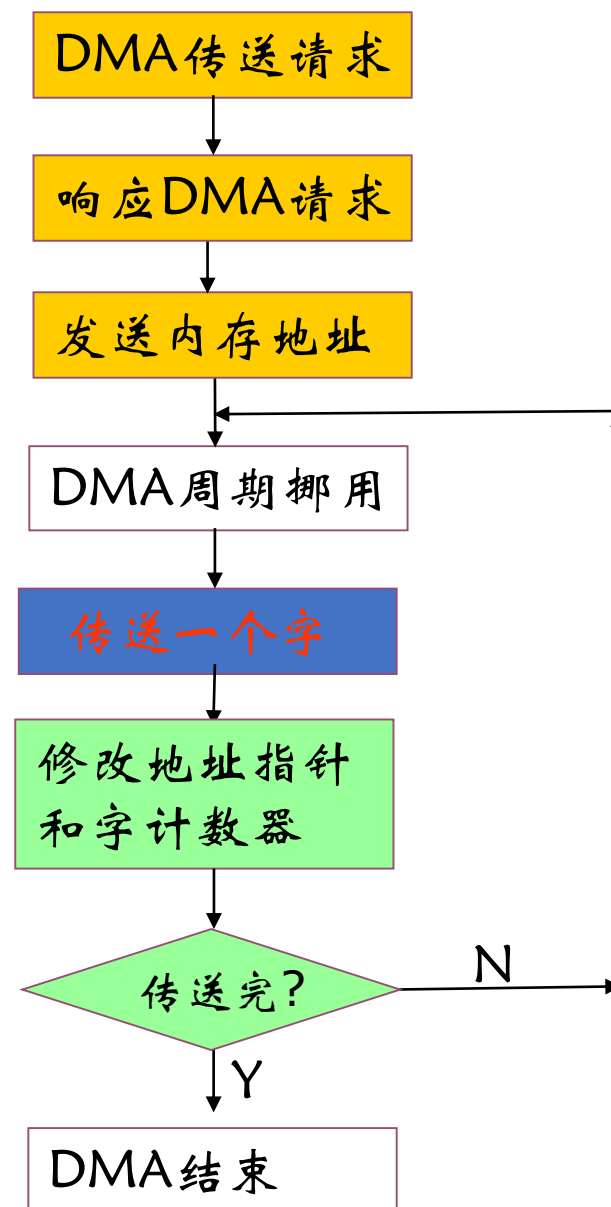
- CPU通过指令向DMA接口发送必要的传送参数，置DMAC内部配置寄存器，并启动DMA工作。这些参数的作用是：
  1. 指出数据传送的**方向**，即指出是主机将数据传送给设备，还是由设备传给主机。
  2. 指出数据块在**主存的首地址**。
  3. 指出数据在外设**存储介质上的地址**。
  4. 指出数据的**传送量**。



# DMA操作过程（传送阶段）

- 当设备准备好一个数据块后，向DMA控制器发出DMA传输请求；
- DMA控制逻辑响应请求，发送内存地址
  1. DMA控制器挪用一個周期，读或者写一个存储字
  2. 内存地址加1，指向下一个要传输数据的地址；  
字数计数器减1，表示剩下的要传输的字数
  3. 判断数据是否全部传输完。若没传完则又开始新的周期挪用，传输下一个字。
- 重复1-3步，直到整个数据块全部在外设和内存之间传输完毕。

**传输过程CPU完全不参与！**

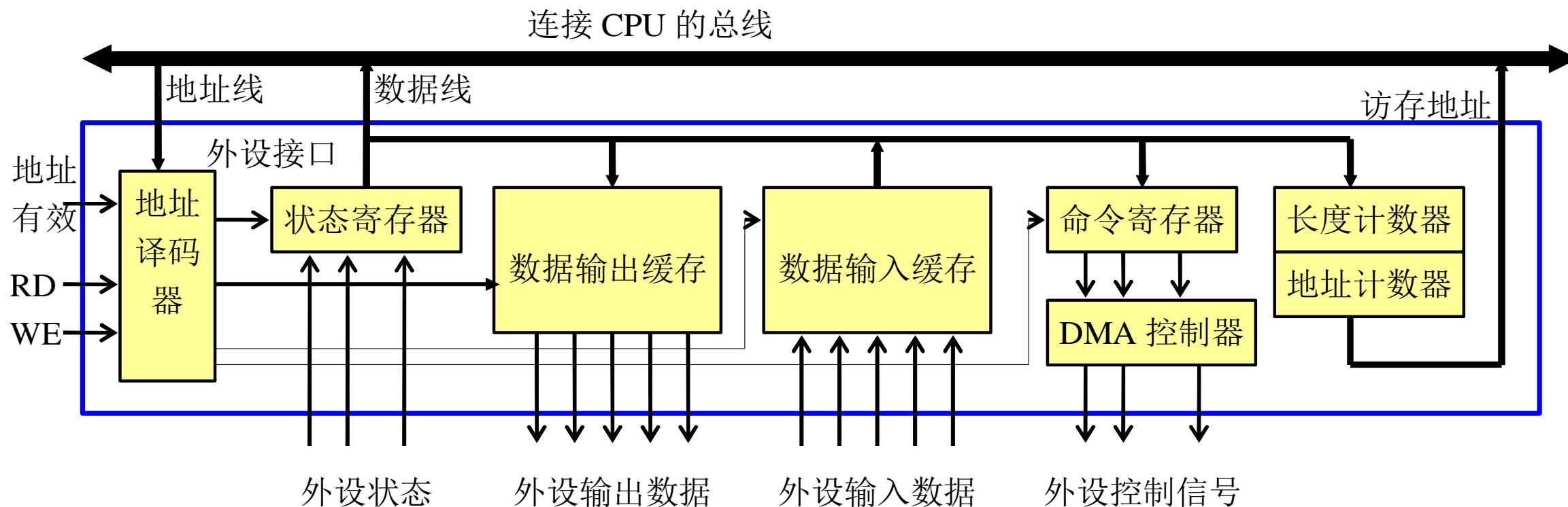
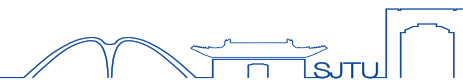


# DMA操作过程（结束阶段）



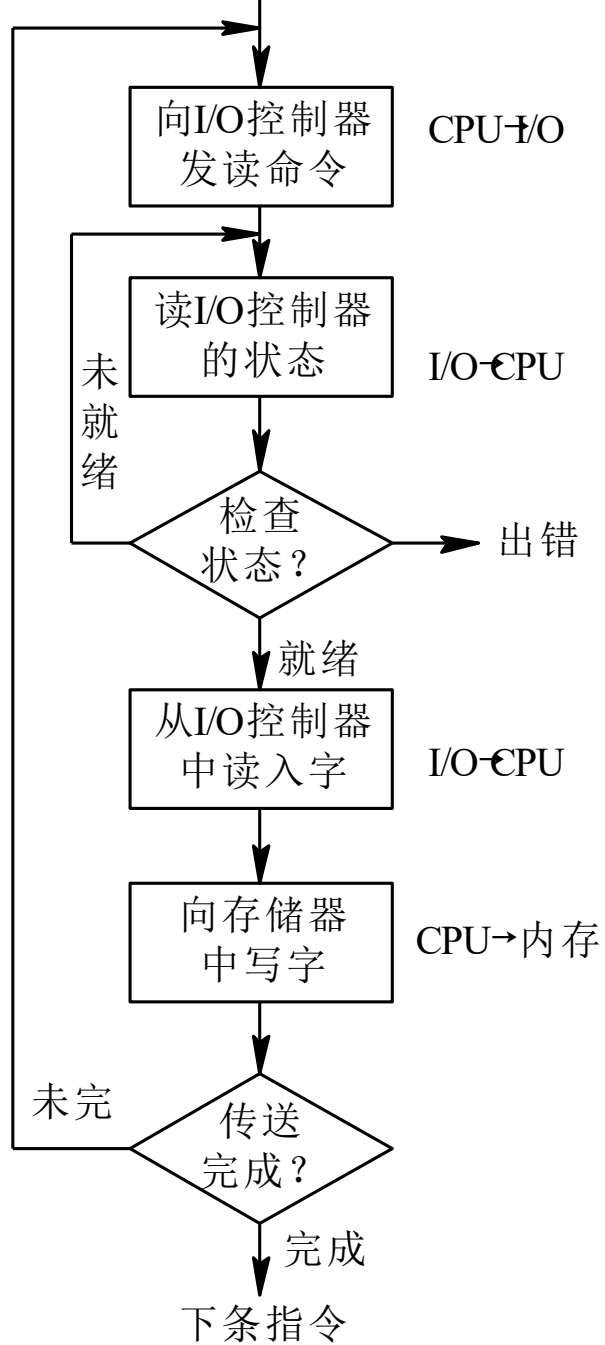
- DMA在两种情况下都进入结束阶段。
- 正常结束，一批数据传送完毕；
- 非正常结束，DMA发生了故障，也要进入结束阶段
- 不论是哪一种情况进入结束阶段，DMAC向CPU发出中断请求，CPU执行服务程序，查询DMA接口状态，根据状态进行不同处理。

# 基本的DMA控制器

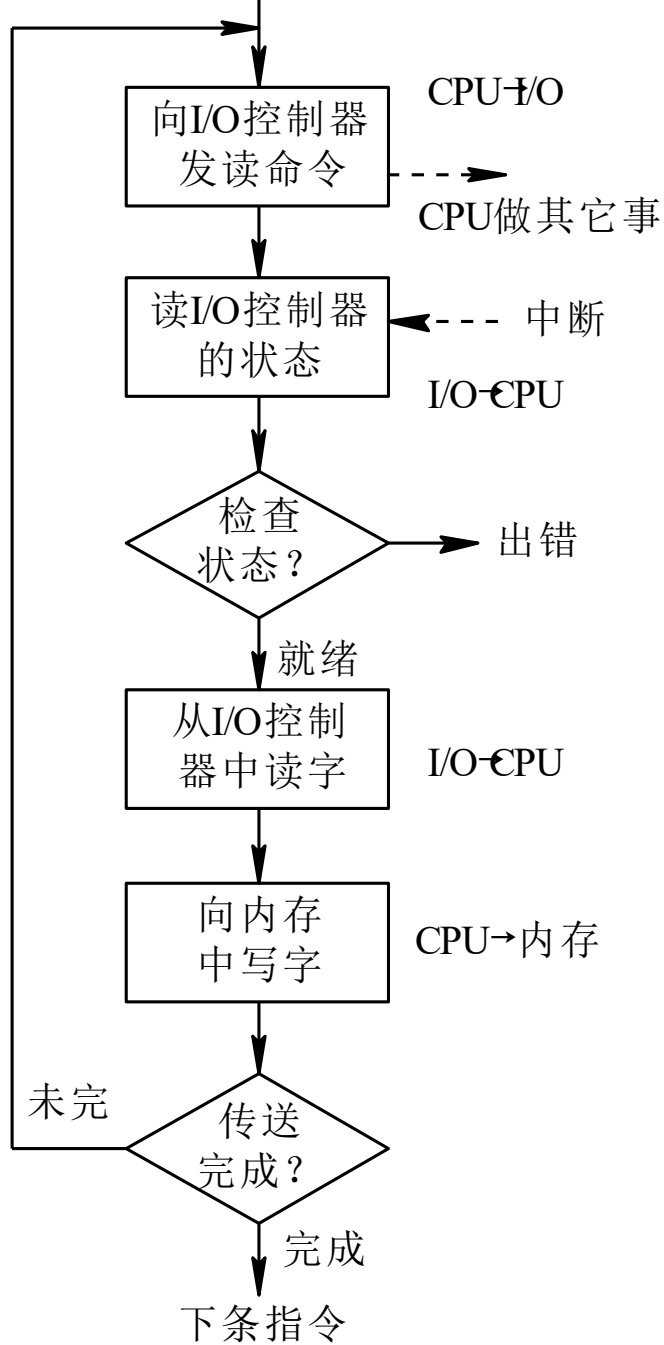


# DMA与程序中断的区别

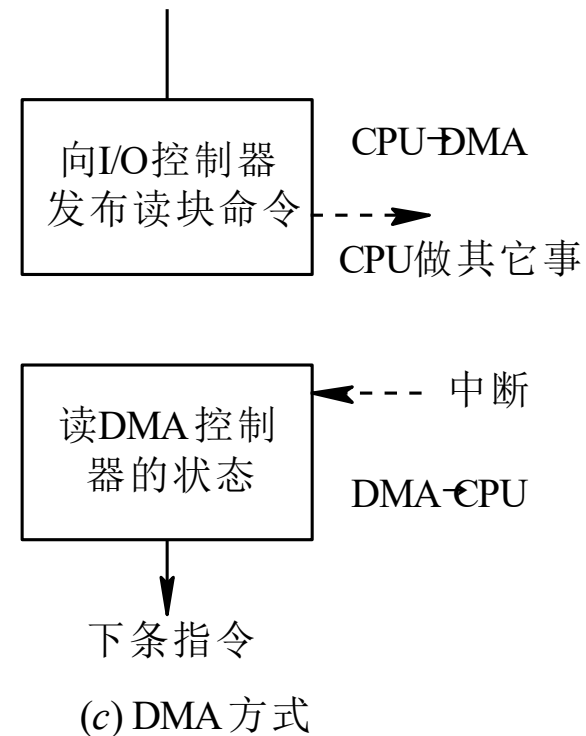
- 中断通过程序实现数据传送，而DMA靠硬件来实现。
- 中断时机为两条指令之间的空操作，DMA响应的时机为两个存储周期之间
- DMA仅挪用了一个存储周期，不改变CPU现场。
- DMA利用了中断技术
- DMA请求的优先权比中断请求高。CPU优先响应DMA请求，是为了避免DMA所连接的高速外设丢失数据。
- 中断不仅具有数据传送能力，还能处理异常事件。DMA只能进行数据传送。
- 中断适合于少量数据的不定时输入和输出的控制，DMA方式适合用于成组交换数据的场合



(a) 程序I/O方式



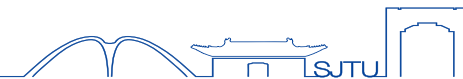
(b) 中断驱动方式



三种输入输出控制方式的对比



# 小结



- 直接存储器访问 (DMA)

谢谢！

