**DMA直接内存访问**

**概述**

* DMA: Direct Memory Access
  + 是 **外部设备** 和 **内存** 之间直接传输数据的方式
  + 是 一种高速的数据传输操作
  + **外部设备** 和 **内存** 之间的数据传输，不通过CPU，也不需要CPU的干预
  + **DMA控制器** 控制数据的传输

**DMA控制器控制控制数据传输的方式**

* 停止CPU访问内存
  + 当外围设备要求传送一批数据时，由DMA控制器发一个停止信号给CPU，要求CPU放弃地址总线，数据总线以及控制总线的使权
  + DMA控制器获得总线控制权以后，开始进行数据传送。
  + 在一批数据传送完毕后，DMA控制器通知CPU可以使用内存，并把总线控制权交还给CPU
* 周期挪用方式
  + 周期挪用是指：利用CPU不访问内存的那些周期来实现DMA操作，而此时DMA可以使用总线而不用通知CPU，也不会妨碍CPU的工作
  + 在这种方法中，每当I/O设备发出DMA请求时，I/O设备便挪用或窃取总线的所有权占用一个或几个主存周期，而DMA不请求时，CPU仍继续访问主存
  + I/O设备要求DMA传送的三种情况
    - 此时CPU不需要访问内存，I/O设备访问内存与CPU不冲突
    - CPU正在访问内存，此时必须等待CPU访问内存的周期结束，CPU才会让出总线所有权
    - I/O设备要求访问内存，此时CPU也要访问内存，产生冲突，此刻I/O访问内存优于CPU访问内存
* DMA与CPU交替访问内存
  + 即在同一个CPU周期内，即发生CPU对内存的访问，又发生DMA对内存的访问，因为一个CPU工作周期，要比内存的存取周期长的多
  + 如CPU的工作周期是1.2us,内存的存取周期小于0.6us，那么一个CPU周期可以分为C1和C2两个周期，CI供DMA控制器访问内存，C2供CPU访问内存
  + 这种方式不需要总线使用权的申请，建立和归还过程

**DMA的组成**

* 主存地址寄存器
  + 用于存放数据的内存地址
* 状态控制寄存器
* 数据缓冲寄存器
* 数据数量计数器
* DMA的控制逻辑
* 中断机制
* DMA请求触发器

**DMA实现数据传输的基本流程**

* **外部设备** 通过 **DMA控制器**向 **CPU** 发出**DMA请求**
* CPU响应DMA请求，交出总线控制权(控制总线，地址总线，数据总线)，此时系统转变为DMA工作方式，由DMA控制器接管
* DMA控制器，发送 内存地址 到地址总线，并传输数据的长度
* 执行DMA数据的传送
* DMA操作结束后，把总线控制权交还给CPU

**DMA的两种数据传输方式**

* 软件对数据的请求
  + 例如系统调用read函数的读取操作
  + 当进程调用read函数的时候，驱动程序函数分配一个**DMA缓冲区**，并让硬件将数据传送到这个缓冲区，然后进程进入休眠状态
  + 硬件将数据发送到DMA缓冲区，当硬件将数据写入DMA缓冲区完毕后，将产生一个中断
  + 在中断处理函数中，获得来自硬件的数据，应答中断，并且唤醒进程，此时进程就可以读取数据了。
* 硬件异步将数据传送给系统
  + 例如 数据采集设备
  + 硬件产生中断，宣告新数据的到来(**这个中断主要是创建一个DMA缓冲区**)
  + 中断处理程序分配一个DMA缓冲区，并且告诉硬件将数据发送到DMA缓冲区中
  + 外部设备将数据写入DMA缓冲区，完成后产生另一个中断
  + 处理程序分发数据，唤醒任何相关进程，然后执行清理工作。

**DMA的工作原理**

* 一个设备接口试图通过总线直接向另一个设备发送数据
* 外设不是直接向CPU发送DMA请求信号的
* 外设通过DMA的一种专门接口电路，即DMA控制器向CPU发出DMA请求信号，这个信号就是向CPU提出接管总线的控制权
* CPU收到该信号后，在当前的总线周期结束后，会按照DMA信号的优先级和提出DMA请求的先后顺讯响应DMA信号
* CPU对某个设备接口响应DMA请求时，会让出总线控制权
* 于是在DMA控制器的管理下，外设和内存直接进行数据交换，而不需要CPU的干预
* 数据传送完毕后，设备接口会向CPU发送DMA结束信号，交换总线控制权