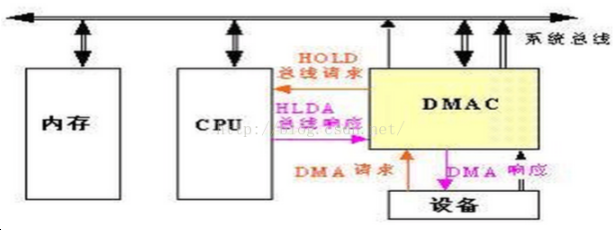
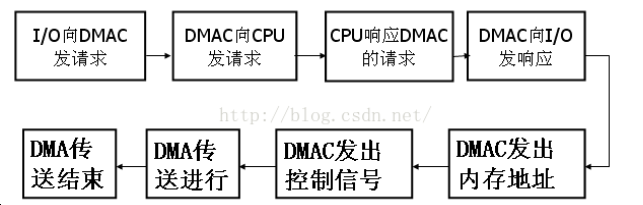
DMA传输



1. DMA请求   
   CPU对DMA控制器初始化，并向I/O接口发出操作命令，I/O接口提出DMA请求。
2. DMA响应   
     DMA控制器对DMA请求判别优先级及屏蔽，向总线裁决逻辑提出总线请求。当CPU执行完当前总线周期即可释放总线控制权。此时，总线裁决逻辑输出总线应答，表示DMA已经响应，通过DMA控制器通知I/O接口开始DMA传输。
3. DMA传输   
     DMA控制器获得总线控制权后，CPU即刻挂起或只执行内部操作，由DMA控制器输出读写命令，直接控制RAM与I/O接口进行DMA传输。    
     在DMA控制器的控制下，在存储器和外部设备之间直接进行数据传送，在传送过中不需要中央处理器的参与。开始时需提供要传送的数据的起始位置和数据长度。  4.DMA结束   
     当完成规定的成批数据传送后，DMA控制器即释放总线控制权，并向I/O接口发出结束信号。当I/O接口收到结束信号后，一方面停 止I/O设备的工作，另一方面向CPU提出中断请求，使CPU从不介入的状态解脱，并执行一段检查本次DMA传输操作正确性的代码。最后，带着本次操作结果及状态继续执行原来的程序。    
     由此可见，DMA传输方式无需CPU直接控制传输，也没有中断处理方式那样保留现场和恢复现场的过程，通过硬件为RAM与I/O设备开辟一条直接传送数据的通路，使CPU的效率大为提高。



对于嵌入式中的DMA，其实是在写数据寄存器的时候用dma的传输来代替。就像i2c设备，在发送和接收数据的时候都是要往数据寄存器中写数据的。比如那个寄存器是I2C\_DATA，如果用cpu来传输的话就是writel（data， I2C\_DATA）;而用dma传输就是配置好要传输的buf长度，然后源地址就是buf的地址，目标地址就是I2C\_DATA。

       这里还要注意经过cpu的是虚拟地址，而dma传输的是物理地址。

       其实dma传输就是dma控制在两个物理地址之间传输数据。

