

DMA（直接存储器访问）

DMA(Direct Memory Access, 直接内存存取) 是所有现代电脑的重要特色, 它允许不同速度的硬件装置来沟通, 而不需要依赖于 CPU 的大量中断负载。否则, CPU 需要从来源把每一片段的资料复制到暂存器, 然后把它们再次写回到新的地方。在这个时间中, CPU 对于其他的工作来说就无法使用。

中文名	直接存储器访问	外文名	Direct Memory Access	缩写	DMA	功能	不同速度的硬件装置来沟通	来源	嵌入式底层驱动
-----	---------	-----	----------------------	----	-----	----	--------------	----	---------

目录

1 原理

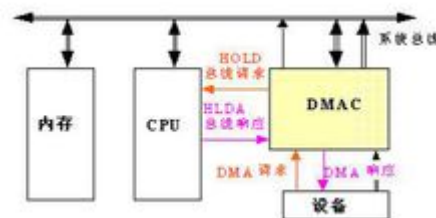
- 请求
- 响应
- 传输
- 结束

2 传送方式

原理

DMA 传输将数据从一个地址空间复制到另外一个地址空间。当 **CPU** 初始化这个传输动作，传输动作本身是由 **DMA** 控制器来实行和完成。典型的例子就是移动一个外部内存的区块到芯片内部更快的内存区。像是这样的操作并没有让处理器工作拖延，反而可以被重新排程去处理其他的工作。**DMA** 传输对于高效能 嵌入式系统算法和网络是很重要的。

在实现 DMA 传输时，是由 DMA 控制器直接掌管总线，因此，存在着一个总线控制权转移问题。即 DMA 传输前，CPU 要把总线控制权交给 DMA 控制器，而在结束 DMA 传输后，DMA 控制器应立即把总线控制权再交还给 CPU。一个完整的 DMA 传输过程必须经过 DMA 请求、DMA 响应、DMA 传输、DMA 结束 4 个步骤。



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。

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。

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，在传送过程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。

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。

传送方式

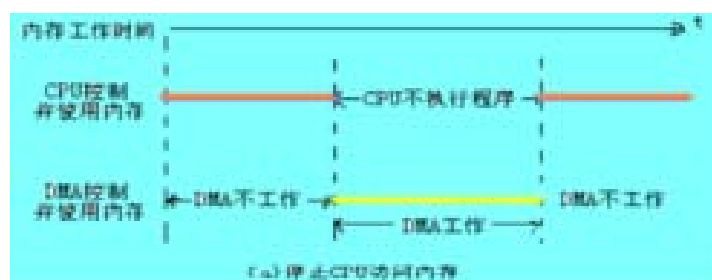
DMA 技术的出现，使得外围设备可以通过 DMA 控制器直接访问内存，与此同时，CPU 可以继续执行程序。那么 DMA 控制器与 CPU 怎样分时使用内存呢？通常采用以下三种方法：(1) 停止 CPU 访内存；(2) 周期挪用；(3) DMA 与 CPU 交替访问内存。

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 放弃对地址总线、数据总线和有关控制总线的使用权。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。图(a)是这种传送方式的时间图。很显然，在这种 DMA 传送过程中，CPU 基本处于不工作状态或者说保持状态。

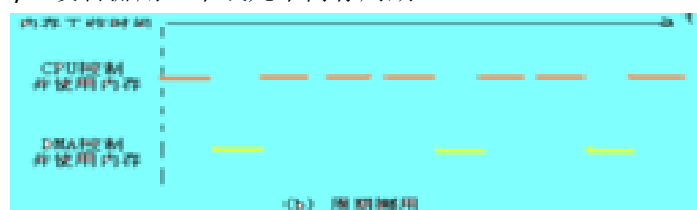
优点：控制简单，它适用于数据传输率很高的设备进行成组传送。

缺点：在 DMA 控制器访内阶段，内存的效能没有充分发挥，相当一部分内存工作周期是空闲的。这是因为，外围设备传送两个数据之间的间隔一般总是大于内存存储周期，即使高速 I/O 设备也是如此。例如，软盘读出一个 8 位二进制数大约需要 32 μ s，而半导体内存的存储周期小于 0.5 μ s，因此许多空闲的存储周期不能被 CPU 利用。



周期挪用

当 I/O 设备没有 DMA 请求时，CPU 按程序要求访问内存；一旦 I/O 设备有 DMA 请求，则由 I/O 设备挪用一個或几个内存周期。



这种传送方式的时间图如下图(b)：

I/O 设备要求 DMA 传送时可能遇到两种情况：

- (1) 此时 CPU 不需要访内，如 CPU 正在执行乘法指令。由于乘法指令执行时间较长，此时 I/O 访内与 CPU 访内没有冲突，即 I/O 设备挪用一二个内存周期对 CPU 执行程序没有任何影响。
- (2) I/O 设备要求访内时 CPU 也要求访内，这就产生了访内冲突，在这种情况下 I/O 设备访内优先，因为 I/O 访内有时间要求，前一个 I/O 数据必须在下一个访问请求到来之前存取完毕。显然，在这种情况下 I/O 设备挪用一二个内存周期，意味着 CPU 延缓了对指令的执行，或者更明确地说，在 CPU 执行访内指令的过程中插入 DMA 请求，挪用了一二个内存周期。与停止 CPU 访内的 DMA 方法比较，周期挪用的方法既实现了 I/O 传送，又较好地发挥了内存

和 CPU 的效率,是一种广泛采用的方法。但是 I/O 设备每一次周期挪用都有申请总线控制权、建立线控制权和归还总线控制权的过程,所以传送一个字对内存来说要占用一个周期,但对 DMA 控制器来说一般要 2—5 个内存周期(视逻辑线路的延迟而定)。因此,周期挪用的方法适用于 I/O 设备读写周期大于内存存储周期的情况。

DMA 与 CPU 交替访问内存

如果 CPU 的工作周期比内存存取周期长很多,此时采用交替访内的方法可以使 DMA 传送和 CPU 同时发挥最高的效率。

这种传送方式的时间图如下:



此图是 DMA 与 CPU 交替访内的详细时间图。假设 CPU 工作周期为 $1.2\mu s$, 内存存取周期小于 $0.6\mu s$, 那么一个 CPU 周期可分为 C1 和 C2 两个分周期, 其中 C1 专供 DMA 控制器访内, C2 专供 CPU 访内。

这种方式不需要总线使用权的申请、建立和归还过程, 总线使用权是通过 C1 和 C2 分时制的。CPU 和 DMA 控制器各自有自己的访内地址寄存器、数据寄存器和读/写信号等控制寄存器。在 C1 周期中, 如果 DMA 控制器有访内请求, 可将地址、数据等信号送到总线上。在 C2 周期中, 如 CPU 有访内请求, 同样传送地址、数据等信号。事实上, 对于总线, 这是用 C1, C2 控制的一个多路转换器, 这种总线控制权的转移几乎不需要什么时间, 所以对 DMA 传送来讲效率是很高的。

这种传送方式又称为“透明的 DMA”方式, 其来由是这种 DMA 传送对 CPU 来说, 如同透明的玻璃一般, 没有任何感觉或影响。在透明的 DMA 方式下工作, CPU 既不停止主程序的运行, 也不进入等待状态, 是一种高效率的工作方式。当然, 相应的硬件逻辑也就更加复杂