

如何使用DMA在低功耗可穿戴设备中加快外设监测

Brandon Hurst, 硬件和嵌入式固件工程师

摘要

本文介绍在嵌入式系统程序中使用直接内存访问(DMA)的用途、优点和缺点。本文描述了DMA如何与外设和内存模块交互,以提高CPU的运行效率。还将为读者介绍不同的DMA总线访问架构,及各自的优点。

嵌入式系统执行的一个常见任务是管理外部输入。管理输入会给处理器带来很多不必要的计算压力,导致处理器处于有功功率模式下的时间更长,响应速度更慢。为了优化功率,保持对事件的快速响应以及管理大量数据连续传输,具有直接内存访问(DMA)的微控制器可提供更好的解决方案。

直接内存访问(DMA)

在涉及外设的系统应用中,微处理器在许多点可能都会遭遇瓶颈。例如,在管理一个不断发送数据的ADC时,处理器可能时常被中断,导致它很难完成其他任务。DMA是一种在大型或快速数据处理事务中移动数据和尽量减少处理器参与的方法。您可以将DMA控制器当做一个协处理器,它的唯一作用就是与存储器和外设交互。通过这种方式,主处理器可以成功管理工作量大的外设,专注于处理其他任务,甚至可以在后台处理数据时进入睡眠状态,以节省功率。例如,在Arm®架构上,DMA模块可以在LP2(睡眠)或LP3(运行)模式下运行。对于需要更长的电池寿命的应用,例如可穿戴传感器中枢和智能手表,这具有明显的优势。

优点和缺点

在许多数字系统中,DMA很有用处,有时甚至用于管理大量总线流量。它还用于网卡、显卡,甚至一些原始的IBM个人电脑中。也就是说,将DMA集成到设计中时,确实需要进行一些权衡考量。

表1. 使用DMA的优点

使用DMA的优点	
CPU时间	DMA可以最大限度减少需要处理器执行和中断的频率,缩短处理数据需要耗费的CPU时间。
功耗	如果DMA允许处理器在DMA传输期间进入睡眠模式,则有机会尽可能降低功耗。
并联操作	根据系统总线的架构情况,处理器可以在执行外设事务时执行其他操作。

表2. 使用DMA的缺点

使用DMA的缺点	
成本	在系统中集成DMA需要用到DMA控制器,这会使系统成本更加高昂。
复杂性	虽然DMA可以降低中断频率,但它会使应用固体的大小和复杂性增加。
对平台的依赖性	不同制造商生产的DMA控制器以及同一制造商生产的不同DMA控制器,其内部架构不同,根据它们本身的总线访问方案,行为特性也会不同。
缓存区的不连续性	在写入存储器体系的缓存层时,DMA事务可能导致逻辑错误。使用缓存连贯的系统架构或在DMA完成时使用缓存无效可以解决这一问题。

总线访问和CPU周期

虽然DMA控制器在节能或加速嵌入式系统方面非常有效,但它们的实现方案还没有高度标准化。可以使用多种方案来确保不会与CPU同时授予内部总线访问权限。总线访问方案的主要目标是避免同时访问相同的存储器位置,以免发生缓存区不连续和逻辑错误。单个DMA控制器通常配置为采用这些方案之一,因为使用每种方案可能需要不同的硬件或固件控制。大多数DMA控制器使用的总线访问方案是突发、周期窃用和透明模式的DMA。

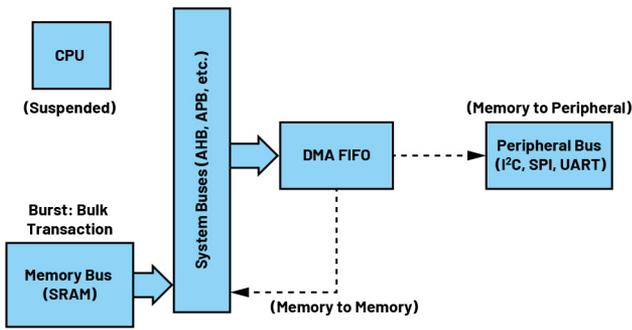


图1. DMA运行期间，突发DMA的架构图。

突发DMA是藉由不常见的大型突发事件发生，在此期间，DMA会根据缓冲区可以存储的容量，向目标缓冲区发送尽可能多的数据。DMA控制器会在短时间内阻止CPU运行，以移动大量内存，然后将总线交还给主CPU，并一直重复该过程，直到传输完成。突发DMA通常被认为是速度最快的类型。

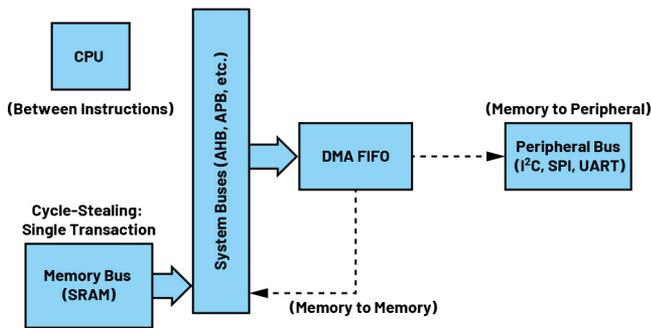


图2. 在DMA运行期间，周期窃用DMA发生在两个CPU周期之间。

相反，单字节传输或周期窃用DMA从CPU获取提示，并且仅在CPU指令之间执行操作。它在两个CPU周期之间插入一个操作，因此实际上“窃取”了CPU时间。由于一次只能执行一项操作的限制，它通常比突发DMA慢。

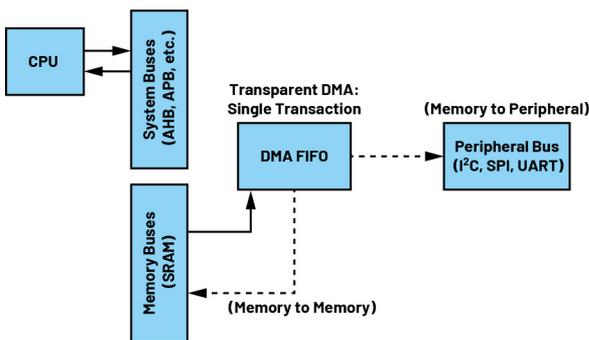


图3. 在DMA运行期间，透明DMA在处理器处理不访问数据或地址总线的任务时发生。

透明DMA一次只能执行一项操作，但是，它也必须等待处理器执行指令，以获得对所需数据或地址总线的访问。验证这种访问限制需要使用额外的逻辑，这种类型DMA的速度通常最慢。在不需访问存储器总线但需要进行额外的处理的应用中，透明DMA可能具有优势。在这种情况下，优势在于消除了对CPU的节流，因为处理器无需完全停止运行。

表3. DMA类型及其优缺点总结

DMA类型	优点	缺点
突发DMA	速度最快的DMA类型	CPU的空闲时间相对很长
周期窃取DMA	CPU不会连续长时间处于空闲状态	比突发DMA慢
透明DMA	无需使用CPU节流	速度最慢的DMA

突发DMA架构示例

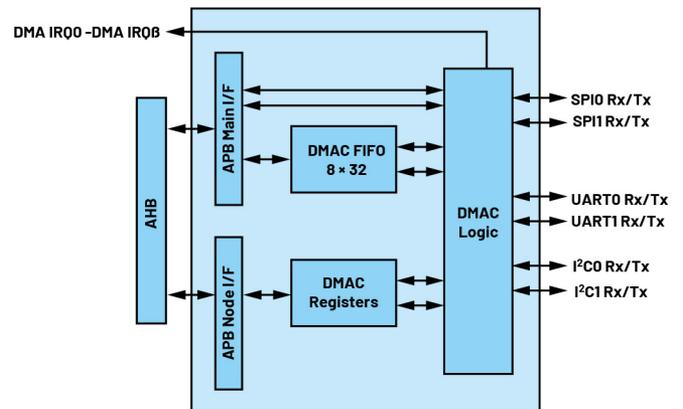


图4. MAX32660上的DMA控制器的架构图。

关于突发DMA控制器示例，可以参见MAX32660（参见图4）。上方路径对应数据流，下方路径代表先进的高性能总线(AHB)和DMA逻辑之间的控制/状态流。DMA控制器可以用作AHB和存储器或外设模块之间的缓冲区接口，具体取决于其配置方式。DMA逻辑位于DMA缓冲区和每个外设之间，用于在处理期间独立管理每个唯一的外设总线。DMA运行一次可以移动多达32个字节，只要源/目标缓冲区中可以包含这么多数据。缓冲区最高可存储16 MB，除了内部存储器传输外，还可以配置为发送或接收I2C、SPI、I2S和UART。编程DMA控制可能会因协议的不同而略有不同，但外设事务完全由DMA控制器管理。仲裁模块控制四个DMA通道和CPU之间的总线访问限制，根据优先级系统批准请求。

现代DMA选项

总之，对于管理大量传感器并要求高吞吐量、高效率和低功耗操作的现代嵌入式系统来说，DMA是一个关键特性。它就像是一个专门用于处理存储器和外设总线事务的协处理器。

许多应用必须使用DMA，以尽可能降低功耗和减轻处理器负载。例如，健康和可穿戴设备可以处理大量的数据吞吐量，但它们也必须尽可能地节省电池电量，同时还要处理敏感数据。ADI公

司在适用于低功耗可穿戴设备的微控制器上采用快速突发DMA架构，例如MAX32660和MAX32670。此外，DARWIN Arm微控制器（例如MAX32666）是为集成Bluetooth® 5的可穿戴设备和物联网应用而设计的。这些器件采用2个8通道突发DMA控制器，支持基于事件的事务。它们甚至配备了出色的安全硬件，具有安全引导加载程序和信任保护单元(TPU)，可以加速ECDSA、SHA-2和AES加密。从早期的IBM电脑到网卡，再到现在的安全、低功耗可穿戴设备和物联网设备，DMA是现代数字系统的基本特征。



作者简介

Brandon Hurst是一名硬件和嵌入式固件工程师，就职于Maxim Integrated（现为ADI公司的一部分）的培训和客户服务团队。他毕业于加州理工大学，拥有电气工程学士学位，于2021年1月加入Maxim。此前，Brandon曾在Maxim的TTS团队和苹果公司的产品安全工程团队实习。联系方式：brandon.hurst@analog.com。

