

Linux 之 DMA

原创 wencunZerotonez 于 2023-03-01 00:13:35 发布 576 收藏 1 版权

文章标签： linux 驱动开发 嵌入式硬件

DMA (Direct Memory Access, 直接内存访问) 是一种 **计算机系统** 中常用的数据传输方式, 它可以让设备在不占用 CPU 时间的情况下, 直接访问内存, 实现高速数据传输。在数据传输量大、速度要求高的场景中, DMA 可以大大提高系统性能。

DMA 传输过程中, 通常需要使用一个特殊的硬件设备——DMA 控制器。DMA 控制器是一种专门的外设, 可以控制数据传输的方向和流程, 通过中断或者 I/O 端口与 CPU 通信。DMA 控制器可以实现从内存到设备、从设备到内存、设备之间等多种传输方式。

1. DMA 控制器驱动程序

DMA 控制器驱动程序是实现 DMA 传输的核心。它需要控制 DMA 控制器的寄存器, 设置 DMA 传输参数, 开启 DMA 传输等操作。DMA 控制器驱动程序需要根据硬件平台的不同进行适当的修改和调整。

在 Linux 系统中, DMA 控制器驱动程序需要实现 dmaengine 驱动框架, 该框架是一个用于支持多种类型的 DMA 控制器和设备的框架。dmaengine 驱动框架提供了 DMA API 和 DMA 设备 API。DMA API 用于管理 DMA 传输, DMA 设备 API 用于管理 DMA 控制器和设备的注册、卸载等操作。

DMA 控制器驱动程序还需要注册中断处理函数, 以便在 DMA 传输完成时能够及时响应中断请求。中断处理函数需要根据 DMA 控制器的具体实现进行相应的修改和调整。

2. 设备驱动程序

设备驱动程序需要使用 DMA 控制器来进行数据传输。在 Linux 系统中, 设备驱动程序通常需要使用 dma_alloc_coherent() 函数来为 DMA 分配内存, 并使用 dma_map_single() 或 dma_map_sg() 函数将 **内存映射** 到 DMA 控制器中。dma_alloc_coherent() 函数可以保证所分配的内存是连续的, 适合 DMA 传输。

在数据传输完成后, 设备驱动程序需要使用 dma_unmap_single() 或 dma_unmap_sg() 函数来解除内存映射, 释放 DMA 控制器和内存资源。

3. DMA 引擎框架

DMA 引擎框架是一个可扩展的框架, 用于管理各种类型的 DMA 控制器和设备。它包含了 DMA API, 用于实现 DMA 传输, 以及 DMA 设备 API, 用于管理 DMA 控制器和设备的注册、卸载等操作。DMA 引擎框架可以方便地与其他驱动程序集成, 提高了 DMA 的可用性和灵活性。

DMA 引擎框架中的 DMA API 是通用的, 适用于各种类型的 DMA 控制器和设备。它包括了 DMA 预处理、DMA 描述符管理、DMA 操作执行等函数。DMA 设备 API 是专门针对某种类型的 DMA 控制器和设备实现的, 可以通过 DMA 引擎框架进行管理。

4. DMA 缓存管理

DMA 缓存管理是为了解决 DMA 传输中的缓存一致性问题。由于 CPU 与 DMA 控制器之间使用的是不同的总线, 因此存在缓存数据不一致的情况。

DMA 缓存管理机制通过 dma_map_single() 和 dma_map_sg() 函数将内存映射到 DMA 控制器中, 以确保缓存数据的一致性。在 DMA 传输完成后, 使用 dma_unmap_single() 和 dma_unmap_sg() 函数将内存解除映射, 以便 CPU 可以访问缓存数据。

除了使用 `dma_map_single()` 和 `dma_map_sg()` 函数，还可以使用 `dma_sync_single_for_device()` 和 `dma_sync_single_for_cpu()` 函数来实现缓存一致性。这两个函数分别用于将 DMA 写入的数据同步到内存中和将内存中的数据同步到 DMA 读取的缓存中，以确保数据一致性。

5. DMA 内存分配

DMA 内存分配是指在 DMA 传输过程中为数据分配内存。由于 DMA 控制器需要直接访问内存，因此必须分配与硬件平台相适应的内存。在 Linux 系统中，可以使用 `dma_alloc_coherent()` 函数分配 DMA 内存。`dma_alloc_coherent()` 函数分配连续的物理内存，确保 DMA 控制器能够访问这些内存。

6. DMA 原语

DMA 原语是用于实现 DMA 操作的基本函数。在 Linux 系统中，DMA 原语包括 DMA `memcpy`、DMA `memset` 和 DMA `xor` 等。DMA `memcpy` 用于在内存和 DMA 控制器之间传输数据，DMA `memset` 用于将内存或 DMA 控制器中的数据初始化为指定的值，DMA `xor` 用于计算校验和等。

7. DMA 测试工具

在进行 DMA 相关的开发和调试时，需要使用 DMA 测试工具来验证 DMA 的正确性和性能。Linux 内核提供了多个 DMA 测试工具，包括 `dma-test`、`dmaengine-test` 和 `dmatest` 等。这些工具可以测试 DMA 传输的正确性、性能和稳定性，以及验证 DMA 控制器驱动程序和设备驱动程序的正确性。

总之，Linux 系统中的 DMA 实现涉及多个方面，包括 DMA 控制器驱动程序、设备驱动程序、DMA 引擎框架、DMA 缓存管理、DMA 内存分配、DMA 原语和 DMA 测试工具等。在开发 DMA 相关的应用程序时，需要对这些方面有一定的了解，并根据硬件平台的不同进行适当的调整和修改。