

基于DMA的多通道大容量数据实时通信设计与实现

周志为

国电南瑞科技股份有限公司

DOI:10.12238/acair.v1i1.6126

[摘要] 针对多通道大容量数据实时通信系统中CPU通过Local bus传输通信数据大量占用CPU处理时间的问题,提出一种基于CPU+FPGA架构采用链式DMA进行多通道大容量数据接收和发送的设计,不同于DMA乒乓缓存机制,该设计仅需在CPU内存开辟单缓存区,大大节约DMA所需的CPU内存资源;成功实现链式DMA数据传输和CPU数据运算的高效并行处理,提升了系统的性能。

[关键词] 链式DMA; 单缓存; 实时通信; 多通道; 大容量

中图分类号: TP274 **文献标识码:** A

Design and Implementation of Multi-channel and Large-capacity Data Real-time Communication Based on DMA

Zhiwei Zhou

NARI Technology Co., Ltd

[Abstract] Aiming at the problem that CPU consumes a lot of CPU processing time when transmitting communication data through local bus in multi-channel and large-capacity data real-time communication system, a design of receiving and sending multi-channel and large-capacity data by chain DMA based on CPU + FPGA framework is proposed, which is different from DMA dual buffer mechanism. This design only needs to open a single cache area in CPU memory, which greatly saves CPU memory resources needed by DMA, successfully realizes the efficient parallel processing of chain DMA data transmission and CPU data operation, and improves the performance of the system.

[Key words] chain DMA; single cache; real-time communication; multichannel; large-capacity

前言

在工业控制领域,采用定时中断进行实时任务处理的嵌入式系统得到了广泛应用。为了满足规定的中断时间内多通道数据实时通信的要求,很多系统采用了中央处理器(Central Processing Unit, CPU)+现场可编程门阵列(Field Programmable Gate Array, FPGA)的架构方式。由于FPGA可以并行地完成多通道通信数据的接收和发送,所以这种方式可以减少CPU的处理时间。

目前FPGA一般通过双端口随机存取存储器(Dual Port Random Access Memory, DPRAM)和CPU进行数据交互。在CPU和FPGA之间有多通道大容量数据需要传输的场合,这种方式CPU读写FPGA数据的时间无法忽略,尤其是当CPU以Local bus异步模式读写FPGA数据时,随着嵌入式系统的功能不断丰富,实时任务不断增加,这种方式甚至有可能造成系统无法在规定的中断时间内完成全部的实时任务。

尽管目前可以采用PCI-E、SRIO等方式实现CPU和FPGA之间的快速数据传输,但是这些方式对CPU和FPGA具有较高的要求,而且可能需要更改现有的硬件设计,在某些成本比较敏感或者

硬件不易更换的场合是无法使用的。

作为CPU的通用配置,直接存储器访问(Direct Memory Access, DMA)可以独立于CPU实现外设与存储器之间,以及存储器与存储器之间的快速数据传输,大大提高系统的数据传输效率和性能。

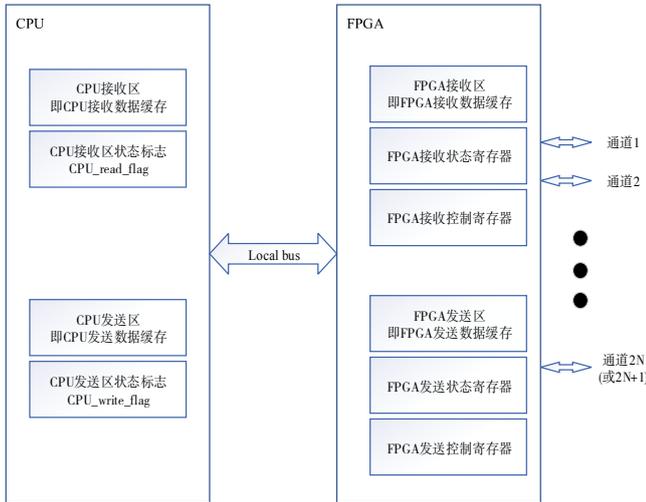
DMA通常有块传输方式和链式传输方式。块传输DMA方式通过直接设置寄存器启动DMA数据传输,如果有多个数据块需要传输,则每个数据块传输完成后需要重新设置寄存器开启新的DMA数据传输。由于数据块之间传输延时较长,这种DMA方式在执行多个数据块传输时效率较低。链式DMA方式是指将所有数据块的DMA配置保存为链表描述符,当一个数据块传输完成后, DMA控制器通过链表描述符指针自动开启下一个数据块的传输。由于数据块之间传输延时较短,这种DMA方式适合地址不连续的多个数据块的传输。

本文提出了一种不需要更改现有硬件设计,采用链式DMA分时复用缓存区读写FPGA数据的方法。不同于普遍采用的DMA乒乓缓存机制需在CPU内存开辟双缓存区,该方法只需在CPU内存开

辟单缓存区,大大节约了DMA所需的CPU内存资源。在CPU和FPGA之间有多通道大容量数据需要传输、CPU内存资源紧张的情况下,采用这种方式实现了链式DMA数据传输和CPU数据运算的高效并行处理,大大降低了实时任务所需的中断时间,提升了系统的性能。

1 嵌入式系统结构与通道数据收发机制

嵌入式系统的基本结构如图1所示。



注: CPU_read_flag为1时表示CPU接收区有新数据, CPU可以读取处理; CPU_write_flag为1时表示CPU发送区空闲, CPU可以写入新数据。

图1 嵌入式系统结构示意图

如图1所示, CPU通过Local bus与FPGA进行数据交互, FPGA与2N(或2N+1)个外部通道并行进行固定频率的数据接收和发送。

1.1 通道数据接收机制

通道数据接收的流程如图2所示。

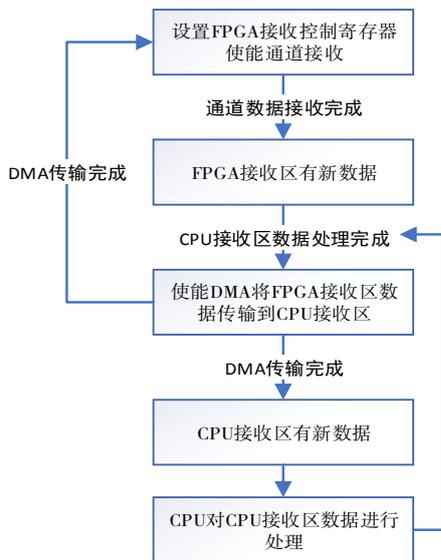


图2 通道数据接收流程图

当CPU通过设置FPGA接收控制寄存器使能通道接收后, FPGA会接收相应通道的数据存放在FPGA接收区, 接收完成后, FPGA会更新FPGA接收状态寄存器。

当FPGA接收区有新数据且CPU接收区空闲(没有数据或数据已经处理过,即CPU_read_flag=0)时, CPU使能DMA将FPGA接收区数据传输到CPU接收区, 传输完成后, 设置FPGA接收控制寄存器再次使能通道接收外部数据, 并且将CPU_read_flag置1(指示CPU接收区有新数据等待处理)。

当CPU接收区有新数据(即CPU_read_flag=1)时, CPU对这些数据进行处理, 处理完成后, 将CPU_read_flag置0(指示CPU接收区数据已经处理完成)。

1.2 通道数据发送机制

通道数据发送的流程如图3所示。

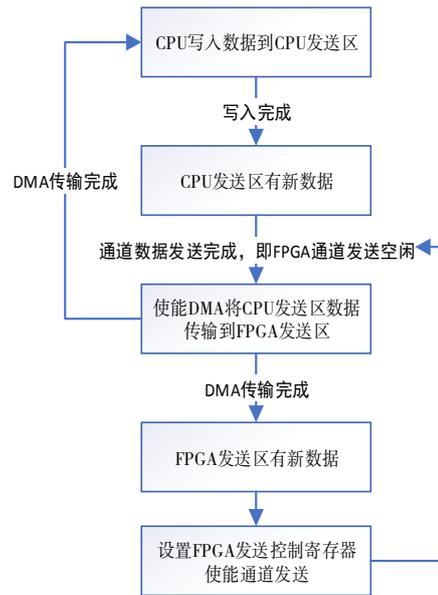


图3 通道数据发送流程图

当CPU发送区空闲(即CPU_write_flag=1)时, CPU将需发送的数据写入CPU发送区, 写入完成后, 将CPU_write_flag置0(指示CPU发送区有新数据等待传输)。

当CPU发送区有新数据(即CPU_write_flag=0)且FPGA通道发送空闲时, CPU使能DMA将CPU发送区数据传输到FPGA发送区, 传输完成后, 将CPU_write_flag置1(指示CPU发送区空闲, 可以写入新数据), 并且设置FPGA发送控制寄存器将FPGA发送区数据发送出去, 发送过程中FPGA发送状态寄存器保持通道发送忙状态, 发送完成后, 更新为FPGA通道发送空闲状态。

2 多通道链式DMA数据收发设计

定时中断的处理时序如图4所示。

由图4可知, 在CPU处理通道N+1~2N(或2N+1)的数据时, DMA进行通道1~N的数据传输; 在CPU处理通道1~N的数据时, DMA进行通道N+1~2N(或2N+1)的数据传输, 这样就实现了DMA数据传输和CPU数据运算的高效并行处理, 大大降低了实时任务所需的中断时间。而且这种设计不同于DMA乒乓缓存机制, 只需在CPU内存开辟单缓存区, CPU和DMA分时复用该缓存区即可, 这样就大大节约了DMA所需的CPU内存资源。

以定时中断1为例, 程序流程具体如下:

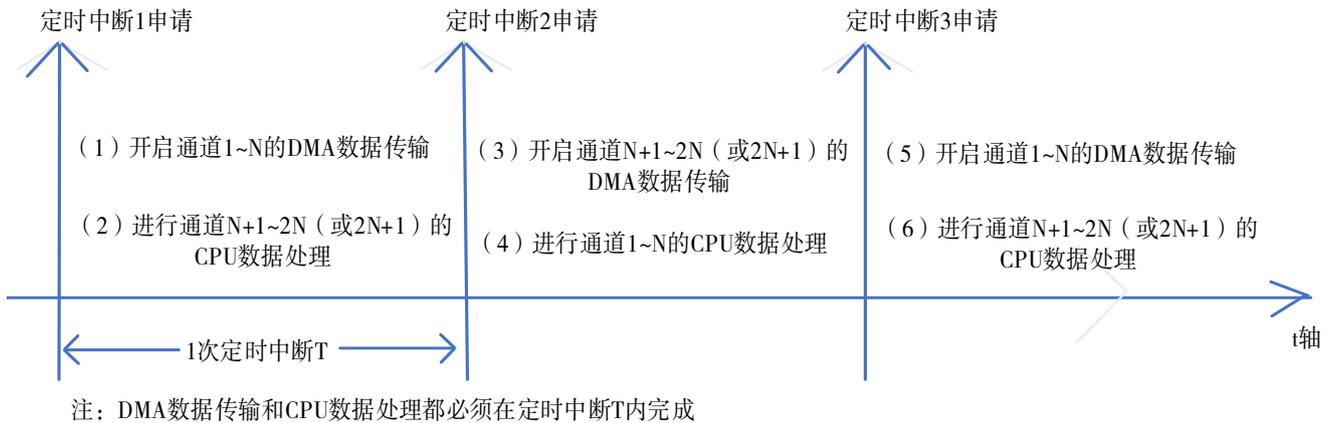


图4 定时中断处理时序图

(1) 开启通道1~N的DMA数据传输, DMA数据传输在定时中断T内完成。进入定时中断服务程序后。

①由于上个中断开启的DMA传输此时已经完成, 将参与上个中断DMA数据接收通道的CPU_read_flag置1(指示CPU接收区有新数据等待处理), 并且设置FPGA接收控制寄存器再次使能该通道接收外部数据; 将参与上个中断DMA数据发送通道的CPU_write_flag置1(指示CPU发送区空闲, 可以写入新数据), 并且设置FPGA发送控制寄存器将该通道的FPGA发送区数据发送出去; ②CPU查看通道1~N的FPGA接收状态寄存器, 如果某一通道的FPGA接收区有新数据且CPU接收区空闲(即CPU_read_flag=0)时, 则设置该通道的DMA配置并通过链表描述符加入链式DMA队列; ③CPU查看通道1~N的CPU_write_flag, 如果某一通道的CPU发送区有新数据(即CPU_write_flag=0)且FPGA通道发送空闲时, 则设置该通道的DMA配置并通过链表描述符加入链式DMA队列; ④使能链式DMA开始数据传输。(2) 进行通道N+1~2N(或2N+1)的CPU数据处理, CPU数据处理在定时中断T内完成。使能链式DMA后, ①CPU查看通道N+1~2N(或2N+1)的CPU_read_flag, 如果某一通道的CPU接收区有新数据(即CPU_read_flag=1)时, 则CPU对该通道的CPU接收区数据进行处理, 处理完成后, 将CPU_read_flag置0(指示CPU接收区数据已经处理完成); ②CPU查看通道

N+1~2N(或2N+1)的CPU_write_flag, 如果某一通道的CPU发送区空闲(即CPU_write_flag=1)时, 则CPU将需发送的数据写入该通道的CPU发送区, 写入完成后, 将CPU_write_flag置0(指示CPU发送区有新数据等待传输)。

3 结束语

该设计已在电力系统安全自动装置中得到运用。链式DMA的使用, 实现了数据传输和数据运算的高效并行处理, 提升了装置的性能。该设计实现简单, 可应用于利用FPGA实现多通道数据采集或实时通信的系统中。

[参考文献]

- [1] 时磊, 张铁军, 王东辉. 高速多通道DMA控制器的设计与实现[J]. 微计算机应用, 2010, 31(05): 50-53.
- [2] 曾文龙, 周游, 贺珊, 等. 基于多通道DMA控制器的媒体播放器SoC设计与验证[J]. 中国科技论文, 2014, 9(01): 45-48.
- [3] 苏永海, 黄莉. 基于PCI Express的多通道动态优先级DMA系统的FPGA设计[J]. 通信技术, 2017, 50(07): 1570-1575.
- [4] 陈骄阳. 高速多通道DMA控制器的设计[J]. 信息与电脑(理论版), 2009, (16): 37-38.
- [5] 石文侠, 吴龙胜, 盛廷义, 等. 一种支持全双工数据传输的多通道DMA控制器设计[J]. 微电子学与计算机, 2015, 32(02): 76-79+83.