

基于 Zynq-7000 的 SRIO 高速数据传输设计与实现

杜金艳, 叶旭鸣

(天津津航计算技术研究所 天津 300308)

摘要: 为了满足 Zynq-7000 系列芯片的 SRIO 数据传输要求, 提出了一种基于 FPGA 控制 DMA 传输进行 SRIO 通信的设计方案, 并完成了 ARM 与 FPGA 核间高吞吐率的数据交互操作。系统的 FPGA 部分主要用来控制 DMA 的数据传输和 SRIO 事务处理, ARM 只进行简单的参数设置, 能够完成 SRIO 各种事务类型的传输。实际应用表明, 该系统具有操作简便、高效的特点, 同时减轻了 CPU 的负担, 达到了预期效果。

关键词: Zynq-7000; 高速数据传输; DMA; 高效

中图分类号: TN79+1

文献标识码: A

文章编号: 1674-6236(2017)24-0030-04

Design and implement of SRIO high-speed data transmission base on Zynq-7000

DU Jin-yan, YE Xu-ming

(Tianjin JinHang Computing Technology Research Institute, Tianjing 300308, China)

Abstract: In order to meet the requirements of SRIO data transmission of Zynq-7000 spectrum, a design scheme of SRIO Communication based on the FPGA control of DMA transmidtion is proposed, and the a high band-width communication between ARM and FPGA core is implemented. The FPGA part of the system is mainly used to control the data transmission of DMA and SRIO transaction processing. ARM only for simple parameter settings, complete the transmission of various types of transaction SRIO. real application shows that the system has the characteristic of simple operation and high efficiency, at the same time release the main CPU, The proposed method achieves the expected effect.

Key words: Zynq-7000; high-speed data transmission; DMA; high efficiency

Xilinx 公司推出的 Zynq-7000 系列全可编程片上系统 (SoC, System on a Chip) 采用了微处理器加可编程逻辑的结构, 集成 ARM Cortex A9 双核 (PS, Processing System) 以及最多可达相当于 500 多万个逻辑门可编程逻辑 (PL, Programmable Logic) 单元, 能够灵活地用于各种目的的应用^[1]。而 SRIO 通信通常用于两个端点器件间数据传输工作, 只需要处理 SRIO 的端点设计, 与以往不同, 系统是要将 SRIO 数据自动传输到 PS 的内存中。因此, 系统采用以集成 ARM 处理器为中心, FPGA 可编程逻辑部分为扩展子系统实现数据传输, 通过 PS 和 PL 软硬件协同处理实现数据的高速传输、存储和处理。

1 系统架构

在设计中选用了 Xilinx 公司的 Zynq 芯片 XC7Z045,

主要实现数据传输和数据处理功能。系统结构原理框图如图 1 所示。在系统的工作过程中, 通过 SRIO 接口将数据接收到集成 ARM 的内存中, 便于 ARM 对数据进行数据处理。

根据软硬件协同设计方法的软硬件划分原则: 高速、低功耗由硬件实现; 多品种、小批量由软件实现^[1]。系统软硬件划分为: 数据传输部分由 PL 完成, 数据处理部分由 PS 完成。因此数据传输不能过多的占用 ARM 的处理时间, 为减轻 PS 的负担, 要尽量用 FPGA 部分来处理。

2 方案设计

根据 Zynq-7000 的处理能力, 基本能满足日常的各种数据处理需求, 在搭建的平台上有两种可选的数据处理方案。

收稿日期: 2016-10-08 稿件编号: 201610088

作者简介: 杜金艳 (1977—), 女, 辽宁黑山人, 硕士研究生, 高工。研究方向: 电路与系统。

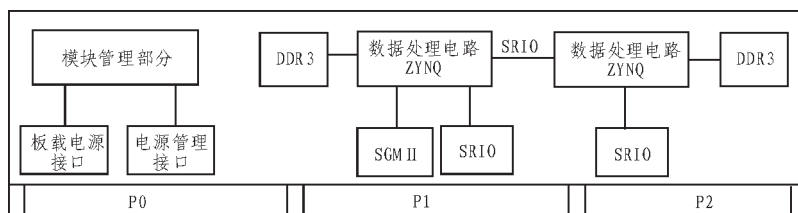


图1 系统结构原理框图

1)应用 MicroBlaze“软”处理器配合 FPGA 完成。

MicroBlaze 有很多不同的架构选项,可以根据目标应用的需求纳入或排除出处理器的具体实现中。ARM 是专用的资源,而 MicroBlaze 位于逻辑部分^[9]。在设计中, MicroBlaze 处理器可以负责协调特定的底层功能与系统之间的配合, FPGA 只做简单的数据协议转换,从而提升整体的性能。

2)完全由 FPGA 完成该部分功能。

应用 FPGA 来进行处理可以提高处理效率,用 FPGA 完成 DMA(Direct Memory Access, DMA) 控制器设计,这样当 DMA 控制器在做传输的时候,处理器就可以从事其他任务了。在这种情况下, DMA 控制器既是总线主机也是总线从机。作为主机, DMA 控制器要和存储控制器通信,也会要请求总线仲裁。作为从机, DMA 控制器回应从总线主机(大多数时候就是处理器)而来的请求,建立起存储传输。

第一种方案具有更高的普遍性与可扩展性, FPGA 接口处理要比第二种方案简单,系统的 SRIO 发送与接收数据都在软核上处理,传输效率较低,在数据处理速度上会慢于方案二。方案二最大的特点是使用 FPGA 架构对数据传输提速,要求设计人员

对 SRIO 协议有一定的了解。因此,方案二适用于对速度和实时性要求比较高的场合,这将比第一种方案在速度上有不少优化。

3 设计实现

采用第二种方案,进行 FPGA 设计。将 SRIO 数据接收到 ARM 的内存中,首先需要考虑集成 ARM 的 DDR 存储器与 FPGA 的数据通道。

集成 ARM 的 DDR 存储器控制器包括 3 个主要的模块^[3]:一个核心存储器控制器和调度器(DDRC)、一个 AXI 存储器端口接口(DDR1)和一个数字 PHY 以及控制器(DDRP)。对 PS 和 PL 的共享存储器的共享访问是通过多端口 DDR1 来支持的,它具有 4 个 AXI 从机端口来满足这个要求:

1) PL 通过两个专用的 64 位端口(AXI_HP)来访问。

2)一个 64 位端口通过 L2 cache 控制器专用于 ARM CPU。这个端口可以被配置为低延迟。

3)所有其他 AXI 主机通过中央互联共享剩下的端口。

PS 部分有 4 个高性能 AXI_HP(High Performance,

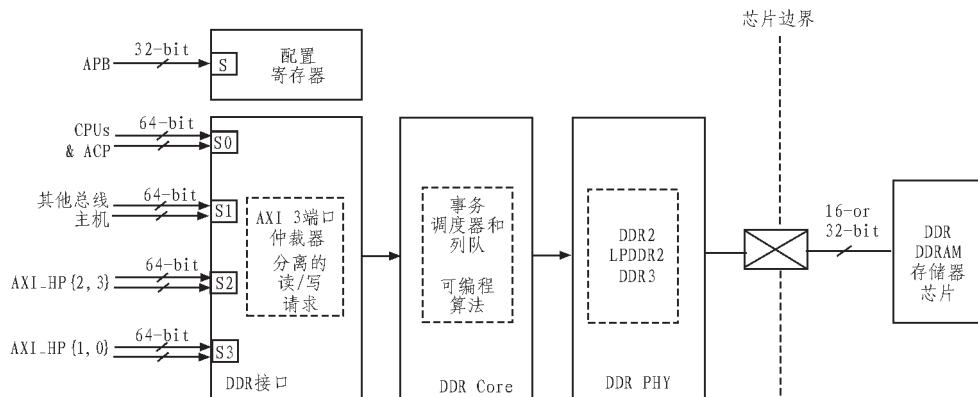


图2 DDR 存储器控制器框图

HP)接口来实现从 PL 总线主机到 OCM(On Chip Memory, OCM)和 DDR 存储器的高带宽数据通路^[8]。PS 部分的 ACP 端口提供了 PL 中所实现的加速器和 PS 之间的低延迟的链路,而且对于 L1 和 L2 cache 带有可选的一致性操作能力。这是一个 64 位的接口,

使得 PL 可以实现一个能访问 OCM 和 L2 cache 的 AXI 主机。因此能保证数据的内存一致性。通过以上分析 PS 各种 AXI 接口, PS 对外发送的数据通道采用 AXI_HP 接口,应用 ACP 通道进行 PS 接收数据的通道。

根据数据流的传输方向和DMA操作方向的映射,即写操作为MM2S(Memory Map to Stream),读操作为S2MM(Stream to Memory Map)操作。具体的设计实现如图3所示。

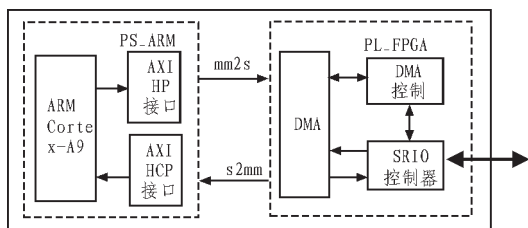
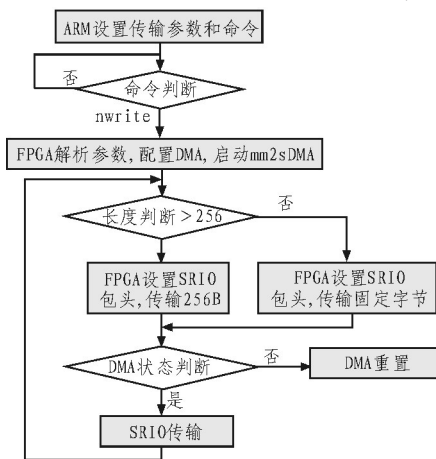


图3 具体设计实现框图

3.1 DMA及DMA控制模块

AXI DMA高速传输的实现采用AXI DataMover引擎IP核实现高速的数据传输功能^[4],而AXI-DMA引擎IP核也是实现从PS内存到PL高速传输高速通道^[5],两者主要在于主动与被动的区别。AXI-DMA中PS主动的,但是AXI DataMover是完全由PL控制的,PS



是完全被动的,因此DMA采用DataMover IP核。

DataMover IP核用来实现存储器接口(AXI4 Memory Mapped)和流接口(AXI4 STREAM)的转换,是数据传输的通道;状态接口主要用来寄存DMA的工作状态等信息,用户可通过该接口读取这些状态信息;配置寄存器模块主要用来对AXI_DataMover进行初始化的配置以使其能正常的工作。

根据SRIO协议的传输特点,SRIO的IP核采用HELLO包格式^[7],每包最大数据负载是256个字节,而对于小于256字节的长度,IP核规定了几种固定的长度传输,因此对于任意长度数据的处理,需要进行数据长度的解析判断,分多次进行传输。

DMA控制模块根据ARM命令配置AXI DataMover的各个参数,并启动DMA操作,同时将各种状态通过状态接口返回给ARM。SRIO主动发起的NWRITE和NREAD事务DMA的操作流程如图4,当SRIO作为被动时的处理方法相似。

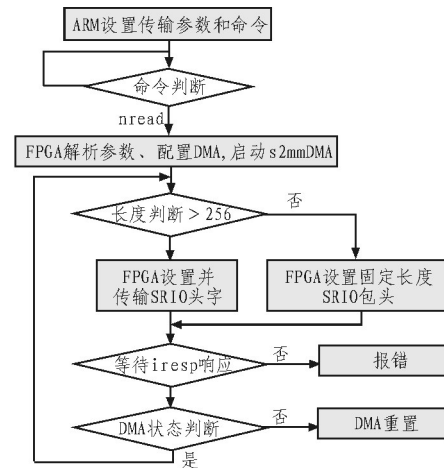


图4 DMA控制模块框图

3.2 SRIO控制模块

SRIO数据传输过程是基于请求和响应机制的。SRIO的IP核应用AXI4 Stream接口,IP核的端口分别可以设置为两种类型,通用I/O和发起(Initiator)/响应(Target);同时又可以分为两种包格式:HELLO和SRIO Stream。通用I/O类型可以减少发送和接收I/O包的通道数,仅用到ioTx和ioRx端口;而发起/响应类型是允许应用到的事务端口分离,分别为IREQ(Initiator Request)、IRES(Initiator Response)、TREQ(Target Request)、TRESP(Target Response)。为便于设计,采用了发起/响应类型端口,HELLO包格式。

SRIO模块中,FPGA根据ARM命令参数,在

SRIO主动发起时根据不同的事务类型(nwrite、nread和swrite,doorbell等)和数据长度。SRIO模块会自动生成头字数据,组织请求数据包;数据包经过SRIO链路发送到目标端口,若是带有响应的事务,还要查询IRES端口,接收响应数据。反之在被动时,要解析SRIO的包头字信息,识别出是何种事务类型、数据长度等信息再进行处理。

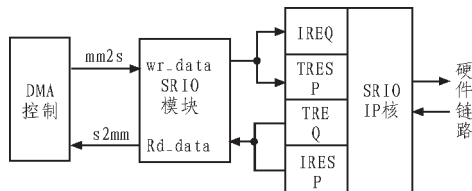


图5 SRIO模块传输示意图

4 功能验证

为了验证所设计的高速SRIO传输的性能,按照SRIO规定的事务类型对其进行性能测试,由于传输过程中的数据解析及DMA状态的判断都需要占用一定时间开销,因此,SRIO实际传输速率跟理论值是有一定差距的。

系统将SRIO总线速率设置为2.5 Gbps模式,在PS侧进行SRIO Nwrite事务测试,将PS_ARM内存1 GB数据进行传输,用定时器进行测试,测试结果为从ARM命令开始到SRIO完成发送,用时约1.7 s,通过计算得到传输效率约为线速的60%左右,已满足应用要求。

5 结 论

针对ZYNQ异构处理器结构特征,通过对ZYNQ芯片及SRIO的接口及协议的分析,设计并实现了一种基于ZYNQ芯片的SRIO数据传输方案,并详细介绍了FPGA的实现过程,最后并对其进行了测试。此方案提高了SRIO数据自动传输到PS的内存中的可靠性和安全性,同时也提高了整个系统工作的效率。随着应用高速通信越来越广泛,对数据实时处理的要求会越来越严格,该方案的实现,对未来高速通信有着深远的影响。

参考文献:

- [1] 陆佳华,江舟,马岷. 嵌入式系统软硬件协同设计实战指南-基于Xilinx Zynq[M]. 北京:机械工业出版社,2012.
- [2] 何宾. Xilinx all programmable Zynq-7000Soc[M]. 北京:清华大学出版社,2013.
- [3] Xilinx Inc.Zynq-7000All Programmable Soc Technical Reference Manual(V1.10),2015.
- [4] Xilinx Inc.AXI DMA v7.1 LogiCORE IP Product Guide[EB/OL].[2014-04-01].http://www.xilinx.com/support/documentation/ip_documentation/axi_dma/v7_1/pg021_axi_dma.pdf.
- [5] Xilinx Inc.AXI DataMover v5.1 LogiCORE IP Product Guide[EB/OL].[2014-04-01].http://www.xilinx.com/support/documentation/ip_documentation/axi_datamover/v5_1/pg022_axi_datamover.pdf.
- [6] Xilinx Inc.AXI Reference Guide v14.3[EB/OL].[2014-11-15][en- tation/ip_documentation/axi_ref_guide/latest/ug761_axi_reference_guide.pdf](http://www.xilinx.com/support/docum).
- [7] Xilinx Inc.Serial RapidIO Gen2 Endpoint v3.2[EB/OL].[2014-10-15].http://www.Xilinx.com/support/documentation/ip_documentation/Serial_RapidIO_Gen2_Endpoint/v3_2/pg007_srio_gen2.pdf.
- [8] Xilinx Inc.Zynq-7000 Technical Reference Manual v1.7[EB/OL].[2014-2].http://www.xilinx.com/support/documentation/user_guides/ug585-Zynq-7000-TRM.pdf.
- [9] Xilinx Inc.LogiCORE IP MicroBlaze Micro Controller System, Product Specification[EB/OL].[2012-4].http://www.xilinx.com/support/documentation/sw_manuals/xilinx14_1/ds865_microblaze_mcs.pdf.
- [10] Xilinx Inc.AXI Interconnect v2.1 LogiCORE IP Product Guide[EB/OL].[2014-10-01].http://www.xilinx.com/support/documentation/ip_documentation/axi_interconnect/v2_1/pg059_axi_interconnect.pdf.
- [11] Xilinx Inc. AXI4 Stream Interconnect v1.1 LogiCORE IP Product Guide[EB/OL].[2014-12-01].http://www.Xilinx.com/support/documentation/ip_documentation/axis_interconnect/v1_1/pg035_axi_interconnect.pdf.
- [12] Xilinx Inc.Zynq-7000 All Programmable Soc Overview v1.6, [EB/OL].[2013-12-01].http://www.xilinx.com/support/documentation/data_sheets/ds190-Zynq-7000-Overview.pdf.
- [13] ARM, Cortex-A9 MPCore Technical Reference Manual, r3p0版, [EB/OL].[2011-07]http://infocenter.arm.com/help/topic/com.arm.doc.ddi0407g/DDI0407G_cortex_a9_mpcore_r3p0_trm.pdf.
- [14] Xilinx, Inc., ISE to Vivado Design Suite Migration Guide, v2014.1[EB/OL].[2014-4].http://www.xilinx.com/support/documentation/sw_manuals/xilinx2014_1/ug911-vivadomigration.pdf.
- [15] ARM, "ARM Generic Interrupt Controller: Architecture Specification" v1.0, [EB/OL].[2008-9].http://infocenter.arm.com/help/topic/com.arm.doc.ih0048a/IHI0048A_gic_architecture_spec_v1_0.pdf.