

基于 FPGA 的高速数据转存系统^①

张鹏飞, 杜雨铭

(成都信息工程学院 电子工程学院, 成都 610225)

摘要: SAR 雷达前端数据采集系统速度与存储系统容量的日益提高, 对数据转存系统的性能提出了更高的要求。以 Xilinx 公司 Virtex-5 系列 FPGA 为硬件平台, 基于 PCI-E 协议与 Aurora 协议提出了一种高速数据转存系统解决方案。在 Aurora 协议基础之上, 自定义了一种可靠的帧格式; 利用双口 RAM 对数据进行缓存, 以乒乓方式操作, 确保了在输入、输出数据传输速率不匹配情况下数据传输的稳定性与可靠性; 转存系统与 PC 通过 PCI-E 总线进行通信, 命令交互采用 PIO 模式, 图像与列表数据传输采用 DMA 模式。经实际测试, 该方案能够满足某型号 SAR 雷达系统对高速数据转存的要求。

关键词: FPGA; PCI-E; Aurora; 高速数据转存; DMA

High Speed Data Transfer and Storage System Based on FPGA

ZHANG Peng-Fei, DU Yu-Ming

(College of Electronic Engineering, Chengdu University of Information Technology, Chengdu 610225, China)

Abstract: With the increasing capacity of the storage system and the speed of the SAR radar front-end data acquisition, the higher request of the performance of data transfer and storage system is put forward. A solution of high speed data transfer and storage system based on the PCI-E and the Aurora protocol using the hardware platform of Xilinx's Virtex-5 series FPGA is proposed. Customizing a reliable frame format on the basis of Aurora protocol. Using the dual-port RAM for data cache and operating in ping-pang mode ensures the stability and reliability of the data transfer in the case of transfer speed mismatch of the input and output data. The transfer and storage system and PC via the PCI-E bus for communicating, command interaction using the PIO mode and image and list data transfer using the DMA mode. Actual test shows this solution can meet the request of a type of SAR radar system for high speed data transfer and storage.

Key words: FPGA; PCI-E; Aurora; high speed data transfer and storage; DMA

合成孔径雷达(Synthetic Aperture Radar, 简称 SAR)是一种主动微波成像雷达, 具有全天时、全天候、分辨率高、多极化、视角可变、穿透性强等特点^[1]。SAR 雷达数据转存系统是 SAR 雷达系统的重要组成部分, 负责将记录的 SAR 回波数据转存至本地硬盘供后期处理^[2]。雷达前端数据采集速度与数据记录设备容量的日益提高, 对数据转存系统的数据转存速度提出了更高的要求。

传统的并行传输技术已接近理论上限, 但仍不能满足需求, 因此串行传输技术重新返回到高速传输领域, 并引领新一代吉比特传输技术。在目前系统级互

连速率已达到 Gbps(Gigabit per second)的设计中, 先进的高速串行传输技术迅速取代传统的并行传输技术, 成为业界的主流。高速串行传输技术不仅能够带来更高的性能、更低的成本和更简化的设计, 克服了并行传输的速度瓶颈, 还节省了 I/O 资源, 使印制板的布线更简单。因此, 被越来越广泛地应用于各种系统设计中, 包括 PC、消费电子、海量存储器、服务器、通信网络、工业计算和控制、测试设备等。本设计以 Xilinx 公司的 Virtex-5 系列 FPGA(Field Programmable Gate Array)为硬件平台, 提出了一种基于 PCI-E 协议和 Aurora 协议的高速数据转存系统解决方案。

^① 收稿时间:2013-07-08;收到修改稿时间:2013-08-19

1 系统概述

机载 SAR 雷达采集到的图像信息或卫星回传的图像信息存储在由一组 NAND Flash 阵列和两个单片 NAND Flash 芯片构成的固态硬盘(数据源)中,地面转存设备通过高速数据转存系统将固态硬盘中的数据转存到本地硬盘供后期处理.系统总体分成四部分:PCI-E 模块、数据流控制模块、Aurora 模块和时钟管理模块.系统总体框图如图 1 所示.

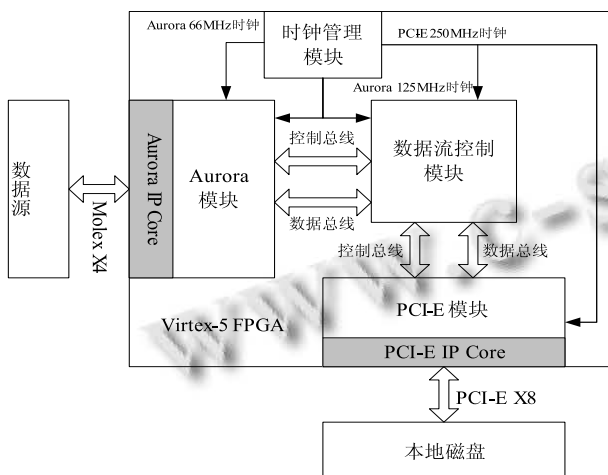


图 1 系统总体框图

FPGA 芯片采用 Xilinx 公司的 Vitex-5 系列的 XC5VSX95T. 该芯片集成了 16 个 RocketIO GTP 硬核模块. RocketIO GTP 硬核模块采用了 CML(Current Mode Logic)、CDR(Clock and Data Recovery)、8B/10B 编码和预加重等技术,极大地减小了时钟扭曲、信号衰减和线路噪声对接收性能的影响,线速率从 100Mbps 到 3.75Gbps^[3].

除了底层的物理层技术,该芯片还集成了基于 Rocket IO GTP 硬核的 Integrated Endpoint for PCI Express IP Core 和带 64bit LocalLink 用户接口的 Aurora 协议引擎的参考设计.

2 系统主要功能模块设计与关键技术

2.1 Aurora 模块设计

数据源与转存系统基于 Aurora 协议进行通信,将数据源内存储的图像数据与列表数据读入转存系统内部,并写入数据流控制模块的双口 RAM.

2.1.1 Aurora 协议简介

Aurora 协议是一个相对简单可定制的协议,只控

制链路层和物理层. Aurora 协议的设计理念是使其它高层协议,例如 TCP/IP 协议和以太网,可以很容易的运行在 Aurora 协议之上. Aurora 协议使用一个或多个高速串行通道构成更高速的链路.协议不仅定义了物理接口,还定义了包结构、嵌入其他协议的推荐程序、数据提取和流量控制.于此同时 Aurora 协议还是一个相对开放的协议,当标准协议不能满足要求,或者标准协议对于用户来说过于宽泛时,用户可以根据实际需要制定属于自己的协议^[4].本设计在基于 Aurora 底层协议的基础上制定了个性化协议.

2.1.2 帧格式定义

由于 Aurora IP Core 对包括物理层和数据链路层的 Aurora 底层协议进行了封装,为了满足转存系统的要求,因此制定了相应的顶层协议. Aurora 模块自定义命令帧格式和数据帧格式分别如表 1 与表 2 所示.命令帧的帧长度总是 64bit,数据帧长度根据传输数据长度的不同而不同.

表 1 命令帧格式

位域	63...32	31...16	15...0
字段	CRC32	LenSel	Cmd

Cmd: 命令字段,区分不同命令;

LenSel: 长度选择字段,在读取数据的命令中用于选择每次传输的数据长度;

CRC32: 校验字段,该字段为前 32 位数据的 CRC (Cyclic Redundancy Check)校验值.数据源端在接收到命令后,对接收的命令进行 CRC 校验,再与该字段进行比较从而验证接收到的数据正确与否.

表 2 数据帧格式

位域	63...0	...	63...0	63...0
字段	Dn	...	D1	Header

Header: 数据头字段,用于区分数据源回传的是图像数据还是列表数据;

D1...Dn: 数据字段,数据源传送的图像数据或列表数据.

2.2 数据流控制模块

完成上位机与数据源之间的命令与数据流向控制以及异步时钟域信号的同步处理.

2.2.1 双口 RAM 设计

根据 Aurora 模块接收到数据帧帧头的不同,分别将来自数据源的列表数据和图像数据写入各自对应的双口 RAM.

在实际工程中,数据源存储的图像数据大小为 500MB 左右,每次按 360KB 连续传输.存储的列表数据大小为 8KB.因此在设计双口 RAM 时,图像数据对应的双口 RAM 设计成 640KB,采用乒乓操作.数据源每次对双口 RAM 写满 320KB 数据,产生一个脉冲信号,用以触发 PCI-E 模块的数据发送状态,将该 320KB 的图像数据传给 PC.整个 500MB 的图像数据全部传输完成后,结束整个 DMA 传输过程.列表数据的双口 RAM 设计成 8KB,只需要一次 DMA 传输就能完成.

2.2.2 异步时钟域信号同步处理

由于 Aurora 模块的数据与控制信号的同步时钟为 125MHz,而 PCI-E 模块的数据与控制信号的同步时钟为 250MHz,二者不同频不同相,所以必须进行同步处理.

在实际设计中,同步单一信号和同步总线信号二者的设计方法是不同的.在本设计中,双口 RAM 写时钟采用 Aurora 模块 125MHz 时钟,双口 RAM 读时钟采用 PCI-E 模块 250MHz 时钟,从而实现对 Aurora 模块和 PCI-E 模块两个时钟域的隔离以及总线信号的同步.对于单一信号采用双锁存器法,即在一个信号从一个时钟域进入另一个时钟域之前,将该信号用新时钟域控制的两个锁存器连续锁存两次.经验证这种方法满足实际的要求,达到了信号同步的目的.

2.3 PCI-E 模块设计

上位机与转存系统之间通过 PCI-E X8 接口互连,基于 PCI-E Rev 1.1 协议,同时采用 DMA(Direct Memory Access)方式实现了转存系统与 PC 之间高速稳定的数据传输.

2.3.1 PCI-E 协议简介

PCI Express (Peripheral Component Interconnect Express)是 Intel 为取代 PCI 总线,支持 AWG(Arphone Working Group)制定的第三代 I/O 技术.采用目前业内最流行的点对点差分串行连接.根据位宽的不同 PCI-E 接口分为 X1、X2、X4、X8、X16 和 X32,其中 X2 用于内部接口而非插槽模式. PCI Express 是一个多层协议,包括事务层、数据链路层和物理层^[5].

2.3.2 PCI-E 接口实现

基于 PCI-E 总线接口的通信有多种实现方式.第一种实现方式使用专用的接口芯片(如 NXP 公司的 PX1011A 芯片).该方式中接口芯片实现了物理层、数

据链路层和事务层的逻辑控制,但该方式使用芯片多、PCB 布局布线复杂、成本较高.第二种实现方式使用可编程器件(如 FPGA)实现 PCI-E 各层的功能.该方式具有灵活的可编程性,但开发难度比较大,开发周期长.因此 Xilinx 公司为 PCI-E 设计提出了第三种快速高效的解决方案,即利用 Virtex-5 系列芯片内嵌的 Rocket I/O 模块、硬核 Endpoint PCI-E 模块及可选的软核 IP,大大降低了 PCI-E 的开发难度^[6].

转存系统设计中,使用 IP 核 Endpoint Block Plus for PCI Express,并配置成 PCI-E X8 链路,INx 传统中断方式.该 IP 核与 PCI Express Rev 1.1 协议完全兼容,线速率达到 2.5Gbps,支持 X1、X4 及 X8 等多种链路模式,支持 32Vector 的 MSI 中断,且具有时钟补偿、自动时钟回复、8B/10B 编码、流控制等多种功能^[7].

PCI Express 端点有两种数据传输模式,即 PIO(Programmed Input Output)模式和 DMA(Direct Memory Access)模式. PIO 模式中,每次数据传输都由 CPU 直接参与.在大数据传输时,这种模式不仅占用了宝贵的 CUP 资源而且传输速度与效率较低.相对于 PIO 模式, DMA 模式直接对内存进行读写,不需要 CPU 的直接参与,其数据传输速度与效率较 PIO 模式高出很多倍.

本设计中,PC 对转存系统相关参数进行配置以及对数据源发送相关命令时,采用 PIO 模式;PC 读取转存系统内图像数据与列表数据采用 DMA 模式.

2.3.3 基于 PCI-E 协议的命令交互

PC 与转存系统之间、转存系统与数据源之间的命令交互具体流程如图 2 所示.

由于转存系统与数据源之间的命令格式为 8 个字节(64 位),而 PIO 模式每次存储器写命令字长为 4 个字节(32 位),所以 PC 每写一个完整的命令或读一个完整命令反馈信息需要两次存储器写操作或读操作.

具体流程如下:

① PC 写命令:PC 向命令数据高 32 位寄存器和命令数据低 32 位寄存器分别写入命令字的高 32 位和低 32 位;

② PC 写命令更新:PC 发送完命令后,向命令更新寄存器写入指定值,用以通知转存系统命令写完成;

③ 转存系统转发 PC 命令:转存系统检测到命令更新后,将高低 32 位的命令数据,组成 64 位的命令数据转发给数据源;



图 2 PC、转存系统和数据源三者间命令交互

④ PC 查询命令反馈更新: PC 在发送完命令更新命令后, 查询命令反馈更新寄存器的值;

⑤ PC 读取命令反馈信息: 如果命令反馈更新寄存器的值与某个指定的值相同, 则表明命令反馈信息成功写入转存系统. PC 发起读取命令反馈信息命令, 转存系统响应命令, 并回传命令反馈信息.

⑥ PC 写命令反馈更新清除: 当 PC 完成读取命令反馈信息操作后, 向命令反馈更新寄存器写入指定值, 清除命令反馈更新标志.

2.3.4 基于 PCI-E 协议的 DMA 传输

DMA 是一种不经过 CPU 直接从系统内存存取数据的数据交换模式. 在 DMA 模式下, CPU 只需向 DMA 控制器下达指令, 让 DMA 控制器来处理数据的传送, 数据传送完成后再把信息反馈给 CPU, 这就在很大程度上减轻了 CPU 资源占有率提高了数据传输效率.

在 PCI-E 系统中有两种 DMA 模式: 系统 DMA 模式和总线控制 DMA 模式(Bus Master Direct Memory Access, BMD). 系统 DMA 模式通常由一个被总线上所有设备共享的 DMA 引擎构成. 目前很少有根复合体和操作系统对系统 DMA 模式提供支持. 然而总线控制 DMA 模式应用最为广泛. 总线控制 DMA 模式将 DMA 引擎集成在 PCI-E endpoint 设备内部^[8]. 这样每一个

PCI-E endpoint 设备对应一个独立的 DMA 引擎, 使系统设计更加灵活、方便、可扩展性强. 本设计采用总线控制 DMA 模式, BMD 设计的逻辑结构如图 3 所示.

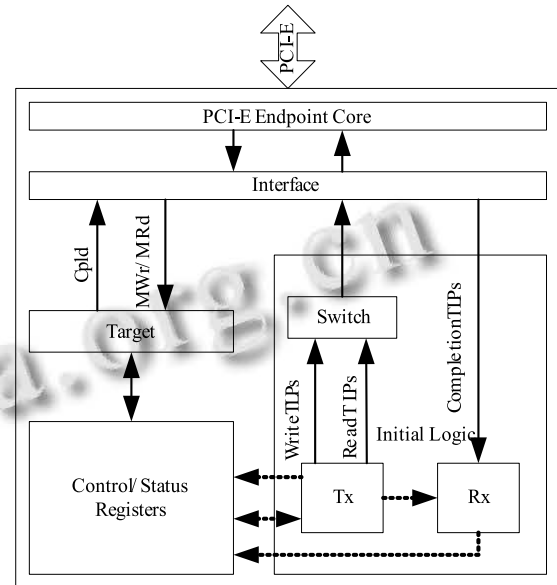


图 3 BMD 设计的逻辑结构

该结构包括接口模块 (Interface)、目标逻辑 (Target)、控制和状态寄存器 (Control/Status Registers)、初始化逻辑 (Initial Logic) 以及 PCI-E endpoint 模块 (PCI-E Endpoint Core). 目标逻辑用来捕获出现在接口上的单个 DW 存储器写或读事务, 也就是对图 3 中对控制寄存器的写和对状态寄存器的读, 目标逻辑实际上就是 PIO 读写操作. 控制和状态寄存器提供了 DMA 控制的各种操作信息. 初始化模块就是根据读写请求来产生存储器读或存储器写事务包. 其内部的 Tx 和 Rx 分别是发送和接收引擎, 用来发送和接收事务包^[9].

图像数据或列表数据 DMA 传输流程:

- ① 写读图像数据命令或读列表数据命令;
- ② 复位与 DMA 传输相关的控制和状态寄存器;
- ③ 取消初始化复位, 等待执行 DMA 相关操作;
- ④ 写 DMA 的 PC 侧存储数据的起始地址;
- ⑤ 写 DMA 事务包大小;
- ⑥ 写 DMA 事务包个数;
- ⑦ 写 DMA 传输启动;
- ⑧ 等待 DMA 传输完成中断, 传输完成.

数据传输是将图像或列表数据从数据源读入转存系统然后上传到 PC 内存最后写入本地硬盘, 也就是从数据源到 endpoint 设备最后通过 DMA 方式传到 PC 内存.

每次 DMA 传输数据量等于 DMA 传输事务包大小乘上每次 DMA 传输事务包个数. 通过改变以上参数可改变一次 DMA 传输数据量, 以满足不同工程实际的要求. 对于大数据传输往往需要多次连续的 DMA 传输才能完成. 实际测得数据传输速度如图 4 所示.

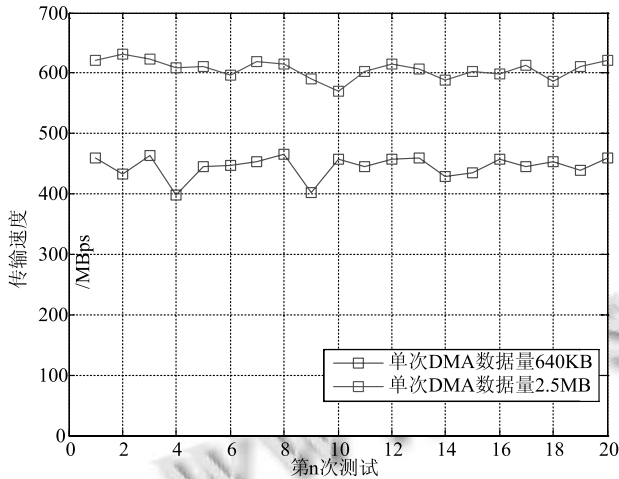


图 4 实测数据传输速度

在数据源端数据极限输出速度为 800MBps 的情况下, 当单次 DMA 传输数据量为 640KB 时, 系统数据总的传输速度在 450MBps 左右; 当单次 DMA 传输数据量为 2.5MB 时, 系统总的传输速度在 600MBps 左右.

3 FPAG资源开销与性能分析

3.1 FPGA 资源开销

该方案在 Xilinx 公司 XC5VSX95T 芯片上实现, 时钟理论最高频率可达到 550MHz, 具体资源开销如图 5 所示.

Number of BUFDSs	2 out of 8	25%
Number of BUFGs	6 out of 32	18%
Number of DSP48Es	1 out of 640	1%
Number of GTP_DUALs	6 out of 8	75%
Number of PCIEs	1 out of 1	100%
Number of PLL_ADVs	2 out of 6	33%
Number of RAMB18X2s	1 out of 244	1%
Number of RAMB36SDP_EXPs	2 out of 244	1%
Number of LOCed RAMB36SDP_EXPs	1 out of 2	50%
Number of RAMB36_EXPs	148 out of 244	60%
Number of LOCed RAMB36_EXPs	4 out of 148	2%
Number of Slices	3912 out of 14720	26%
Number of Slice Registers	8551 out of 58880	14%
Number used as Flip Flops	8543	
Number used as Latches	8	
Number used as LatchThrus	0	
Number of Slice LUTs	6987 out of 58880	11%
Number of Slice LUT-Flip Flop pairs	10999 out of 58880	18%

图 5 FPGA 资源消耗

设计中的 Slices 资源只用到 30% 左右, Bolock 利用率在 60% 左右, 节约了系统资源, 具有极大地可扩展性.

3.2 性能分析

理论上, 单通道的 Rocket I/O 支持最高 3.125Gbps 的线速率, 为 8B/10B 编码、同步、差错控制等操作预留 20% 的损耗, 仍然能够支持单通道 3.0Gbps 的线速率. 尽管受到 PCI-E Rev 1.1 协议中单方向单通道 2.5Gbps 线速率的限制, 同时考虑 8B/10B 编码, 实际单方向最大吞吐率为 16Gbps, 但足以满足该雷达系统总的转存速度 4Gbps 的要求.

4 结语

SAR 技术的发展对数据转存系统的传输速度和可靠性提出了更高的要求. 本文提出的基于 FPGA 的高速数据转存系统解决方案可有效地提高数据传输速率, 在某型号 SRA 雷达系统中得到了成功应用, 解决了数据传输的速度瓶颈. 同时本文给出了主要功能模块的设计方法, 详细地阐述了所涉及的关键技术. 方案的设计思想与方法也可为其他通信方面的应用提供有价值的参考, 具有广阔的应用前景.

参考文献

- 皮一鸣, 杨建宇, 付毓生, 杨晓波. 合成孔径雷达成像原理. 成都: 电子科技大学出版社, 2007: 1-10.
- 李建兵, 徐向辉. 基于 PCI-E 总线的高速数据传输卡的设计与实现. 计算机测量与控制, 2011, 19(10): 2581.
- Xilinx Inc. Virtex5 FPGA RocketIO GTP Transceiver User Guide. USA, Xilinx Inc. 2009.
- 田耘, 胡彬, 徐文波. Xilinx ISE Design Suite 10.x FPGA 开发指南. DSP、嵌入式与高速传输篇. 北京: 人民邮电出版社, 2008: 382-383.
- 汤琦, 蒋军敏. Xilinx FPGA 高级设计及应用. 北京: 电子工业出版社, 2012: 82.
- 陈雪, 李佩玥, 章明朝, 隋永新, 杨怀江. 基于 PCI Express 总线的高速光纤接口卡的设计与实现. 电子技术应用, 2011, 37(5): 65.
- Xilinx Inc. Endpoint Block Plus V1.14 for PCI Express DS551. USA, Xilinx Inc. 2010.
- Xilinx Inc. Bus Master Performance Demonstration Reference Design for the Xilinx Endpoint PCI Express Solutions XAPP1052. USA, Xilinx Inc. 2011.
- 任连芳. 基于 PCI Express 总线的数据传输与存储[学位论文]. 南京: 南京理工大学, 2010.