

文章编号: 2095-4980(2022)04-0385-08

基于 FPGA 的 PCIe 接口逻辑设计与实现

李龙乾, 方 华, 冯 姣*, 李 鹏

(南京信息工程大学 电子与信息工程学院, 江苏 南京 210044)

摘 要: 为了提升高速串行计算机扩展总线标准(PCIe)总线互联设备在高速通信过程中的系统性能, 减少对中央处理器(CPU)资源的占用, 基于 Kintex-7 系列现场可编程逻辑门阵列(FPGA)平台进行总线主控式直接存储访问(DMA)设计, 通过 PCIe 接口实现了主机设备(PC)与 FPGA 设备之间的高性能数据传输。同时, 基于 Root Port 仿真平台设计 DMA 读写测试用例, 仿真结果验证 PCIe 接口逻辑的正确性。通过连接上位机和配置驱动进行实际传输速率测试, 结果表明, DMA 写速率最高可达 1 620 MB/s, DMA 读速率最高可达 1 427 MB/s, 带宽最大值能够达到 PCIe 接口理论带宽值的 84%。设计方案成本低, 可靠性高, 能够满足高性能、低延时的数据采集要求。

关键词: 高速串行计算机扩展总线标准; 现场可编程逻辑门阵列; 直接存储访问; 数据传输

中图分类号: TP397.1

文献标志码: A

doi: 10.11805/TKYDA2020281

Design and implementation of PCIe interface logic based on FPGA

LI Longqian, FANG Hua, FENG Jiao*, LI Peng

(School of Electronic & Information Engineering, Nanjing University of Information Science & Technology, Nanjing Jiangsu 210044, China)

Abstract: The bus master Direct Memory Access(DMA) controller is implemented based on the Kintex-7 series Field Programmable Gate Array(FPGA) platform in order to improve the system performance of Peripheral Component Interconnect express(PCIe) bus interconnection devices in the process of high-speed communication and reduce the consumption of Central Processing Unit(CPU) resources. The high performance data transmission between Personal Computer(PC) and FPGA via PCIe is realized. DMA test cases are designed through the Root Port simulation platform, and the simulation results verify the correctness of the PCIe interface logic. The actual transmission rate is tested through the master computer and driver, and the experimental results show that the measured highest transfer rates are 1 620 MB/s on write and 1 427 MB/s on read, which reaches 84% of the theoretical maximum. The design scheme bears the advantages of low cost and high reliability, and can meet the requirements of data acquisition with high performance and low delay.

Keywords: Peripheral Component Interconnect express; Field Programmable Gate Array; Direct Memory Access; data transmission

信息采集与数据传输技术具有高带宽、低延时的特点, 对数据处理设备的性能提出更高要求^[1]。由于 FPGA 具有可编程、高集成度、高保密性、速度快等优点, 因此多数信息采集系统的数据处理平台使用 FPGA 板卡^[2]。但板卡与主机设备之间的数据传输链路是影响系统性能的主要因素^[3-5], 所以选择适当的总线技术至关重要。PCIe 是一种高速串行计算机扩展总线标准^[6], 相比 PCI 总线在拓扑结构方面取得重大改进, 由并行总线变为串行总线, PCIe 采用点对点的差分信号传输方式, 以减少有害串扰和电磁干扰的影响, 保障端对端传输的可靠性。此外 PCIe 具备更快的速度、灵活的扩展性, 用户可针对不同的应用选择 x1 到 x32 的接口模式, 如何利用 PCIe 总线最大化传输效率成为当前的研究热点^[7]。

在数据传输过程中, DMA 方式不需要依赖 CPU 的大量中断负载, 能有效提高系统性能。Kavianipour 等人在

收稿日期: 2020-06-16; 修回日期: 2020-07-16

基金项目: 国家自然科学基金资助项目(61501244; 61501245); 江苏省自然科学基金资助项目(BK20150932)

*通信作者: 冯 姣 email:jiao.feng@nuist.edu.cn

2014年设计了基于Xilinx IP核的DMA数据传输平台，包括DMA控制器、Linux操作系统下的驱动及上位机，并在不同的FPGA开发板测试，其最高的读/写速度分别为748 MB/s、784 MB/s^[8]。这虽然解决了PIO(Programmed Input/Output)读/写带宽低的问题，但由于配置的PCIe版本以及数据负载的大小限制，该方案没有充分利用PCIe总线带宽。李晓宁等基于Virtex-6系列FPGA设备开发了DMA数据传输平台，其PCIe总线配置为Gen1 x8模式，DMA写速率最高为1.5 GB/s，达到PCIe总线理论带宽的80%^[9]。2016年，蒲恺等人设计了PCIe总线接口逻辑，DMA读/写带宽分别达到187 MB/s和198 MB/s^[10]。何广亮在2018年设计的PCIe总线DMA控制器，与PC通信时收发速率均可达到850 MB/s以上^[11]。2019年，秦固平等设计了一种PCIe接口逻辑，但该方案仅实现了DMA写传输方式^[12]，未充分利用PCIe总线全双工的特点。还有其他学者基于不同的FPGA开发板设计了DMA控制器^[13-15]，但其DMA控制器中状态机转移过程较复杂，且主要都是在比PCIe Gen2版本传输速率低一倍的PCIe Gen1模式下的设计方案。目前，还没有学者基于Root Port模型搭建的DMA读写测试平台验证接口逻辑的正确性。

针对以上不足，本文提出一种基于PCIe Gen2版本下的DMA控制器设计方案，简化状态机转移过程，并基于Root Port仿真平台设计DMA传输测试用例，快速验证数据传输的正确性。

1 PCIe 协议概述

1.1 PCIe 设备的层次

PCIe协议规定PCIe总线为3层^[16]，如图1，分别为事务层(Transaction Layer, TL)、数据链路层(Data Link Layer, DL)和物理层(PHysical Layer, PHY)。事务层的主要任务是管理数据包和包流量控制。数据链路层负责其他两层之间的链路和错误管理。物理层主要完成链路训练、接口初始化以及8b/10b编码。

1.2 PCIe 的数据包结构

不同的协议层对应不同的数据包格式，如图2所示。根据事务层包(Transaction Layer Packet, TLP)头标可以将PCIe事务分为：存储器读/写事务、IO读/写事务、配置事务以及消息事务^[17]。用户在设计接口逻辑时，只需根据事务类型添加对应头标和数据，其余结构由IP核添加。

1.3 PCIe 理论传输带宽

PCIe Gen1 每条链路的理论带宽为 2.5 GT/s^[18]，同样是基于 8b/10b 编码，PCIe Gen2 将速率翻倍，达到了 5.0 GT/s。根据图 2 可知，各层数据包有额外开销，所以 PCIe 理论上的最大带宽可以根据公式(1)计算出，其中 P_L 为最大数据负载， O_V 为协议开销(以 32 bit 存储器寻址为例，该值为 20 Byte)， N 为通道数， Gen 为 PCIe 版本(Gen1 或者 Gen2)。

$$T = \frac{P_L}{P_L + O_V} \cdot N \cdot Gen \cdot (250 \text{ MByte/s}) \tag{1}$$

例如，对于 PCIe Gen2 模式，通道数为 4，且最大数据负载为 512 Byte 的理论最大带宽为 1 925 MB/s。

2 PCIe 传输系统设计

2.1 系统结构及功能

本文基于 Xilinx PCIe IP 核实现总线主控式 DMA 控制器设计，简化了接收引擎和发送引擎状态机转移过程。IP 核实现了 PCIe 物理层和数据链路层的功能，并提供了事务层接口，DMA 控制器只需根据事务层包格式完成数据包发送、接收和解析等操作，缩短了开发周期。系统总体设计如图 3 所示，主要包括接收引擎、发送引擎、AXI-TRN 转换模块、中断控制器、控制状态寄存器、接收 FIFO 以及发送 FIFO。用户逻辑与两个异步 FIFO 模块相连，解决了 DMA 控制器与其他模块之间数据传输的跨时钟域处理问题。

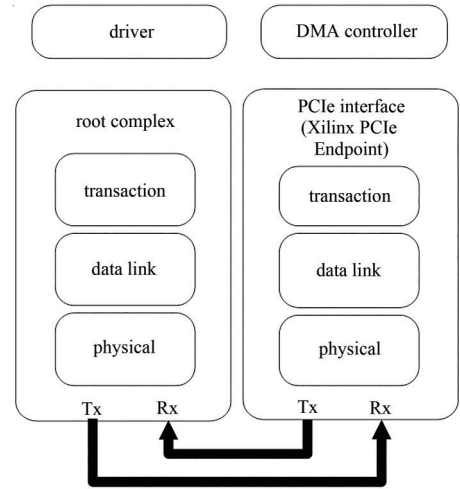


Fig.1 Hierarchy of PCIe
图 1 PCIe 的分层结构

PHY	DL	TL	DL	PHY	
start	ID	header	payload	LCRC	end
1 Byte	2 Byte	12/16 Byte	128-4 096 Byte	4 Byte	1 Byte

Fig.2 Each layer of PCIe packet structure
图 2 PCIe 各层数据包结构

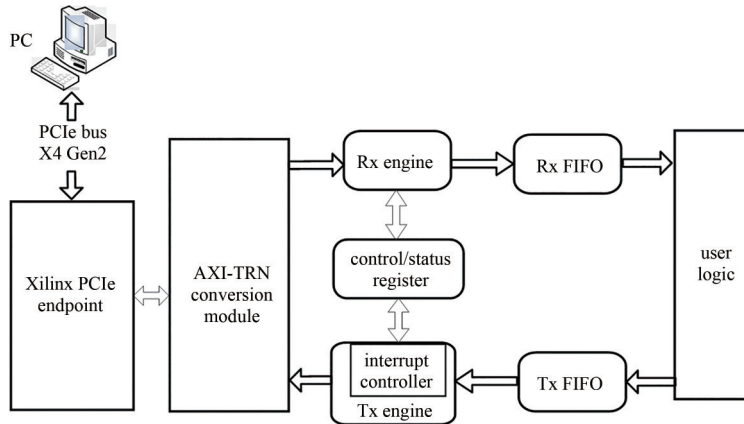


Fig.3 Overall design of the system
图3 系统总体设计图

本系统主要实现功能是主机端以PIO方式将DMA传输的地址、数据大小和数量等参数写入控制状态寄存器中，再由端点设备发起DMA读/写请求。在DMA读请求过程中，DMA控制器中的发送引擎向主机端发送存储器读请求(Memory Read Request, MRd)TLP，然后等待接收主机发送带数据的完成包(Completion with Data, CplID)，并将CplID中的有效数据写入接收FIFO中，最后由中断控制器提交中断告知主机设备；当端点设备发起DMA写请求时，发送引擎从发送FIFO中提取数据并按照存储器写请求(Memory Write Request, MWr)TLP格式组包，通过PCIe链路发送至主机设备，当传输完成时，同样会以中断方式告知主机端。

2.2 接收引擎设计

接收引擎作用为接收来自主机设备的PIO读/写请求TLP和CplID。当端点设备收到PIO读请求TLP时，接收引擎将解析其中的地址等信息，并通知发送引擎根据地址从控制状态寄存器中组织有效数据发送CplID至主机设备。若接收的是PIO写请求TLP，接收引擎将解析出的配置信息写入控制状态寄存器。若接收到来自主机端的CplID，接收引擎将CplID的有效数据写入接收FIFO中。本文通过状态机实现接收引擎设计，共包含4个状态，图4为接收引擎状态转移图。

各状态主要操作为：

- 1) DMA_RX_RST：处于该状态时，接收引擎解析收到的TLP，根据其头标判断是PIO读/写请求TLP或者CplID，并分别跳转到DMA_RX_MRD_WT、DMA_RX_MWR_WT和DMA_RX_CPLD_QWN状态做下一步处理。由于数据总线位宽为128位，接收引擎在此状态可以直接解析数据负载为1个双字(Double Word, DW)的CplID，并将其有效数据写入接收FIFO，完成后中断控制器提交中断请求。
- 2) DMA_RX_MRD_WT：在此状态中，表明接收引擎已经收到来自主机设备的PIO读请求TLP，并等待发送引擎将CplID发送到主机端，之后回到DMA_RX_RST状态。
- 3) DMA_RX_MWR_WT：接收引擎根据PIO写请求TLP中的数据对控制状态寄存器进行配置，并等待写入完成后返回DMA_RX_RST状态。
- 4) DMA_RX_CPLD_QWN：接收引擎在当前状态继续接收数据负载大于1DW的CplID，并将有效数据写入接收FIFO，完成后中断控制器提交中断并跳转至DMA_RX_RST状态。

2.3 发送引擎设计

发送引擎功能是向主机设备发送MRd TLP、MWr TLP以及PIO读完成包。发送引擎状态转移过程如图5所示，总共包括3个状态。

各状态含义分别为：

- 1) DMA_TX_RST：在该状态下，根据控制信号判断当前执行发送MRd TLP、MWr TLP或者PIO读完成包操

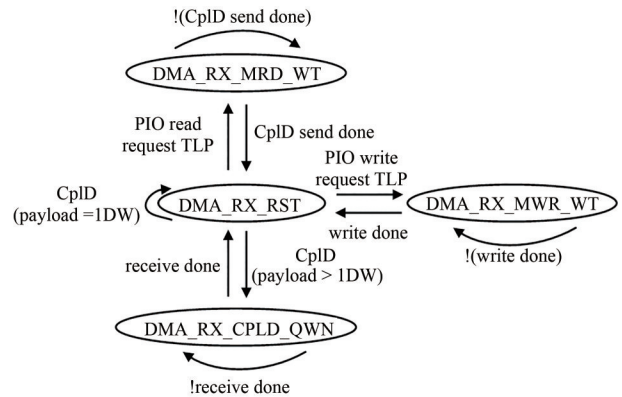


Fig.4 Finite state machine diagram of the RX engine
图4 接收引擎状态转移图

作。当确定操作方式时，发送引擎会将 TLP 前 4 个 DW 的数据传输到总线上，并根据状态机的转移做进一步处理。发送引擎可以直接在当前状态完成 MRd TLP 和数据负载为 1DW 的 MW_r TLP 的发送。若当前发送的 MW_r TLP 的数据负载大于 1DW，则进入 DMA_TX_MWR_QWN 状态。若为 PIO 读完成包，则跳转到 DMA_TX_CPLD_WT 状态。

2) DMA_TX_CPLD_WT: 此状态表明端点设备接收到主机设备的 PIO 读请求，发送引擎正在等待 IP 核接收 CplD，接收完毕将回到 DMA_TX_RST 状态。

3) DMA_TX_MWR_QWN: 该状态持续发送 MW_r TLP 的数据负载，直到数据负载的长度与控制状态寄存器中的参数一致，由中断控制器提交中断请求后返回 DMA_TX_RST 状态。

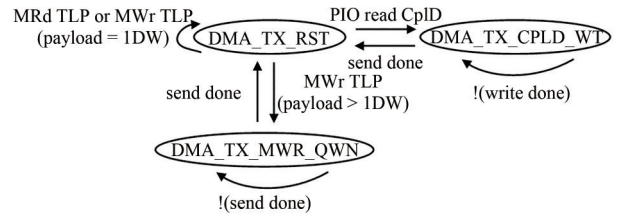


Fig.5 Finite state machine diagram of the Tx engine
图 5 发送引擎状态转移图

3 基于 Root Port 模型的 DMA 传输仿真测试

3.1 Root Port 模型简介

本文使用 Xilinx 提供的 Root Port(RP)模型搭建 DMA 传输测试平台，该模型提供了一个功能强大的测试环境，用户可以通过其测试程序接口(Test Program Interface, TPI)快速验证逻辑设计的正确性。RP 仿真模型主要功能为生成下行 TLP 流量提供源机制，从而驱动用户逻辑，并作为目的机制接收来自用户设计产生的上行 TLP 流量。RP 平台还提供了配置空间初始化、生成 TLP 事务、输出日志文件以及用于创建和验证设计的接口等功能。用户可以集中精力编写相关任务及函数，不需要耗费时间开发基于端点设备的仿真平台，缩短开发周期。

RP 模型包括 dsport、usrapp_tx、usrapp_rx 和 usrapp_com 个模块，图 6 是 Root Port 模型基本架构。dsport 模块与 usrapp_tx 和 usrapp_rx 模块相连并向端点设备传输 TLP 或者从 PCIe 链路接收 TLP，该模块与 IP 核实现 PCIe 数据链路层和物理层的功能。usrapp_tx 模块通过 TPI 调用测试程序向 dsport 模块发送 TLP。usrapp_rx 模块接收并解析来自端点设备的 TLP。usrapp_tx 和 usrapp_rx 模块共享 usrapp_com 中的函数，例如 TLP 的处理和日志文件的输出。但 RP 模型不能直接用于 DMA 读/写测试，本文通过编写功能完善的仿真测试用例，验证 DMA 传输过程的正确性。

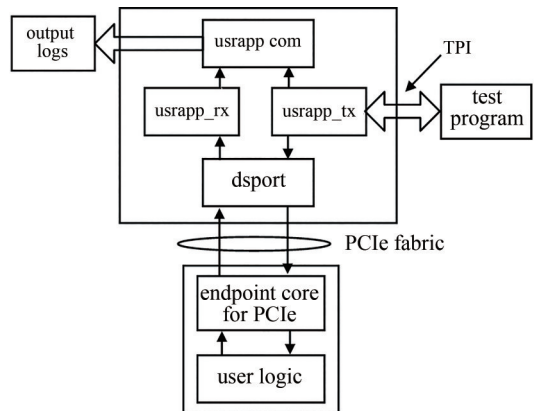


Fig.6 Root Port model structure
图 6 Root Port 模型结构

3.2 DMA 读传输仿真

本文通过 DMA 读测试用例模拟主机端的应用程序对 PCIe 设备进行操作。图 7 为 DMA 读/写测试程序流程图。

首先，依次调用 usrapp_tx 模块中的任务 TSK_SIMULATION_TIMEOUT, TSK_SYSTEM_INITIALIZATION, TSK_BAR_INIT，分别为设置仿真超时时间、系统初始化和基地址寄存器(Base Address Register, BAR)空间初始化。在仿真初始阶段模拟 BIOS 枚举 PCIe 设备，对 FPGA 设备的配置寄存器进行赋值。之后，通过发送 PIO 写请求 TLP 对 DMA 读传输相关的寄存器进行设置，执行 DMA 控制器复位、取消复位、配置 TLP 地址、大小、数量和开始 DMA 传输操作，具体配置流程及对应值如表 1 所示。最后，当 usrapp_rx 模块收到来自端点设备的 MRd TLP 时，通知 usrapp_tx 模块发送相应的 CplD。端点设备接收到所有 CplD 时，中断控制器发送中断请求，结束 DMA 读传输过程。

如图 8 所示为配置控制状态寄存器过程。信号线 a_i[6:0]、wr_d[31:0] 的数值代表相应控制状态寄存器的地址与设置值，写使能信号 wr_{en}_i 连续出现了 6 个高电平，分别对应表 1 中的 6 个步骤。初始化复位信号 init_{rst}_o 由低电平变为高电平的过程说明 DMA 控制器完成复位。随着 DMA 读开始信号 dma_{mrd}_{start} 的拉高，发送引擎将向主机设备发送 MRd TLP。

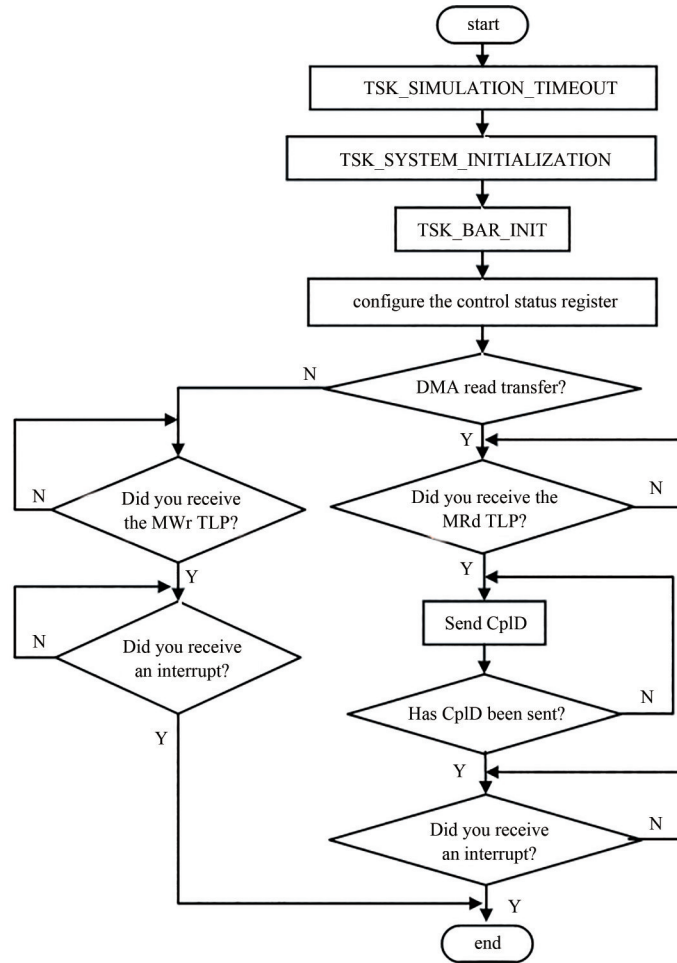


Fig.7 Flow of DMA read/write test program

图 7 DMA 读/写测试程序流程图

表 1 配置控制状态寄存器步骤

Table1 Steps to configure control status register

step	operation	value
1	reset DMA controller	0x00000001
2	cancel reset	0x00000000
3	configure DMA R/W address	0xeeee0000/0xffff0000
4	configure DMA R/W size	0x00000020
5	configure DMA R/W count	0x00000010
6	start DMA R/W transfer	0x00010000/0x00000001

图 9 是 DMA 读传输开始后，DMA 控制器工作的仿真过程。图中包括接收引擎接口(Rx_Engine_Ports)、发送引擎接口(Tx_Engine_Ports)、接收 FIFO 接口(Rx_Fifo_Ports)以及中断控制器接口(Interrupt_Control_Ports)的仿真时序。随着 dma_mrd_start 信号拉高，表明 DMA 读传输开始，发送引擎开始发送 MRd TLP。当 mrd_count 信号值为 16'h10 时，在发送接口 trn_td[127:0] 总共传输了 16 个 MRd TLP。随后，接收引擎在 trn_rd [127:0] 总线上收到了从 usrapp_tx 模块发送的 CplD，每个 TLP 的有效数据负载为 128B，接收引擎解析收到的 CplD，使能 wr_en 信号，并将数据成功写入接收 FIFO 中。在接收引擎收到 16 个 CplD 后，将 dma_mrd_done 信号置为高电平，此时中断控制器将信号 cfg_interrupt_n_o 拉低，结束 DMA 读传输过程。从整个仿真过程来看，接收引擎可以正确接收并解析来自主机端的 PIO 读/写 TLP 和 CplD。



Fig.8 Timing diagram of configuring control status register

图 8 配置控制状态寄存器时序图

3.3 DMA 写传输仿真

DMA 写传输的控制状态寄存器配置流程与 DMA 读传输基本相同。同样需要利用仿真平台通过 PIO 写请求 TLP 对控制状态寄存器进行配置，见表 1，这里不再赘述。DMA 写测试程序流程如图 7 所示，与 DMA 读传输测试不同的是，发送引擎向主机端发送 MWr TLP 之后，中断控制器直接提交中断请求，不需要主机设备返回 CpID。

图 10 为 DMA 写传输仿真时序图。DMA 写传输开始的标志是 dma_mwr_start 信号被拉高，rd_en 信号变为高电平，发送引擎从发送 FIFO 中读出数据并按照 MWr TLP 格式进行组包。从图中能够看到，mwr_count[15:0]信号值为 16'h10 时，在发送接口 trn_td[127:0]一共传输了 16 个数据负载为 32DW 的 TLP，与控制状态寄存器中配置的 TLP 大小、数目一致。当 dma_mwr_done 信号变为高电平时，表示发送引擎完成所有 MWr TLP 的传输，并提交中断信号结束 DMA 写传输过程。综合 DMA 读/写仿真过程来看，发送引擎能够根据控制状态寄存器中的配置信息正确发送对应的 MRd/MWr TLP。本文基于 RP 平台提出一种 DMA 读/写测试用例，仿真结果验证了 DMA 控制器的逻辑功能完整并符合预期。

4 实验结果

4.1 实验环境

本文中使用的 FPGA 开发板采用 Xilinx 公司的 Kintex-7 xc7k325tffg900-2 作为主控芯片，PCIe 链路配置成 Gen2 x4 模式。主机设备配置为 Windows 10 64 位操作系统，CPU 为 i5 处理器，主频 3.4 GHz。本文的整体系统设计基于 Verilog 语言，采用自顶向下的模块化设计思路，使用 Vivado 2017.4 开发软件，经过综合实现步骤生成比特流文件下载到开发板。结合对应的上位机与驱动进行 DMA 读写传输速率测试。

4.2 DMA 传输速率测试

本文分别进行了 DMA 读传输和 DMA 写传输的速率测试。在上位机程序中，调用函数 Query Performance Counter()得到一次 DMA 传输所用的计算机周期数；再通过函数 Query Performance Frequency()得到计算机时钟频率。根据 DMA 带宽计算公式(2)，得到实际的传输带宽。

$$T = \frac{N \times P}{n \times t_c} \tag{2}$$

式中：N 为传输的 TLP 数目；P 是 TLP 的最大有效负载；n 为计算机周期数；t_c 为计算机时钟周期。

图 11 显示了 DMA 读写传输带宽与传输的数据量关系。图中虚线为 PCIe Gen2 x4 配置下的理论最大带宽，由于读写过程中 TLP 的数据负载不同，根据式(1)得出的理论最大带宽也不相同。随着 DMA 传输数据量的增大，DMA 读写传输带宽都持续增加，而且当数据量从 10 KB 增大到 1 MB 时，DMA 传输带宽增长幅度最明显，超过 1 MB 时，增长速度趋于平缓，并接近传输带宽的最大值。DMA 读带宽为 1 427 MB/s，DMA 写传输带宽可达 1 620 MB/s，峰值带宽达到 PCIe 理论带宽值的 84%。

根据分析，造成 DMA 实际带宽低于理论带宽值的原因有：PCIe 各层数据包的格式会消耗部分传输带宽；并且主机设备对中断请求的处理时间同样会影响带宽值。表 2 展示了本文方法与其他文献方法在 DMA 传输带宽及带

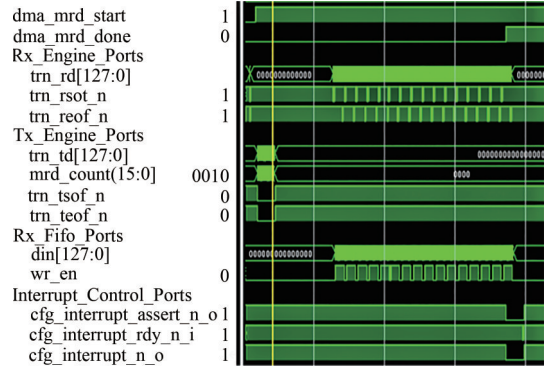


Fig.9 Simulation timing diagram of DMA read transmission

图 9 DMA 读传输仿真时序图

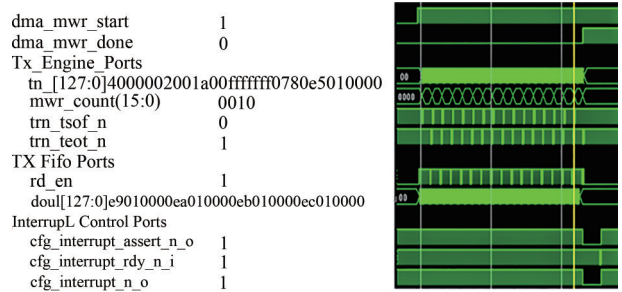


Fig.10 Simulation timing diagram of DMA write transmission

图 10 DMA 写传输仿真时序图

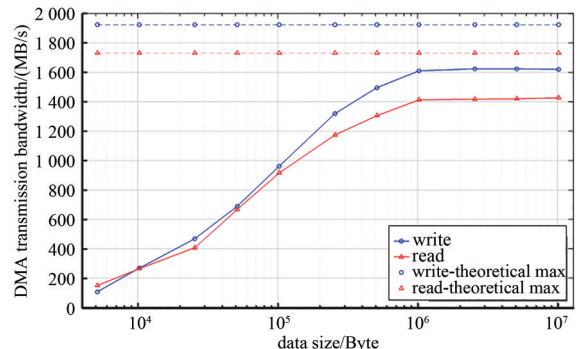


Fig.11 Curves of DMA transmission bandwidth variation

图 11 DMA 传输带宽变化曲线

宽利用率方面的对比。文献[8]中 PCIe 采用 Gen2 x4 模式下得到的 DMA 读带宽为 784 MB/s，DMA 写带宽为 748 MB/s。文献[9]的 PCIe 链路配置模式为 Gen1 x8，DMA 传输速率为 1.5 GB/s。文献[10]在 Gen1 x1 模式下测得的 DMA 读/写带宽分别达到 187 MB/s、198 MB/s。文献[12]在 PCIe Gen1 版本链路数为 x8 配置下，测得 DMA 写传输速率为 1.06 GB/s。将其全部换算为 PCIe Gen2 x4 模式下进行比较，本文在传输速率及带宽利用率方面均有明显提高。

表 2 DMA 传输带宽与峰值带宽利用率对比

Table 2 Comparison of DMA transmission bandwidth and peak bandwidth utilization

	DMA write speed/(MB/s)	DMA read speed/(MB/s)	peak bandwidth utilization/(%)
KAVIANIPOUR (ref [8])	784	748	52
LI Xiaoning (ref [9])	1 500	1 470	80
PU Kai (ref [10])	1 581	1 500	82
QIN Guping (ref [12])	1 060	—	55
this paper	1 620	1 427	84

5 结论

为了满足高速数据采集系统的要求，设计了一种 PCIe 接口逻辑，并简化 DMA 控制器中发送引擎和接收引擎的状态机转移流程。同时基于 RP 模型提出一种 DMA 传输测试用例，快速验证 DMA 读写传输的正确性。结合上位机与驱动程序的硬件测试平台，完成实际速率测试，结果表明，DMA 写传输最大带宽为 1 620 MB/s，DMA 读传输最大带宽可达 1 427 MB/s，与其他文献方法相比，本文的设计方案有效提升了带宽利用率，带宽最大值可达理论带宽的 84%。不足之处在于，目前 DMA 控制器是总线主控式，以后考虑使用分散聚合式 DMA(Scatter Gather DMA，SGDMA)进一步提升系统性能。

参考文献：

- [1] BITTNER R,RUF E,FORIN A. Direct GPU/FPGA communication via PCI express[J]. Cluster Comput, 2014,17:339-348.
- [2] 赵赫,黄志洪,余乐,等. 适用于FPGA的浮点型DSP硬核结构设计[J]. 太赫兹科学与电子信息学报, 2019, 17(3):524-530. (ZHAO He,HUANG Zhihong,YU Le,et al. A suitable design of floating-point DSP hard core structure in FPGA[J]. Journal of Terahertz Science and Electronic Information Technology, 2019,17(3):524-530.)
- [3] ROTA L,CASELLE M,CHILINGARYAN S,et al. A PCIe DMA architecture for multi-gigabyte per second data transmission[J]. IEEE Transactions on Nuclear Science, 2015,62(3):972-976.
- [4] 李胜蓝,姜宏旭,符炜剑,等. 基于PCIe的多路传输系统的DMA控制器设计[J]. 计算机应用, 2017,37(3):691-694,716. (LI Shenglan,JIANG Hongxu,FU Weijian,et al. Design of DMA controller for multi-channel transmission system based on PCIe[J]. Journal of Computer Applications, 2017,37(3):691-694,716.)
- [5] JIANG H X,LI S L,LI H Y,et al. A high-performance and hardware-efficient PCIe transmission for a multi-channel video using command caching and dynamic splicing on FPGA[J]. Journal of Circuits,Systems and Computers, 2018,27(4):1-19.
- [6] 王齐. PCI express 体系结构导读[M]. 北京:机械工业出版社, 2010. (WANG Qi. Introduction to PCI express architecture[M]. Beijing:China Machine Press, 2010.)
- [7] 马鸣锦,朱剑冰,何红旗. PCI、PCI-X 和 PCI express 的原理及体系结构[M]. 北京:清华大学出版社, 2007. (MA Mingjin,ZHU Jianbing,HE Hongqi. The principle and architecture of PCI,PCI-X and PCI express[M]. Beijing:Tsinghua University Press, 2007.)
- [8] KAVIANIPOUR H,MUSCHTER S,BOHM C,et al. High performance FPGA-based DMA interface for PCIe[J]. IEEE Transactions on Nuclear Science, 2014,61(2):1-3.
- [9] 李晓宁,姚远程,秦明伟. 基于PCIe的高速接口设计[J]. 信息技术与网络安全, 2016,35(1):27-29,32. (LI Xiaoning,YAO Yuancheng,QIN Mingwei. Design of high-speed interface based on PCIe[J]. Information Technology and Network Security, 2016, 35(1):27-29,32.)
- [10] 蒲恺,唐庆,田园. 基于IP核的PCIe总线接口逻辑的设计和实现[J]. 航空计算技术, 2017,47(1):116-120. (PU Kai,TANG Qing,TIAN Yuan. Designing and implementation based on IP core of interface logic for PCI express[J]. Aeronautical Computing Technique, 2017,47(1):116-120.)
- [11] 何广亮. 基于PCIe总线的DMA控制器设计与实现[J]. 信息技术与网络安全, 2018,37(4):131-135. (HE Guangliang. Design and implementation of DMA controller based on PCIe bus[J]. Information Technology and Network Security, 2018, 37(4): 131-135.)
- [12] 秦固平,毛瑞娟,杨小勇,等. 一种PCIe接口逻辑的FPGA实现[J]. 无线电工程, 2019,49(4):351-356. (QIN Guping,MAO

- Ruijuan, YANG Xiaoyong, et al. An FPGA implementation of PCIe interface logic[J]. Radio Engineering, 2019,49(4):351–356.)
- [13] ZHAO Y X, LIU X, YANG J. A resource and timing optimized PCIe DMA architecture using FPGA internal data buffer[J]. IEICE Electronics Express, 2019,16(1):1–12.
- [14] 李木国, 黄影, 刘于之. 基于FPGA的PCIe总线接口的DMA传输设计[J]. 计算机测量与控制, 2013,21(1):233–235,249. (LI Muguo, HUANG Ying, LIU Yuzhi. Design of DMA transmission with PCIe bus interface based on FPGA[J]. Computer Measurement and Control, 2013,21(1):233–235,249.)
- [15] 陈刚, 张京, 唐建. 一种基于FPGA的PCIe总线及其DMA的设计方法[J]. 兵工自动化, 2014,33(5):75–77. (CHEN Gang, ZHANG Jing, TANG Jian. A design of PCIe bus and DMA based on FPGA[J]. Ordnance Industry Automation, 2014,33(5):75–77.)
- [16] MBAKOYIANNIS D, TOMOUTZOGLU O, KORNAROS G. Energy-performance considerations for data offloading to FPGA-based accelerators over PCIe[J]. ACM Transactions on Architecture and Code Optimization, 2018,15(1):1–24.
- [17] SHIM C, SHINDE R, CHOI M. Compatibility enhancement and performance measurement for socket interface with PCIe interconnections[J]. Human-centric Computing and Information Sciences, 2019,9(10):1–18.
- [18] PCI-SIG. PCI express 2.0 base specification revision 0.9[S/OL]. [2020-07-16]. Peripheral Component Interconnect Special Interest Group(PCI-SIG), 2006. <http://www.pcisig.com/specifications/pciexpress/base2/>.

作者简介:

李龙乾(1996–), 男, 河南省洛阳市人, 在读硕士研究生, 主要研究方向为嵌入式系统设计. email:20181219041@nuist.edu.cn.

冯 姣(1984–), 女, 吉林省延吉市人, 博士, 副教授, 硕士生导师, 主要研究方向为嵌入式系统设计. email:jiao.feng@nuist.edu.cn.

方 华(1977–), 男, 安徽省黄山市人, 硕士, 副教授、硕士生导师, 主要研究方向为嵌入式系统设计.

李 鹏(1984–), 男, 山西应县人, 博士, 教授、硕士生导师, 主要研究方向为嵌入式系统设计.