

基于 IP 核的 PCI Express 接口

徐天¹, 何道君², 徐金甫¹

(1. 解放军信息工程大学电子技术学院, 郑州 450004; 2. 北方信息技术研究所, 北京 100072)

摘要: 提出一种基于 IP 核的 PCI Express 接口设计方案, 采用分层的体系结构, 具有高性能、高可靠性、软件兼容等特点。在 Xilinx 公司的 xc5vlx30t FPGA 器件中, 运用 PCI Express 硬核端点模块, 设计并实现了 PCI Express 总线接口, 同时在接口内部设计了 DMA 控制器。仿真实验结果表明, 该接口能够有效提高总线的数据传输速率, 充分发挥总线的性能。

关键词: PCI Express 总线; 现场可编程门阵列; DMA 控制器; IP 核

PCI Express Interface Based on IP-core

XU Tian¹, HE Dao-jun², XU Jin-fu¹

(1. Institute of Electronic Technology, PLA Information Engineering University, Zhengzhou 450004;
2. North Information Technology Institute, Beijing 100072)

【Abstract】 A design scheme for PCI Express interface based on IP-core is proposed, which uses storied structure and has such characteristics as high performance, high reliability and software compatibility. The PCI Express bus interface is designed and implemented by using PCI Express endpoint block hard IP-core in Xilinx xc5vlx30t Field Programmable Gate Array(FPGA). A DMA controller is also designed inside the interface logic circuit. Simulation experimental results show this interface speeds up the frequency of data transfer and the high performance of PCI Express bus is demonstrated adequately.

【Key words】 PCI Express bus; Field Programmable Gate Array(FPGA); DMA controller; IP-core

现代测控系统和通信领域对数据传输速率的要求越来越高。相比 PC 中其他技术的发展, 总线技术的发展显得相对缓慢, 总线性能已经成为制约系统性能发挥的瓶颈。传统的 ISA, EISA 总线等已无法适应高速数据传输的要求, PCI 总线技术虽然经过了不断的修正和发展, 但是由于它固有的缺陷, 使其应用领域受到限制^[1]。PCI Express 总线以其优异的性能和低廉的造价引起了业界的广泛关注, 具有广阔的应用前景。本文介绍 PCI Express 总线接口的设计方法, 并实现一个基于 IP 核的 PCI Express 总线接口。

1 PCI Express 总线简介

PCI Express 总线^[2]是 Intel 公司于 1997 年提出的第 3 代 I/O 技术, 是种全新的串行总线技术。与 PCI 总线相比, PCI Express 总线具有以下特点:

(1)在数据传输模式上, PCI Express 总线采用双工串行传输模式, 一条 PCI Express 通道由 2 对 LVDS 差分线对组成: 一对负责发送, 另一对负责接收, 单向数据传输速率为 2.5 Gb/s。

(2)具有很好的灵活性, 一个 PCI Express 物理连接可以根据实际需要配置成 $\times 1, \times 2, \times 4, \times 8, \times 16, \times 32$ 个并行的数据通道, 满足不同设备之间通信带宽的要求。

(3)在软件层与 PCI 总线完全兼容, 原有 PCI 总线的驱动程序可以完全移植到 PCI Express 总线架构的系统中。

(4)串行连接采用自同步时钟技术, 时钟内嵌于串行数据的 8 bit/10 bit 编码中, 可实现数据传输率的自适应调整。

PCI Express 协议定义了 3 层结构: 物理层, 数据链路层和事务层。每个层次按照协议中规定的内容, 完成相应的数据处理功能。

2 PCI Express 接口设计方法

PCI Express 总线的接口的设计方法大体有 2 种: 使用专用接口芯片或者使用可以实现 PCI Express 物理接口的可编程器件。

专用接口芯片实现了 PCI Express 总线的物理层、数据链路层和事务层的控制逻辑。例如, PEX 8311 是 PLX 公司推出的第 1 款本地总线到 PCI Express $\times 1$ 的桥接芯片^[3-4], 符合 PCI Express 1.0 规范, 支持自动极性反转、CRC 校验、链路设备电源管理, 具有直接主模式、直接从模式、DMA 和数据预取等功能。

使用 PEX 8311 实现 PCI Express 接口, 用户仅需实现相对简单的本地总线接口即可, 降低了开发难度。但是这种方法受限于专用接口芯片的性能, 用户无法根据自身的需求提升接口的性能。

可编程器件设计方案的基本思想是由 FPGA 实现 PCI Express 的上层电路, 利用 Philip 公司或 TI 公司提供的 PHY(物理层)器件实现物理层接口, 通常需要购买 Xilinx 公司或 Altera 公司提供的 IP 核与 PHY 器件配合使用。在有些 FPGA 产品中, 集成有 PCI Express 接口的硬核模块, 可以采用相应的产品实现 PCI Express 接口的功能。例如, Xilinx 公司 Virtex-5 LXT/SXT/FXT/TXT 系列 FPGA 器件中, 集成有

作者简介: 徐天(1982-), 男, 助理工程师、硕士研究生, 主研方向: 专用集成电路设计应用, SOC 系统设计; 何道君, 高级工程师; 徐金甫, 副教授

收稿日期: 2009-08-20 **E-mail:** xutian_1982@163.com

PCI Express 硬核端点模块,能够自动完成数据链路层和物理层的数据处理功能。采用这种方法,用户只需要设计处理事务层数据包的逻辑电路即可。还可以根据自己的需求,设计具有特殊功能的电路结构,发挥接口的性能,但是这种方法的开发难度较大,开发周期相对较长。

3 PCI Express 接口实现

本设计采用 Xilinx 公司 Virtex-5 系列 xc5vlx30t 芯片,使用该芯片内部的 PCI Express Endpoint 硬核端点模块,设计了处理事务层数据的逻辑电路,实现了计算机内存与用户逻辑之间数据的正确传输。其中,控制状态机的设计和 DMA 控制器的设计是重点、难点,下面详细介绍具体的实现过程。

3.1 接口电路的设计

在接口电路中,PCI Express 物理层和数据链路层的电路采用 Xilinx 公司的 PCI Express Endpoint Block plus v1.5 硬核端点模块实现,能够有效完成接口物理层和数据链路层的数据处理功能,提供给上层一个事务层数据接口。

上层电路主要由 5 部分电路构成,分别是核配置与辅助控制模块、数据发送控制器、数据接收控制器、数据输入 FIFO 和数据输出 FIFO,如图 1 所示,最终提供给用户 FIFO 类型的用户逻辑接口。

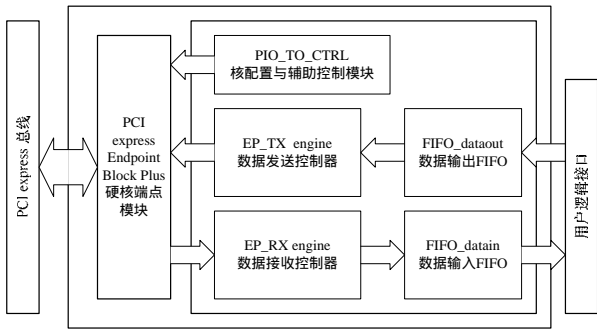


图 1 PCI Express 接口电路结构

核配置与辅助控制模块与硬核端点模块紧密联系,完成硬核端点模块中通道宽度、时钟频率、设备号、版本号、存储空间类型与大小等一系列可控制参数的配置与一些基本功能的控制。

硬核端点模块提供的事务层数据接口为 64 bit 的数据发送接口和 64 bit 的数据接收接口以及控制数据发送和接收的一系列控制信号接口。数据接收控制器将收到的事务层包进行解析,根据数据包的类型进行相应的处理。数据发送控制器用于将所要发送的数据按照事务层包的格式进行封装后,发送到硬核端点模块。

数据输入 FIFO 和数据输出 FIFO 共同构成了数据通道,均为双端口 FIFO,用于连接 PCI Express 接口和用户逻辑。数据输入 FIFO 用于保存从接口收到的数据;用户逻辑电路输出的数据保存到数据输出 FIFO 中后发送到接口电路。

3.2 控制状态机的设计

数据发送控制器和数据接收控制器负责事务层数据的封装和解析工作,是本设计中的关键电路模块。这两个模块均采用有限状态机方式进行设计,下面详细介绍这 2 个模块状态机的设计。

接收控制器从硬核端点模块中收到事务层包(TLP),根据 TLP 的格式,第 56 bit~第 62 bit 为表示数据包类型的标志位。按照标志位中的信息,对 TLP 的内容进行解析,进行相应的数据操作。所设计的状态机的状态转换图如图 2 所示,各个

状态的说明如表 1 所示。

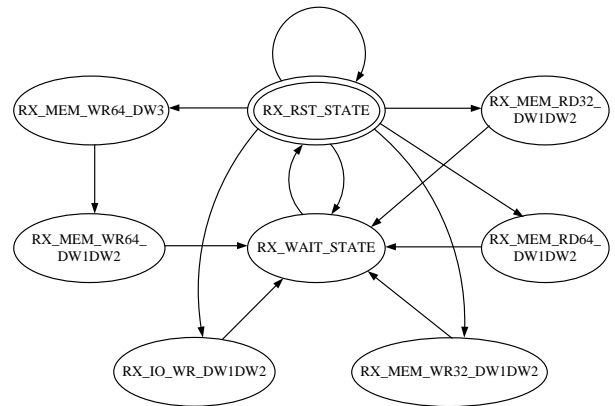


图 2 接收控制器状态机

表 1 接收控制器状态说明

状态	说明
RX_RST_STATE	状态机空闲状态
RX_MEM_RD32_DW1DW2	存储器读操作, 32 bit 模式, 收到第 1 和第 2 个双字
RX_MEM_WR32_DW1DW2	存储器写操作, 32 bit 模式, 收到第 1 和第 2 个双字
RX_IO_WR_DW1DW2	IO 写操作, 收到第 1 和第 2 个双字
RX_MEM_RD64_DW1DW2	存储器读操作, 64 bit 模式, 收到第 1 和第 2 个双字
RX_MEM_WR64_DW1DW2	存储器读操作, 64 bit 模式, 收到第 1 和第 2 个双字
RX_MEM_WR64_DW3	存储器写操作, 64 bit 模式, 收到第 3 个双字
RX_WAIT_STATE	状态机等待其他操作完成状态

系统复位后,接收控制器状态机处于空闲状态,等待接收数据包,收到 TLP 后跳转到相应的状态进行存储器或者 I/O 的读写操作。在操作过程中,对存储器或者 I/O 的状态进行监控,如果目标设备没有准备好,则进入等待状态直到目标设备就绪。在存储器或者 I/O 的读写操作完成后,状态机又返回空闲状态。在各个状态中接收控制器产生相应的控制信号完成数据的写入以及对发送控制器的操作。

与接收控制器状态机相比,发送控制器状态机相对简单,只有 3 个状态,状态转换图见图 3,各个状态的说明见表 2。

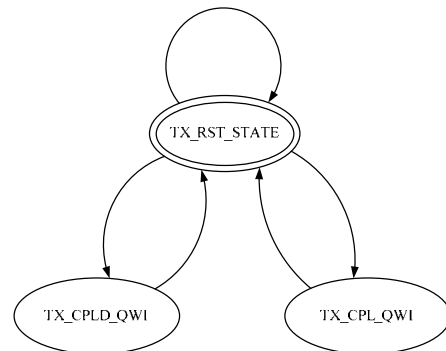


图 3 发送控制器状态机

表 2 发送控制器状态说明

状态	说明
TX_RST_STATE	状态机空闲状态
TX_CPLD_QW1	有载荷数据的发送状态
TX_CPLD_QW1	无载荷数据的发送状态

系统复位后,发送控制器状态机处于复位状态。当从接收控制器收到发送数据的命令后,如果需要发送的是一个载荷数据,则将从存储器或者 I/O 中取出的载荷数据填充到 TLP

中的相应位置后发送到端点模块；如果需要发送的只是一个响应或者标志信息，则直接封装成一个 TLP 发送到端点模块。

3.3 DMA 控制器的设计

由于串行总线结构与传统的并行总线结构不同，因此存储器的读写操作都是以数据包的形式传输。普通的读写操作一次最多只能进行 64 bit 数据的读写，使得总线多数时间处于等待状态，严重影响了 PCI Express 总线的数据传输速率。本文设计出相应的 DMA 控制器，实现了 DMA 方式的数据传输。

DMA 控制器的主要电路结构为 14 个与 DMA 操作相关的寄存器，通过对这些控制寄存器的读写，设定 DMA 方式数据读写操作的地址范围和数据大小，完成 DMA 数据读写操作。这些寄存器的定义和功能描述如表 3 所示。

表 3 DMA 控制寄存器说明

编号	名称	说明
0#	复位寄存器	用于复位 DMA 控制器 0 bit 用于启动 DMA 写操作， 8 bit 表示 DMA 写操作完成，
1#	控制寄存器	16 bit 用于启动 DMA 读操作， 24 bit 表示 DMA 读操作完成
2#	写地址寄存器	设置写数据的起始地址
3#	写包长度寄存器	设置写数据包的大小
4#	写包数量寄存器	设置写数据包的数量
5#	保留	
6#	保留	
7#	读地址寄存器	设置读数据的起始地址
8#	读包长度寄存器	设置读数据包的大小
9#	读包数量寄存器	设置读数据包的数量
10#	写性能寄存器	只读，检测写操作性能
11#	读性能寄存器	只读，检测读操作性能
12#	读完成状态寄存器	只读，可读出 DMA 读操作完成时有关寄存器的状态
13#	读完成数据检测寄存器	只读，可读出 DMA 读操作完成时的最后一个数据
14#	读完成数据大小寄存器	只读，可读出 DMA 读操作完成时传送数据的多少
15#	中断服务及中断控制寄存器	用于检测、设置和清除 DMA 操作的中断

在设计了上述 14 个 DMA 控制寄存器后，接收控制器和发送控制器的状态机也要进行相应的调整，以适应 DMA 操作的需要。在接收控制器状态机中加入了 2 个状态，分别用于接收数据传输过程中的第 1 个 DMA 数据包和剩余的 DMA 数据包。在发送控制器状态机中也加入了 2 个状态，分别用于发送数据传输过程中的第 1 个 DMA 数据包和剩余的 DMA 数据包。为了达到尽可能高的数据传输速率，在接收和发送过程中都使用了 64 bit 数据传输模式。

在设计了如上所述的 DMA 控制器后，PCI Express 总线接口就可以用 DMA 方式进行数据的读写操作。进行 DMA 读/写操作的具体过程如下：(1)写中断控制寄存器，打开中断；(2)写读/写地址寄存器，设置读/写数据的起始地址；(3)写读/写包长度寄存器，设置读/写数据包的大小；(4)写读/写包数量寄存器，设置读/写数据包的数量；(5)写控制寄存器，启动 DMA 读/写操作；(6)等待中断服务寄存器发出中断；(7)收到中断，读中断服务寄存器；(8)写中断控制寄存器，关闭中断。

4 仿真与性能测试

使用 Verilog 语言编写代码实现本设计，使用 Modelsim SE 6.2g 对该设计进行功能仿真。从数据写入 FIFO 的仿真波形图中可以看到，未采用 DMA 方式时，每次只能将一个

数据写入到 FIFO 中，在 2 次写操作之间存在较长的等待时间；采用 DMA 方式时，可以根据用户的设置连续写入多个数据，消除了 2 次操作之间的等待时间，大大提高了数据传输速率。

自行设计了采用 Xilinx 公司 xc5vlx30t 芯片的 PCI Express×4 接口数据传输卡；使用 Xilinx 公司的 ISE 9.2 对设计代码进行综合和布局布线，生成配置文件，下载到 xc5vlx30t 芯片中；使用 DELL Precision 690 工作站，在 Windows 环境下编写测试程序对数据传输卡 PCI Express 接口数据传输的速率进行测试。

采用多种数据量对 PCI Express 接口进行 DMA 方式读数据测试，得到的结果如图 4 所示，硬件层数据传输速率稳定在 840 Mb/s。

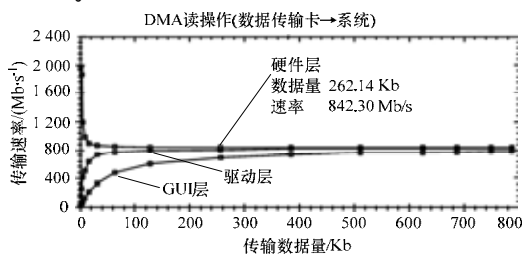


图 4 实际测试读数据速率

采用多种数据量对 PCI Express 接口进行 DMA 方式写数据测试，得到的结果如图 5 所示，硬件层数据传输速率稳定在 720 Mb/s。

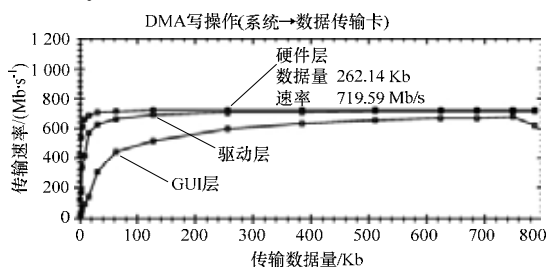


图 5 实际测试写数据速率

从图 4、图 5 可以看出，DMA 控制器有效减少了数据传输过程中的等待时间，大幅度地提高了数据传输速率，发挥了 PCI Express 总线的优越性能。

5 结束语

本文采用基于 Xilinx 公司硬 IP 核的方法设计实现了 PCI Express 总线接口及数据的传输。对于 DMA 控制器的设计，能提高数据传输的速率，使 PCI Express 总线的性能得到充分发挥。

参考文献

- [1] 熊 彬. 走进 PCI Express 的世界[J]. 微型计算机, 2004, 10(9): 112-117.
- [2] PLX Technology Inc.. PEX 8311 Product Brief[Z]. (2005-12-06). http://www.plxtech.com/pdf/technical/expresslane/TechnologyBrief_PCIExpress.pdf.
- [3] PLX Technology Inc.. PEX 8311 Express Lane PCI Express-to-Generic Local Bus Bridge Data Book Version 0.90[Z]. (2006-04-20). <http://www.plxtech.com/products/expresslane>.
- [4] 刘 豫. Xilinx 可编程解决方案应对市场变化和成本挑战[J]. 电子设计应用, 2005, 8(5): 37-39.

编辑 陈 文