

# FPGA 实现 PCIe 总线 DMA 设计

胡传皓 黄虎 颜浩 邢雪霞

成都理工大学信息科学与技术学院 四川成都 610059

## 【文章摘要】

本文对 PCIe 总线提出新的 DMA 设计,并在 Xilinx Virtex6 系列 FPGA 平台上实现。通过实际测试,验证了 DMA 模式数据传输的正确性及传输带宽。

## 【关键词】

FPGA;PCIe 总线;DMA

## 0 引言

现代信息通信技术的发展对总线的带宽以及传输速率有了更高的要求。PCIe 总线是因特尔公司提出的高速差分总线,其 3.0 协议中单通道的峰值带宽接近 8Gb/s。Xilinx Virtex6 系列 FPGA 芯片内置 PCIe IP 核,能够满足 DMA 的设计需求。

## 1 FPGA 设计方案简介

Virtex6 系列 FPGA 提供了成熟、稳定的 PCIe 传输方案。在其 PCIe IP 核中,有物理层,数据链路层和传输层。物理层面向硬件结构;数据链路层完成纠错、重传、拆解、封装包等功能;应用层直接面向用户,负责初步封装包和最后一步拆解包,控制命令,检测状态等。本文提出一种基于该 IP 核传输层的片上系统的 PCIe 设计方案。

## 2 高速数据传输系统设计

### 2.1 PCIe 接口设计

使用 Virtex6 系列的 FPGA 开发板与 PC 间通过 PCIe 总线传输数据, FPGA 通过 DMA 方式对外部存储器进行数据读写操作,实现开发板与 PC 间的高速数据传输。

对 FPGA 进行 PCIe 接口设计,主要包括 PCIe 端点硬核、面向传输层的应用逻辑设计两部分。Xilinx Virtex6 PCIe IP 核提供了从 BAR0 到 BAR5 共 6 个 32 位基地址寄存器 (BAR),我们在设计的时候可以按需使用。本设计将 BAR0 和 BAR1 组成一组,将 BAR2 和 BAR3 组成一组,两组都有 64 位地址空间。前者用来存放 CPU 访问 FPGA 寄存器的地址,后者用来存放 DMA 控制器和缓存器地址。在实际的使用过程中,可以通过 RBAR[6:0] 的值来判别 BAR0 和 BAR2,其程序代码如下:

```
begin
  BAR0<=~RBAR[0];
  BAR2<=~RBAR[2];
end
```

为了更易理解,这里用一个实际的例子来解释其工作方式:当 PC 机的 CPU 读取 FPGA 的内部寄存器时, FPGA 把从 PCIe 总线上收到的数据通过传输层送出, PCIe 总线的应用层再通过接收状态机判断需要访问的是哪个地址,然后再将请求

数据值存放在 FIFO\_1 中,用户逻辑读取该请求值后将其存放在 FIFO\_2 中,最后发送状态机将这个值发送给 IP 核的传输层。

### 2.2 DMA Engine 设计

DMA 即直接内存操作,操作过程中省去了 CPU 取指令、取数、送数等操作。因此具有速度快的优点。采用 DMA 的方式访问通过 PCIe 总线连接的外部设备时, DMA 控制器会向 CPU 提出接管控制 PCIe 总线控制权的请求,当 CPU 完成当前总线周期后,会让出总线控制权,从而使外设和存储器之间直接进行数据交换。

对 PCIe 总线数据帧类型进行修改并且加入 DMA 数据帧,修改后的部分代码如下:

```
parameter DMATYPE=7'b1001010;
parameter
DMADATA1=10'b0100000000;
case (ST)
  RT: begin
    (省略)
  case (dmatype)
    (省略)
  DMATYPE: begin
    (省略)
  ST<=DMADATA1;
  end
  (省略)
  DMADATA1: begin
    (省略)
  end
  (省略)
endcase
参考上述方式,同样可以在传输状态机中添加 DMA 请求,其部分程序代码如下:
```

```
parameter DMAREQ=10'b0100000000;
(省略)
case (ST)
  RT: begin
    if(~dmaready) begin
      ST<=DMAREQ;
      dmaen<=1'b1;
    end
    (省略)
  DMAREQ: begin
    (省略)
  if(~dmadata)
    ST<=TXDONE;
  else
    ST<=DMAREQ;
  end
```

DMA Engine 设计包括两部分,分别为:读操作和写操作。现对这两部分进行详细说明,当 DMA Engine 的状态为 RT 时,需要判断当前是否为空闲状态,此时可以

根据读取到的 DMA 信息标识来判断。然后在从 FPGA 的 RAM 中读取 DMA 控制信息,该信息包括读和写,然后又根据相应的值来进入对应的读状态或者是写状态。

为了更易理解,此处也使用一个实际的例子来解释其工作方式:当 PC 机的 CPU 对 FPGA 外部存储器写入一些数据时, CPU 会通过 FPGA 中的 PCIe IP 核,向基地址寄存器 (BAR0) 写入地址数据,然后再根据该数据值配置、开启 DMA Engine。如果 DMA Engine 成功开启,那么 FPGA 会向 PC 机的 CPU 发出接管控制 PCIe 总线的 DMA 读请求,并且等待 CPU 发送数据。此时 DMA Engine 等待从 PCIe 接口发来的 DMA 数据。同样,当 PC 机的 CPU 对外部存储器进行读操作的时候, FPGA 向 PC 机的 CPU 发出接管控制 PCIe 总线的 DMA 写请求,并且等待 PCIe 接口发送数据。

## 3 设计仿真验证

对 FPGA 外部 64 位 DDR2 DRAM 进行 32 个双字读写操作,仿真结果如下。图 1 为采用 DMA 进行写操作时 PCIe 接口的波形。图 2 为采用 DMA 读操作时 PCIe 接口的波形。经过验证可以得到, PCIe x1 写操作为 1720Mb/s,读为 1684Mb/s。

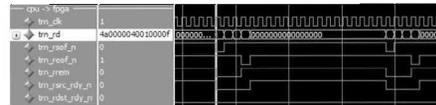


图 1 通过 PCIe 对 DMA 写操作

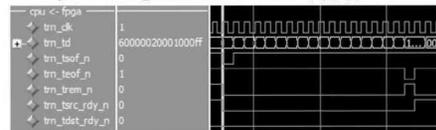


图 2 通过 PCIe 读 FPGA 外部 DRAM

## 4 总结

基于 Xilinx Virtex6 系列芯片的 PCIe IP 核实现 PCIe 总线的 DMA 设计很好的解决了速率与带宽的问题,整个设计占用 FPGA 的 Slice 以及 BRAM 资源较少,具有很好的性能,可满足大多系统数据高速传输的要求。

### 【参考文献】

- [1] 王齐. PCI Express 体系结构导读[M]. 北京:机械工业出版社, 2011:99-103
- [2] PCI Express Base specification, 1.0a ed[S]. PCI SIG, 2003  
<http://www.xilinx.com>, 2010
- [3] 宋宇, 周文远, 陈磊, 叶青, 田建国. 基于 PCI 总线的 DMA 高速数据传输系统[J]. 光电子·激光, 2004, 15(10): 1181-1184
- [4] 徐君明, 裴先登, 王海卫, 等. 高性能计算机 I/O 技术 PCI Express 分析[J]. 计算机工程, 2004, 30(12): 7-8

### 【作者简介】

胡传皓(1989-),男,河南省郑州市人,硕士研究生,研究方向:系统集成。

# 嵌入式资源免费下载

## 总线协议:

1. [基于 PCIe 驱动程序的数据传输卡 DMA 传输](#)
2. [基于 PCIe 总线协议的设备驱动开发](#)
3. [CANopen 协议介绍](#)
4. [基于 PXI 总线 RS422 数据通信卡 WDM 驱动程序设计](#)

## VxWorks:

1. [基于 VxWorks 的多任务程序设计](#)
2. [基于 VxWorks 的数据采集存储装置设计](#)
3. [Flash 文件系统分析及其在 VxWorks 中的实现](#)
4. [VxWorks 多任务编程中的异常研究](#)
5. [VxWorks 应用技巧两例](#)
6. [一种基于 VxWorks 的飞行仿真实时管理系统](#)
7. [在 VxWorks 系统中使用 TrueType 字库](#)
8. [基于 FreeType 的 VxWorks 中文显示方案](#)
9. [基于 Tilcon 的 VxWorks 简单动画开发](#)
10. [基于 Tilcon 的某武器显控系统界面设计](#)
11. [基于 Tilcon 的综合导航信息处理装置界面设计](#)

## Linux:

1. [Linux 程序设计第三版及源代码](#)
2. [NAND FLASH 文件系统的设计与实现](#)
3. [多通道串行通信设备的 Linux 驱动程序实现](#)
4. [Zsh 开发指南-数组](#)
5. [常用 GDB 命令中文速览](#)
6. [嵌入式 C 进阶之道](#)

## Windows CE:

1. [Windows CE.NET 下 YAFFS 文件系统 NAND Flash 驱动程序设计](#)
2. [Windows CE 的 CAN 总线驱动程序设计](#)
3. [基于 Windows CE.NET 的 ADC 驱动程序实现与应用的研究](#)
4. [基于 Windows CE.NET 平台的串行通信实现](#)
5. [基于 Windows CE.NET 下的 GPRS 模块的研究与开发](#)
6. [win2k 下 NTFS 分区用 ntldr 加载进 dos 源代码](#)

## PowerPC:

1. [Freescale MPC8536 开发板原理图](#)

## ARM:

1. [基于 DiskOnChip 2000 的驱动程序设计及应用](#)
2. [基于 ARM 体系的 PC-104 总线设计](#)
3. [基于 ARM 的嵌入式系统中断处理机制研究](#)
4. [设计 ARM 的中断处理](#)
5. [基于 ARM 的数据采集系统并行总线的驱动设计](#)
6. [S3C2410 下的 TFT LCD 驱动源码](#)

## Hardware:

1. [DSP 电源的典型设计](#)
2. [高频脉冲电源设计](#)
3. [电源的综合保护设计](#)
4. [任意波形电源的设计](#)