

PCIE3.0 接口的 SSD 硬盘的 FPGA 实现

宋鹏程, 张 春

(清华大学 微电子所, 北京 100084)

摘 要: 设计并实现了一个高性能固态盘的硬件原型系统. 该固态盘以 NAND 闪存为存储介质, 满足 ONFI4.0 相关标准. 与主机通过第三代 PCIe 总线接口进行通信. 主控部分基于 FPGA 实现. 在实现上, 采用了双总线结构, PCIE3.0 接口并介绍了 NAND 闪存转换层的设计.

关键词: 固态盘; 闪存; 现场可编程逻辑门阵列; 高速互联总线

中图分类号: TN4

文献标识码: A

文章编号: 1000-7180(2017)12-0063-04

Design of High-speed Solid State Drive in FPGA

SONG Peng-cheng, ZHANG Chun

(Institute of Microelectronics, Tsinghua University, Beijing 10084, China)

Abstract: A high-performance solid state drive(SSD) is designed and implemented in the essay. NAND flash chip that obey the standard of ONFI4.0 is used as the storage meida in SSD. SSD communicate with the host through PCIE bus. The control part is implemented based on FPGA. A dual-bus structure and PCIE3.0 IP core are introduced. NAND flash translation layer(FTL) is also described in detail.

Key words: solid state drive; NAND flash; FPGA; PCI express

1 引言

固态硬盘主要由主控芯片, 存储芯片和缓存芯片组成. 除了少数采用 SD 卡为存储介质的主控^[1]外, 主流 SSD 的储存介质均为 NAND 闪存^[2]. NAND 闪存具有两个重要特性: 一是在写入前要对整块进行擦除操作; 二是有限的擦写次数. 所以针对上述的两个特性, 在软件层需要实现地址映射, 垃圾回收, 损耗均衡算法.

随着 NAND 闪存器件的不断发展, 3D 堆叠与新工艺的出现, ONFI 协议的不断更新, NAND 闪存的极限读写速度越来越快. 在 ONFI4.0 的协议中, 通过 NV-DDR3 的模式, 设置 timing mode 为 Mode 10, 可以实现最高 400 MHz 的 800 MT/s 的高速读写. 传统的总线协议 SATA3.0 协议 600 MB/s 已经很难满足需要^[3]. PCIE3.0 的高速度高带宽已经成为固态硬盘接口协议的新选择, 但是以往的设计^[4]

是基于 PCIE2.0 接口实现的, 已经成为限制硬盘速度的瓶颈. 在硬盘传输协议也开始由传统的 AHCI 转向新的 NVME 协议, 以减少延迟和提高硬盘的 IOPS^[5].

2 SSD 设计与实现

2.1 整体结构

目前 SSD 的实现主要有基于专用芯片和 FPGA 两种方案. FPGA 在具有灵活、可重复编程的特性, 可以方便地升级新的 ECC 算法、接口协议及数据安全算法. 在本设计中选用 Xilinx 的 XC7VX690T 芯片, 不仅有丰富的逻辑资源, 还集成了 PCIE3.0 接口, 设计结构如图 1 所示.

NAND 闪存芯片选用镁光公司的 MT29F128G08CFAAB, 容量为 128 GB. 其页大小为 8 kB, 每页有 448 字节的冗余位.

FPGA 是整个设计的核心, 实现 NAND 闪存读

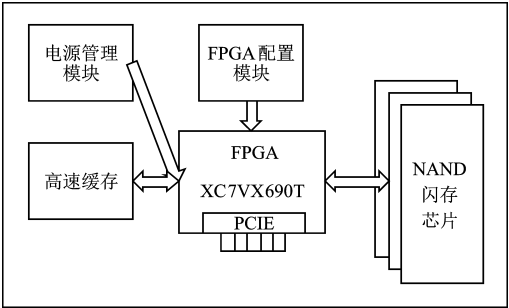


图 1 SSD 结构图

写控制、PCIE 传输、DRAM 内存控制器等. 电路包括以下几个模块: 基于 AXI 接口的 NAND FLASH 读写控制 IP 模块, CPU MICROBLAZE 模块, PCIE3.0 模块, DMA 模块, UART 调试模块. 不同于文献[6]在主机端实现闪存管理算法, MICROBLAZE 可以为固件层的磨损平衡, 垃圾回收等算法提供硬件基础, 减小主机 CPU 负载. UART 模块用于调试.

PCIE 接口采用 PCIE3.0 的 4 通道模式, 单通道速度为 8 Gb/s, 协议开销约为 5%. 采用 4 通道模式理论极限可以达到 30.5 Gb/s.

在数据总线设计上, 在 200 MHz 的时钟频率下, 总线位宽为 256 bit, 其带宽为 47.68 G/s. 在考虑到 PCIE3.0 各层传输开销的情况下, 总线带宽约为 PCIE3.0 带宽的两倍, 可以满足数据传输与 DMA 搬移的带宽要求. DMA 传输采用突发式模式, 在总线位宽为 256 bit 下, 传输长度设置为 256. 这样每次传输数据为 8 kB, 与 NAND 闪存的页大小相同, 实现 NAND 闪存的最高利用率和 DMA 传输的最高效率^[7].

固态硬盘工作时, 首先将数据写入内存缓冲区中, 然后将数据的源地址, 目的地址和数据长度写入寄存器堆中, 通知 MICROBLAZE. MICROBLAZE 在读取相应寄存器内容并存入相应链表中. 最后将通过中断通知主机写入完成. 具体流程如图 2 所示.

在数据烧写进 NAND 闪存时, 首先通过 DMA 进行数据搬移, MICROBLAZE 给出烧写命令和烧写地址. 在烧写完成后, FTL 返回完成信号通知 MICROBLAZE.

下面将分别介绍各个子模块的具体内容.

2.2 NAND 闪存读写控制

ONFI(Open NAND Flash Interface)标准是由英特尔、镁光、海力士等为首宣布的统一制定的连接 NAND 闪存和控制芯片的接口标准. 本文中采用的

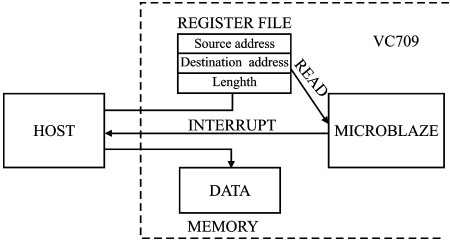


图 2 SSD 写入与应答

镁光芯片也符合这一协议. NAND FLASH 读写控制模块支持 7 种相关命令, 从而实现对 NAND 闪存芯片的读写. 读取 ONFI 标识, 重置, 读取特征值、设置特征值、读取数据、烧写数据、擦出块. 支持 SDR 和 DDR 两种数据传输模式及其切换. 其信号接口管脚如表 1 所示.

表 1 NAND 闪存管脚定义

异步信号	同步信号	描述
ALE	ALE	地址锁存
CE#	CE#	片使能
CLE	CLE	命令锁存
DQx	DQx	数据输入
—	DQS	数据选通
RE#	W/R#	读使能, 读/写
WE#	CLK	写使能, 时钟
R/B#	R/B#	忙闲

其中 CLE 为命令锁存信号, 当为高电平时, 代表输入为命令. ALE 为地址锁存信号, 当为高电平时, 代表输入 DQ 为地址数据复用信号

闪存的烧写数据的基本过程如图 3 所示. (1) 写入一个周期的命令 80H; (2) 写入 5 个周期的地址, 三个周期的行地址两个周期的列地址; (3) 写入数据; (4) 检测 RB 信号拉低, 开始讲数据烧写入; (5) 检测 RB 信号拉高, 结束烧写过程.

在完成一次命令后, 模块引脚 done 端口产生一个中断脉冲, 用于通知 MICROBLAZE 命令完成.

2.3 PCIE3.0 接口

PCIE 总线为端对端的连接方式. 连接在两端互为数据发送端和接收端.

如图 4 所示, PCIE 总线物理链路的一个通路由两组差分信号组成. 发送端的 TX 部件与接收端的 RX 部件使用同一组差分信号, 为发送端的发送链路. 同样, 发送端的 RX 部件与接收端的 TX 部件组成发射端的接收电路.

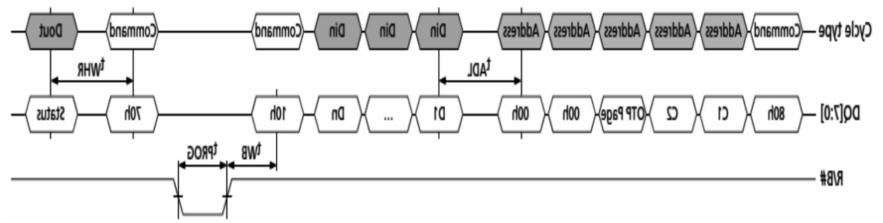


图 3 NAND 闪存烧写时序图

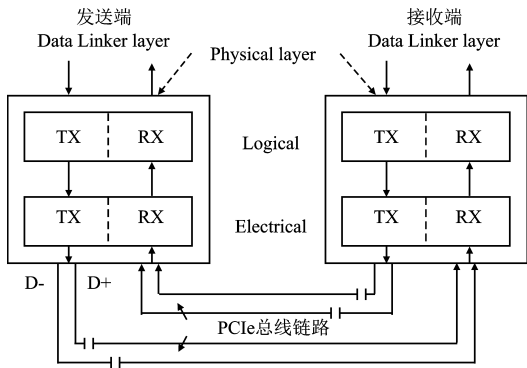


图 4 PCIE 物理层链路管脚

PCIE 总线采用了串行连接的方式,并且使用数据包来进行信号传输.如图 5 所示,在 PCIE 结构中,数据包在核心层产生,然后依次通过事务层,链路层和物理层发送出去.与第二代总线相比,第三代 PCIE 总线在物理层进行了改进,由原每字节添加两位冗余改进为每 128 比特添加两位冗余位,提高了传输效率.

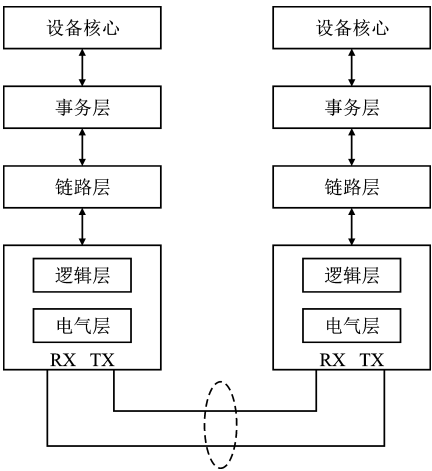


图 5 PCIE 设备的分层

基址寄存器 (BAR) 是与路由相关的一组基址寄存器,所有需要将系统存储器,IO 或存储器映射的 IO (MMIO)地址分配给他们作为目标设备都必须实现基址寄存器.在硬核配置过程中,只采用两个基址寄存器, BAR0, BAR1. BAR0 对应 PCIE 硬核中的 AXI_

LITE 接口,用于写入控制命令. BAR1 对应 PCIE 硬核中的 AXI MASTER 接口,用于数据的传输.在 VIVADO 开发套件中完成地址分配后,将相应的地址映射到与之相对应的基址寄存器中.在本设计中,尽管 VC709 开发板提供了 4 G 的外部内存,但是由于需要为 MICROBLAZE 软核 CPU 及其上运行的各种算法留下一定的内存空间,所以两个 BAR 并不能占用全部内存.所以为数据缓存只分配 1 G 的内存空间,对于 NAND 闪存中大小为 2 MB 的块,可以同时存放 512 块,缓冲大小合适.

2.4 AXI 双总线结构

AXI 为片上互联总线体系结构,赛灵思公司的 IP 核均采用 AXI 接口,支持 AXI 互联.并且 AXI 总线也为 ARM 架构所支持,当需要提供更强的 CPU 时,设计易于移植.

AXI 总线是一种多通道体系结构,有读地址,写地址,读数据,写数据,写完成五条通道.每条通道都有 VALID 和 READY 信号,进行握手通信,然后完成通道内的系统传输.由于信息流的单向传输,减少了延时,使得以更小的面积,更低的功耗,获得更好的性能.

在以往的设计^[8]和 VIVADO 自动连线算法中,均采用单总线结构,即所有的 IP 核挂载在一条 AXI 总线上.单总线结构中,过多不必要的互联,一方面导致 AXI INTERCONNECT 的 MUX 选择器的数量以 2 的倍数增加;另一方面导致过长的关键路径,运行时钟频率较低,性能较差.而采用数据控制的双总线结构,可以减少无信息交互的互连线,从而有效的解决了面积和性能两方面的问题.

如图 6 所示,由 PCIE 接口进入的数据经由数据总线写入片外内存,CDMA 经由数据总线将片外内存数据搬移到 AXI-FTL (FLASH TRANSLATION LAYER) 模块,并最终烧写入 NAND 闪存中.其余接口均连接至另一条控制总线,由 MICROBLAZE 对各模块进行配置和控制.采用双总线的模式,可以提高时钟频率,提高性能.

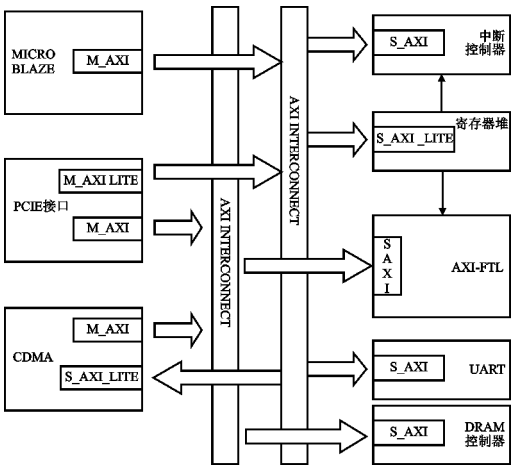


图 6 SSD 双总线结构

3 验证过程

由于 NAND 闪存的特性,在 MICROBLAZE 上需要编写相应的算法,磨损平衡,垃圾回收等.在数据写入过程,软件层必须获知以下几件事情:(1)PCIE 何时完成数据传输;(2)CDMA 何时完成数据搬移;(3)NAND 闪存芯片何时完成数据烧写.在本设计中,采用中断方式通知软件层三件事情.

为此,搭建如图 7 所示的验证平台.主机端与 VC709 开发板通过 PCIE 插槽连接.制作具有 FMC 接口的 NAND 闪存芯片组的子板.通过开发板上的 FMC 插槽连接.

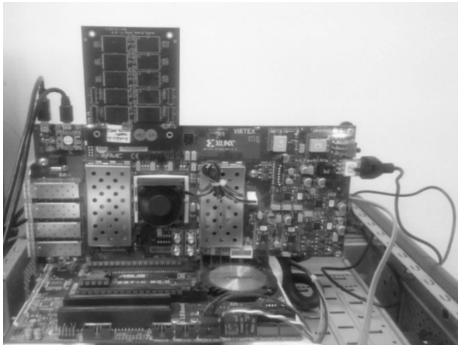


图 7 FPGA 验证平台

验证具体操作如图 8 所示.

- (1)在 FPGA 烧写完成,主机端识别出 PCIE 后,在主机端加载 PCIE 驱动.
- (2)运行 FPGA 固件程序,完成对 NAND 闪存的初始化.
- (3)在主机端运行测试程序,首先通过 PCIE 接口 BAR1 向内存中写入数据,之后通过 BAR0 将数据写入的内存地址,数据长度及将要烧写的 NAND

闪存逻辑地址写入寄存器中,并产生中断信号通知 MICROBLAZE. MICROBLAZE 读取相应寄存器中的值,并保存到相应变量中.

(4)MICROBLAZE 将内存中数据地址和长度,以及 AXI-FTL 的相应地址写入 CDMA 模块,开始 DMA 传输.在传输完成后,DMA 通过中断通知 MICROBLAZE 搬移完成.

(5)MICROBLAZE 将要烧写的 NAND 闪存物理地址写入寄存器中,并给 AXI-FTL 模块一个 start 脉冲信号,在完成烧写后,AXI-FTL 模块 done 信号产生一个脉冲中断,通知 MICROBLAZE.

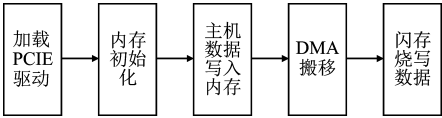


图 8 SSD 验证流程

4 实验结果

在测试程序中,记录传输开始时间与结束时间,并计算传输速度.结果如图 9 所示,设计的 SSD 固态硬盘原型可达到约 1 500 MB/s 的传输速度,并且随着传输块大小的增加,速度仍有提升的空间.

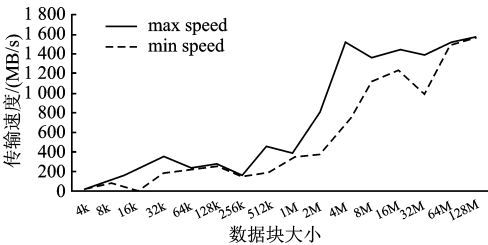


图 9 不同传输文件大小下的传输速度

综合实现后 FPGA 的资源使用如表 2 所示.

表 2 电路逻辑资源使用量

	资源使用	利用率/%
LUT	40 406	9.33
REGISTERS	42 317	4.88
F7 MUX	956	0.44
BLOCK RAM	76	5.17

5 结束语

本文设计并实现了基于 PCIE3.0 的 SSD 固态硬盘.采用 FPGA 实现,兼顾了灵活性和速度.PCIE 接口的最高读写速度约为 1 500 MB/s,远超过 SA-TA 接口的理论极限速度.相信这一设计将有广泛的应用前景.

- 文本自动分类技术上的比较研究[J]. 河北农业大学学报, 2008, 31(3): 120-123.
- [3] 余鹰, 苗夺谦, 刘财辉, 等. 基于变精度粗糙集的 KNN 分类改进算法[J]. 模式识别与人工智能, 2012, 25(4): 617-623.
- [4] 苏毅娟, 邓振云, 程德波, 等. 大数据下的快速 KNN 分类算法[J]. 计算机应用研究, 2016, 33(4): 1003-1006.
- [5] Yao B, Li F, Kumar P. K nearest neighbor queries and knn-joins in large relational databases (almost) for free[C]//Data engineering (ICDE), 2010 IEEE 26th international conference on. IEEE, 2010: 4-15.
- [6] Indyk P, Motwani R. Approximate nearest neighbors: towards removing the curse of dimensionality[C]//Proceedings of the thirtieth annual ACM symposium on Theory of computing. ACM, 1998: 604-613.
- [7] Jagadish H V, Ooi B C, Tan K L, et al. iDistance: An adaptive B+-tree based indexing method for nearest neighbor search[J]. ACM Transactions on Database Systems (TODS), 2005, 30(2): 364-397.
- [8] Datar M, Immorlica N, Indyk P, et al. Locality-sensitive hashing scheme based on p-stable distributions [C]//Proceedings of the twentieth annual symposium on Computational geometry. ACM, 2004: 253-262.
- [9] Zhu P, Zhan X, Qiu W. Efficient k-Nearest Neighbors Search in High Dimensions Using MapReduce[C]//Big Data and Cloud Computing (BDCloud), 2015 IEEE Fifth International Conference on. IEEE, 2015: 23-30.
- [10] 陈佑雄, 向阳, 张骐, 等. 基于 LSH 和 MapReduce 的近邻模型推荐算法[J]. 微电子学与计算机, 2013, 30(12): 47-49.
- [11] Gao Yanjie. Data Processing with Spark[M]. Beijing: China Machine Press, 2015.
- [12] Andoni A, Indyk P. E2lsh: Exact euclidean locality-sensitive hashing [EB/OL]. [2017-01-09]. <http://web.mit.edu/andoni/www/LSH/>. 2004.
- [13] Park Y, Hwang H, Lee S. A Fast k-Nearest Neighbor Search Using Query-Specific Signature Selection [C]//Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. ACM, 2015: 1883-1886.
- [14] 彭凯, 汪伟, 杨煜普. 基于余弦距离度量学习的伪 K 近邻文本分类算法[J]. 计算机工程与设计, 2013, 34(6): 2200-2203.
- [15] 戴上平, 冯鹏, 刘盛英杰, 等. 基于余弦距离的敏感哈希的 KNN 算法在中文文本上的快速分类[J]. 计算机工程与科学, 2015, 37(10): 1971-1976.

作者简介:

赵彤男, (1992-), 硕士研究生. 研究方向为数据挖掘.
E-mail: laokuazi@163.com.

刘斌男, (1966-), 博士, 硕士生导师. 研究方向为模糊技术、人工智能、数据挖掘.

李涛男, (1990-), 硕士研究生. 研究方向为数据挖掘.

(上接第 66 页)

参考文献:

- [1] Luo J, Fan L, Chen Z, et al. A solid state drive architecture with memory card modules[J]. IEEE Transactions on Consumer Electronics, 2016, 62(1): 17-22.
- [2] Bez R, Camerlenghi E, Modelli A, et al. Introduction to flash memory[J]. Proceedings of the IEEE, 2003, 91(4): 489-502.
- [3] Seong Y J, Nam E H, Jin H Y, et al. Hydra: A block-mapped parallel flash memory solid-state disk architecture [J]. IEEE Transactions on Computers, 2010, 59(7): 905-921.
- [4] 雷雨. 基于 PCIE 固态硬盘的嵌入式存储技术[D]. 北京: 中国科学院, 2014.
- [5] Kim H, Choi H, Kim Y et al. MS2: NAND-flash SSD-based multi-layered storage management system [C]//2016 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC). Chengdu, 2016: 100-104.
- [6] Huh J, Kim J S, Jung D, et al. Parameter-aware I/O management for solid state disks (SSDs) [J]. IEEE Transactions on Computers, 2012, 61(5): 636-649.
- [7] Cai Y, Haratsch E F, McCartney M, et al. FPGA-based solid-state drive prototyping platform [C]//IEEE, International Symposium on Field-Programmable Custom Computing Machines. IEEE, 2011: 101-104.
- [8] 李硕, 刘芳, 肖依. 基于 FPGA 的 PCIe SSD 设计与实现[J]. 中国科技论文, 2014, 19(4): 403-406.

作者简介:

宋鹏程男, (1993-), 硕士. 研究方向为数字电路设计.
E-mail: songpc14@mails.tsinghua.edu.cn.

张春男, (1972-), 博士, 副研究员. 研究方向为 RFID 技术与新型触觉传感器.