

基于 FPGA 的 SATA IP CORE 设计^①

Design of Serial ATA Based on FPGA

唐纯杰 黎福海 欧阳科文 (湖南大学 电气与信息工程学院 湖南 长沙 410082)

摘要: 在深入研究 Serial ATA 1.0 协议的基础上, 分析讨论了协议各层的状态变迁与设计难点, 并按照标准设计了基于 Xilinx FPGA 的高性能 SATA IP CORE, 提出了可重用性的设计结构。搭建了一个 SATA IP 验证平台, 在此基础上实现了 SATA 接口的固态电子硬盘 SSD。

关键词: Serial ATA FPGA RocketIO 固态电子硬盘

20 世纪 90 年代以来, 并行 ATA 已统治个人计算机、服务器存储设备总线协议十余年。然而, 随着各种应用对存储性能需求成指数级增长, 其已逐渐成为限制设备性能的瓶颈。鉴于并行总线性能的提升空间有限, 2001 年, 由 Intel 等众多厂商组成的 Serial ATA 委员会正式研究确立了新的串行存储总线标准 Serial ATA, 为大规模数据存储提供更高的性能与可靠性。

经过众多业界厂商不遗余力的推行, 目前 Serial ATA 已经在个人计算机、服务器、网络存储(NAS)以及独立磁盘冗余阵列(RAID)等诸多领域普及。然而, Serial ATA 总线协议非常复杂, 目前仍然只有少数国外厂商能提供 Serial ATA IP, 而且授权费用昂贵。本文试图解决使用 Serial ATA 接口设备研发费用高、时间长的问题, 使用 Xilinx 的 FPGA 设计实现了 Serial ATA 协议。它完全遵从 Serial ATA 1.0 标准, 支持 48 位 LBA 寻址、输入输出缓冲、错误处理等特性, 并且嵌入高速 DMA 引擎。

1 Serial ATA 协议概述

相比前代总线标准 ATA, Serial ATA 有着革命性变化。它是点对点的串行互联总线, 采用差分信号线传输数据。Serial ATA 1.0^[1]总线标准中将整个协议明确地分为四层, 典型结构拓扑图如图 1 所示。

层与层之间有明确的分工。应用层(Application)

负责所有的 ATA 命令执行, 包括对控制命令模块寄存器的访问; 传输层(Transaction)主要负责以帧信息结构 FIS(Frame Information Structure)的形式传输主机和设备之间的控制命令与数据; 链路层(Link)主要负责信道编解码, 包括 8B/10B 编解码与 CRC 校验; 物理层(Physical)主要负责串/并转换、发送和接收串行数据流。

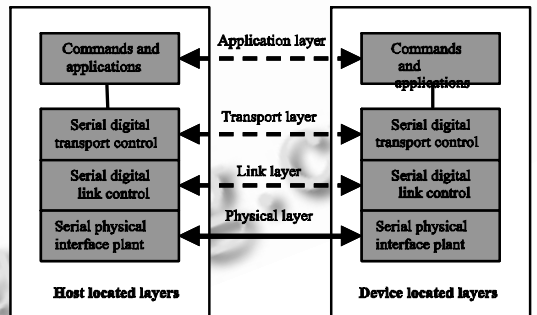


图 1 Serial ATA 典型拓扑图

2 协议设计与实现

系统采用 Virtex-4 FX^[2]系列 FPGA 设计实现。Virtex-4 FX 系列 FPGA 是 Xilinx 公司推出的高端产品, 逻辑集成度高, 拥有 500MHz 系统时钟控制, 且集成 PowerPC 与 622 Mbps~6.5 Gbps RocketIO? 收发器, 并针对嵌入式处理和高速串行连接功能进行了优化, 特别适用于高速串行通信的开发。系统结构框图^[3]如图 2 所示。

① 收稿时间:2008-12-31

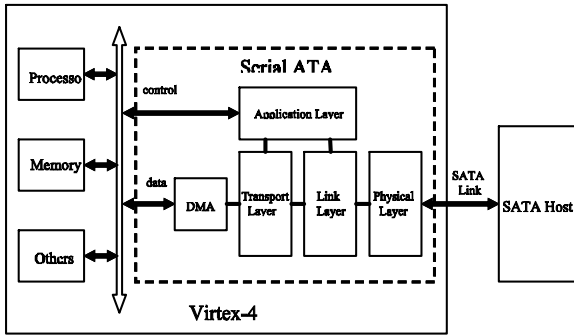


图 2 系统结构框图

2.1 物理层

正文物理层的主要任务是建立正常通讯链接，包括提供给上层连接情况、串/并转换、传输数据流。

Serial ATA 1.0 传输的是 1.5Gb/s 差分 LVDS NRZ 串行数据流，对于高速串行数据流传输以及串/并转换部分，采用 Virtex-4 FX 内部的千兆位级收发器硬核 RocketIO MGT(Multi-Gigabit Transceiver)^[4]实现，通过对 RocketIO 时钟、参数选择，可得到 20bit 数据流，RocketIO 主要配置参数如表一。

表 1 RocketIO 时钟参数配置

TX 属性	设置	RX 属性	设置
TXPLLNDIVSEL	20	RXPLLNDIVSEL	20
TXOUTDIV2SEL	4	RXOUTDIV2SEL	4
TXASYNCDIVIDE	2' b01	RXUSRDIVISOR	1
TXCLKMODE	4' b0100	RXCLKMODE	6' b000011
TXOUTCLK1_USE_SYNC	FALSE	TXOUTCLK1_USE_SYNC	FALSE
TXCLKO_FORCE_PMACLK	FALSE	TXCLKO_FORCE_PMACLK	FALSE
TX_CLOCK_DIV1_DER	2' b01	RX_CLOCK_DIV1_DER	2' b01
TXABPMACLKSEL	REFCLK1	RXABPMACLKSEL	REFCLK1

通讯链接的建立是靠对段外控制信号 OOB(Out Of Band)的检测实现的，并且向上层提供了物理层的链接情况。OOB 信号主要有三个：COMRESET、COMINIT 和 COMWAKE，都是由四个 D/K 控制字的不同序列组成，D/K 控制字包括 K28.5，D10.2，D27.3。对它们的检测是通过检测 ALIGN 数据流的宽度和 ALIGN 数据流的间隙达到的。MGT 接收模块的

输出信号 RXSIGDET 低电平对应为 ALIGN 数据流的宽度，而高电平则对应为 ALIGN 数据流的间隙。在设计中，系统时钟为 75MHz，间隙宽度为 260~420ns(19~32 cycles)时为 COMRESET，间隙宽度为 80~200ns(6~13 cycles)时为 COMWAKE。对 OOB 信号的发送则预先将要发送的 OOB 信号固化在程序中，然后根据状态变迁提交数据，并控制发送时间。

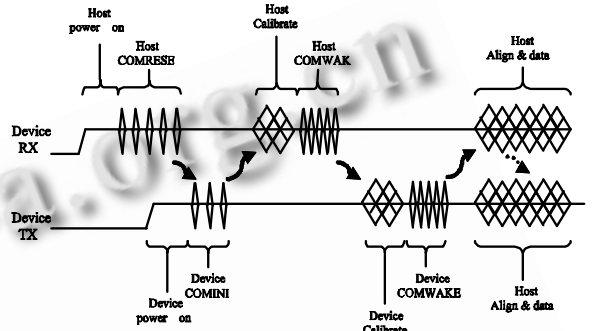


图 3 系统通讯链接建立序列

物理层通讯链接的状态变迁如图 3 所示，系统由有限状态机实现，下面详细介绍通讯链接建立时状态变迁过程：

- A: 主机复位，发出 COMRESET；
- B: 设备检测到 COMRESET 后以 COMINIT 响应，连续发送 6 个传输周期。(COMINIT 为重新连接开始信号，设备可以在任意时间发送 COMINIT 重新建立连接)；
- C: 主机校准，发出 COMWAKE；
- D: 设备接收到 COMWAKE 后，发送 COMWAKE，接着连续发送 ALIGN 数据流，
- G: 主机锁存数据，检测到 ALIGN 后以接收的速度转发，之前一直发送控制字 D10.2。
- H: 设备锁存数据，检测到 ALIGN 后发送同步信号 SYNCp，通讯建立成功，进入到正常操作模式。如在 54.6us(2048 个基元传输时间)内未检测到 ALIGNp，则进入错误处理，等待重新连接。

2.2 链接层

链接层的主要任务是信道编解码。当传输层有数据需要传送时，链接层计算 CRC 值，为其加上包头 SOF(start of frame)和包尾 EOF(end of frame)。经过打包的数据通过 CRC、加扰和 8B/10B 编码后，传递给物理层，并在包的传输过程中通过插入各种

primitive 基元进行流量控制, 在成功传递完有效数据后, 连接层等待对方的应答, 并把结果传给上层; 当链接层从物理层接收到有效数据时就对其进行 8B/10B 解码、解扰以及 CRC 效验, 然后将数据提交到传输层。

通过 8B/10B 编解码、扰码器与 CRC 效验, 通很好地改善数据码流的直流基线漂移与抖动, 能非常有效的增强数据传送的有效性。系统所采用的扰码器抽头系数为:

$$x^{16} + x^{15} + x^{13} + x^4 + 1。CRC 伴随式: x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1。$$

链接层模块如图 4 所示, 主要包括: CRC 值计算/效验模块、加扰/解扰模块、8B/10B 编码解码模块^[4], primitive 生成/译码模块和链接层状态机控制模块, 各个模块都接收控制状态机的控制信号, 并反馈自己的状态信号, 控制状态机完成链接层状态变迁, 同时和传输层、物理层的控制模块交流控制和状态信息。由于物理层数据流位宽为 20bit, 采用两通道 8B/10B 编码解码模块对数据进行并行操作, 保证了数据流连续性。primitive 生成/译码模块将 primitive 序列固化, 根据状态机的控制信号对各 primitive 基元进行发送和译码, 实现数据的流量控制。primitive 基元是由四个 D/K 控制字组成的序列, 系统应用的 primitive 类型及其组成如下表 2。

表 2 Primitive 类型及其组成

primitive 类型	组成
Align	{D27.3,D10.2,D10.2,K28.5}
Cont	{D25.4,D25.4,D10.5,K28.3}
DMAT	{D22.1,D22.1,D21.5,K28.3}
Receiver Ready	{D10.2,D10.2,D21.4,K28.3}
Receive-In-Progress	{D21.2,D21.2,D21.5,K28.3}
Sync	{D21.5,D21.5,D21.4,K28.3}
Hold	{D21.6,D21.6,D10.5,K28.3}
Recevier OK	{D21.1,D21.1,D21.5,K28.3}
Start Of Frame	{D23.1,D23.1,D21.5,K28.3}
Wait for frame Termination	{D24.2,D24.2,D21.5,K28.3}
End Of Frame	{D21.6,D21.6,D21.5,K28.3}
Reception Error	{D22.2,D22.2,D21.5,K28.3}
Transmission data ready	{D23.2,D23.2,D21.5,K28.3}
Hold Ack	{D21.4,D21.4,D10.5,K28.3}

去除封装, 提交给应用层。FIS 有不同的格式, 分别用不同的 FIS 号标示。传输过程中主要用到的为 Data FIS (46H)和 Register FIS(27H、34H)两种。

对于 DATA FIS, 发送时应用层直截将 DMA 输出的数据以 FIS 格式传输给应用层, 接收时将 FIS 数据经过 64Dword 的 FIFO 缓存, 以备 DMA 读取数据。传输过程中伴随流量控制。

对于 Register FIS 以及其它非数据 FIS, 发送时根据要求传输的 FIS 类型在寄存器堆里提取 FIS 的内容, 以 8 Dword 作为缓存将提取的内容放在适当的位置, 然后通知链接层开始传输 FIS, 传输完成后向上层返回传输结果。接收时同样以 8Dword 作缓存存储从链接层获取到 FIS, 判断 FIS 类型, 根据 FIS 类型把 FIS 内容放在寄存器堆的适当位置, 最后向上层返回传输结果。

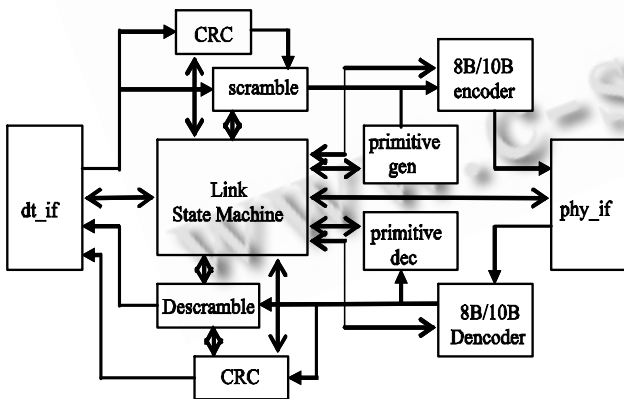


图 4 链接层模块结构框图

2.3 传输层

传输层不必关心需要传输或接收信息的多少, 只需把要传输的数据封装成 FIS(Frame information Structures)格式, 发送到链接层, 或者把收到的 FIS

2.4 应用层

应用层设计兼容 PATA 的寄存器窗口, 包括 Sector count、Sector number、Cylinder low、Cylinder high、Device /Head、Data、Command、Feature、Error、Status。另外, 应用层设置有设备

控制(包括中断控制)、设备配置等寄存器。

应用层模块如图五所示,包括: ATA 接口、控制、任务、状态机和传输层接口 5 大模块。其中, ATA 接口模块主要产生读写逻辑,完成处理器对寄存器的访问;控制模块负责接口控制与中断控制;状态机负责完成对命令的译码,实现 PIO 和 DMA 传输。任务模块包含有各类寄存器,并根据状态机译码进行数据传输,执行命令;传输层接口模块向下层提供寄存器访问逻辑,实现与传输层接口。

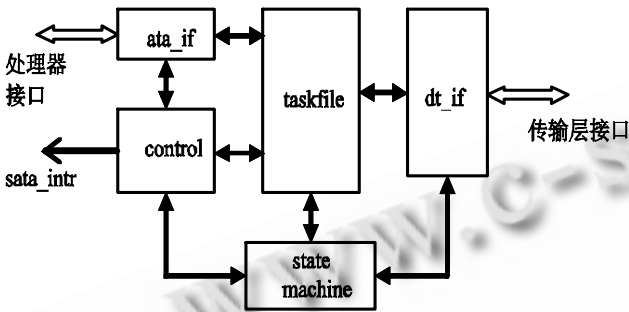


图 5 应用层模块结构

当处理器收到中断后,通过接口模块读取命令,然后根据命令进行参数设置,命令执行由任务模块与状态机自动完成。如若命令要求传输数据,处理器还需设置启动 DMA,数据传输完成后,DMA 通过中断方式通知处理器。

3 Serial ATA 验证

由于 SATA 协议的复杂性,要完全实现非常困难,设计主要实现了 SATA 基本通信功能,并在此基础上利用外扩 FLASH 做存储,设计了一个 8GB 的固态电子硬盘 SSD,实现硬盘功能,并安装操作系统。

将系统安装上 Windows XP 后,以主盘方式启动,通过总线分析仪捕出的数据(如图六所示)分析可以得出,串行通信良好,能实现 DMA 传输模式,协议得到了很好的实现。

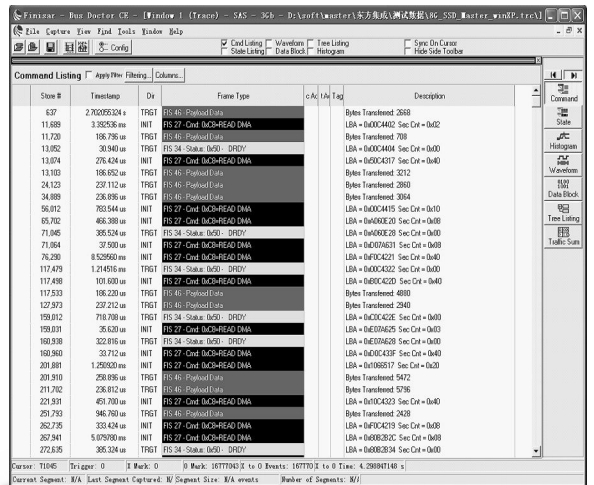


图 6 系统数据传输

4 结语

本文所介绍的基于 FPGA 的 SATA IP CORE 设计在 Xilinx Virtex-4 FX 器件上得到了很好的结果。系统工作在 150MHz 频率下,总共占用 11900 个 Slices、18600 个四输入 LUT。设计中的 SATA IP 具有使用方便、集成度高、成本低等优点,在固态电子硬盘 SSD 或其它基于 SATA 接口的系统开发中应用本设计,将使开发变得方便快捷,更能够适应市场需求。

参考文献

- 1 Serial ATA. High speed serialized AT Attachment Revision 1.0a, 2003.
- 2 Xilinx Inc. Virtex-4 User Guide. Xilinx Inc, 2006.
- 3 Xilinx Inc. 嵌入式串行 ATA 存储系统. Xilinx Inc, 2006.
- 4 Xilinx Inc. Virtex-4 RocketIO Multi Gigabit Transceiver User Guide. USA: Xilinx Inc, 2007.
- 5 A.X. Widmer, P.A. Franzsek, ADC-Balanced Partitioned-Block, 8B/10B Transmission Code, 1983.
- 6 Ciletti MD. Advanced Digital Design with the VERILOG HDL, 2005.