

嵌入式 SATA2.0 接口控制器中的非对称缓冲器设计

李 晔, 刘元安, 袁东明, 胡鹤飞

(北京邮电大学 电子工程学院, 北京 100876)

摘 要: 本文描述了基于 Xilinx Virtex-5 FPGA 的嵌入式 SATA2.0 主机接口控制器中, 物理层与链路层之间数据位宽转换缓冲器的设计方法. 针对转换过程中可能会出现的数据错序问题, 采用多比特移位寄存器组设计了应用于物理层 16bit 与链路层 32bit 位宽数据之间的转换电路. 仿真和板级验证结果表明, 该逻辑电路与 Xilinx FPGA 内嵌的 FIFO 相比, 平均时延降低了 70%, 在完成相同功能的情况下, 使用了更少的芯片资源和控制逻辑.

关键词: SATA; 主机控制器; Rocket I/O; 数据缓冲

中图分类号: TP302

文献标识码: A

文章编号: 1000-7180(2012)02-0178-05

Design of Asymmetric Buffer in Embedded SATA2.0 Interface Control System

LI Ye, LIU Yuan-an, YUAN Dong-ming, HU He-fei

(School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China)

Abstract: This paper depicts a data width transfer buffer design that interfaces physical layer with link layer of embedded SATA2.0 host bus adapter (HBA) system which based on Xilinx Virtex-5 FPGAs. For the purpose of eliminating the disorders from data transferring, the logical circuit, which is composed of multi-bit shift register group, defined between physical layer and upper layers has been simulated and board level verified. Compared with the existing Xilinx FIFOs, the design's average latency is reduced by 70 percent. Results show that the asymmetric buffer can save more chip resources and control logic under the same throughput capability.

Key words: SATA; HBA; rocket I/O; data buffer

1 引言

SATA (Serial Advanced Technology Attachment) 协议在大容量、高可靠的磁盘接口传输方面, 具有可扩展、易集成、传输高效等特点, 相对于以往的并行传输接口 PATA (Parallel Advanced Technology Attachment), SATA 接口更简单, 仅使用两对差分线进行传输, 减小了信号间串扰, 能提供更高的传输速率, 目前广泛应用的 SATA2.0 标准线速率可达 3.0Gbps, 可满足普通用户存储和工业应用的要求.

Xilinx 公司在其 Virtex-5 FX 系列的 FPGA 中集成了 Rocket I/O GTX (以下简称 GTX) 串行数据

收发器^[1], 该收发器支持多种通信接口协议 (SATA, CPRI, PCI Express, XAUI 等), 支持线速率 750Mbps~6.5Gbps. GTX 还集成了 8B/10B 编解码模块和 SATA 带外信号 (Out of Band, OOB) 的检测功能, 能方便的应用于 SATA 磁盘主机侧控制器 (Host Bus Adapter, HBA) 的物理层接口中. SATA 协议提倡 HBA 链路层以上的逻辑工作时以双字 (32 bit) 作为数据宽度, 而 Xilinx 公司提供的 SATA 物理层参考设计^[2] 配置将 GTX 收发器的并行数据位宽设置为单字 (16 bit), 当数据在 HBA 物理层 (16bit) 和 HBA 链路层 (32bit) 之间交换时, 需要进行位宽转换.

针对在进行位宽转换时可能会出现数据的高、

收稿日期: 2011-10-10; 修回日期: 2011-10-27

基金项目: 国家重大专项 (2011ZX03001-004-02)

低单字错序问题,本文提出了一种非对称缓冲器,以完成 HBA 两个层次间的速率匹配。

2 SATA 主机控制器物理层

HBA 的物理层时钟产生模块与 GTX 组件的连接方式如图 1 所示。

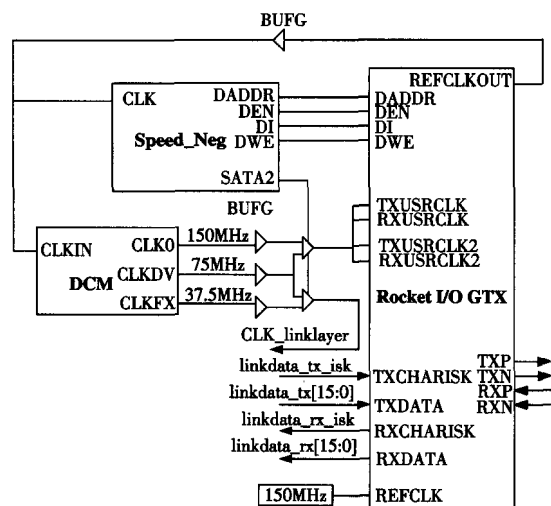


图 1 HBA 物理层时钟结构

数字时钟管理单元 (Digital Clock Management, DCM) 利用其输入端的 150MHz 时钟产生 3 个同相位不同频率的时钟输出: 150MHz、75MHz 和 37.5MHz, 用于 HBA 物理层和上层逻辑的时钟。

为了使 HBA 能向下兼容 SATA1.0 (线速率 1.5Gbps), 需要能动态调整 GTX 的时钟工作频率。图 1 中, Speed_Neg 模块有两个作用: ① 通过 GTX 的 DRP 端口, 修改 GTX 内部的共享 PLL 分频比, 动态调整 GTX 的工作频率; ② 读取 GTX 当前工作状态, 判断处于 SATA2.0 或 SATA1.0 速率下, 从而选择 DCM 模块向 GTX 输出的工作时钟频率。

当 HBA 物理层工作在 SATA1.0 时, DCM 输出至 GTX 的时钟频率 TXUSRCLK/RXUSRCLK2 (用于 GTX 内部发送和接收通路数据时钟) 均为 75MHz; HBA 物理层工作在 SATA2.0 时, 上述四个时钟的频率则为 150MHz。GTX 对外的数据端口 TXDATA/RXDATA 和 HBA 链路层进行数据交互, 数据位宽均为 16bit, 此外还有指示数据类型的 TXCHARISK/RXCHARISK 端口, 位宽均为 1bit。

SATA 协议推荐 HBA 在处理数据时以 Dword (32 bit) 为单位^[3], 以便进行时延控制和状态机跳转, 能有效降低设计复杂度, 提高 HBA 工作效率。

当 GTX 将数据与 HBA 链路层进行数据交互时, 需要进行 16bit 与 32bit 的数据转换。HBA 链路层以上的数据位宽是物理层的 2 倍, 其工作时钟频率仅需为物理层的 1/2, 即 SATA1.0 时, HBA 链路层时钟频率为 37.5MHz, SATA2.0 时, 时钟频率为 75MHz, 该时钟也是通过 Speed_Neg 模块对 DCM 模块的输出时钟进行选择的。

3 物理层和链路层间数据格式

目前大多数高速串行数据传输协议都是用 8B/10B 编码, 该编码有很多优点: 直流平衡, 消除线路上发送端连续发送多个 0 或 1 导致接收端由于电压位阶关系无法识别引起接收出错, 也利于接收端调整其与发送端之间的时钟同步和纠正错误比特^[4]。同样 SATA 协议也使用 8B/10B 编/解码提高线路的稳定性。按照协议描述, 该过程在 HBA 链路层将数据提交至物理层之前进行。GTX 中内嵌了 8B/10B 编解码模块, 只需在配置 GTX 时使能该功能即可。下面结合 8B/10B 的码字类型讲述 SATA 物理层和链路层之间的数据格式。如图 2 所示。

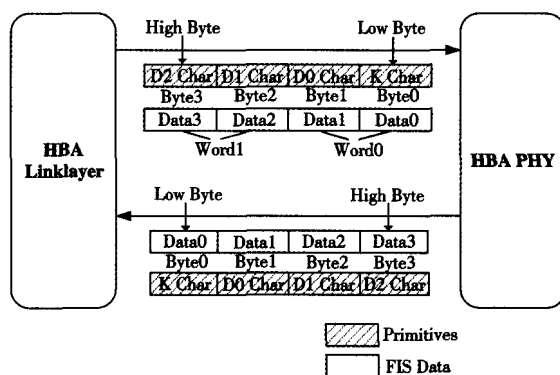


图 2 HBA 物理层与链路层间数据格式

主机控制器 HBA 和磁盘之间传递两类数据: 控制原语 (以下称为 primitive) 和数据 (以下称为 FIS Data), 每个单位数据的宽度均为 1 个 Dword (32 bit)。在物理层处理时, 为了进行 8B/10B 码组映射, 将 1 个 Dword 数据分成 4 个 8 bit 的码组, 每组 8bit 数据在物理层编码时映射成 10bit 数据然后发送出去, 接收时先将 10bit 码组反映射成 8bit 数据再分组传送给 HBA 链路层。8B/10B 编码根据输入编码器前的数据类型 (普通数据 D Char 和控制信号 K Char) 选择不同的映射方式。如图 2 所示, primitive 以一个控制字符 K Char 开头, 后接三个数据字符 D Char; FIS Data 则按 8bit 分组后以数据字符 D Char 的方式映射。由于 K Char 和 D Char 的

码组映射方式不同,故需要指示物理层哪个字符是 K 字符,剩余不指示的便是 D 字符,同样的,物理层接收时也会告诉链路层哪些字符解码后是 K 字符.该功能对应图 1 中 GTX 的 TXCHARISK/RXCHARISK 端口, TXCHARISK 指示物理层待发送的 linkdata_tx[15:0] 中低 8bit 是 K 字符,高电平有效. RXCHARISK 链路层接收到的 linkdata_rx[15:0] 中低 8bit 是 K 字符.

如图 2 所示, HBA 链路层处理的数据宽度是 Dword, primitive 最高字节是 D2 Char, 最低字节是 K Char; FIS Data 最高字节是 Data3, 最低字节是 Data0. HBA 物理层数据发送端口 linkdata_tx[15:0] 和数据接收端口 linkdata_rx[15:0] 每个时钟处理该 Dword 的高字 (Word1) 或低字 (Word0). 当数据从物理层传输至链路层时,总是先传输 Word0, 后传输 Word1, 重组时必须正确的将其组成为 {Word1, Word0} 格式, 若将前一个 Dword 的 Word1 和后一个 Dword 的 Word0 组成 {Word0cur, Word1pre} (下标 cur 表示当前物理层正在传输的 Dword, pre 表示物理层前一个处理的 Dword) 就会发生数据错误, HBA 上层的逻辑也将不能正常工作. 同理, 在 HBA 链路层向物理层发送数据时, 也需要进行数据转换, 先发送 Dword 的 Word0, 后发送 Word1, 否则磁盘不能识别 HBA 发送的数据, 通信将失败.

本文提供了一种非对称缓冲器设计, 确保能将数据的高低 Word 转换成所需的 Dword, 且该缓冲器比其他常用的位宽转换器件有一定的优势.

4 非对称缓冲器设计

进行数据位宽转换时, 通常使用非对称异步 FIFO 或双端口 RAM, Xilinx 公司都提供了这两种模块的 IP 核, 但是笔者在设计中发现这两个类型的模块都不适用于 SATA 磁盘控制器的物理层和链路层数据位宽转换: ① 非对称异步 FIFO^[5] 从写使能有效到 FIFO 指示数据有效需要时延为: $1 \text{ WR_CLK} + 4 \text{ RD_CLK} (+1 \text{ RD_CLK})^a$, 其中 WR_CLK 为 FIFO 写入时钟周期, RD_CLK 为 FIFO 读取时钟周期, a 取值 0 或 1, 是由异步时钟域读写而产生的不确定延时. 在 FIFO 指示数据有效后, 读使能有效并输出数据至少需要 1 个 RD_CLK 的时延. SATA 协议规定, HBA 和磁盘进行流量控制过程中, 磁盘发出流量控制 primitive 的请求后, 必须在 20 个链路层时钟之内收到主机控制器 HBA 对此原语

的响应, 故非对称异步 FIFO 的时延太大, 不能满足此项应用. ② 双端口 RAM^[6] 需要添加额外的读写端口地址控制电路, 数据的输出相对于地址的变化也至少有 1 个 RD_CLK 的时延, 增加了芯片资源的同时并没有提升效率. ③ 链路层和物理层写入和读取具有异步性和随机性, 无论是非对称异步 FIFO 还是双端口 RAM 的 IP 核, 都无法保证数据的高低 Word 能正确转换.

严格来说, 本文设计的缓冲器并不是异步的^[7], 图 1 所示, 该缓冲器的读写时钟均来自同一个数字时钟管理单元 DCM, 缓冲器两侧的时钟 150/75MHz 和 75/37.5MHz 是同相位的, 即每个 75/37.5MHz 时钟周期里必然有两个 150/75MHz 时钟的上升沿, 这非常重要, 否则缓冲器可能不能正确寻找到高低 Word 的边界, 从而导致转换错误.

4.1 物理层 16bit 至链路层 32bit 转换模块

该模块将物理层 GTX 接收到的 16bit 并行数据转换成链路层可处理的 32bit 并行数据, 模块设计结构如图 3 所示.

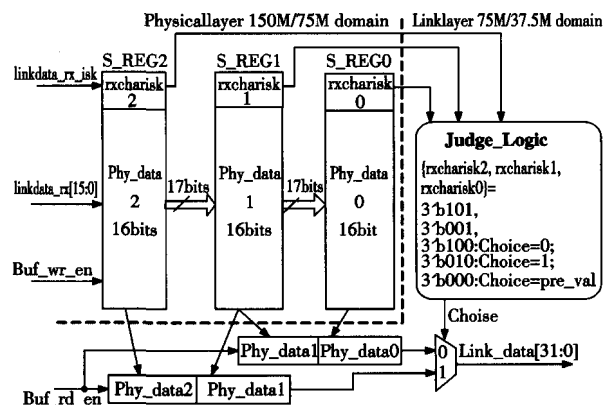


图 3 16bit 转 32bit 模块结构图

虚线划分开了两个时钟域, 物理层接收到的 16bit 并行数据 linkdata_rx[15:0] 和 K 字符指示位 linkdata_rx_isk 在 150/75MHz 时钟作用下, 存入移位寄存器 S_REG2, 该寄存器每周期将 17bit 数据右移动, 即 $S_REG2 \rightarrow S_REG1 \rightarrow S_REG0$, buf_wr_en 是寄存器的写入使能, 这样, S_REG2、S_REG1 和 S_REG0 中就有三个 Word 的数据, 其中必有两个数据属于同一个 Dword. 两种可能, 正确重组后的 Dword 应该由 {S_REG2[15:0], S_REG1[15:0]} 或 {S_REG1[15:0], S_REG0[15:0]} 构成.

判断逻辑 Judge_Logic 工作在 75/37.5MHz 时钟下, 根据 S_REG 移位寄存器最高比特 (RXCHARISK) 的值来判断选择哪一种组合方式. 读使

能 Buf_rd_en 有效后, Judge_Logic 开始工作:① rxcharisk1 为 1 时,选择 {S_REG2[15:0], S_REG1[15:0]} 作为 Link_data[31:0] 的输出,同时置位选择标志为 Choice;② rxcharisk1 为 0 且 rxcharisk2 和 rxcharisk0 不全为 0 时,选择 {S_REG1[15:0], S_REG0[15:0]} 作为 Link_data[31:0] 的输出,同时复位选择标志位 Choice;③ rxcharisk2、rxcharisk1 和 rxcharisk0 均为 0 时,选择标志位 Choice 保持不变,根据 Choice 的值选择重组数据来源,即保持在接收 primitive 时的重组方式。

芯片上电工作后,磁盘一直向 HBA 发送数据 (primitive 或 FIS Data),而 HBA 仅当物理层 GTX 上电复位准备好之后才能接收数据,故第一个收到的 Word(16bit)是不确定的,可能是某个 Dword 的高 16bit,也可能是低 16bit,通过 primitive 中 K 字符的标志判断 Dword 的重组方式,接下来的数据重组中均保持此规则。

4.2 链路层 32bit 至物理层 16bit 转换模块

链路层向物理层的数据位宽转换传输时,需要保证传送给物理层时均是先发送数据的 Word0,下一个时钟周期发送该 Dword 的 Word1,接收端磁盘方能正确解析发送的数据,模块设计结构如图 4 所示。

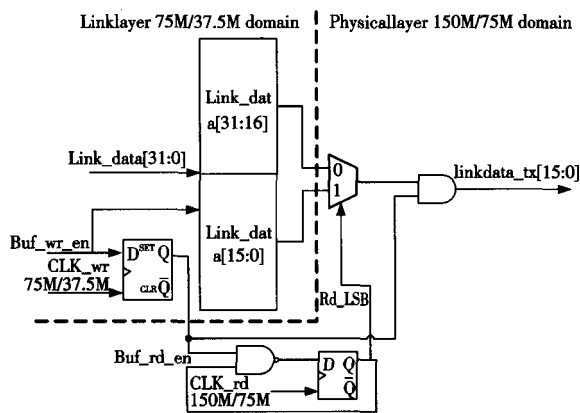


图 4 32bit 转 16bitS 模块结构图

待进行位宽转换的数据 link_data[31:0] 在链路层时钟 (75/37.5MHz) 的驱动下更新转换模块内的 32bit 寄存器 buf_data_reg。Rd_LSB 每个物理层时钟 (150/75MHz) 翻转一次,选定 32bit 寄存器 buf_data_reg 高、低字进行输出。

尽管链路层时钟和物理层时钟来自同一个 DCM 模块,两个时钟的相位是对齐的,但高频率物理层时钟的上升沿有可能对准低频率链路层时钟的

下降沿,若此时置位寄存器的读使能 buf_rd_en,每个 Dword 都是先输出高位的 Word1,后输出低位的 Word0,从而发生高低位错序。为防止上述情况的发生,必须保证寄存器的读使能 buf_rd_en 有效沿对准物理层和链路层时钟的上升沿,采用图 3 中的结构,保证了读出的 Dword 高低 16bit 的顺序正确。

图 4 未画出检测 linkdata_tx[15:0] 低 8bit 是否 8B10B 编码的 K 字符的逻辑, SATA 协议中 K 字符只有两个: K28.3 (7Ch) 和 K28.5 (BCh),控制原语的低 16bit 也只有四种组合,如表 1 所示。

表 1 SATA 控制原语低 16bit 组合类型

Byte1	Byte0	对应的 16 进制数
D10.2	K28.5	4A_BCh
D10.5	K28.3	AA_7Ch
D21.5	K28.3	B5_7Ch
D21.4	K28.3	95_7Ch

工作在物理层时钟下的判断逻辑只需找到这四个码组,即可置位 K 字符标志位 TXCHARISK。

5 验证及结果分析

该非对称缓冲器已应用于笔者设计的嵌入式高速 SATA 磁盘接口控制器中,控制器工作在 Xilinx Virtex-5 XC5VFX70T-FF1136 平台上,图 5 和图 6 为该缓冲器功能仿真结果:

图 5 显示了系统复位进入工作状态后从物理层先收到 3 个 primitive (2 个 ALIGNp (7B_4A_4A_BCh) 和 1 个 CONTp (99_99_AA_7Ch)), 然后收到 FIS Data 的情况。数据的写入到读出仅有 1 个链路层时钟周期的时延,比 Xilinx 非对称异步 FIFO IP 核的 6.5 个链路层时钟短了 84.6%。

图 6 显示了系统复位进入工作状态后从链路层依次收到 2 个 SYNCp (B5_B5_95_7Ch) 原语, 1 个 CONTp (99_99_AA_7Ch) 原语, 2 个 R_RDYp (4A_4A_95_7Ch) 原语, 1 个 CONTp 原语和 5 个 FIS Data 的情况。数据的写入到读出时延为 3 个物理层时钟周期的延时,比 Xilinx 非对称异步 FIFO IP 核的 8 个物理层时钟短了 62.5%。

表 2 为非对称缓冲器与 Xilinx 非对称异步 FIFO IP 核 (写入深度 16) 的综合结果对比,综合工具为 Synplify Pro 9.6.2。

