

USB3.0 物理层中弹性缓冲的设计与实现

朱小明, 王小力, 程 曾

(西安交通大学 电信学院, 陕西 西安 710049)

摘 要: 弹性缓冲用于在不同的时钟域中同步数据以保持数据的完整性, 在 USB、PCIE 等高速串行总线的物理层中普遍应用。通过分析弹性缓冲的作用机制, 根据 USB3.0 的协议要求, 采用具有写指针屏蔽、指针跳跃、断点保存与握手、输出控制等具有创新功能的异步 FIFO 来设计弹性缓冲, 很好实现了时钟频率补偿的目的。所设计的弹性缓冲采用并行 10 位数据, 读写时钟可达到 500 MHz 的频率。该研究结论可用于满足 USB3.0 协议的弹性缓冲等高速弹性缓冲的场合, 具有一定的工程应用价值。

关键词: 弹性缓冲; 常半满; USB3.0; 异步 FIFO

中图分类号: TP393

文献标识码: A

文章编号: 1000-7180(2012)06-0117-05

Design and Implementation of Elastic Buffer for USB3.0 PHY

ZHU Xiao-ming, WANG Xiao-li, CHENG Zeng

(School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China)

Abstract: Elastic Buffers which are applied in USB3.0 and PCIE physical layer are used to ensure data integrity when bridging two different clock domains. By the research of elastic buffer mechanism and USB3.0 specification, an elastic buffer is designed by using asynchronous FIFO which has innovative function including write pointer mask, breakpoint restored, pointer jump, handshake methods and output control, and the frequency delta is managed perfectly by this elastic buffer. The width of input and output data is 10 bits, and the write clock and read clock can achieve 500 MHz. The research result can be applied to elastic buffer of USB3.0 and other system where high speed elastic buffer is needed, and is valuable to engineering application to some extent.

Key words: elastic buffer; normal half full; USB3.0; asynchronous FIFO

1 引言

通用串行总线 3.0 (Universal Serial Bus 3.0, USB3.0) 是由 Intel、Microsoft、NEC 等公司于 2008 年 11 月公布的标准^[1]。USB 是近些年来应用在 PC 领域的新型接口技术^[2]。相对于 USB2.0, USB3.0 依然属于高速, 串行, 源同步的传输协议。在发送端采用差分对的方式发送串行数据, 在接收端利用时钟和数据恢复电路 (Clock and Data Recovery, CDR) 从串行数据中恢复出串行数据和时钟。由于 USB3.0 的收发两端支持独立的参考时钟源^[3], 所以接收端恢复出的时钟频率和其本地时钟频率存在

差别。为了补偿两个时钟的频率差别, 需要一个弹性缓冲 (Elastic buffer), 将 CDR 恢复的时钟域数据有效地同步到本地时钟域中。

弹性缓冲由 Maurice Karnaugh 在电话网络中传输脉冲编码调制 (Pulse Code Modulation, PCM) 信号中提出来的^[4]。弹性缓冲通过插入和删除某一特殊的符号来实现时钟补偿的目的, 并能够稳定地在两个时钟域间同步数据, 因此, 弹性缓冲在很多技术应用中被采用, 例如 USB、PCIE、以太网等协议中都采用了弹性缓冲技术来同步数据。本文依据 USB3.0 协议, 设计并实现了适用于 USB3.0 物理层的弹性缓冲。

2 弹性缓冲的作用域及其容量

2.1 弹性缓冲的作用域

USB3.0 的数据传输采用全双工 (Full duplex), 物理层有两个差分信号对, 一对为发送差分信号对, 另一对为接收差分信号对。由于弹性缓冲只用于接收端, 如图 1 给出物理层的接收端的框图, 它包括差分接收、时钟数据恢复电路、串并转换、8B10B 解码和解扰电路等。

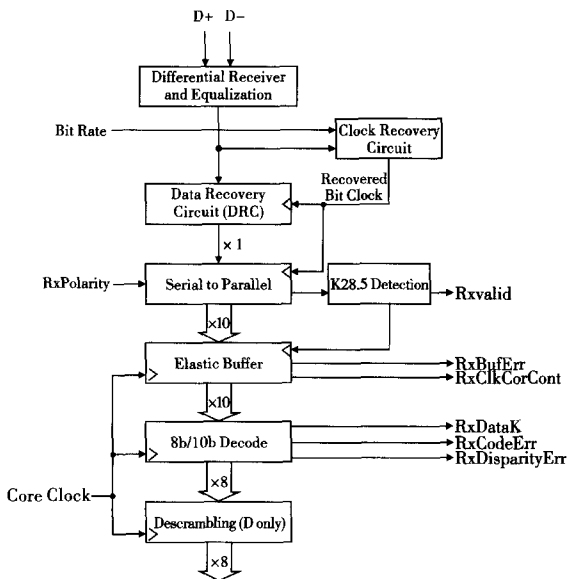


图 1 USB3.0 物理层接收部分结构

如图 1 所示, 差分信号由差分接收端接收, 经过 CDR 恢复出时钟和数据, 然后数据经过串并转换为 10 比特的数据, 通过序列检测器检测出包的起始标志符 K28.5, 当检测到 K28.5 时, 使符号锁定 (symbol lock) 信号有效, 当 symbol lock 有效时, 弹性缓冲开始写入数据。弹性缓冲的写时钟为 CDR 恢复的时钟, 称为接收时钟, 而其读时钟为接收端的系统时钟。通过弹性缓冲将接收时钟域的数据有效的同步到系统时钟域, 并且经过 8B10B 译码和解扰器将数据传递给系统。

2.2 弹性缓冲容量

为了补偿接收时钟和系统时钟的频率差别, 在发送端平均每发送 354 个字节插入一个 SKP 序列对 (SKP ordered set), 一个 SKP ordered set 是由 2 个 SKP (K28.1) 构成, 当接收端弹性缓冲检测到 SKP ordered set 时根据此时弹性缓冲中数据的个数来删除或插入 SKP ordered set。由于 USB3.0 的发送和接收端采用独立的参考时钟, 所以, 在接收端

通过 CDR 恢复出来的接收时钟和本地的系统时钟会有一个时钟偏差。为了匹配两端的频率差别, 需要一个弹性缓冲, 通过插入和删除 SKP 来达到速率匹配。

本文设计的参考时钟精度 -300×10^{-6} 到 $+300 \times 10^{-6}$, 扩频时钟 (Spread Spectrum Clocking) 的时钟变化为 $-5\,000 \times 10^{-6}$ 到 0, 因此, 发送端和接收端的时钟偏差为 $-5\,300 \times 10^{-6}$ 到 300×10^{-6} , 最大的偏差为 $5\,600 \times 10^{-6}$ 。不管参考时钟频率如何, 两者之间的偏差为 $5\,600 \times 0.000\,001 = 0.0056T$ 。在 USB3.0 中最大数据包长度为 1056 字节, 并且协议要求在一个数据包中不能插入 SKP。最差的情况下两个时钟发送数据的最大数据偏差为: $(T \times 1056 \times 10 - T(1 - 0.0056) \times 1056 \times 10) / T = 59$ 比特 = 8 字节, 这里乘 10 是考虑了 8b/10b 编码, 因此弹性缓冲容量最小为 8 字节。

弹性缓冲深度为其可以存放的数据个数, 通过计算可知, 弹性缓冲的深度应该为 8。本文设计采用常半满^[5] (Normal half full) 模式来设计弹性缓冲, 其容量为 16, 在通常情况下弹性缓冲中有 8 个数据, 剩下的 8 个为缓冲空间, 故称常半满。常半满模式首先要向弹性缓冲中写入 8 个符号 (symbol, 在 USB3.0 中一个 8 比特位宽或者一个 10 比特位宽的数据为一个 symbol), 达到半满, 然后读使能有效, 常半满模式只有在 symbol 队列中出现 SKP 对时, 才能添加或者删除 SKP 对。

3 弹性缓冲的原理和结构

3.1 弹性缓冲的原理

弹性缓冲实质上是一个异步 FIFO, 其写时钟为接收时钟, 读时钟为系统时钟, 数据位宽为 10 比特, FIFO 深度为 16, 当 symbol lock 有效时, 在接收时钟域, 向 FIFO 中写入 8 个来自串并转换器的数据。当读写时钟一样快慢时, FIFO 中有 8 个有效数据, 维持在半满状态。

当读时钟快于写时钟, 读出的数据多于写入的数据, 经过一段时间后导致 FIFO 中的数据数量少于 8, 甚至有可能被读空。如图 2 所示, 当输入数据出现 SKP 时, FIFO 中有效数据为 4, 比常态少了 4。此时, 弹性缓冲应该添加 4 个 SKP, 使得 FIFO 维持半满状态, 以此来调节时钟。此时写指针向前跳跃 4 个间隔, 并保存跳跃点的位置, 当读指针读到跳跃区间时, 通过输出控制, 完成 SKP 添加, 也就是说, 并没有将新插入的 SKP 确实写入 FIFO 中, 跳跃区间

内的地址数据为无效的数据,并且添加的 SKP 必须满足运行差异参数(Running Disparity)^[6]的要求,否则会导致 8B10B 译码时出错。

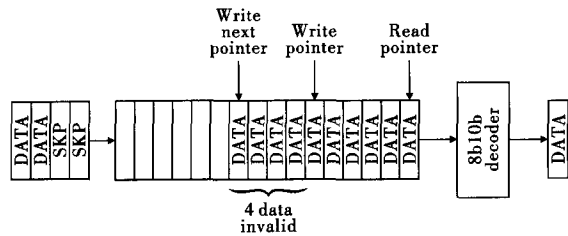


图2 常半满模式中读快于写

当读时钟慢于写时钟,写入的数据多于读出的数据,经过一段时间后将导致FIFO中的数据数量多于8,甚至有可能被写满。如图3所示,在出现SKP窗口时,FIFO中有效数据为10,比常态多2。此时,弹性缓冲应该删除2个SKP,使得FIFO维持半满,以此来调节时钟。此时写指针应该暂停2个时钟周期,完成SKP删除。

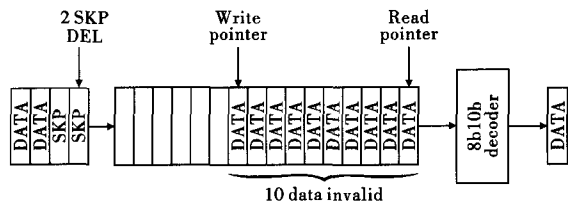


图3 常半满模式中写快于读

在弹性缓冲读写控制的过程中,写先于读。当FIFO中数据达到8个后,读写同时有效。当写结束,即一个包接收完毕,但是读不一定结束,直至读到空,即所有数据已经同步到本地系统时钟域中,表示此次读写任务结束。这种流程控制就保持了系统数据的完整性。

3.2 弹性缓冲的结构设计

本文设计的常半满模式的弹性缓冲,其结构如图4所示,其结构可以分为输入检测模块,二进制和格雷码读写指针产生模块,格雷码读写指针在两个时钟域间同步模块,读写控制模块,判断弹性缓冲中有效数据的阈值监测模块以及输出控制模块。本文采用同步比较来判断弹性缓冲中的有效数据个数,避免了异步比较的毛刺和不稳定,采用了一种新的满空标志产生方法,采用写指针屏蔽,写指针跳跃和读写指针握手的方式实现SKP的删除和插入。

3.2.1 输入检测模块设计

输入检测模块的功能是检测输入数据是否为

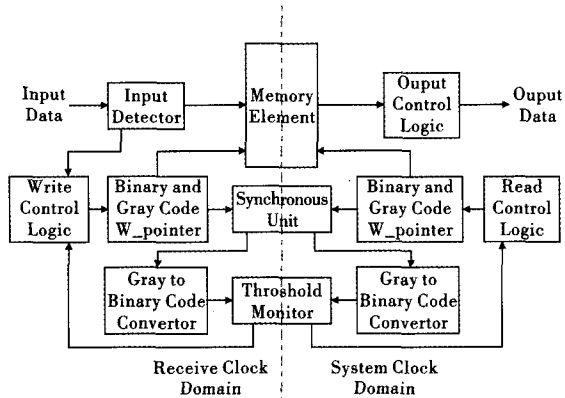


图4 弹性缓冲的结构

SKP,并根据此时弹性缓冲中有效数据个数,选择是否删除输入的SKP或插入新的SKP。为了使新插入的SKP满足8B10B编码要求,还应保存输入SKP的运行差异参数。

3.2.2 读写指针的产生和同步模块设计

在判断弹性缓冲中有效数据个数和产生满空标志时,由于异步FIFO的读写是由异步时钟控制的^[7],不能将两个时钟域的指针直接比较,这样会产生较大的亚稳态。本文设计中利用二进制码和格雷码指针产生模块,产生二进制码和格雷码,其中用二进制码作为存储单元的地址输入。将格雷码的读指针,用写时钟采样两次,即经过两个写时钟控制的寄存器同步到写时钟域中,由于格雷码每次只变化一位,故经过同步后,可将亚稳态的可能性降到最低,然后将写指针和同步到写时钟域的读指针做比较产生FIFO满的标志。同样,将写指针同步到读时钟域中产生FIFO空的标志。本文采用新的满空标志产生方法,只将读写指针地址的次高位到最低位作为存储单元的地址,当读写地址相同时弹性缓冲为空,当读写指针最高位不同而其余所有位相同时弹性缓冲为满^[8]。

3.2.3 阈值监测模块设计

阈值监测模块的功能是用来判断弹性缓冲的有效数据个数,直接用格雷码无法判断缓冲中的数据,因此,需将同步到写时钟域中的格雷码读指针转化为二进制码与二进制的写指针比较来判断弹性缓冲中的有效数据。由于二进制的读指针由同步到写时钟域中的格雷码的读指针转换而来,此时不会产生亚稳态的问题。当弹性缓冲中的数据少于等于6时,添加SKP,add_skp信号有效,当弹性缓冲中的数据大于等于10时删除SKP,dele_skp信号有效。

3.2.4 读写指针控制模块和输出控制模块设计

当 `dele_skp` 信号有效时,并且输入检测模块检测到输入数据为 SKP 时,写指针控制模块使写使能无效,暂停写指针,或称指针屏蔽(pointer mask),则完成 SKP 的删除。

当 `add_skp` 信号有效时,并且输入检测模块检测到输入的数据为 SKP 时,根据弹性缓冲的数据个数,插入新的 SKP. SKP 添加通过断点保存,写指针跳跃与握手和输出控制模块来实现。

在 SKP 窗口出现和添加阈值标志有效时,写指针此时计算 FIFO 中的有效数据个数,根据 FIFO 中有效数据的个数与 8 的差距,来决定 `write_addr_b` 所指向的下一个指针点,在写指针模块的控制下完成写指针的跳跃,即为写指针跳跃(write pointer jump). 并且在写时钟域把当前的写指针和下一个指针点保存到 `current_write_addr_b` 和 `next_write_addr_b` 中. 同时在写时钟域给出 `inform_rp` 信号,由于读指针总是落后于写指针,当读指针读到断点的起始地址时, `inser_skp_en` 信号有效,并反馈 `ack` 信号,当写时钟域检测到 `ack` 信号有效时撤销 `inform_rp` 信号,输出控制模块检测到 `inser_skp_en` 信号有效后,根据此时输入的 SKP 的 RD 的值插入新的 SKP,输出控制模块保证插入所有的 SKP 均满足 8B10B 编码的要求;当读指针读到断点的结束地址时, `inser_skp_en` 信号变为无效,同时撤销 `ack` 信号,完成 SKP 的插入,使 FIFO 维持半满状态。

读指针控制模块的功能是控制读使能信号,当 `symbol lock` 信号有效时,此时写时钟开始向 FIFO

中写入数据,读使能需要等到 FIFO 写入 8 个数据后才能有效,当 `symbol lock` 无效后,即一个数据包已经接收完毕,写使能无效,但此时读使能不一定立刻无效,直到 FIFO 被读空为止,所有的数据都已经同步到本地,读使能才无效。

4 弹性缓冲设计的仿真

本文采用 Verilog HDL 设计弹性缓冲,用 NC-Verilog 做功能仿真验证。

本设计采用台积电 TSMC 65 nm 的工艺库用 Design Compiler 进行综合,并且给予较为保守的时序约束,读写的时钟频率均可达到 500 MHz. 用 Encounter 进行后端的布局和布线,得到精确的时序信息,然后采用 Prime Time 进行静态时序分析,最终采用 NC-Verilog 做时序仿真,其仿真结果如图 5 和图 6 所示。

仿真结果表明,本文设计的弹性缓冲可以正确地写入和读出数据,能够正确删除和插入 SKP,插入的 SKP 满足 8B10B 编码的要求. 能够正确地产生弹性缓冲满空标志,并可在两个时钟域间同步数据. 设计完全符合 USB3.0 协议的要求,验证了本文设计的可靠性。

5 结束语

本文系统分析了弹性缓冲的作用原理和结构模块. 根据 USB3.0 协议要求,采用异步时钟 FIFO 来设计弹性缓冲,弹性缓冲采用并行 10 位数据,读写可以达到 500 MHz 的频率. 经过仿真验证,设计完

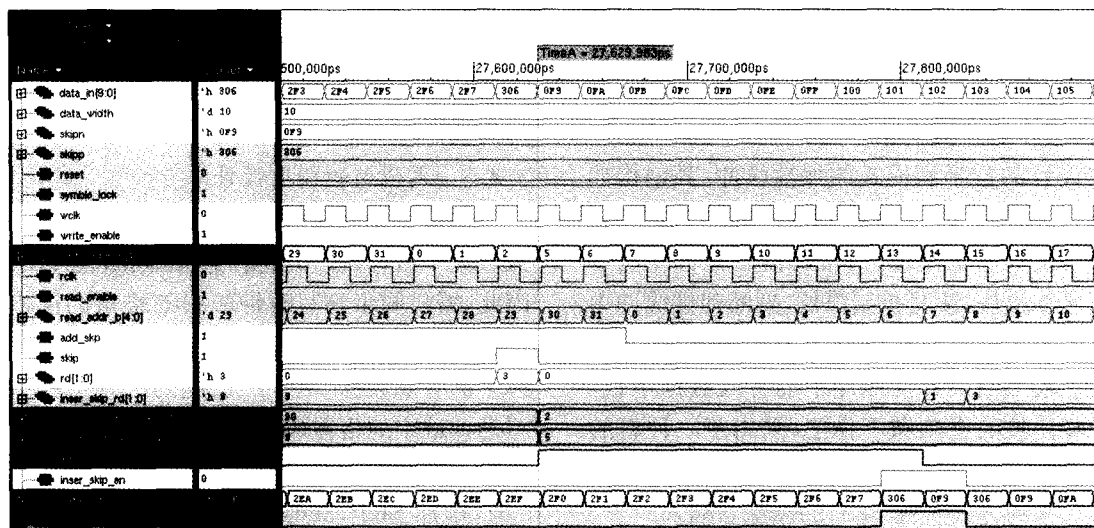


图 5 SKP 对的添加

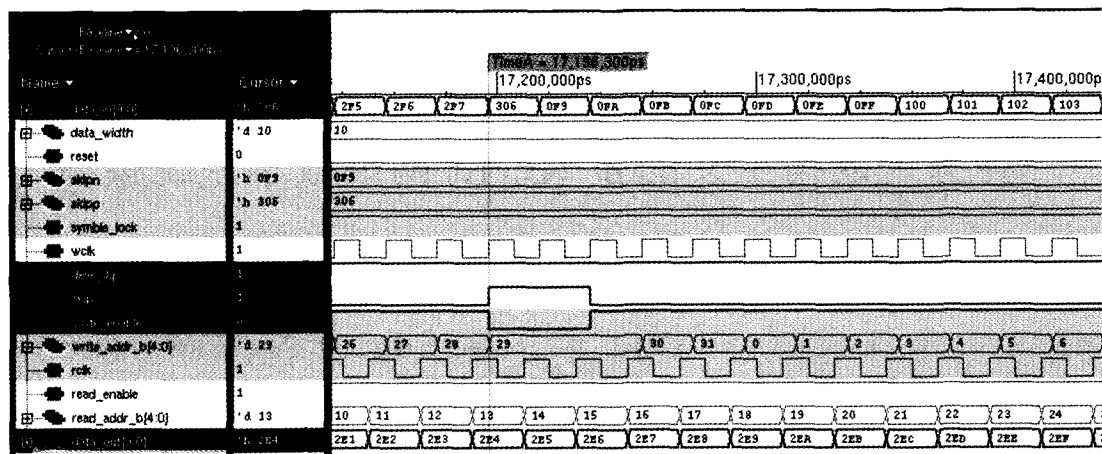


图6 SKP对的删除

全满足 USB3.0 的协议要求,实现了时钟频率补偿的技术要求.将本文设计的结构稍加修改也可作为 PCIE 物理层的时钟频率补偿,并可满足其它需要高速弹性缓冲的技术应用,本文研究结论具有一定工程应用价值.

参考文献:

- [1] Wikipedia, USB[EB/OL]. [2011-09-28]. <http://en.wikipedia.org>.
- [2] 彭琰,曾云.基于 HID 类 USB 外设功能控制器的 ASIC 设计[J].微电子学与计算机,2009(4):15-18.
- [3] Universal Serial Bus 3.0 Specification[S]. 美国:USB 组织,2008.
- [4] Joe Winkles. Elastic Buffer Implementations in PCI Express Devices [M]. Mindshare Inc, 2003.
- [5] PHY Interface for PCI Express and USB 3.0 Version3.0[S]. 美国:英特尔,2009.
- [6] 张民,许进,黄学田.光以太网[M].北京:北京邮电大学出版社,2003.
- [7] 刘洪波,龙娟,郝晓莉,等.异步 FIFO 状态判断的研究与设计[J].微电子学与计算机,2007(24):81-84.
- [8] Cummings C E. Simulation and synthesis techniques for asynchronous FIFO design [R]. SNUG, 2002.

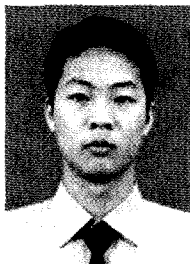
作者简介:



朱小明 男,(1985—),硕士研究生.研究方向为计算机高速串行接口技术、视频编解码与通信.



王小力 男,(1956—),教授,博士生导师.研究方向为高性能 VLSI 与优化设计、集成化芯片系统设计与基础研究、光纤通讯与光波分复用技术.



程曾 男,(1987—),硕士研究生.研究方向为 SOC 低功耗设计与验证.