

基于串行 RapidIO 的 Buffer 层设计

任雪倩^{1,2}, 李金城², 赵建中¹, 张 锋¹

(1 中国科学院 微电子研究所 智能感知研发中心, 北京 100029;

2 北京交通大学 电子信息工程学院, 北京 100044)

摘 要: RapidIO 是一种新兴的高性能互联总线, 主要应用于嵌入式系统. 作为串行 RapidIO 结构的重要组成部分, Buffer 层负责保证数据包的可靠传输, 并且在链路级流量控制中起着重要作用. 对此提出了一种改进的 buffer 层结构, 同时支持发送端流量控制和接收端流量控制两种模式. 仿真结果表明, 该 buffer 层有效地提高了 buffer 的空间利用率和传输效率. 该设计在高性能嵌入式系统中具有一定的工程应用价值.

关键词: buffer; RapidIO; 流量控制

中图分类号: TN4

文献标识码: A

文章编号: 1000—7180(2016)09—0047—04

A Buffer Layer Design in Serial RapidIO

REN Xue-qian^{1,2}, LI Jin-cheng², ZHAO Jian-zhong¹, ZHANG Feng¹

(1 Smart Sensing R&D Centre, Institute of Microelectronics of Chinese Academy of Sciences, Beijing 100029, China;

2 School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China)

Abstract: RapidIO is an emerging high-performance interconnect technology for embedded system interconnection. As a very significant part of serial RapidIO, buffer layer helps to guarantee reliable packet delivery and plays an important role in link level flow control. In this paper, the design of an improved buffer layer in serial RapidIO endpoint is described in detail. Both transmitter-controlled flow control and receiver-controlled flow control are supported in this design. Simulation results show that the proposed buffer layer has better space utilization and higher transmission efficiency, compared with the referenced designs. It is valuable for high performance embedded system applications to some extent.

Key words: buffer; RapidIO; flow control

1 引言

近年来, PCI Express、以太网、RapidIO 等高性能总线标准相继出现, 逐渐填补了这个差距. RapidIO 是一种高性能、点到点的互联技术, 可以被用于芯片级、板级互联^[1]. 相比其他新兴的互联标准, RapidIO 具有高可靠性、低延迟、高带宽、低引脚数和完善的流量控制机制等特点, 更加适合应用于嵌入式系统当中^[2].

RapidIO 的层次结构从上至下分为以下三层: 逻辑层、传输层和物理层. 逻辑层定义了互联的整体

规范和包的格式, 并且包含终端器件处理事物所必要的信息. 传输层协议定义了标准的传输机制, 定义了 RapidIO 数据传输的地址空间以及数据包在终端期间中传输的路由信息. 物理层位于 RapidIO 分级结构的最底层, 描述了全双工的器件接口, 包括数据包和控制符的传输, 流量控制, 错误管理和其他器件间的电气特性^[3].

串行 RapidIO 的结构图如图 1 所示, 可以看出, buffer 层是传输层和物理层之间的桥梁. Buffer 层负责发送和接受数据包的缓存, 对数据包的可靠传输和链路级流量控制至关重要^[4]. 本文提出了一

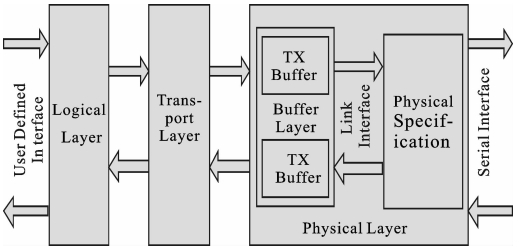


图 1 串行 RapidIO 层次结构

种基于 RapidIO 系统的改进的 buffer 层结构,该 buffer 具有更大的空间利用率和更高的传输效率。

2 相关工作

RapidIO 是一种新兴的基于包交换的高带宽互联技术。在串行 RapidIO 系统中主要有两种数据:数据包和控制符。在控制符的协作下,buffer 层通过重传机制来实现错误恢复和链路级流量控制。Buffer 层包含 TX buffer 和 RX buffer 两部分。参考文献^[5]提出了一种四列式的存储结构,用四个空间固定的存储队列去分别存储四种优先级的数据包。然而,在实时网络密集应用中^[6],buffer 层发送和接收的主要是优先级为 0 和 1 的数据包,在这种情况下,四列式结构的高优先级数据包存储队列大部分时候都被浪费了。因此,buffer 层最好可以不为每个优先级的数据包预设存储空间。根据协议,物理层将数据包发送至传输层的顺序应该与 RapidIO 端口收到数据包的顺序一致^[3],因此 RX buffer 不需要存储接收到数据包的优先级信息。文献^[5,7]中提出了两种不同的 buffer 结构,然而处理优先级信息方式却使得这两种结构较为复杂。为了解决这个问题,本文提出了一种新型的 buffer 层结构,以更简单的结构实现更好的空间利用率和更高的传输效率,该 buffer 同时支持接收端流量控制和发送端流量控制两种流量控制模式。

3 结构设计

作为传输层和物理层之间的连接,buffer 层对数据包的可靠传输以及链路级流量控制起着重要作用。Buffer 层分为 TX buffer 和 RX buffer 两部分。其中,TX buffer 负责管理当前队列当中的数据包,并将其发送到链路当中;RX buffer 负责缓存从链路中接收到的数据包,然后将其发送给传输层^[4]。

(1)TX Buffer

TX buffer 主要由两部分组成,如图 2 所示,左边部分主要负责处理数据包,右边部分是主要的控

制单元。TX buffer 用 32 个 FIFO 去存储数据包,每个 FIFO 可以存储一个最大长度的数据包,最大数据负载长度为 256 字节。空闲 FIFO 管理模块对空闲 FIFO 的标号进行管理,当传输层向 TX buffer 发送事物包时,空闲 FIFO 管理模块会选择空闲的 FIFO 将数据包进行写入。所有的 FIFO 都没有优先级属性,由优先级管理模块去统一管理这些 FIFO 中数据包的优先级信息,必要时优先级管理模块会提升部分数据包的优先级信息。AckID 管理模块负责以循环增序方式为每个数据包添加 AckID 值。当重传事件发生时,仲裁模块根据数据包的优先级和事务类型信息,将队列中的数据包进行重新排序发送,ackID 管理模块也需要在特定的情况下去重新调整数据包的 ackID 值。如果要求重传的数据包是响应包,则这个数据包需要被重新发送,并且这个包的优先级要加 1。TX buffer 必须保留每一个发出去的数据包,直到这个数据包成功被链路伙伴接收以后,TX buffer 才可以将其丢弃。由于数据包的 ackID 值表明了数据包的传输顺序和应答顺序,因此,收到 PA(packet-accepted)控制符则表示 PA 控制符对应的数据包和在这个数据包之前发送的数据包都已经被链路伙伴成功的接收。

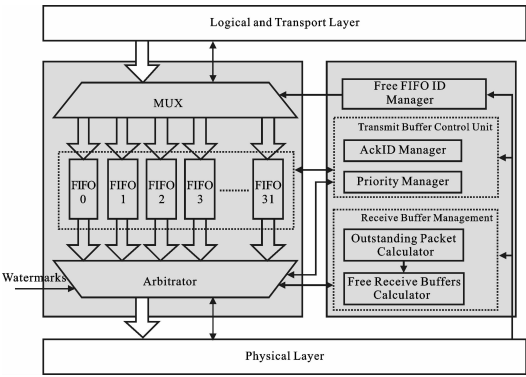


图 2 TX buffer 结构图

接收端缓存管理区域工作在发送端流量控制模式下。未确认数据包计算模块负责计算已经发往链路但还没有收到链路伙伴确认的数据包的数量。工作在发送端流量控制模式下的端口,在正常传输模式下,需要至少在特定的周期内向其链路伙伴发一个包含自身可用接收 buffer 数量信息的控制符。通过这个接收到的数值和计算出的未确认数据包的数量,接收端可用缓存计算模块计算出链路伙伴的可用接收 buffer 数量。然后,仲裁模块可以根据这个数值,待发送队列中数据包的类型和优先级,以及水标值,来选择当前应该发送哪个数据包。

(2)RX Buffer

RX buffer 负责接收物理层数据包并传送给传输层. 若 RX buffer 认为当前优先级的数据包可被接收,则该数据包可缓存到 RX buffer 中,RX buffer 也不必再记录该数据包的优先级信息. RX buffer 的结构可被分成两部分,如图 3 所示. 图中,上面的部分负责缓存数据包并将其传送给传输层,下面的部分负责判断当前物理层数据包能否被缓存到 RX buffer 中来. 数据包存储模块最多可以容纳 32 个最大长度的数据包,指针模块则负责读和写的操作. 可用缓存空间计算模块负责计算当前可以用来接收数据包的可用缓存空间的数量,为了方便起见,下文用 FB(free buffers)去指代这个数值. 数据包接收模块主要负责依据数据包的优先级、流量控制模式和 FB,来判断当前数据包是否可以被接收.

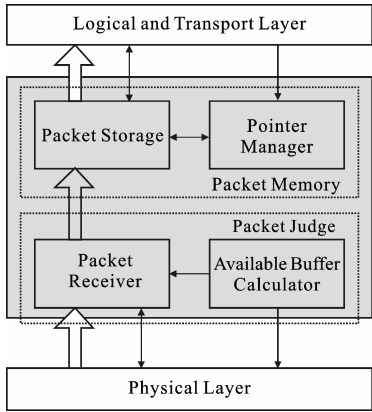


图 3 RX buffer 结构图

若采用的是发送端流量控制模式,则只要 FB 大于 1,数据包就可以被 RX buffer 接收;若采用的是接收端流量控制,则 RX buffer 能否接收这个数据包完全取决于当前是否有足够的缓存空间去存储这个优先级的数据包. 为了避免死锁,最低优先级的数据包最容易被拒收,特别是当 RX buffer 空间逐渐被填满时.

采用空间共享机制可提升 buffer 空间利用率. 图 4 是共享机制下数据包存储的示例. 灰色阴影部分表示该部分空间已经被数据包占用,非阴影部分表示空间可用. 图 4(a)中,上一个接收的数据包记为 a,优先级为 m. 此时,优先级为 n 的数据包 b 到达,若 FB 满足要求,则数据包 b 可以被写入到数据包存储模块,而优先级 m 和 n 的关系并不影响数据包的接收,如图 4(b)所示. 所有接收的数据包都被连续的存到数据包存数模块,这样就实现了空间的共享,这种共享的优势在数据包很短时更加明显.

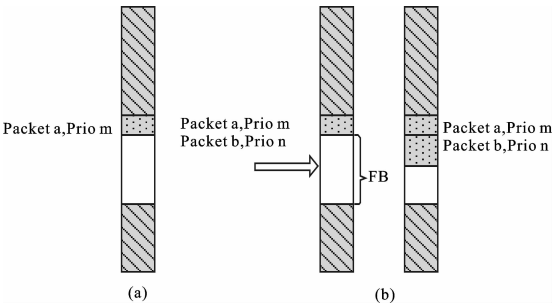


图 4 RX buffer 的共享机制

4 性能分析

本文通过直接连接两个 RapidIO 终端的方式来验证该 buffer 层的性能,其中一个终端作为发送方,另一个作为接收方,并采用 buffer 空间利用率和传输效率作为衡量性能的两个指标. 以下所有的结论都是基于该验证环境下的正常传输模式.

(1)Buffer 空间利用率

TX buffer 空间利用率用写入 TX buffer 的字节和 TX buffer 总空间的比值来表示,验证过程中 TX buffer 存储的数据包不允许被移除. 为了分析 TX buffer 的空间利用率,本文用 32 个最低优先级的确定数据负载长度的 NWRITE 事物包作为要发送的内容. 该 TX buffer 结构至多可以存储 32 个任意优先级的数据包,而参考文献[5]中的四列式结构和参考文献[7]中的结构只能用固定的空间去存储特定优先级的数据包. 如图 5 所示,本文的 TX buffer 空间利用率比四列式结构高出 65%.

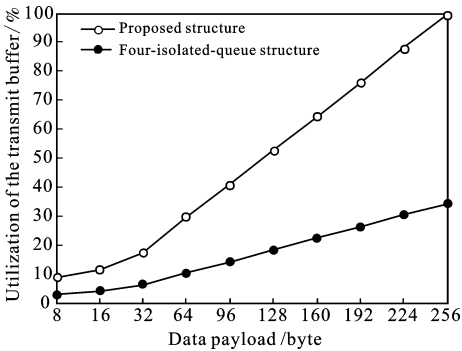


图 5 TX buffer 空间利用率

RX buffer 空间利用率的计算方式和 TX buffer 一样. 为了阐述空间共享机制的好处,将本文 RX buffer 结构与没有应用共享机制的相同结构进行对比. 没有应用共享机制的结构至多可以存储 32 个数据包,不论数据包的长短. 而本文的 RX buffer 可以通过空间共享的方式实现更好的空间利用率,特别

是在数据包很短的情况下. 图 6 和图 7 分别是本文 RX buffer 结构的空间利用率和没有共享机制的相同结构的空间利用率.

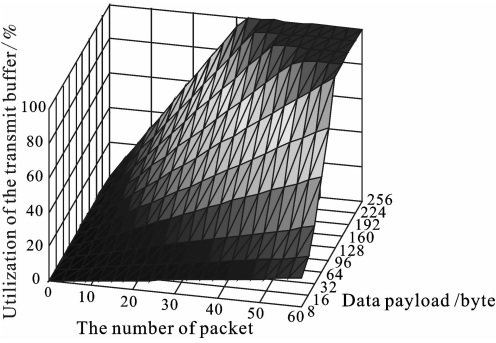


图 6 本文 RX buffer 空间利用率

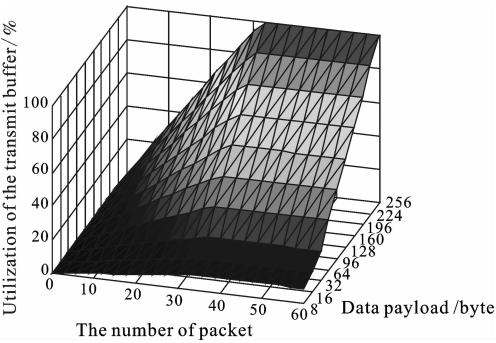


图 7 无共享机制结构 RX buffer 的空间利用率

(2)Buffer 传输效率

本文将测试环境设定成连续产生和发送具有最长数据负载和最低优先级的 NWRITE 事物包,直到 100 个数据包已经成功的被链路伙伴的传输层接收为止. 链路伙伴的传输层延迟固定周期才开始接收数据. Buffer 层的传输效率用链路伙伴传输层收到的数据字节数与其自身传输层发送的数据字节数的比值来表示. 从图 8 可以看出,相比文献[7]中的结构,本文提出的 buffer 结构具有更高的传输效率.

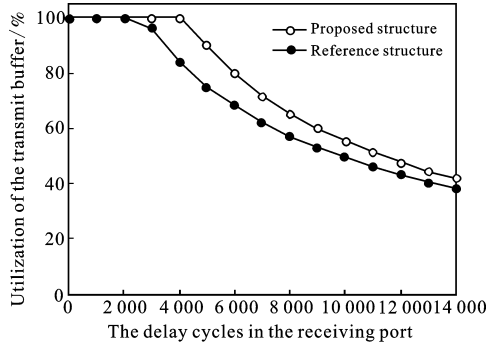


图 8 Buffer 传输效率

5 结束语

本文提出了一种基于串行 RapidIO 系统的改进的 buffer 层结构,其中,TX buffer 将优先级信息和数据包分开来存储和管理,RX buffer 将进来的数据包进行统一存储. 通过改善数据包的存储结构 and 应用更加简单的共享机制,该结构实现了更好的 buffer 空间利用率,更好地保障了数据包的可靠传输,并且增加了传输效率. 分析结果表明,该结构适用于串行 RapidIO 系统,可以被应用于高性能嵌入式系统中.

参考文献:

[1] Syed Ijlal Ali Shah, Shashank Khanvilkar, Ashfaq Khokhar, RapidIO traffic management and flow arbitration protocol [J]. IEEE Communications Magazine, 2006,44(7) :45-51.

[2] Sam Fuller, RapidIO: The embedded system interconnect [M]. UK: John Wiley & Sons, 2005.

[3] The RapidIO trade association, RapidIO interconnect specification part6[S]. Rev 2. 1. 2009.

[4] Xilinx, LogiCORE IP serial rapidIO Gen2 endpoint v2. 0. product guide for vivado Design Suite[R]. 2013.

[5] X B Zhao, S Jia, Y Wang, et al. Buffer design based on flow control in rapidIO[C]//2010 Asia Pacific conference on Postgraduate Research in Microelectronics and Electronics, Shanghai; IEEE, 2010;150-153.

[6] David Bueno, Adam Leko, Chris Conger, et al. Simulative analysis of the rapidIO embedded interconnect architecture for real-time, network-intensive applications [C] // Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks, Tampa, FL, USA; IEEE, 2004;710-727.

[7] Wu Fengfeng, Jia Song, Wang Yuan. An effective buffer space management in serial rapidIO endpoint[C]//2011 Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics, Macaa; IEEE, 2011;135-138.

作者简介:

任雪倩 女,(1992-),硕士. 研究方向为高速接口与 VLSI 体系结构设计. E-mail:renxueqian@ime. ac. cn.

李金城 男,(1967-),博士,副教授. 研究方向为卫星导航原理与芯片设计.

张 锋 男,(1977-),博士,副研究员. 研究方向为高速低功耗接口电路和高速低功耗存储器的设计.

赵建中 男,(1983-),博士,助理研究员. 研究方向为高速串行接口.