

网络交换中的缓存管理研究与实现

李 平¹, 杨婷婷¹, 韩俊刚²

(1 西安邮电学院 计算机学院, 陕西 西安 710061; 2 西安邮电学院 电子工程学院, 陕西 西安 710061)

摘 要: 缓存管理的效率严重制约着网络交换芯片的存储转发速度, 如何选取合适的缓存管理方法成为关键. 本文提出了链式存储和环形链式存储两种管理算法, 二者皆采用链表的形式实现. 通过在 ALTERA 系列 FPGA 开发板上的测试分析, 证明了链式存储算法节省资源, 而环形链式存储算法可以带来更高的工作速率, 两种算法都可广泛应用于网络交换芯片的设计当中.

关键词: 存储转发; 链式存储; 环形链式存储; FPGA

中图分类号: TP391

文献标识码: A

文章编号: 1000-7180(2012)01-0131-04

The Research and Realization of Buffer Management in Network Switching

LI Ping¹, YANG Ting-ting¹, HAN Jun-gang²

(1 School of Computer Science, Xi'an University of Posts and Telecommunications, Xi'an 710061, China)

2 School of Electronic Engineering, Xi'an University of Posts and Telecommunications, Xi'an 710061, China)

Abstract: The efficiency of buffer management constrains the store-and-forward speed of network switch chip seriously, how to select the suitable cache management methods become the key. This paper puts forward the Linked Storage and Circular Linked Storage management algorithm. They all realized using the linked list. The test and analysis on ALTERA's FPGA development board demonstrate that Linked Storage algorithm saves resource and Circular Linked Storage algorithm brings higher work rate, the two algorithms all can be widely applied to the design of network switch chip.

Key words: store-and-forward; linked storage; circular linked storage; FPGA

1 引言

互连网的发展对交换机的性能, 如交换的容量、速率和服务质量等提出了更高的要求^[1-2]. 研究高速、大容量、可扩展的新型交换网络结构是网络发展的重要课题之一. 共享缓存结构是一种较常用的交换结构, 其特点是缓存资源可以得到最大化利用, 并能获得很好的时延性能和很高的吞吐率^[3].

存储转发^[4]的基本结构, 可以划分为基于输入端口排队(IQ)、基于输出端口排队(OQ)以及联合输入输出排队(CIOQ)^[5]. 基于输入端口的排队是按照不同的输入端口进行入队操作, 同一个输入端口

排成一个队列; 基于输出端口的排队是按照不同的输出端口进行入队操作, 同一个输出端口排成一个队列; 联合输入输出排队则分别在输入和输出端口同时都具有缓存队列^[5]. 在基于共享缓存的交换芯片中, 输入端口和输出端口都共享一个缓存模块, 所有需要经过交换机的数据都要经过缓存存储转发, 其核心是将存储划分为多个链表, 形成基于不同输出端口的队列. 共享缓存所交换的仅仅是包的指针, 显著提高了交换容量, 因此, 共享缓存在网络交换芯片中得到了广泛应用. 但是缓存分配释放算法的优劣在很大程度上制约着缓存的使用效率, 因此, 必须有一种合理的缓存分配释放算法来管理有限的缓存

收稿日期: 2011-05-06; 修回日期: 2011-06-22

基金项目: 陕西省 13115 科技创新工程重大科技专项(2009ZDKG-43); 陕西省教育厅科研计划项目(2010JK840)

空间。

本文研究的交换芯片,其共享缓存单元为一块宽度为 143bit、深度为 24Kbit 的静态 RAM。通过在 ALTERA 系列 FPGA 开发板上的设计验证,系统对比了两种缓存分配释放算法,以确定在相应环境下的最佳管理方法。

2 共享缓存排队结构

共享缓存排队结构的设计思想如下^[6]:在存储的内部,单元被有序地存入不同的逻辑队列,每一个队列对应一条输出线路,去往同一个输出端口的单元被链接到同一个逻辑队列中。另一方面,通过连续地检索每一个输出队列的头片来确定单元的输出生流;输出流是时分复用的,要传送的单元被送往输出线路。每一个队列有两个指针来控制,一个头指针和一个尾指针。头指针是逻辑队列的第一个单元,尾指针是逻辑队列的最后一个单元或者是用来存储下一个到达的单元的空位置。

3 链式存储(LS)算法

为了实现对数据缓存地址单元的管理,链式存储(Linked Storage)算法需要使用 4 个存储器和 2 个寄存器,其定义见表 1。要实现地址单元的合理分配和释放,必须严格控制 4 个存储器的读写时序。

表 1 链式存储算法所使用的存储器和寄存器

名称	深度/宽度定义	用途	初始数据
存储器 1	143/24K	数据缓存	0
存储器 2	15/24K	空地址链表	1~24K-1
存储器 3	15/64	头地址	0
存储器 4	15/64	预分配地址	0~63
头寄存器	15bit	空链表头	64
尾寄存器	15bit	空链表尾	24K-1

存储器 ram1 共有 24K 个单元,即 24K 个地址。初始化的时候把 ram1 的地址单元 0~63 预分配给 64 个端口,这些预分配地址缓存在 ram4 中。把 ram1 的地址单元 1~24k-1 依次存入 ram2 的地址 0~24K-2 中,这时 ram2 的地址单元 64 为空地址链表头,ram2 的地址单元 24k-1 为空地址链表尾。把空地址链表的头存在寄存器 head 中,空地址链表的尾存在寄存器 tail 中。当接收到单元使用请求时,如果当前数据块是包头或者包中,把当前空地址链表头与数据块拼接作为 ram1 的写数据,把 ram4 中

预分配的地址作为该数据块的写地址,然后把空地址链表头作为新的预分配地址存入 ram4 中。同时,当前的空地址链表头将作为 ram2 的读地址去读取空地址链表,下一拍读出的值将作为新的空地址链表头。如果当前数据块是包尾或者短包不做预分配地址的拼接。在分配地址的过程中要把每个端口的包头地址缓存到 ram3 中,等相应端口的包尾到来时,将头地址读出发送给下级发送模块。当接收到释放请求时把当前释放数据块的地址写到空地址链表尾,并且将该地址更新为新的空地址链表尾。

地址的分配和释放主要在于对空地址链表缓存 ram2 的控制。图 1 模拟了一组数据存取过程,从中可以看到空地址链表缓存 ram2 的变化。依次向缓存中存入 6 个数据块:包 1 头,包 1 中,包 2 头,短包,包 2 尾,包 1 尾。则读出的顺序应该是短包,包 2,包 1。其中包 1 来自端口 0,包 2 来自端口 1,短包来自端口 3。图(a)是初始化之后空地址链表缓存的状态,图(b)地址分配的示意图,图(c)是释放短包时地址回收的示意图。

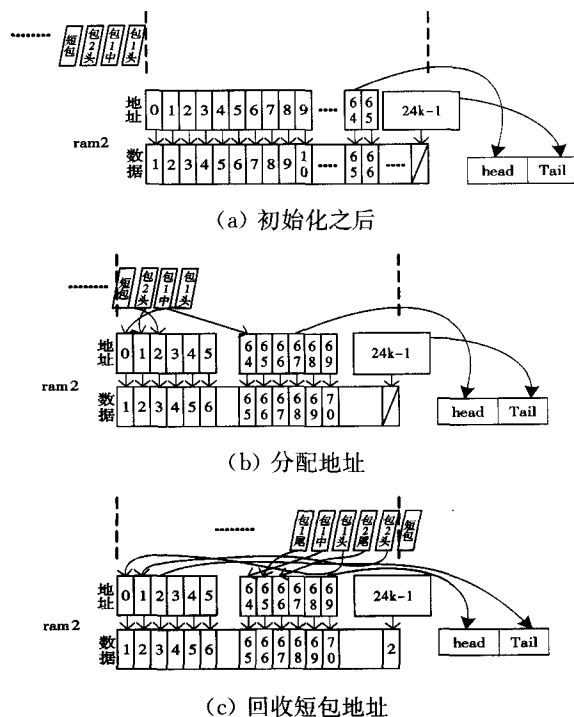


图 1 空地址链表缓存变换示意图

根据 LS 算法的设计思想,构建缓存管理实现框图如图 2 所示。包含预分配处理、数据缓存处理、头地址缓存和空地址链表 4 个模块。

预分配处理模块:检测地址单元申请使能,以源端口号为读地址,对 ram4(预分配地址缓存)进行读操作,在下一个时钟节拍以源端口为写地址将寄存

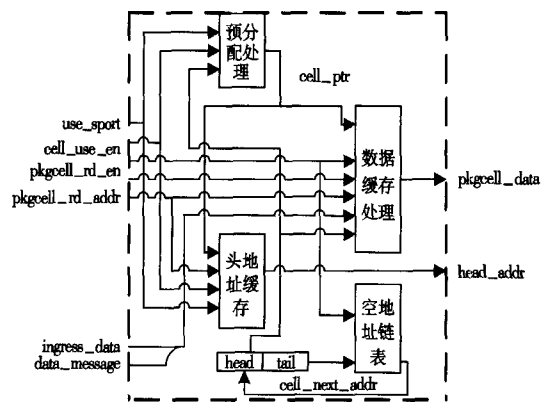


图 2 基于 LS 算法的缓存管理模块实现框图

器 head 中的值写入 ram4 中。

数据缓存处理模块:为数据块拼接预分配地址并把结果写入数据缓存。当检测到写地址申请时对当前数据块的状态进行判断,如果是包头或者包中则把 head 与数据块拼接,然后以 cell_ptr 为 ram1 的写地址将拼接后的数据写入 ram1;如果是包尾或者短包则直接以 cell_ptr 为 ram1 的写地址将数据块写入 ram1;当检测到读使能的时候则以 pkgcell_rd_addr 为 ram1 的读地址对 ram1 进行读操作,并将读出的数据送给下级模块。

头地址缓存模块:每分配一个地址就要检测当前数据块的状态,如果是包头则以源端口为写地址把头地址写入 ram3 中;如果是包尾则以源端口为读地址从 ram3 中读出头地址送给下级模块;如果是短包则直接把短包的写地址送给下级模块。

空地址链表模块:每分配一个地址,以当前空地址链表头作为 ram2 的读地址,从 ram2 中读出一个数据作为新的空地址链表头;当检测到下级模块送来的释放使能时,则把当前释放数据块的地址写入到空地址链表尾指向的地址,并且将该地址更新为新的空地址链表尾。

4 环形链式存储(CLS)算法

环形链式存储(Circular Linked Storage)算法需要使用 3 个存储器和 1 个 FIFO,其定义见表 2。

表 2 环形链式存储算法所使用的存储器和 FIFO

名称	深度/宽度定义	用途	初始数据
存储器 1	143/24K	数据缓存	0
存储器 2	15/64	队列头地址	0~63
存储器 3	15/64	队列尾地址	0~63
FIFO	15/24K	空地址链表	640~24K-1

与链式存储(LS)算法一致,表 2 中的数据存储器 ram1 的深度为 24K,宽度为 143bit。环形链式存储(CLS)算法将 ram1 的可用地址使用一个 FIFO 来存储。FIFO 是一个先入先出队列,它具有一个头指针和一个尾指针。当需要向 ram1 中写入数据时,将可用地址从头指针指向的地址单元取出;当需要从 ram1 中释放数据时,将释放的地址写入尾指针指向的地址单元。在任何时刻,ram1 中的可用地址都将连接在 FIFO 中,形成一个环形的链表,相应的读写过程如下:

(1) 读数据过程

读数据的过程模拟如图 3 所示。

- Step1:从 ram2 中读出头地址信息;
- Step2:根据头地址信息从 ram1 中读出数据及下一地址信息;
- Step3:将下一地址信息更新为队列新的头地址。

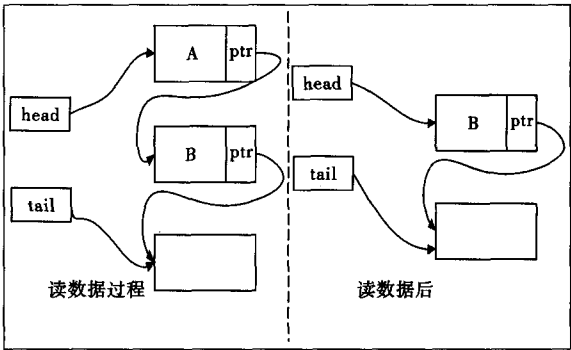


图 3 读数据过程

(2) 写数据过程

写数据的过程模拟如图 4 所示。

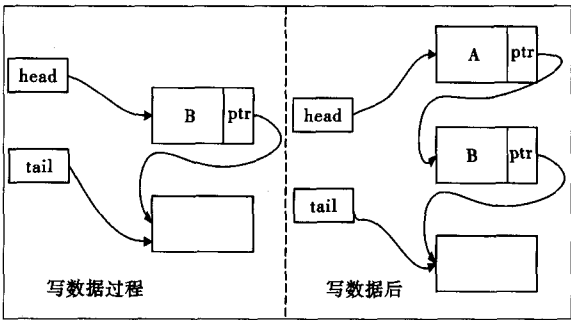


图 4 写数据过程

- Step1:从 ram3 中读出尾地址信息;
- Step2:根据尾地址信息将分配好的下一地址和数据写入到相应的队尾;
- Step3:将下一地址信息更新为队列新的尾

地址.

根据 CLS 算法的设计思想,构建缓存管理实现框图如图 5 所示.包含写入控制、数据缓存处理、读出控制和可用地址链表 4 个模块.

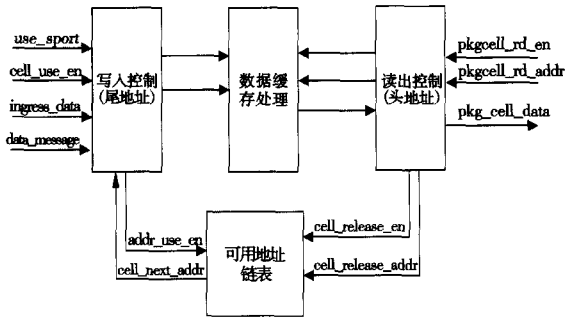


图 5 基于 CLS 算法的缓存管理模块实现框图

写入控制模块:检测地址单元申请使能,若为高,则以端口号为读地址,从 ram3 中读出相应的队列尾作为数据缓存的写地址,将当前数据与可用地址链表模块分配出的下一跳地址拼接作为数据缓存的写数据,同时产生相应的写使能信号.

数据缓存处理模块:接收写入控制模块送来的写信号将数据写入 ram1;接收读出控制模块送来的读信号将数据从 ram1 中读出.

读出控制模块:检测下级模块送来的读信号和释放信号.若读信号为高,则以端口号为读地址从 ram2 中读出相应的队列头作为数据缓存的读地址,同时产生相应的读使能信号;若释放使能为高,则将当前的读地址和释放使能送往可用地址链表模块.

可用地址链表模块:预分配可用地址送往写入控制模块,若有地址单元申请使能,产生相应的下一跳地址(可用地址 FIFO 首);检测读出控制送来的释放使能,若为高,将释放地址写入可用地址 FIFO 尾.

5 算法分析

两种算法均采用 Verilog 的 RTL 级代码实现,并使用 ALTERA 公司的 EP4SGX230KF40C2ES 系列芯片进行 FPGA 开发板上的测试.本文从 4 个方面对两种算法的性能进行了比对,这 4 个比对项目分别为 ALUT(高级查找表)数目、逻辑寄存器数目、存储单元数目以及工作速率.其中工作速率在很大程度上影响着网络交换芯片的交换速率,其余 3 个项目则反映了资源的占有情况.在测试中,用于工作速率分析的模型为 ALTERA 公司提供的 Slow

950mV 85C Model.

表 3 给出了最终的测试结果.从中可以看出,链式存储算法使用了较少的逻辑资源,工作速率为 177MHz;环形链式存储算法则使用了相对较多的逻辑资源,换来了 256MHz 的高速率.

表 3 算法测试结果

	ALUT (个)	寄存器 (bit)	存储单元 (bit)	速率 (MHz)
LS	561	151	3 884 928	177.62
CLS	652	203	3 884 928	256.81

6 结束语

本文针对网络交换芯片中缓存的分配与释放问题,通过分析 ALTERA 系列 FPGA 开发板上得出的测试结果,系统对比了链式存储和环形链式存储两种管理方法,最终证明链式存储算法节省资源,而环形链式存储算法具有较高工作速率.设计人员可依据不同环境选取不同的实现算法.

参考文献:

- [1] Daniel A Menasc a, Honglei Ruana, Hassan Gomaa. QoS management in service-oriented architectures[J]. Performance Evaluation, 2007, 64(7): 646-663.
- [2] N M Mosharaf Kabir Chowdhury. Adaptive analysis of high-speed router performance in packet-switched networks[J]. Literature Review, 2007: 1-10.
- [3] 刘 宇,王玉艳.以太网交换控制芯片的缓存结构[J]. 计算机工程, 2010, 36(10): 248-250.
- [4] 安学军,祝明发,高文学,等. 虫洞路由交换及其缓冲区设计[J]. 微电子学与计算机, 2002(9): 15-18.
- [5] 吕丹阳,解永良.高性能交换结构中的输入排队调度算法综述[J]. 微电子学与计算机, 2004, 21(1): 86-89.
- [6] H Jonathan Chao, Bin Liu. High performance switches and routers[M]. Hoboken, New Jersey: Published by John Wiley & Sons, Inc, 2007.

作者简介:

李 平 男,(1987—),硕士研究生.研究方向为 VLSI 系统与计算机辅助设计.

杨婷婷 女,(1986—),硕士研究生.研究方向为 VLSI 系统与计算机辅助设计.

韩俊刚 男,(1943—),教授,硕士生导师.研究方向为 VLSI 系统与计算机辅助设计.