

# 大型传感器网络中高速连续数据流的缓存策略

汪海山 康建斌 耿立中 马 骋 熊剑平 贾惠波

(清华大学精密测试技术及仪器国家重点实验室 北京 100084)

**摘要** 大型传感器网络是指一种规模巨大的传感器网络,其传感器节点多达上万个(设为  $n$ ),如果每个传感器的采样率为 1ms,每个采样点为 24bits,则整个网络产生的数据总流量可高达  $24n$ kbps(因为  $n$  可达上万,所以流量可达几百 Mbps)。如何实现这个巨大网络中传感器节点到基站的高速连续数据流的缓存,是保证整个网络顺利运行的关键所在。采用 FPGA 和异步 FIFO 的结合,完美地解决了这个问题。

**关键词** 大型传感器网络,高速连续数据流,二级缓存,双缓存

## Buffer Strategy of High Speed and Continuous Data Stream in Huge Sensor Networks

WANG Hai-shan KANG Jian-bin GENG Li-zhong MA Cheng XIONG Jian-ping JIA Hui-bo

(State Key Laboratory of Precision Measurement Technology and Instruments, Tsinghua University, Beijing 100084, China)

**Abstract** Huge sensor network is a large sensor network in scale. In this system, total sensor nodes can reach up to several ten thousands(suppose the number is  $n$ ). If the sample of each sensor node is 1 millisecond, and every sample point has 24 bits, so the total speed of whole network is  $24n$  kbps (because  $n$  may be up to several ten thousands, the speed can reach up to several hundred Mbps). How to realize the high speed and continuous data stream from sensor nodes to sink nodes in this huge network, is the key to ensure the normal running of the whole network. This paper adopted a combination of FPGA and asynchronous FIFO, which solves the problem perfectly.

**Keywords** Huge sensor networks, High speed and continuous data stream, Two-level buffer, Double buffer

### 1 概述

FIFO 有同步和异步两种。同步 FIFO 的读写用同一个时钟,逻辑简单,速度快,但其应用受到时钟限制,一般用在逻辑比较简单的单时钟模块中做(高速)数据流的缓存。异步 FIFO 读写可用不同的时钟,可用作不同时钟域之间数据传输的速度匹配,其端口还可以采用不同的数据宽度<sup>[1]</sup>。这种特性使数据采集系统中控制器(一般为单片机或 DSP 等,8 位、16 位或 32 位系统)对数据缓存 FIFO 的操作变得简单,将相当一部分软件功能转移到硬件,提高了软件效率,也提高了整个系统的可靠性。

FIFO 再加上一些附属的控制逻辑,可以实现更高级的帧控制<sup>[2]</sup>,极大地加强了 FIFO 在各个领域中的应用。随着现场可编程门阵列(FPGA)和复杂可编程逻辑器件(CPLD)的飞速发展,为开发 FIFO 的附属控制逻辑提供了更为广阔的空间。尤其是新型的 FPGA,应用其内嵌的存储器模块和生产厂商提供的稳定 IP 核,可以方便地完成绝大多数设计。因此,FPGA/CPLD+FIFO 的完美组合在数据采集系统中应用极为广泛<sup>[3]</sup>。

对高速、连续数据流的缓存处理,一般用双缓存<sup>[3,4]</sup>或半满请求技术。双缓存技术在图像处理等对数据流量处理要求比较高的领域已经取得了很多优秀的成果<sup>[4]</sup>,它稳定性好、系统出错后容易恢复。半满请求技术的控制逻辑虽然简单一

些,但存储器选用 FIFO 时,系统一旦出错,将很难恢复。

以传感器阵列为基础的大型传感器网络内,信道流量为每个传感器信息的总和,要将所有的信息实时传输到中央控制器,对数据通信、节点的数据缓存和中转都提出了很高的要求。本文针对这种大型传感器网络节点内的数据特性,设计了一种由两级 FIFO 构成的缓存结构,可以辅助实现链路层的检错处理,并对高速数据流进行缓存,极大地提高了节点处理器的 IO 效率。

### 2 大型传感器网络的结构

该大型传感器网络结构如图 1 所示。

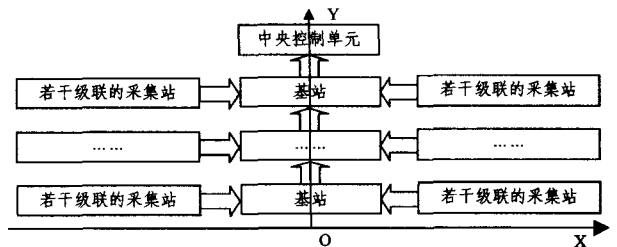


图 1 大型传感器网络结构框图

采集站——负责传感器信号的预处理、A/D 转换,并按照预定义的协议对采集到的数据进行组包、发送,同时转发下级节点过路的数据包。

到稿日期:2008-06-25

汪海山(1982-),男,博士研究生,主要研究方向为大型传感器网络,E-mail:whs05@mails.tsinghua.edu.cn.

X线——位于X方向上的若干采集站组成的一维结构。

基站——负责缓存其所在X线向上发送的数据包,然后合并、重组,并转发到中央控制单元。类似采集站,基站也要同时转发下级节点过路的数据包(因为数据量巨大,所以设多基站)。

Y线——位于Y方向上的若干基站组成的一维结构。

本系统为一个分布式大型传感器信息采集系统,假设所有传感器信号的采样频率为1ms,采样精度为24bits,每条X线最多可以接1000个传感器,因此信道最大数据量可达24Mbps(不计协议效率)。假设采集站和基站之间采用的是RS-485通信协议(该协议与缓存策略设计无关,只需保证FPGA内的收发器设计与物理层、链路层一致),一条用作低速的命令传输,两条用于高速的数据传输。其中,基站可能位于X线的任何位置。也就是说,每个基站两侧的数据流总和虽然为24Mbps,但不一定是对称的。

假设Y线上的基站最多有10个(也就是10条X线),则基站之间通信的最高速率可达240Mbps(不计协议效率),采用千兆以太网可以满足要求。基站在系统中处于非常关键的位置,每个基站都要具有从X线获取24Mbps的RS-485协议数据、处理24Mbps的协议转换数据并实时发送24Mbps的以太网数据的能力,同时要具有转发过路的240Mbps以太网数据的能力。

为了提高基站的数据处理能力,提高其IO效率,我们在基站内的主控CPU的外部总线上挂接一块FPGA,专门用于做RS-485协议的解析和数据缓存。

### 3 FPGA内数据缓存结构设计

#### 3.1 两级缓存结构设计

FPGA内部的两级FIFO缓存结构如图2所示,基站两侧的X线分别称为小端和大端。小/大端1/2#数据RS-485通道的1级FIFO缓存主要负责缓存1个正确的数据帧和剔除CRC校验出错的数据帧;2级FIFO缓存主要负责多个RS-485数据帧的收集,当数据量累积到一定程度后,向主控CPU发起读请求,以提高CPU的IO效率和系统的整体性能。

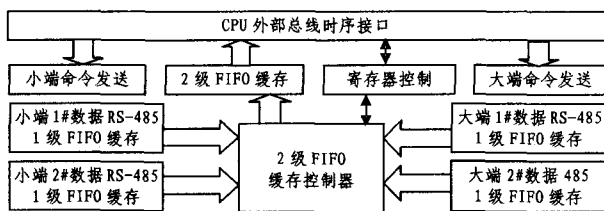


图2 基站内数据缓存设计

#### 3.2 RS-485链路层数据特性及两级缓存结构设计方案

RS-485链路层的帧格式是根据需求自定义的。假设每个数据帧的总长度为206字节,其中数据长度为192字节(4个传感器16个采样点的数据),协议长度为10字节,CRC校验为4字节。缓存数据时应去掉CRC,因此每个数据帧只有202个字节。

当数据帧充满信道时,下一个帧的帧头会紧邻上一个帧的帧尾。当上一个帧写入FIFO后,还没来得及处理下一个帧马上就要求写入时,显然单FIFO的缓存不能满足要求。

鉴于此,针对每个RS-485接收器设计两个FIFO,每个FIFO存放一帧数据。将当前接收的数据帧写入一个FIFO,控制逻辑则对另一个FIFO内的数据进行处理。这样,两个FIFO不停地切换,只要控制逻辑的处理时间小于接收一个数据帧的时间,就可以实现连续数据流的接收、处理和初步缓存。

由于每个基站内有4个RS-485接收器,而且RS-485信道上的数据帧相对较短。如果每接收到一个帧就向CPU产生一次中断请求,势必会因频繁中断(每16ms产生250个中断)而对CPU的正常工作产生消极影响,再加上对数据的读取、处理和转发,极大地加剧了CPU的负担。因此,采用二级缓存,将4个RS-485接收器的数据缓存到一定数量,或缓存到一定时间后,向CPU产生一次读中断请求。这样,CPU可以在一次中断内读取较长的数据,有效地提高了系统的整体效率。

假设4个RS-485接收器共要接收 $F_{total}$ 个数据帧,CPU对外部中断的响应时间为 $t_{irq}$ (包括中断等待时间和保存CPU状态时间),CPU读取一个数据帧的时间为 $t_{read}$ ,如果2级FIFO缓存每收到 $n$ 个数据帧向CPU产生一次读中断请求,则CPU读取这 $F_{total}$ 个数据帧的总时间为

$$t_{total} = \frac{F_{total}}{n} \times (t_{irq} + n \times t_{read}) = F_{total} \left( \frac{t_{irq}}{n} + t_{read} \right)$$

在50MHz的外部总线时钟驱动下,读取一个数据帧的时间约为

$$t_{read} = 20\text{ns} \times 10 \times 202 / 2 = 20200\text{ns} \approx 20\mu\text{s}$$

因为外部总线的数据宽度为16位,所以只需读 $202/2=101$ 次,10为外部总线每读一次需要的时钟数。

CPU内运行的是Linux 2.6版本内核,中断响应频率为1000Hz,取 $t_{irq} \approx 1\mu\text{s}$ , $F_{total} = 1024$ ,则

$$t_{total} = 1024 \left( \frac{1}{n} + 20 \right)$$

CPU读取数据的总时间与缓存数据帧数的关系如图3所示。可见,如果不进行2级缓存,即 $n=1$ ,则有约4.7%的时间浪费在中断响应的等待和保存CPU状态上; $n$ 越大,所花费的时间越少,当 $n=10$ 时, $t_{irq}$ 从原来的4.7%猛降到0.49%,且趋势趋向平缓。

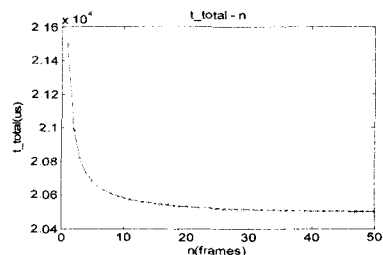


图3 CPU读取数据总时间与缓存数据帧数的关系

$n$ 虽然越大越好,但是由于FPGA内部存储器容量和实时性的约束, $n$ 不可能做到太大,因此 $n$ 应该选择一个适当的值。

CPU上运行的是Linux 2.6版本的内核,其中断响应频率为1000Hz,即1ms,而一条级联链平均每16ms产生250个数据帧,即15.625帧/ms,因此至少缓存16帧才能达到系统的中断响应能力。

极端情况下,基站的两侧各连接125个采集站。这样在一次数据采集过程中,可能产生周期为16ms的48Mbps的超

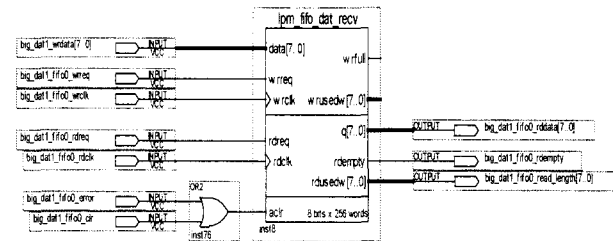
高速数据流,即前 8ms 数据流速度为 48Mb/s,后 8ms 空闲,总的平均速度仍为 24Mbps。这种情况下,至少要将上述的 16 帧扩大到 2 倍,即 32 帧,才能满足峰值流量的缓存需求。

考虑到系统的性能余量和 FPGA 片上存储器可开出 FIFO 容量的大小,每个 2 级 FIFO 缓存设为 4k 字节(19 个帧),两个共 8k 字节(38 个帧)。

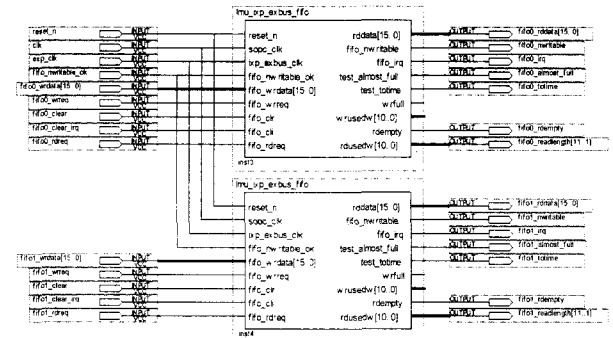
#### 4 方案的实现

选用 intel 的 IXP460 网络处理器作为基站的主控 CPU, altera 的 CycloneII 系列 FPGA EP2C35F484 做逻辑处理和数据缓存,采用 50MHz 的外部总线时钟驱动,该二级缓存结构设计用 Quartus II 的实现如图 4 所示。

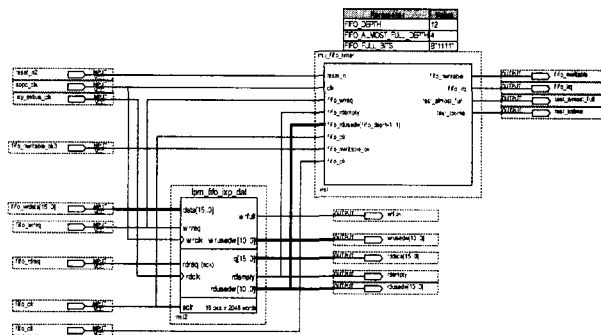
基站两侧的 X 线同时加上 24Mbps 的数据流,数据经过两级缓存后,最终基站通过 IXP460 外部总线读取的数据最高可达 40Mbps,远大于预设的 24Mbps,同时还实现了数据的实时处理和发送,完全达到了系统的设计要求。



(a) 一级缓存 (8 个, 每个 256 字节)



(b) 二级缓存 (2 个, 每个 4096 字节)



(c) 二级缓存的内部结构: FIFO 及其控制

图 4 二级缓存设计用 Quartus II 的实现

**结束语** 大型传感器网络是一种特殊的传感器网络,它由规则的大规模阵列传感器组成,网络中传感器数目多达上万个,产生的采集数据总流量可高达几百 Mbps,同时要实现数据采集的同步、采集数据的实时传输和处理等,是一个十分复杂的工程。

本实验中,运行在采集站和基站之间的是 RS-485 协议,该协议与缓存设计没有关系,只要保证 FPGA 内的接收器和相应的物理、链路层协议一致,该缓存策略适用于所有基站的缓存设计,有极大的扩展性。

#### 参考文献

- [1] 王纲毅,王振华,田金文,等. 读写数据宽度不同的异步 FIFO 设计[J]. 计算机与数字工程,2005,33(6):107-110
- [2] 房海东,潘长勇,杨知行. 处理整帧数据的 FIFO 的巧妙控制设计[J]. 电讯技术,2003(4):63-67
- [3] Thor J, Akos D M. A direct RF sampling multifrequency GPS receiver [J]. Position Location and Navigation Symposium. IEEE,2002;44-51
- [4] Zinner C, Kubinger W. ROS-DMA: A DMA Double Buffering Method for Embedded Image Processing with Resource Optimized Slicing[C]// Real-Time and Embedded Technology and Applications Symposium. 2006;361-372

(上接第 20 页)

- [16] Lee G, Yeh M S, Lo S C, et al. A Strategy for Efficient Access of Multiple Data items in Mobile Environments[C]// International Conference on Mobile Data Management. 2001
- [17] Lee G, Lo S C. Broadcast Data Allocation for Efficient Access of Multiple Data Items in Mobile Environments[J]. Mobile Networks and Applications, 2003, 8: 365-375
- [18] Chung Y D, Bang S H, Kim M H. An Efficient Broadcast Data Clustering Method for Multipoint Queries in Wireless Information Systems[J]. The Journal of Systems and Software, 2002, 64: 173-181
- [19] Tsaih D, Wu G M, Wang C B, et al. An Efficient Broadcast Scheme for Wireless Data Schedule Under a New Data Affinity Model
- [20] Chang Y I, Hsieh W H. An Efficient Scheduling Method for Query-Set-based Broadcasting in Mobile Environments[C]// International Conference on Distributed Computing Systems Workshops, 2004
- [21] Sun W W, Zhang Z Y, Yu P, et al. Skewed Wireless Broadcast Scheduling for Multi-item Queries[C]// The 3rd IEEE International Conference on Wireless Communications, Networking and Mobile Computing. 2007

- [22] Goldberg D E. Genetic Algorithms in Search, Optimization and Machine Learning. Addison Wesley, 1989
- [23] Huang J L, Chen M S. Broadcast Program Generation for Unordered Queries with Data Replication[C]// ACM Symposium on Applied Computing. 2003
- [24] Prabhu N, Kumar V. Data Scheduling for Multi-item and Transactional Requests in On-demand Broadcast[C]// International Conference on Mobile Data Management. 2005
- [25] Huang J L, Chen M S. Dependent Data Broadcasting for Unordered Queries in a Multiple Channel Mobile Environment[J]. IEEE Transactions on Knowledge and Data Engineering, 2004, 16(9): 1143-1156
- [26] Huang J L, Peng W C, Chen M S. SOM: Dynamic Push-pull Channel Allocation Framework for Mobile Data Broadcasting [J]. IEEE Transactions on Mobile Computing, 2006, 5(8): 974-990
- [27] Xu J, Lee W C, Tang X. Exponential Index: A Parameterized Distributed Indexing Scheme for Data on Air [C]// Second ACM/USENIX International Conference on Mobile Systems, Applications, and Services. 2004
- [28] Lo S C, Chen A L P. Efficient Index and Data Allocation for Wireless Broadcast Services[J]. Data Knowledge, 2007, 60(1): 235-255