

基于FPGA的航空全双工以太网交换芯片

王鹏, 姚明旸, 鲍民权, 邱智亮

(西安电子科技大学综合业务网国家重点实验室, 西安 710071)

摘要: 基于ARINC664规范第7部分, 提出符合该规范的基于FPGA的AFDX交换机整体设计方案及其核心交换芯片中关键模块的Verilog HDL实现, 并通过功能仿真、时序仿真、网络仿真等手段对交换芯片的功能进行验证。实验结果证明, 该交换芯片可为航空器中的数据通信设备提供有保障的基于以太网数据帧的交换通道, 具有较高的交换性能及稳定性。

关键词: 全双工交换式以太网(AFDX); 现场可编程门阵列; 虚链路

Switch Chip of Avionics Full Duplex Ethernet Based on FPGA

WANG Peng, YAO Ming-wu, BAO Min-quan, QIU Zhi-liang

(National Key Lab of Integrated Service Networks, Xidian University, Xi'an 710071)

【Abstract】Based on the analysis of ARINC664 specification part 7, this paper provides a solution of AFDX switch and key modules of switch chip using FPGA and Verilog HDL description are presented as well. The scheme has past functional, timing and networks simulation. Experimental result shows that this chip can provide guaranteed data communication channels based on switched Ethernet frames, also has high performance and stability.

【Key words】 AFDX; FPGA; virtual link

1 概述

ARINC664规范是通过航空电子技术委员会审议的新一代机载以太网标准。该规范第7部分定义了一个全双工的、确定性的机载数据交换网络(AFDX), 它通过使用虚链路模拟一个点到点的、具有确定性的网络, 并通过平行冗余结构来提高网络的可靠性^[1]。美国波音和欧洲空客公司已研制出相关的成熟产品并投入使用。Actel公司也发布了基于反熔丝式FPGA的AFDX交换芯片设计方案。随着“大型飞机”项目的启动, 研制具有我国自主知识产权的新型机载数据网络交换芯片已成为一个现实而迫切的任务。

本文介绍符合ARINC664规范的AFDX交换芯片和试验整机的设计方案。交换芯片使用单片FPGA设计。FPGA可充分发挥其设计周期短、投资小、灵活性高等优点, 但由于FPGA逻辑和存储资源以及速度的限制, 也给设计带来一些困难。

2 AFDX特点介绍

AFDX引入虚链路的概念实现一个具有确定性的网络。该网络采用星型拓扑结构, 一个交换机连接着数十个端系统, VL是在2个端系统之间建立的一条单向的虚链路, 每个端系统可使用多条VL与其他端系统进行多个应用之间的通信, 每一个交换机最多能支持4 096条VL。

带宽分配间隔(BAG)和时延抖动(Jitter)参数明确了VL应具有的服务质量。BAG定义当Jitter为0时, 一条VL中连续2个分组的最小间隔时间, 其最小单位是1 ms, 取值为 2^N ms(0 ≤ N < 8)。BAG类似于一个时隙, 在这个时隙内至多有一个分组发送。VL的BAG越小, 表示分配给它的带宽越大。Jitter是分组实际到达时刻与分组预期到达时刻之间的差值, 反映了分组发送时延的变化范围, 一条VL有一个预设的最大

Jitter, 必须满足下列关系^[1]:

$$Jitter_{max} = 40 \mu s + \frac{\sum_{i \in \{set\ of\ VLs\}} (20 \text{ Byte} + L_{max}) \times 8 \text{ bit/Byte}}{Nbw \text{ bit/s}} \quad (1)$$

其中, Nbw为传输介质的带宽; L_{max}为允许最大分组长度。

VL的相关参数在整个网络开始运行之前由系统集成者预置到交换机和端系统中, 不需要复杂的信令来实现连接, 符合航空数据交换网络在系统稳定运行期间无需进行修改的要求。

3 交换机及核心交换芯片的功能结构

3.1 AFDX交换机的组成

图1为AFDX交换机硬件架构。

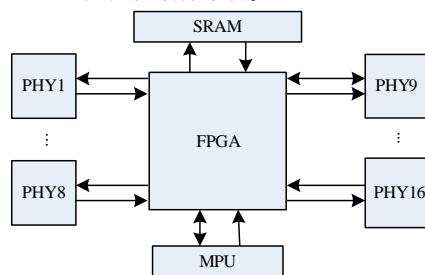


图1 交换机的硬件架构

作为交换芯片的设计载体, FPGA选用Altera公司的高端产品EP2S180和以太网MAC IP核来加速设计进度^[2]; MPU选用Freescall公司的MPC8270; 采用Cypress公司的SRAM作为

作者简介: 王鹏(1983 -), 男, 硕士研究生, 主研方向: IC前端设计; 姚明旸, 博士; 鲍民权, 副教授; 邱智亮, 教授、博士生导师

收稿日期: 2008-04-06 **E-mail:** wp1983roc@163.com

交换机共享存储器；嵌入式操作系统选用Windriver公司的VxWorks；处理器接口使用60X总线接口。MPU作为SNMP的代理者，实现系统管理者和交换机的交互。通过MPU将连接参数表配置到FPGA内部寄存器的方式来完成交换芯片的配置。交换机的管理包括配置管理、端口管理、VL管理、性能管理和故障管理。配置管理用于设置系统属性、运行时间和交换机工作状态等；端口管理包括更改端口名、工作状态和工作速率等。VL管理包括VL资源的分配及VL表项信息的更新。性能管理负责统计端口输入分组数、输出分组数、单播和多播分组数等。故障管理主要负责对设备和网络运行的故障进行实时监控，并给出告警信息。交换机的初始化和配置主要由硬件完成，管理功能将主要由软件来实现。

3.2 核心交换芯片的组成

交换芯片的组成如图2所示。芯片采用共享缓存方式实现交换功能^[3]，可将性能要求与FPGA特点较好结合。

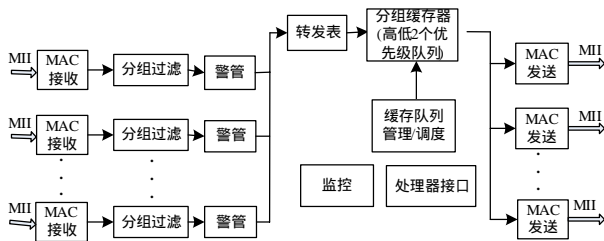


图2 AFDX 交换芯片的功能结构

分组进入交换芯片后，首先由MAC IP核接收并进行CRC校验，然后进入接收端口缓存中接受分组过滤和警管操作，实现接纳控制。其中，过滤和警管模块从预设的连接参数表中根据到达分组的VL号索引到相关配置参数。当调度模块轮询到该输入端口并判定有数据需要转发时，控制接收缓存和输入总线，将该分组发送到片外共享存储器中。调度模块从存储队列管理模块中得到相应的存储空间指针，并通知后者将分组的地址、长度、目的地址等信息加入到相关输出端口的队列链表中。在输出端，发送调度单元按调度策略不断轮询各个输出队列，若某队列有待发的分组，调度单元进而询问相应输出端口的发送缓存中是否有足够存储空间，若有，调度器就通过总线将分组转移到输出缓存中。输入输出缓存各由2个并行FIFO组成，实现乒乓操作^[4]。若端口未能就绪，则该输出队列等待下一次发送机会。输出端口缓存通过MAC发送模块将分组发送到片外输出链路上。由于总线采用较高的加速比，在轮询效率较高的情况下出现输出缓存队列阻塞的概率很小。通过对总线和端口缓存器的适当修改可实现监测端口功能。交换单元中的处理器接口用来连接MPU，实现对内部寄存器的存取操作。

4 交换芯片中的关键模块

4.1 警管模块

本文采用基于字节的令牌桶算法对业务流进行流量控制。初始化时，每条虚连接VL对应的AC寄存器被赋予一个最大值作为上限。对一个分组，如果字节计数器减去分组的字节长度后仍有余量，则认为该分组遵守系统要求的带宽、抖动特性，通过警管；如果计数的值小于分组长度，则认为该分组违约，将其丢弃。字节计数器以25 μs为间隔实现一次自增运算，增加的幅度与带宽有关，在自增后与AC寄存器中的初始值进行比较，如果当前字节计数器的值大于初始最大值，则将初始最大值赋给字节计数器；如果AC寄存器

小于初始最大值，则AC可继续进行自增运算。运算到警管单元后的第4个时钟周期，模块给出处理结果，如图3所示。

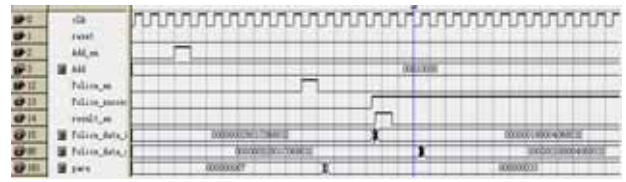


图3 警管模块时序仿真

4.2 分组转发

初始化时，依据连接参数表对端口寄存器进行配置，同时对所有VL警管计数器设初值。对于转发信息的查找，与其他基于算法的转发解决方案相比，CAM方式具有占用空间小，查找速度快的优点。按内部逻辑百兆的工作频率，由于一个最短分组64 Byte通过MAC核和与内部缓存之间的接口(32位宽)需要16个时钟周期，要求检索时间尽量小于160 ns。根据设计目标和规范要求，每个端口可拥有多达256条VL，则应在每个端口处配置一个深度为256项、宽度为128位的CAM查找表。此处CAM以VL号作为输入，输出的内容是该VL的配置信息，如分组长度范围、转发端口号、优先级信息等。时序仿真显示，输入VL ID后的第4个周期，CAM模块即可输出稳定内容，如图4所示。

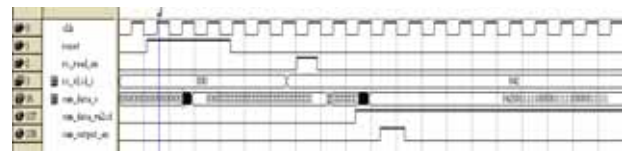


图4 转发模块时序仿真

4.3 存储队列管理模块

4.3.1 队列管理组织形式

队列管理通过链表操作来实现。基于链表节点管理方式用虚拟地址将指针物理地址和数据物理地址统一起来，简化了链表的操作过程，实现高效队列管理。链表节点由数据单元和指针单元构成，数据单元保存在片外SRAM中，指针单元保存在FPGA内的M4K存储资源中。根据一个虚拟地址可找到一个内部的指针单元和一个外部的数据单元。片外SRAM的一个物理地址对应一个32位的数据单元，而每个数据单元可保存一个最大分组，则一个虚拟地址对应SRAM的380个物理地址。每个指针单元的宽度设为64 bit。一个虚拟地址对应着M4K中的一个物理地址。因此，仅当访问SRAM的时候，须将虚拟地址转换为物理地址。

4.3.2 队列管理工作过程

图5表示队列管理功能模块的动作流程，下文以分组的写入操作为例描述队列管理的工作流程。

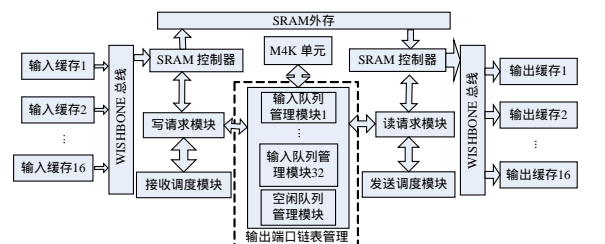


图5 队列管理功能模块

接收调度器接收分组转发信息，写请求模块检查输出端口号并输出到端口队列管理模块(OSQM)。OSQM检查端口

队列长度寄存器,判断本端口队列是否已满。如果队列已满,则 OSQM 向写请求模块发送已满信号,写请求模块丢弃该分组;否则,由 OSQM 给空闲队列管理模块(FQM)发送信号请求空闲地址。FQM 从空闲队列头指针寄存器中读取队列头地址(假设为 a),并通过 OSQM 将该空闲地址送入写请求模块。写请求模块根据这个地址从 Input Buffer 中将等待分组写入到 SRAM 中。与此同时,FQM 进行空闲队列指针调整操作。FQM 读出该空闲地址后,以地址 a 访问 M4K 存储器,从对应指针单元中得到地址 a 的 Next Address 值(假设为地址 c),然后用地址 c 更新空闲队列头节点指针寄存器中,完成地址 a 在空闲队列中的出队操作。

OSQM 读取输出队列的队尾节点指针寄存器的内容(假设为 m)将地址 a 写入到地址 m 所对应指针单元的相应位置,并更新队尾节点指针寄存器的内容为地址 a ,从而完成地址 a 在输出端口模块的入队操作。

4.4 调度算法分析及调度器设计

4.4.1 调度算法分析

本设计在输出队列中设置高低 2 个优先级,采用静态优先级调度器。任何一个供调度器调度的分组业务流都符合 $(X_{\min}, X_{\text{ave}}, I, S_{\max})$, 其中, X_{\min}, X_{ave} 分别表示分组的最小和平均到达间隔; I 表示测量和计算平均间隔的时间长度; S_{\max} 表示该业务流中的最长分组长度。在特定的业务量下,经过该调度器处理后的输出流的分组时延都有确定的上限。经过警管的分组业务流的属性中包括分组到达间隔 T_{BAG} 、允许的最大抖动 J_{\max} 、最大分组长度 S_{\max} 。根据设计中采用单级漏桶机制的流量管制方案,分组到达间隔的平均值可认为是 2 倍的 T_{BAG} 。由于在该方案中不允许连续出现 2 个最小到达间隔,因此,与式(1)中的各个流量描述之间,存在如下关系:

$$\begin{cases} X_{\min} = T_{BAG} - J_{\max} \\ X_{\text{ave}} = T_{BAG} \\ I = 2T_{BAG} \\ S_{\max} = S_{\max} \end{cases} \quad (2)$$

根据该式可得到如下结论:采用 2 级静态优先级的调度器,输出端口速率为 l ,到达业务流 VL 的业务属性 $(T_{BAG_j^q}, J_{\max_j^q}, S_{\max_j^q})$, $\overline{S_{\max}}$ 为链路中允许的最大分组长度,如果有 $\sum_{q=0}^1 \mu_{\text{ave}}^q = 1$, $\mu_{\text{ave}}^q = \frac{S_{\max_j^q}}{\sum_{j \in C_q} T_{BAG_j^q} \times l}$,则到达优先级队列 m 中的分组的最大时延不超过 $\overline{d_m}$,其中

$$\overline{d_m} = \begin{cases} \min(\overline{d_m}, \overline{d_m}') & \sum_{q=0}^{m-1} \mu_{\text{peak}}^q < 1 \\ \overline{d_m}' & \sum_{q=0}^{m-1} \mu_{\text{peak}}^q = 1 \end{cases} \quad (3)$$

上述结论表明,当进入交换机的业务流得到很好的规则和管理时,静态优先级调度算法可给出确定的时间延迟上限。

4.4.2 调度模块设计

调度器接收来自 16 个输入端口的分组,通过轮询实现分组的并串转换,提交给队列管理模块并分配存储空间,生成调度信息,控制数据传输通路由输入缓存模块发送输入分组到片外存储器;通过轮询的调度算法访问 16 个输出端口队列,提交队列管理模块释放存储空间,生成调度信息控制数据传输通路由存储器发送输出分组至输出缓存模块。在输出端口的子队列中按静态优先级调度算法进行调度,实现快速将各输出端口队列的排头分组发送到输出端口缓存中,并在输入业务流量整形的配合下实现确定性能的发送调度。调度器设计的基本思想是分组调度与分组传输相互独立,实现调

度器对输入分流水调度。高度功能模块如图 6 所示。

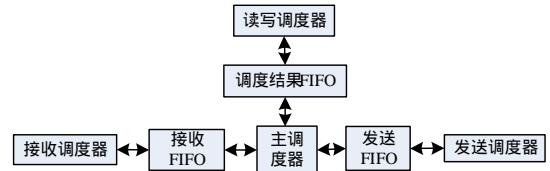


图 6 调度功能模块

各模块功能如下:

接收调度器:轮询 16 个输入端口中的分组缓存接口,若轮询到的端口中有待发送的分组,从该端口接口获取该分组的相应信息,如目的端口、优先级、多播/单播以及长度等,将这些信息写入接收 FIFO 中供总线控制模块使用。

接收 FIFO:暂存放接收调度器的调度结果,并等待主调度器的访问。

发送调度器:轮询 16 个输出端口队列,将等待输出分组的信息写入发送缓存。

发送 FIFO:暂存发送调度器的调度结果,并等待主调度器的访问。

调度结果 FIFO:接收主调度器的最终调度结果,并等待读写调度器的访问。

读写调度器:该模块连接调度模块和片外 RAM 的接口控制器,从 FIFO 中读取最终调度结果,并将分组传递至片外 RAM,为发送 FIFO 提供当前传输分组的长度信息。

主调度器:访问接收 FIFO,将调度信息发送到队列管理模块,并给写请求模块分配存储空间,将该地址空间提交至调度结果 FIFO;访问发送 FIFO,将调度信息发送到队列管理读请求模块,释放存储空间,接收分组长度信息并将调度结果提交至调度结果 FIFO。

根据网络仿真,在系统工作时钟为百兆的情况下,该设计方案能达到的最快的处理能力,相当于每秒可处理约 2.70 MB 个分组。而 16 个端口的 AFDX 交换机理论上的最大满负荷状态为每端口每秒到达 148 810 个最短长度的以太网帧,即每秒共注入 2.38 MB 个以太网分组。因此,调度器的处理能力满足设计需要。

5 结束语

本文介绍一种符合 ARINC664 规范所要求的 AFDX 网络交换机整体解决方案,描述交换芯片中部分模块的设计思想,可较好发挥 FPGA 的优势,各功能模块的 Verilog 代码已使用 QuartusII 进行功能和时序仿真,并在交换机验证平台上通过初步调试。使用 SmartBits 200 对 16 端口交换芯片进行性能测试,吞吐率达 98%,丢包率在 100 Mb/s 时为 0。测试结果表明,本文所提出的设计方案可行。

参考文献

- [1] Courtney R. Aircraft Data Network, Part 7-avionics Full Duplex Switched Ethernet(AFDX) Network[M]. Stockholm, Sweden: Airlines Electronic Engineering Committee, 2004.
- [2] Altera Corporation. StratixII Device Handbook[Z]. 2006.
- [3] 刘增基, 鲍民权, 邱智亮. 交换原理与技术[M]. 西安: 西安电子科技大学出版社, 2003.
- [4] 吴继华, 王 诚. Altera FPGA/CPLD 设计(高级篇)[M]. 北京: 人民邮电出版社, 2005.