

星载 SpaceWire 路由器的研究与设计

毛春静^① 关永^① David Jungwirth^②

^①(首都师范大学信息工程学院 北京 100048)

^②(维也纳工业大学 维也纳 奥地利 A-1040)

摘要: 为满足航天器有效载荷间高速数据多路传输的未来发展需求, 该文研究提出了基于 SpaceWire 的星载数据路由的设计方案。在 SpaceWire 总线标准网络层分析的基础上, 结合虫孔路由原理构建路由器内部结构, 提出了采用无阻塞交叉开关的 SpaceWire 路由器设计方案。在 Simulink 环境下建立了路由器仿真模型, 并通过 BWR 解决了虫孔路由阻塞机制带来的刹车问题, 使路由器的满负荷通信下的平均丢包率减少了 70%。

关键词: 卫星通信; 空间辐照; SpaceWire; 虫孔路由; 带缓冲的虫洞路由

中图分类号: TN927

文献标识码: A

文章编号: 1009-5896(2010)08-1904-06

DOI: 10.3724/SP.J.1146.2009.00464

Research and Design of on-Board Dynamic SpaceWire Router

Mao Chun-jing^① Guan Yong^① David Jungwirth^②

^①(College of Information Engineering, Capital Normal University, Beijing 100048, China)

^②(Vienna University of Technology, Karlsplatz 13 | A-1040 Vienna, Austria)

Abstract: In order to satisfy the requirement of aircraft's multi-transmission for further development, this paper leads to a resolution about on-board SpaceWire routing technology. Based on the network level analysis of SpaceWire standard, it built the architecture of router with wormhole routing theory and non-blocking switch mechanism. A simulation model of router had been set to verify the design, and braking-problem from wormhole routing-blocking had been solved with BWR (Buffered Wormhole Routing), it reduces 70% average packet-losing ratio on full load communication.

Key words: Satellite communication; Space radiation; SpaceWire; Wormhole routing; BWR (Buffered Wormhole Routing)

1 引言

随着空间技术、电子技术的不断发展, 以及探测任务的需求, 卫星有效载荷的数量、集中程度、数据类型和数据量等方面比以前有了更大的扩展。传统空间领域数据下行传输大多只有一个通道^[1], 而多载荷飞行器须将所有载荷的数据复接为一串信号, 下行模式单一, 不能灵活控制多路数据输入、多路数据输出, 并且 CPU 无法应对不断增加的数据通信需求。因此, 需要专用的路由设备来控制、管理载荷间的数据传输; 同时考虑到空间环境有较强的辐照、电磁干扰、单粒子事件等因素, 需要专用的路由设备及相应的数据通信总线来实现该特殊环境下的数据通信需求。

SpaceWire 高速总线标准^[2]是欧洲空间局 (European Space Agency) 基于两个商用标准 IEEE1355-1995 和 IEEE1596.3, 通过对其可靠性、功耗等方面的改进, 面向航空航天应用提出的一种高速数据通信标准。该标准具有通信速率高、容错性强、维护方便等特点, 并具有良好的空间苛刻环境适应能力。

本文在 SpaceWire 总线标准下点对点链路通信的基础上^[3], 结合总线标准的网络层定义及虫孔路由原理, 设计了一种能灵活控制多有效载荷间的数据传输的星载路由器。

2 网络层分析

2.1 网络

SpaceWire 高速总线网络由大量的连接、节点 (node) 和路由器 (router) 组成; 节点是数据包的源和目的; 连接为数据包传输提供通道; 路由器将多个节点连接在一起, 并为数据包从一个节点到其他多

2009-04-02 收到, 2010-05-27 改回

国家自然科学基金 (60873006) 和北京市自然科学基金 (4082009)

资助课题

通信作者: 毛春静 mcjing@163.net

个可能的节点提供路由的方法。图1是总线标准中的网络拓扑结构^[2]。

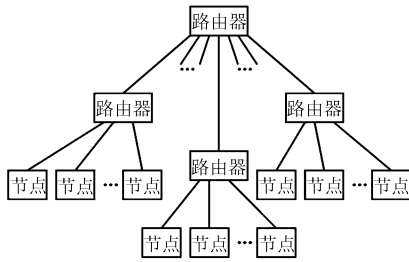


图1 SpaceWire高速总线网络拓扑结构

2.2 数据包

数据被分割成数据包是为了以易于处理的块的形式经过一个网络进行传送，数据包是网络层可以处理的最小数据单元。SpaceWire数据包由3个数据节片(Flit)^[4]组成：一个目的地址，一个内容和一个包结束标记，即EOP(End of Packet)或EEP(Error End of Packet)。

<目的地址><内容><结束标记>

目的地址：目的节点或任务的地址。

内容：传送的数据。

数据包结束标记：一个用来表明数据包结尾的特殊字符。

其中，目的地址包含一个零列表或者更多的目的标识符；内容包含从源传向目的的数据字符；EOP将用来作为正常情况下数据包的结束标记，EEP将用来表示在错误发生的情况下数据包的结束。

2.3 虫孔路由

虫孔路由是数据包路由的一种有效路由形式^[2]。每个数据包包含一个头，该头中含有目的节点地址。一个数据包的头被路由器接收，通过检查目的地址来决定该数据包的路由输出端口。如果被请求的输出端口是闲置的，那么该数据包会立即被路由到该端口。而该输出端口会一直标记为忙直到该数据包的最后一个字符被路由器传送——即数据包的结束标志被路由器检测到。与一个完整的数据包在被路由器发送前先被接收并存储的早期技术相比，虫孔路由削减了在每个路由器内使用的缓冲器的数量，减少了硬件开销，降低了数据包传输延迟，容易实现片上网络(Network on Chip)^[5]。其中一个数据包网络延迟 T_{lower} 可描述为

$$T_{lower} = (L_f / B_{ch}) \cdot D_{sd} + L_p / B_{ch} \quad (1)$$

L_f 是每个数据节片的长度， D_{sd} 是源节点和目的节点之间消息传递路径长度， B_{ch} 是通道带宽， L_p 是数据包长度，如果存在： $L_f \ll L_p$ ，则 D_{sd} 对 T_{lower}

的影响可忽略不计。

图2对虫孔路由进行了描述。该例是一个数据包正在经过路由器从一个节点发送到另一个节点。黑色标注段是数据包的头，其余部分是灰白色。当路由器接收到头，它就会检测请求的输出端口。如果输出端口闲置的，则路由器在输入端口和输出端口间建立链路，随后数据包流经路由器。当数据包的结束标志(EOP或EEP)被路由器接收到时，路由器就终止该连接并将输出端口置为闲置状态等待接收下一个数据包。

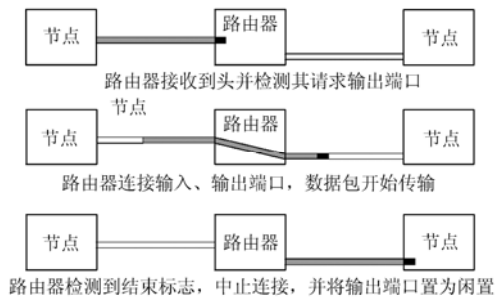


图2 虫孔路由

3 路由器设计

3.1 路由器结构

根据对网络层的分析，结合实际仿真环境及硬件环境，原理样机设计拟采用4个本地接口加上1个用于级联的外部扩展接口。路由器内部结构及模块划分如图3所示。

如图3所示，路由器核心是一个基于虫孔路由的低时延、无阻塞、交换开关(Crossbar-Switch)，通过控制逻辑按照路由表的路由配置将数据包传送到SpaceWire接口以及外部扩展接口。各SpaceWire接口按照SpaceWire标准^[2]提供高速、双向通信及数

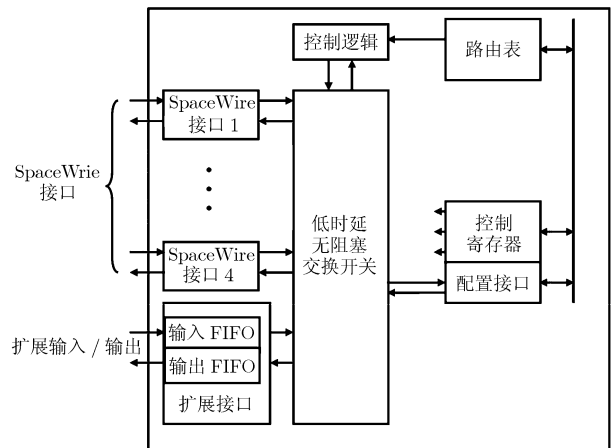


图3 路由器内部结构

据编、解码。配置接口能通过任何一个SpaceWire接口以及扩展接口访问，它是一个根据SpaceWire接口标准内置的一个专用端口，用于读取控制寄存器中的配置信息。端口映射及优先位(priority bits)^[6]等信息可通过配置端口在路由表中设置。

3.2 数据包寻址

数据包寻址的方式有两种：路径寻址和逻辑寻址。考虑到逻辑寻址方式下路由表开销大、复杂度、组网方式不灵活等因素^[6]，不适宜片上网络设计；而路径寻址方式相对简单，路由表开销小，组网灵活；由于路径寻址是基于网络中相对源节点的目的节点的定位，因此通过源节点和路由器处理的数据包寻址的复杂性也相对较低。

图4描述了一个SpaceWire网络。各SpaceWire节点被从N1到N9进行编码，且每个节点只有一个链路接口。SpaceWire路由器被从R1到R4进行编码，且每个路由器有4个链路接口。使用路径寻址方法从节点N1传送数据包的内容到节点N3的路径设置如下：

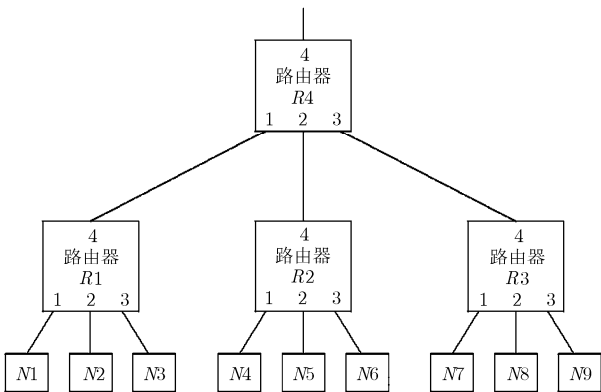


图4 SpaceWire网络实例

<3><内容><EOP>

从节点N1传送数据包的内容到节点N8的路径设置如下：

<4><3><2><内容><EOP>

3.3 头删除

为传输任意大小的数据包，需对数据包进行头删除处理^[7,8]。图5(a)对头删除进行了描述。数据包的第1个头数据字符(目的标识)被用来标识路由输出端口的地址。当路由器收到一个数据包，检测该数据包的第1个目的标识来决定数据包路由经过的输出端口。数据包头的第1个目的标识随即被删除，原数据包头的第2个目的标识符被接下来路由所使用。

图5(b)描述了1个数据包经过两个路由器的情况。数据包的目的地址包括3个目的标识符。在第1

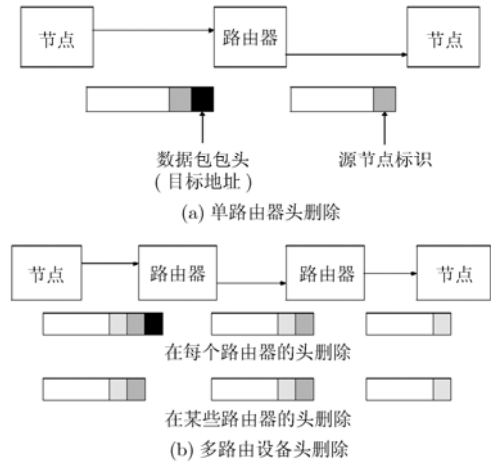


图5 路由器头删除原理

个路由器处第1个目的标识符用来决定使用哪个输出端口，随后该标识符被删除。在第2个路由器处第2个目的标识符被使用随后也会被删除。最后数据包根据在数据最前面的第3个目的标识符到达目的节点。

数据包网络传输的每个阶段，它都被认为是包含了一个单一的目的标识头，一个内容和一个数据包结束标记的数据包。当这个单一的数据标识在路由器被删除时，内容的第1个数据符就变成了新的目的标识头。所有阶段都用相同的方式对待数据包，使用第1个数据标识符来决定目的地址(输出端口)，然后在向前传送数据包之前删除第1个数据标识符。

3.4 无阻断交叉开关

路由器核心部分是一个多路无阻断交叉开关，负责整个通信总线的数据调度、转发、管理等功能，其内部结构如图6所示。

如图6所示，无阻断交叉开关在从任何一个输入到任何一个输出都有完备的连接；同时，通过专门的时序控制各输入/输出间的连接。选择模块对各输入数据包进行目标地址提取、识别、删除，并根据目标地址建立与目标输出的通路，传送数据包。

3.5 设计实现

在Matlab/Simulink仿真环境下构建仿真模型如图7：

参照路由器内部结构(图3)，设计无阻断交叉开关、控制逻辑、路由表、控制寄存器、配置接口等模块并集成在Route模块中，外部提供与SpaceWire接口及扩展接口的通信连接。各SpaceWire接口及扩展接口按照通信协议规定^[2]对输入/输出数据进行编码/解码。各SW_n_S(n=1,2,3,4,EX)模块用于模拟各SpaceWire节点的数据发送；数据发送采用2×N矩阵形式，其中第1列为时间码，第2列为对应时间码的节点发送数据，N的大小取决于仿真步长值和仿真

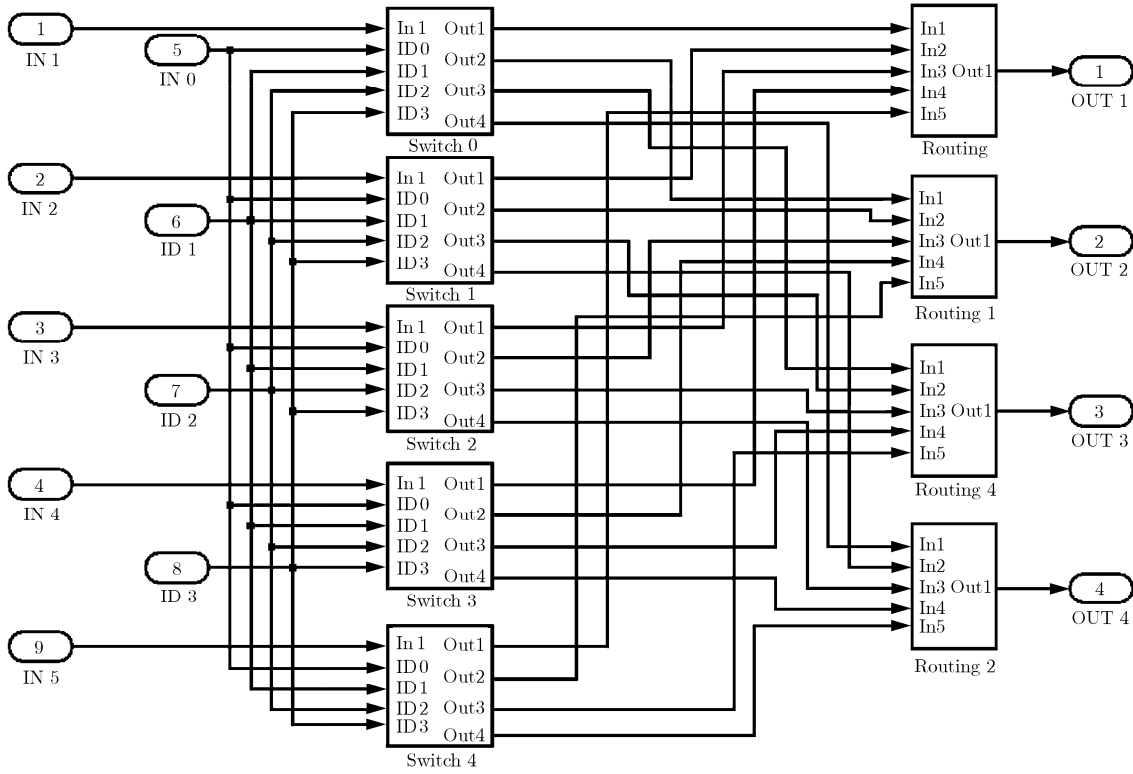


图6 无阻塞交叉开关内部结构

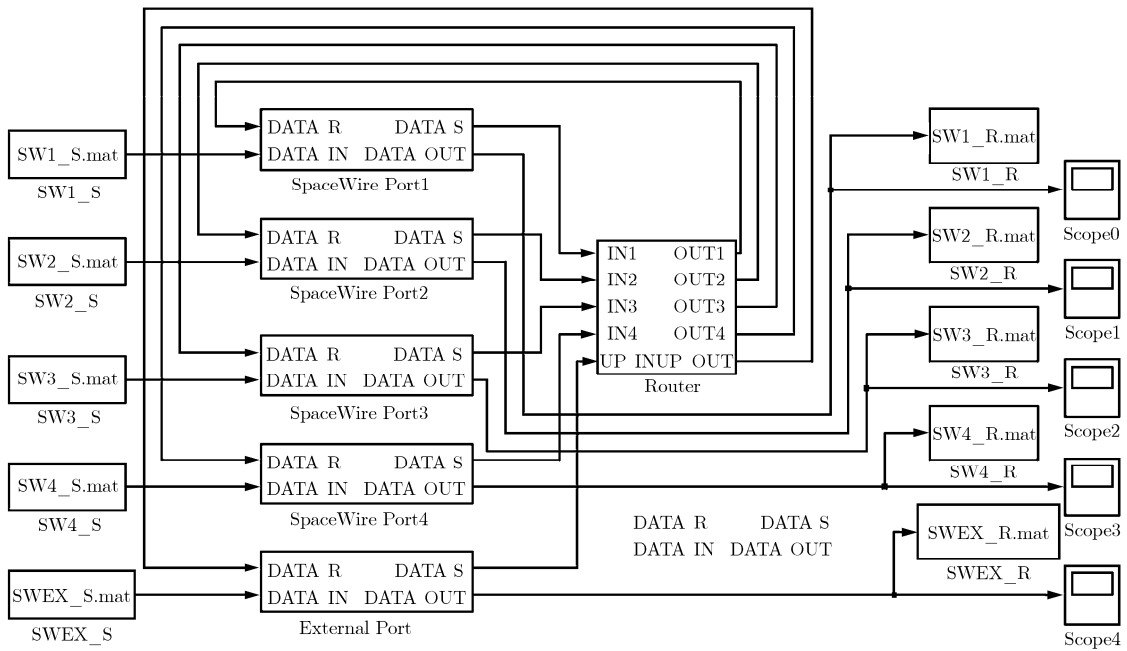


图7 Simulink环境路由器仿真模型

时间的乘积。各SW_n_R($n=1,2,3,4,EX$)模块用于模拟各SpaceWire节点的数据接收,数据格式与发送模块相同,并在各接收端接入一路示波器用于实时、直观观测各节点数据接收情况。

多次仿真分析结果显示,传输速率较高时容易发生数据包丢失。进一步分析发现:由于采用阻塞

机制实现流量控制的虫孔路由,存在固有的刹车问题(Breaking-Problem)^[9,10]。当头节片到达路由器,如果申请的下一个通道被占用,头节片则必须等待。记该数据包从头节片占有的节点传输到尾节片占有的节点时间为 T_b , T_b 可表示为

$$T_b = T_{AD} + T_{ARB} + M \times T_s \quad (2)$$

其中 T_{AD} 为地址译码器译码时间; T_{ARB} 为仲裁器的仲裁时间; M 为数据包中数据节片的长度; T_s 为阻塞信号在路由器间的传输时间。由于所有的数据节片类似流水的方式传输, 所以时钟周期 T_c 必需大于 T_b 即式(3), 否则就会发生数据包丢失。

$$T_c > T_{AD} + T_{ARB} + M \times T_s \quad (3)$$

由于路由器的建立延迟为毫秒级^[10], 信号传输可以在纳秒级时间内完成, 因此时钟周期 T_c 至少为毫秒级。从式(3)可看出, 虫孔路由带来的刹车问题限制了工作频率的提高, 而工作频率是提高传输速率的基础之一。

为解决刹车问题对传输速率的限制, 考虑在设计中引入带缓冲的虫孔路由 BWR(Buffered Wormhole Routing)^[10]。BWR 在传统虫孔路由的基础上, 为路由器增加缓冲区用于缓存被阻塞的数据。在 BWR 技术中, 后续节片的传输将由前一节点的缓冲区是否满决定。当前节点只需要告知下一节点上的节片是否可以传输, 因此并不存在刹车问题。满信号由最后一个缓冲区中是否有数据而形成, 记该信号的形成时间为 T_B 。故时钟周期 T_c 需满足:

$$T_c > T_B + T_s \quad (4)$$

满信号的形成较简单, 在纳秒级时间内可完成, 因此在 BWR 下的时钟周期可达到纳秒级。对比式(3)和式(4)可看出, BWR 可较好的解决刹车问题, 且可使物理通道工作在较高的频率下, 有利于提高通道利用率和传输速率。

按照 BWR 原理为路由器模型设计加入多段缓冲区, 用于缓存阻塞数据。在不同传输速率(2 Mbit/s~50 Mbit/s)下再次进行多次仿真, 数据传输稳定, 不存在刹车问题。

在此基础上, 进一步考虑向实际应用平台的移植。由于有效载荷间数据传输速率较高, 如果完全用软件来实现高速数据路由, 则需要高性能处理器的支持, 势必增加功耗, 对星载设备来说实现代价太大^[1]; 而 FPGA 有设计周期短, 功耗低, 可重构等特点, 较容易实现载荷间数据的高速传输和灵活管理。参照路由器仿真模型, 以 Xilinx XUP Virtex-II Pro 为开发平台, 实现了原理样机的设计。

SpaceWire 路由器原理样机主要由以下几部分组成: 核心路由模块、SpaceWire 协议数据链路模块及两模块间的内部接口模块, 如图 8 所示。核心路由模块控制数据的调度、转发、管理等功能; SpaceWire 协议数据链路模块提供与外部 SpaceWire 协议通信标准的数据节点的通信; 而内部接口模块则用于连接上述两者模块, 并提供两者间的数据格式转换。

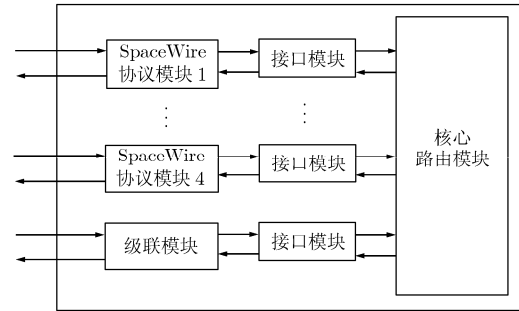


图 8 SpaceWire 路由器原理样机内部结构

在此基础上, 搭建验证系统如图 9 所示。系统由两个路由器级联, 每个路由器下连有若干个数据源测试、模拟设备, 数据源类型包括视频、图像、音频、控制指令等; 并且每个路由器下连有 1 个 PCI 数据源节点卡, 可通过在 PC 机上编写相关软件实时、直观地监测、控制总线工作。路由器完全负责数据路由, 各数据源节点能自行“决策”数据传输目的节点, 路由器分析各节点数据包根据数据包信息转发数据; 而 PC 机通过 PCI 数据源节点卡与路由器相连, 用于监控路由器工作情况, 并可通过其访问路由器控制、配置路由器工作。

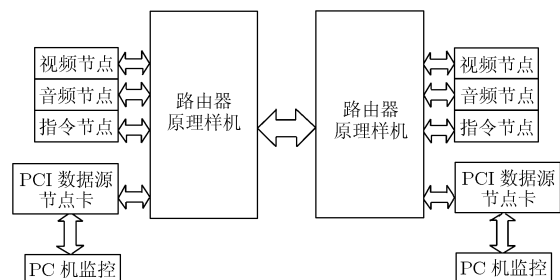


图 9 SpaceWire 路由器原理样机验证系统

通过验证系统测试, SpaceWire 路由器原理样机在连接各种数据源节点(如: 视频、音频、控制指令等)及各种速率(2 Mbit/s~50 Mbit/s)下均能完成数据的调度、转发、管理。路由器在满负荷通信情况下(50 Mbit/s)的丢包率为 0.21%; 同等情况下, 在引入 BWR 技术后的平均丢包率为 0.03%。由此可见, 引入 BWR 技术对减少丢包率有较为显著的效果。

4 结束语

本文在 SpaceWire 总线标准网络层分析的基础上, 结合虫孔路由原理及动态可重构技术, 提出了一种面向航天应用的路由器的设计方法。在 Simulink 环境下建立路由器仿真模型, 通过 BWR 解决了虫孔路由阻塞机制所带来的刹车问题; 并以

FPGA为平台设计了路由器原理样机,在此基础上搭建验证系统,验证了原理样机设计。

基于SpaceWire总线标准路由器的研究有助于简化有效载荷间的连接,提高有效载荷间的通信速率和通信可靠性。同时便于星上设备接口研制的模块化,提高设备的双重能力,也提高地面实验时的可维修性,从而降低开发成本,提高研发效率。星载SpaceWire路由器的研究不仅在航天领域有应用的需要,并且在放射医疗、工业探伤、矿井作业等领域也有一定的应用前景和实际价值。

参考文献

- [1] 张东伟,王宇,王善从. 星载高速数据路由技术的应用研究[J]. 微机计算信息, 2008, 24(6): 132-133.
Zhang Dong-wei, Wang Yu, and Wang Shan-cong. The research of on-board high speed routing technology[J]. *Microcomputer Information*, 2008, 24(6): 132-133.
 - [2] Parkes S M. Standard ECSS-E-50-12A, SpaceWire, links, nodes, routers and networks[S]. European Cooperation for Space Standardization, 2003.
 - [3] 万玛宁. 特殊环境高速总线技术的研究与实现[D]. [硕士学位论文], 首都师范大学, 2007.
Wan Ma-ning. Research and realize of high-speed bus technology in special environment[D]. [MA. dissertation], Capital Normal University, 2007.
 - [4] 计辉,冯萍,吴延昌,张志芬. 一种基于Wormhole Routing的寻径技术[J]. 计算机应用于研究, 2005(5): 233-234.
Ji Hui, Feng Ping, Wu Yan-chang, and Zhang Zhi-fen. A wormhole routing-based routing technology[J]. *Application Research of Computers*, 2005(5): 233-234.
 - [5] 艾真. 基于黑总线传输技术的虫洞路由设计[D]. [硕士学位论文], 电子科技大学, 2007.
Ai Zhen. Design of wormhole routing based on black-bus[D]. [MA. dissertation], University of Electronic Science and Technology of China, 2007.
 - [6] Parkes S, McClements C, Kempf G, Fischer S, and Leon A. Spacewire router[C]. International SpaceWire Seminar (ISWS 2003), Noordwijk, The Netherlands, 2003, 4: 125-132.
 - [7] Fang Tsun-wei. Integrated IPv6 and MPLS data plane on high performance network processor platform[D]. [MA. dissertation], Chung Cheng University, 2005.
 - [8] Parkes S M, *et al.* Remote memory access protocol[S]. ECSS-E50-11 draft F, 2006, 12.
 - [9] 安学军,祝明发,高文学,吴冬冬. 虫洞路由芯片的伸缩缓冲区设计与实现[J]. 计算机工程与应用, 2002, 7: 17-19.
An Xue-jun, Zhu Ming-fa, Gao Wen-xue, and Wu Dong-dong. The design and implementation of the elastic buffer in wormhole routing chips[J]. *Computer Engineering and Applications*, 2002, 7: 17-19.
 - [10] 肖小强,姜玉琴,金士尧,何鸿君. BWR——带缓冲的虫孔路由技术[J]. 计算机学报, 2001, 24(1): 78-83.
Xiao Xiao-qiang, Jiang Yu-qin, Jin Shi-yao, and He Hong-jun. BWR—buffered wormhole routing switching[J]. *Chinese Journal of Computer*, 2001, 24(1): 78-83.
- 毛春静: 男, 1983年生, 硕士生, 研究方向为智能信息处理与嵌入式系统设计.
- 关永: 男, 1966年生, 教授, 研究方向为智能信息处理与嵌入式系统设计.
- David Jungwirth: 男, 1984年生, 研究方向为医学与计算机科学.