

基于光纤的 SpaceWire 节点设计

秦 炜¹, 冯凯强²

(1. 上海创景信息科技股份有限公司, 上海 200000;

2. 中北大学 电子测试技术国防科技重点实验室, 太原 030051)

摘要: 为实现航天器各 SpaceWire 仿真子系统分布式部署的能力, 并且降低仿真节点使用国外收发芯片的高昂成本, 提出一种实现分布式 SpaceWire 仿真节点卡的设计方法, 仿真节点卡能够实现 SpaceWire 节点间 200 Mbps 链路速率下的数据收发功能, 并且通过使用多模光模块实现 SpaceWire 节点 2 000 m 远距离传输, 同时实现了多个 SpaceWire 仿真节点的分布式部署, 利用 FPGA 可编程技术优势, 以 SpaceWire IP 核为核心, 降低了 SpaceWire 仿真节点的成本; 通过仿真软件结合光纤采集卡对仿真节点卡进行了节点功能及光纤传输功能的验证; 经实验测试仿真节点卡链路 200 Mbps 速率连接正常, 数据收发稳定可靠, 实现了分布式部署的能力、并大大降低了 SpaceWire 使用成本。

关键词: SpaceWire; IP 核; 光纤; 仿真节点

Design of SpaceWire Nodes Based on Fiber Optics

QIN Wei¹, FENG Kaiqiang²

(1. Shanghai Vision Microsystems Co., Ltd., Shanghai 200000, China;

2. Science and Technology Electronic Test & Measurement Laboratory, North University of China,
Taiyuan 030051, China)

Abstract: In order to achieve the ability of distributed deployment of various SpaceWire simulation subsystems in spacecraft, and reduce the high cost of foreign transceiver chips for simulation nodes, a design for implementing a distributed SpaceWire simulation node card is proposed. With a link rate of 200 Mbps, the simulation node card can achieve data transmission and reception between SpaceWire nodes, and achieve long-distance transmission of SpaceWire nodes at a distance of 2 000 m through multi-mode optical modules. Meanwhile, the distributed deployment of multiple SpaceWire simulation nodes is also achieved. By utilizing the advantages of FPGA programmable technology and SpaceWire IP core as the core, the cost of SpaceWire simulation nodes is reduced; The node and fiber transmission functions of the simulation node card are verified by combining simulation software with fiber optic acquisition card; After experimental testing, the results show that with a speed of 200 Mbps, the simulated node card link is connected normally, and the data transmission and reception are stable and reliable, achieving distributed deployment and greatly reducing the cost of SpaceWire usage.

Keywords: Spacewire; ip core; optical fiber; simulation node

0 引言

随着宇航事业的快速发展, 航天器电子系统各功能模块对分布式和网络化的需求日益加剧。航天器内部各模块间的通信总线互联及数据交互的重要性尤为突出。鉴于宇航器执行任务呈现多种类及任务繁重的特点, 航天器内部处理器系统的运算量不断增大, 进而导致系统中各个模块节点间的通信数据量持续增长。相比传统的通信总线, 比如 1553B 通信总线技术, 其总线传输率低 (1 Mbps), 并且支持的节点数量有限 (单条总线支持 32

个终端), 如采用传统总线会成为制约航天器系统性能进一步提高的瓶颈^[1]。因此, SpaceWire 宇航总线的诞生克服了传统总线传输速率以及节点拓展受限的不足。

SpaceWire 在 20 世纪 90 年代后期在欧空局的主持下开发, 于 2003 年标准化, 现由 SpaceWire 联盟维护^[2]。SpaceWire 是一种用于空间应用数据通信的协议, 具有高速、点对点、全双工的特点, 可用于各种天基系统及地面模拟仿真, 例如航天器测试和模拟设备。开发 SpaceWire 节点仿真模拟设备, 有助于实现卫星航天器地面功能的快速组装、仿真验证、快速测试, 进而

收稿日期: 2024-06-20; 修回日期: 2025-02-08。

作者简介: 秦 炜 (1973-), 男, 硕士。

引用格式: 秦 炜, 冯凯强. 基于光纤的 SpaceWire 节点设计[J]. 计算机测量与控制, 2025, 33(3): 287-294, 329.

提高开发效率，缩短开发周期，更快地推进航天事业的发展。因此本文根据 SpaceWire 仿真节点自主可控及分布式部署的需求，提出了一种基于光纤传输的 SpaceWire 仿真节点的设计方法。

1 SpaceWire 仿真节点的硬件设计

图 1 为仿真节点的硬件系统结构框图，系统由 7 个模块构成，主要包括：LVDS 接口模块、FPGA 模块、DDR3 模块、Flash 模块、SFP 光纤模块、时钟/复位模块以及供电模块组成。其中 LVDS 接口模块为 SpaceWire 总线的物理接口；FPGA 模块为整个系统的核心，其内部逻辑运行 SpaceWire IP 核以及光纤收发逻辑；DDR3 模块为数据的高速缓存区；Flash 模块为 FPGA 的程序存储区；SFP 光纤模块为光纤收发的物理接口；时钟及复位模块为 FPGA 模块提供系统时钟及系统复位；供电模块负责为整个系统提供供电。

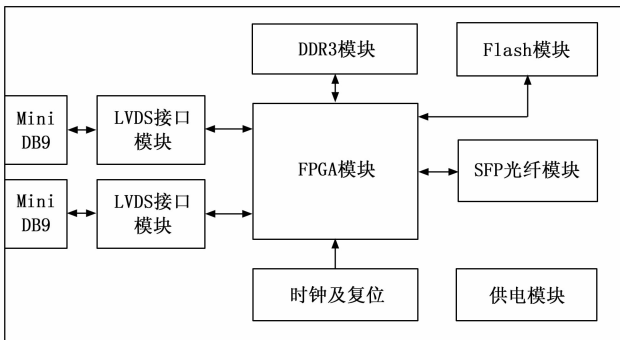


图 1 仿真节点硬件系统结构框图

1.1 仿真节点接口设计

SpaceWire 总线具有以下优点，总线支持全双工双向通信，可进行点对点或者通过路由器进行网络拓扑的数据传输。欧空局制定的协议标准中规定其支持的通信速率最高可到 400 Mbps；可支持的最大传输距离为 10 m；标准中规定了物理层面采用电压为 350 mV 的 LVDS^[3] 技术对信号进行传输，来满足总线高传输能力，标准规定了总线的编解码为 DS 编码^[4] 在降低时钟的情况下保障信号高传输及数据的稳定性。为实现接口数据收发的高速率、稳定可靠，同时避免输入 LVDS 信号对本系统造成的电气干扰，传输接口采用电气隔离的 LVDS 收发器。基于以上传输特性本设计中选用 ADI 公司的 ADN4651BRSZ 芯片^[5] 作为 SpaceWire 接口收发器，该芯片具有 5 kVrms 的 LVDS 电气隔离，最高支持 600 Mbps 的收发速率，能够满足接口链路最高 200 Mbps 的速率要求。

图 2 为两通路 SpaceWire 节点设计，由于选用了隔离 LVDS 收发器，仿真节点卡具有与外部设备电气隔离的特点，不会对外部设备造成干扰，同时不会受外部设备及环境对自身的干扰，提高了仿真节点卡的稳定性

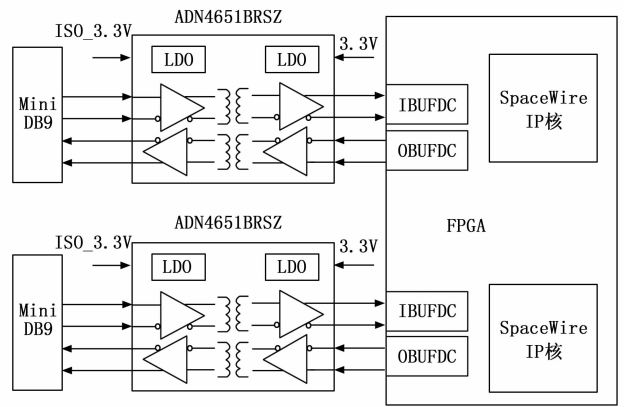


图 2 LVDS 接口设计

及可靠性。

1.2 仿真节点 FPGA 设计

FPGA 作为整个仿真节点的主控制器，其需要运行 SpaceWire IP 核逻辑，并能够满足 200 Mbps 的收发速率，同时还要支持光纤的高速传输。基于以上需求选用 Xilinx 的 Spartan6 系列的 XC6LX45T 芯片作为控制核心^[6]，其具有 4 通路的 GTP 高速收发器^[7]，能够满足光纤 2.5 Gbps 的光纤传输速率^[8]。同时该芯片内部集成 DDR3 控制器能够提供大数据的缓存，进一步提高数帧的收发性能。芯片内部具有 43 K 的逻辑资源，并具有 2 Mbit 的内部 SRAM，芯片内部时钟支持最高 375 MHz，支持发送模块的 200 Mbps 的发送速率。

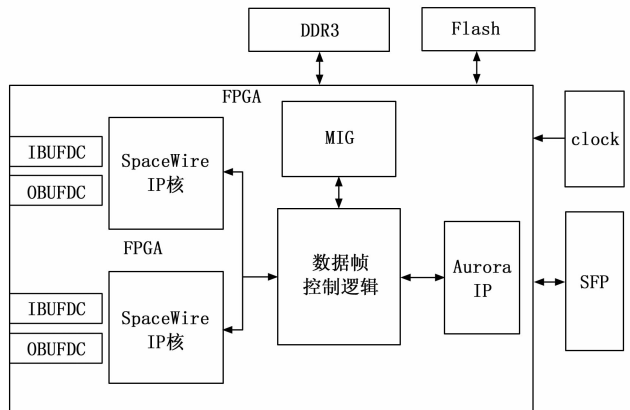


图 3 FPGA 硬件设计

FPGA 硬件逻辑包括：2 通路 SpaceWire IP 核，MIG（DDR 控制器）、数据帧控制逻辑、光纤 IP 核。其中 SpaceWire IP 核负责总线的 DS 编解码，将要发送的数据进行 DS 编码并通过串行总线发送出去，将串行总线输入的数据进行 DS 解码并将解码的数据和控制字发送给数据帧控制逻辑；MIG（DDR 控制器）负责对 DDR 芯片的读写操作，包括 SpaceWire 收发数据缓冲、光纤通道的数据收发缓冲；光纤 IP 核负责控制光模块与外部光纤接口的 Link 及仿真数据收发；数据帧控制

逻辑负责对 SpaceWire 数据收发及缓存, 对光纤数据的帧数据收发及缓存、监控数据的发送缓存、光纤数据帧的解析及编码以及对 DDR 控制器的数据存取。

1.3 光纤接口设计

光纤接口设计采用 FPGA 的 GTP 接口连接多模光纤模块实现, 光纤模块选用思科的 SFP-OC48-SR 多模光电转换收发器, 该收发器最高支持 2.5 Gbps 的传输速率, 传输距离 2 千米, 能够满足光纤 IP 核的 2.5 Gbps 收发速率以及远距离传输的技术要求。

1.4 仿真节点电源设计

仿真节点卡支持 SpaceWire 接口光电隔离, 因此整个板卡除了给 FPGA 主控制器供电之外, 还需要一个隔离电源给 LVDS 收发器供电。其中 FPGA 的供电包括: 内核 VCCINT 供电、VCCAUX 辅助供电、VCCO 接口供电、MGTA VCC 供电等, 图 4 为整个仿真节点卡的供电架构。主要包括 6 种电源, 分别为 1.2 V 为内核供电、1.5 V 为 DDR3 供电、0.75 V 为 DDR3 供电、3.3 V 为 IO 供电、1.2 V 为 GTP 供电、3.3 V 为 LVDS 接口隔离供电。

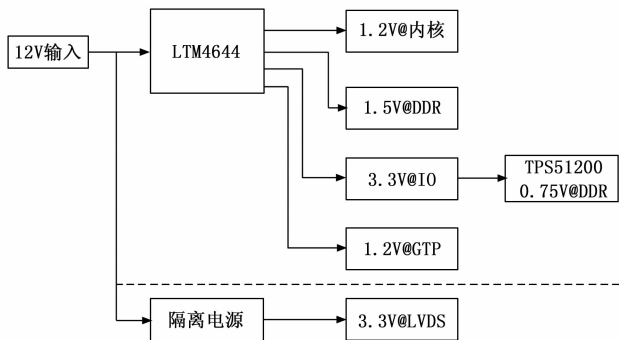


图4 系统电源设计

图4为整个仿真节点板的电源架构, 采用ADI公司的供电芯片LTM4644作为主供电芯片^[9], 该芯片支持4通路的电源输出, 每通路支持4A持续电流, 能够满足FPGA的供电要求。采用TI公司的TPS51200芯片为DDR3提供0.75V的VTT及REF供电。采用金升阳的VRB1203S-6WR3隔离电源作为LVDS输出供电, 该电源输出3.3V, 最大输出功率为6W, 能够满足LVDS接口的供电要求。整个仿真节点卡的上电顺序为: 1.2V电源的FPGA内核供电优先供电, 其次为1.2V的GTP内核供电, 然后是1.5V为DDR及FPGA的IO接口供电, 3.3V为FPGA的接口供电以及TPS51200的输入电源, 0.75V作为DDR3的参考供电, 最后为隔离3.3V作为LVDS芯片的输出供电^[10]。

2 SpaceWire IP核设计

2.1 SpaceWire 核心逻辑框图

图5为SpaceWire IP核整体架构^[11], 作为SpaceWire

总线数据的编解码及链路控制核心, 包括5个逻辑部分: 时钟逻辑、time code逻辑^[12]、link状态机逻辑、发送逻辑、接收逻辑。其中link状态机控制是整个IP核的关键, 控制着SpaceWire总线链路层的通信建立^[13]。

其中, 时钟模块负责整个逻辑的时钟驱动, 包括Time Code的时间码, 时钟模块具有3种时钟频率1、100及200 MHz, 其中1 MHz的时钟作为Link的初始时钟, 100 MHz作为整个逻辑的系统时钟, 200 MHz作为SpaceWire串行发送逻辑的时钟以及接收逻辑的时钟; Time Code逻辑负责产生时间码, 发送给发送逻辑; 发送逻辑负责将发送数据的DS编码以及流控信用值的维护, 其包括3个部分: Credit Counter (信用值)逻辑、发送数据缓存逻辑以及发送逻辑控制; 接收逻辑负责将串行数据进行DS解码, 将串行数据转换为数据字及控制字, 并存入缓存, 其包括两个部分: 接收逻辑控制、数据缓存逻辑; Link控制负责整个链路层的控制, 包括初始Link、接收和发送控制、错误检测以及时间控制逻辑, Link控制逻辑是整个系统的核心, 是整个IP核实现的关键。

IP核的信号说明由图5所示, 包括: Spw_Din和Spw_Sin为SpaceWire的输入信号; Spw_Dout和Spw_Sout为SpaceWire的输出信号; Data in发送的数据信号; Data out接收到的数据信号; TimeCode数据信号; 状态机控制信号以及故障注入信号。

Data in发送的数据信号包括:

1) tx_ready: 输出信号, 低电平有效, 有效时外界可以写数据到fifo中;

2) tx_data [7: 0]: 输入数据总线;

3) tx_write: 输入信号, 低电平有效为写信号。

Data out总线接收的数据信号包括:

1) rx_ready, 输出信号, 低电平有效, 有效时外界可以读fifo中的数据;

2) rx_data [8: 0]: 输入数据总线。当bit8 = 0时候, bit [7: 0]就是数据。当bit8 = 0时候, bit [7: 0]代表的是EOP, 反之代表的是EEP。EOP: bit [7: 0] 011P xxxx; EEP: bit [7: 0] 101P xxxx;

3) rx_read: 输入信号, 低电平有效, 读信号。

TimeCode数据信号包括:

1) tick_in: 输入信号, 低电平有效。当tick_in有效时马上会发送time code;

2) tick out: 输出信号, 低电平有效, 指示IP核有收到time code。

3) time out [7: 0]: 收到的time code;

状态机控制信号包括:

1) reset: 高电平有效系统的外部复位信号;

2) link_start: 控制SpaceWire总线连接启动信号;

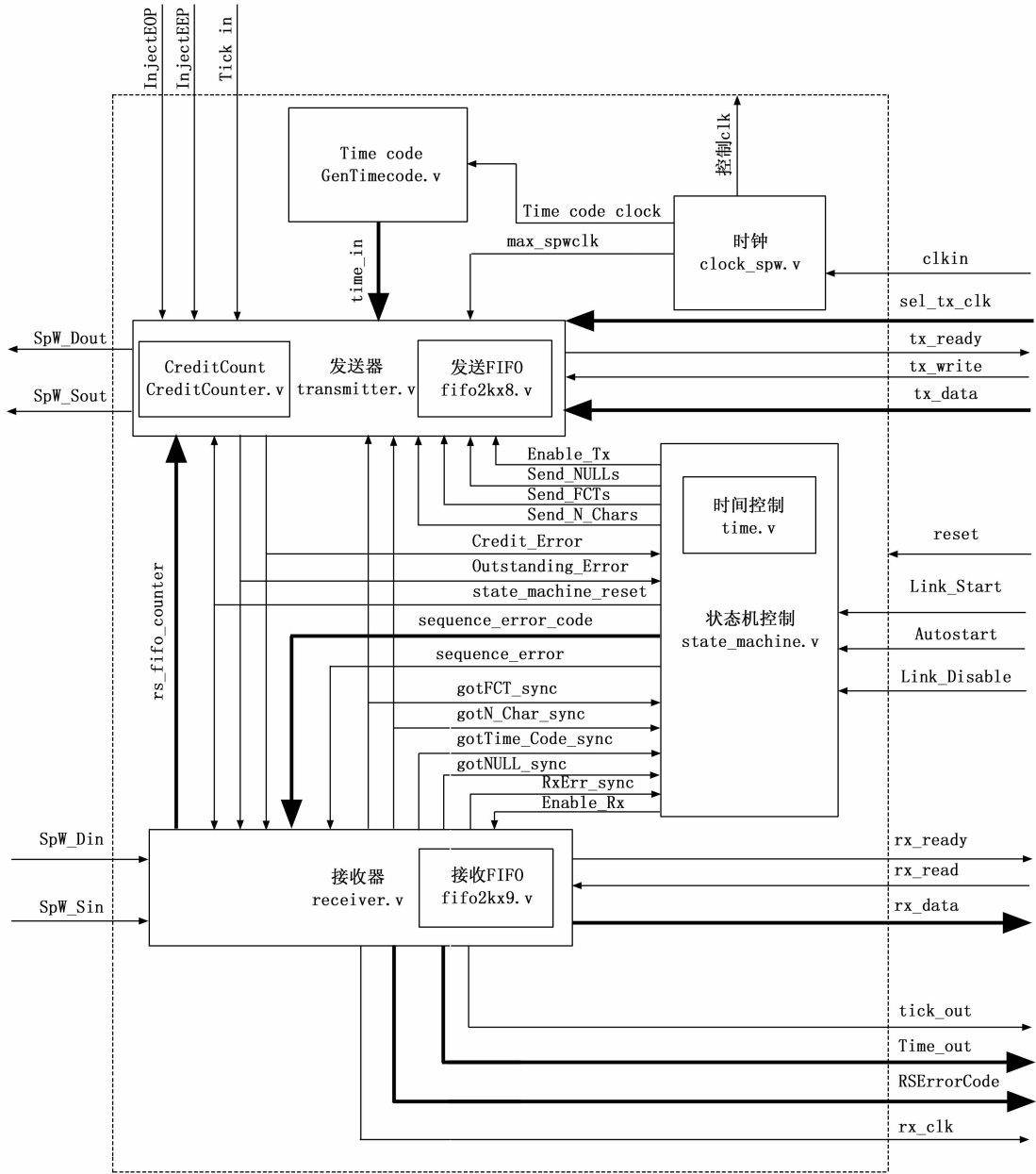


图 5 SpaceWire IP 核架构

3) Link_Disable: 控制 SpaceWire 总线停止链接的控制信号;

4) AutoStart: 控制 SpaceWire 总线的自动启动信号。

故障注入信号包括:

1) InjectEEP: 插入 EEP 信号, 高电平有效, 该信号为正常数据帧中插入帧错误结尾, 用于测试使用。

InjectEOP: 插入 EOP 信号, 高电平有效, 该信号为正常数据帧中插入帧结尾, 用于测试使用。

2.2 Link 状态机设计

链路层的 Link 状态机^[14]控制整个 SpaceWire IP 核的正常工作, 其控制链路接口模块的发送状态机以及接收状态机的工作状态。如果接收模块发生了接收错误,

Link 状态机会对整个接口模块进行控制, 并对发送状态机以及接收状态机进行链路复位及恢复。Link 状态机定义的状态转换如图 6 所示^[15]。其包括 6 个状态分别为: 错误复位态 (Error Reset)、错误等待态 (Error Wait)、待命态 (Ready)、起始态 (Started)、连接态 (Connecting) 及运行态 (Run)。

其中, 错误复位状态负责接收外部系统复位, 进行发送及接收的复位, 当进入错误复位状态 6.4 μs 后进入错误等待态; 错误等待态负责使能接收逻辑, 复位发送逻辑, 当在此状态下等待 12.8 μs 后进入准备态, 此状态期间如果收到接收错误、FCT (流控)、Nchar (字符)、Time Code (时间码) 会进入错误复位状态; 当进入准备态后继续使能接收逻辑并且复位发送逻辑, 并且

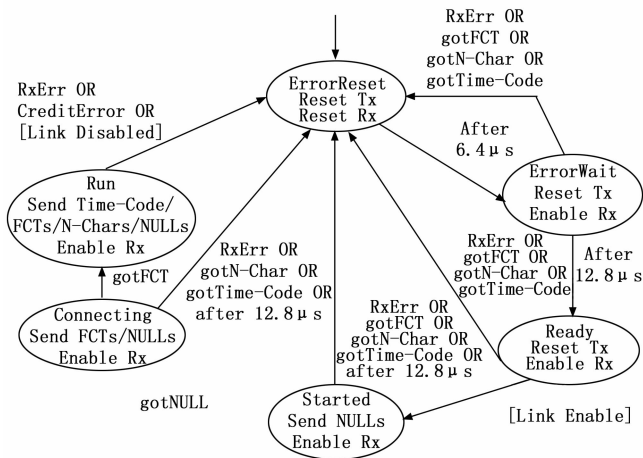


图6 Link状态机设计

接收外部链接启动命令, 启动 $12.8 \mu s$ 定时器后进入启动态, 在此状态期间接收到接收错误、FCT、Nchar、Time Code 都会进入错误复位状态; 进入启动态后, 使能发送 Null、使能接收, 在该状态下当接收到 Null 时会进入链接态, 如果在此状态期间接收到错误、FCT、Nchar、Time Code 以及 $12.8 \mu s$ 超时会进入错误复位状态; 进入链接态后使能发送 FCT (流控)、Null 及使能 $12.8 \mu s$ 定时器, 在此状态下如果接收到 FCT 后进入运行状态, 相反如果在此状态下接收到错误、Nchar、Time Code 以及 $12.8 \mu s$ 超时会进入错误复位状态; 当进入运行状态后, 会使能 Time Code、FCT、Nchar、Null 的发送并且使能接收逻辑, 此时该状态下能够正常地进行收发操作, 包括接收发送时间码、流控、Nchar 及 Null。在此状态期间如果接收到接收错误、信用值错误以及外部的链接关闭命令则将进入错误复位状态。以上为整个状态机的状态流转, 发送逻辑以及接收逻辑接受 Link 状态机的控制, 同时接收逻辑又将接收状态反馈给 Link 状态机, 进而维护整个状态机的正常运转。

2.3 发送状态机设计

发送状态机是用来控制 SpaceWire 节点发送数据、FCT 控制字、时间码的逻辑, 受控于 Link 状态机, 当 Link 状态机处于发送使能时才能进入发送的状态转换。如果链路发生了错误, 接收到来自 Link 状态机的复位信号, 则进入复位状态。设计中定义模块的状态转换图如图 7 所示。

发送状态机由 4 个状态组成包括: 发送复位态 (Tx Resrt)、发送 Null 状态 (Tx Null)、发送 FCT 及 NULL 状态 (Tx Null FCT)、发送所有字状态 (Tx ALL)。

发送复位态是由系统复位结束后会进入该状态, 当该状态下, 来自 LINK 状态机的使能发送并且发送 Null 命令时, 会进入发送 Null 态; 在发送 Null 态下, 会向

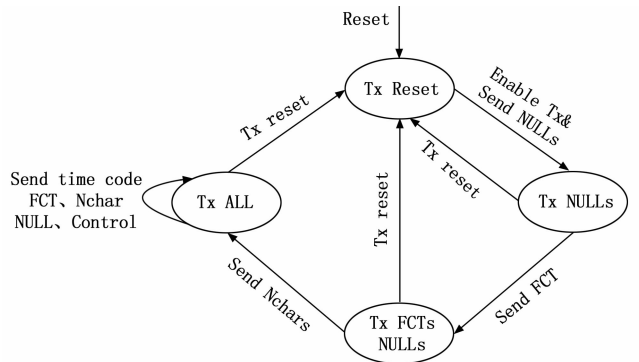


图7 发送状态机设计

接口发出发送 NULL 码, 如果在该状态下接收到 LINK 状态机发送 FCT 命令后将进入发送 FCT Null 态, 该状态下如果接收到 LINK 状态机的复位命令则进入发送复位态; 在发送 FCT Null 态下, 会向接口发送 FCT 流控, 在该状态下如果收到 LINK 状态机发出的发送 Nchar 命令, 则进入发送 ALL 态, 如果收到 LINK 状态机发出的复位命令, 则进入发送复位态; 在发送 ALL 态下, SpaceWire 发送模块可以进行 Time Code、FCT、Nchar 以及 NULL 的发送功能, 如果在该状态下接收到 LINK 状态机发出的发送复位命令, 则会进入发送复位态, 否则会一直在该状态下。

2.4 接收状态机设计

接收状态机是用来控制 SpaceWire 节点接收数据、控制字、时间码的逻辑, 受控于 Link 状态机, 当 Link 状态机处于接收使能时, 才能进入接收的状态转换^[17]。如果链路发生了错误, 接收到来自 Link 状态机的复位信号, 则进入复位状态。设计中定义模块的状态转换图如图 8 所示。

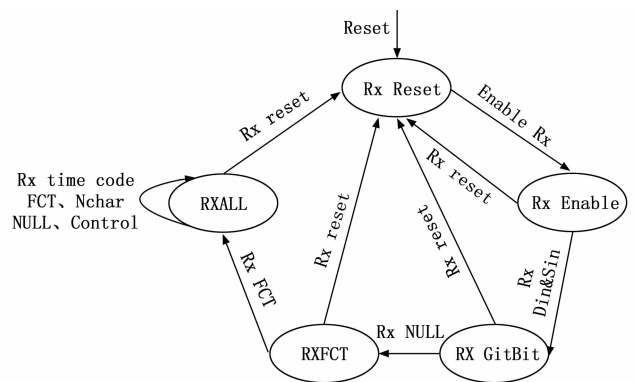


图8 接收状态机设计

接收状态机由 5 个状态组成, 包括: 接收复位态 (Rx Reset)、接收使能态 (Rx Enable)、接收 Bit 态 (Rx GitBit)、接收 FCT 态 (Rx FCT) 以及接收 ALL 态 (Rx ALL)。

接收复位态是由系统复位后进入该状态, 该状态下, 如果接收到 LINK 状态机的接收使能命令, 则进入

接收使能态，否则一直在接收复位态；在接收使能态下，如果接收 DS 编码则进入接收 Bit 态，如果接收到 LINK 状态机发出的接收复位命令，则进入接收复位态；在接收 Bit 态下，可以接收 Null，如果在该状态下接收到 NULL，则进入接收 FCT 态，如果接收到 LINK 状态机发出的接收复位命令则进入接收复位态；在接收 FCT 态，可以接收 FCT，在该状态下如果接收到 FCT，则进入接收 ALL 态，如果接收到 LINK 状态机发出的接收复位命令则进入接收复位态；在接收 ALL 态下，可以接收 Time Code、FCT、Nchar、NULL，如果没有接收到 LINK 状态机发出的接收复位则一直在该状态下，如果接收到 LINK 状态机发出的接收复位则进入接收复位态，等待 LINK 状态机的发出命令。

2.5 监控逻辑设计

监控逻辑主要负责对链路数据的统计，将统计数据组织成数据统计帧，并发送给光纤的发送模块。Speed 信号为链路上的连接速率；Credit counter 为信用值；Outstd counter 为信用余量；GotTimeCode 信号为 SpaceWire IP 核接收到的 Time Code；GotData 信号为 SpaceWire IP 核接收到 Nchar 指示；GotFCT 信号为 SpaceWire IP 核接收到的流控指示；GotNULL 信号为 SpaceWire IP 核接收到空指示；GotEOP 信号为 SpaceWire IP 核接收到帧结尾指示；GotEEP 信号为 SpaceWire IP 核接收到帧错误结尾指示；GotEscEOP 信号为 SpaceWire IP 核接收到的退出帧结尾指示；GotEscEEP 信号为 SpaceWire IP 核接收到的退出错误帧结尾指示；GotDisconnect 信号为 SpaceWire IP 核产生的断连指示；GotParityError 信号为 SpaceWire IP 核产生的奇偶校验错误指示。

其中状态统计逻辑的功能是统计出链路上各个关键参数，包括：链路速率、信用值、时间码信息、数据计数、流控计数、NULL 计数、帧计数、真错误计数、重连计数及奇偶校验错误，如图 9 所示。其中链路速率为当链路上有数据收发时的有效数据载荷；信用值为当前状态下接收端可接收的 Nchar 值。帧包组织上传逻辑负责对统计的状态信息组织成帧的格式发送给光纤的发送逻辑，通过光纤发送给上位机，并将统计信息在上位机软件上呈现，其呈现窗口为软件的 SpaceWire info 窗口。

2.6 光纤收发设计

光纤的收发逻辑如图 10 所示，主要包括 3 个部分：光纤 IP 逻辑^[18]、MIG 逻辑、收发控制逻辑。

其中 MIG 逻辑为控制 DDR3 数据写入以及读取的逻辑，其作为收发数据缓存，包括 SpaceWire 的收发数据、监控逻辑的发送数据以及光纤逻辑的收发数。当数据进行下发流程时，光纤 IP 核接收到来自上位机的有效数据后，通过控制逻辑将有效数据缓存至 DDR3，并

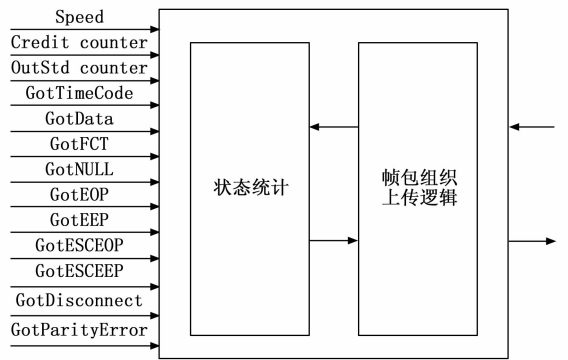


图 9 监控模块逻辑设计

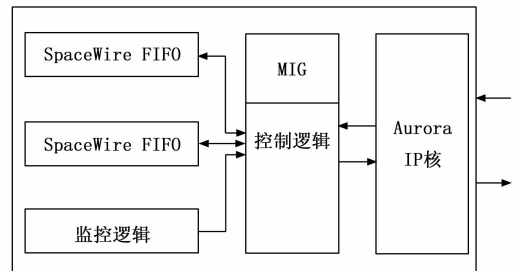


图 10 光纤收发逻辑设计

进行数据帧标识，当控制逻辑空闲时检测到缓存数据帧时，会将帧读出并解析，发送给相应 SpaceWire 的发送 FIFO 数据缓存；当数据进行上传流程时，SpaceWire 的接收 FIFO 有数据时，则控制逻辑会将数据从 SpaceWire 接收 FIFO 中读出并组织成帧，写入 DDR3 的光纤发送缓存区，并标记帧计数，如果光纤的发送逻辑空闲，则将发送缓存区的数据读出，发送给光纤发送模块；同样当监控逻辑有数据时，则控制逻辑会将监控数据读出并组织成帧，写入 DDR3 的光纤发送缓存区，并标记帧计数，如果光纤的发送逻辑空闲，则将发送缓存区的数据读出，并发送给光纤发送模块。

3 软件设计

3.1 PCIE-光纤驱动设计

PCIE-光纤卡为 PCIE 2.0 支持 x4 lane 最高速率为 20 Gbps，同时支持 4 通路的光纤接口，接口速率 2.5 Gbps。为满足 SpaceWire 光纤链路上有效载荷数据传输 200 Mb。

PCIE 光纤卡的驱动采用如图 11 架构的驱动设计。

数据的传输由驱动的用户接收接口和 DMA engine 模块来实现。DMA 接收缓存收集上位机来得有效数据，收集完成后由 DMA 控制器启动一次 DMA 传输，将缓存区的数据打包发给 PCI express 端点。DMA 的发送缓存通过用户发送接口将 DMA 发送过来的数据，上传给应用层。

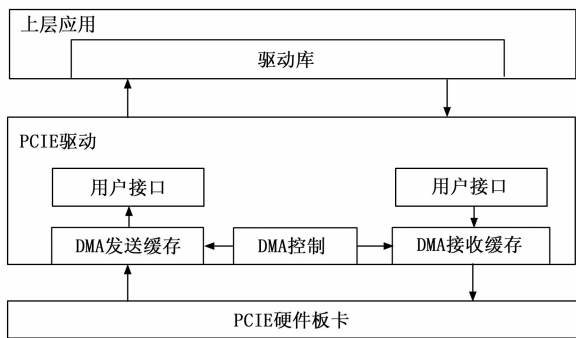


图 11 驱动架构图

应用启动接收函数后，驱动将申请一块连续内存作为接收缓存区，并将内存的地址以及内存深度记录。用户应用程序可通过 DeviceIoControl 将缓冲区地址和数据长度下传给硬件板卡。当硬件板卡有足够的足够的数据时，会启动一次 DMA 的发送，发送的地址和数据长度为应用程序下发的参数。当 DMA 发送全部的数据到缓存区后，产生一次发送中断通知驱动，驱动程序响应中断并通知应用程序，一次 DMA 操作完成，可对缓存区的数据进行读取。

应用启动发送函数后，驱动将申请一块连续内存作为发送缓存区，应用程序通过 DeviceIoControl 函数将缓冲区地址和要写入的数据长度下传给硬件板卡，之后应用程序将要发送的数据写入缓存，并通知硬件板卡可以启动 DMA 读操作，当硬件板卡 DMA 读操作完毕后，产生中断给驱动，至此一次 DMA 读操作完成，应用程序可进行下一次的 DMA 读操作。

3.2 SpaceWire 节点仿真测试软件

SpaceWire 节点仿真测试软件为 SpaceWire 定制的自动化测试软件，如图 12 所示，通过该软件结合实际硬件设备可以实现 SpaceWire 真实数据的收发、存储和分析功能。

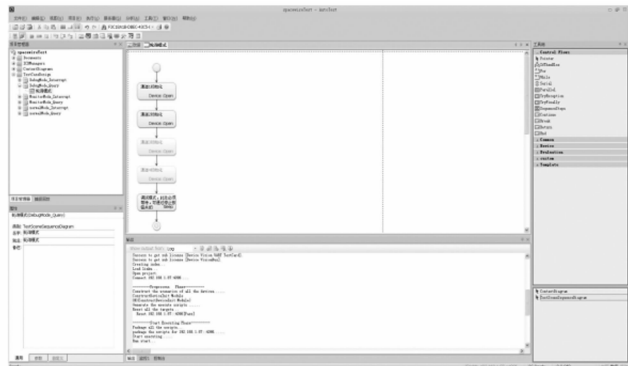


图 12 节点仿真测试软件

该软件采用图形拖拉方式来快速搭建测试场景图和自动化测试环境，主要特点有：条目化用户的测试需求，软件提供了结构化文档的编辑界面，在文档中用户

以条目化的形式来管理所有的测试需求，设置需求点与测试场景图各元素的关联，覆盖所有的测试需求；灵活的测试场景设计器，软件提供了非常灵活的场景图编辑界面，用户通过拖拉图形的方式就能设计出支持用户特定设备的测试场景图；支持场景图运行调试，工具提供了场景图的调试功能，通过在场景图中设置断点控制场景图的运行和多设备的数据收发节奏。同时软件工具提供了详细的日志输出，通过界面用户配置日志输出信息、设备端口的输入输出等信息，方便系统调试及问题定位。

图 13 为节点仿真软件工作流程图，启动软件后扫描仿真节点设备，当扫描出节点硬件设备后打开设备，当设备正确打开后可以设置仿真节点设备的发送参数以及接收参数。设置发送线程以及发送数据，设置接收线程以及接收数据的缓存区。启动接收及发送命令，并且等待发送结束，完成整个发送及接收流程。

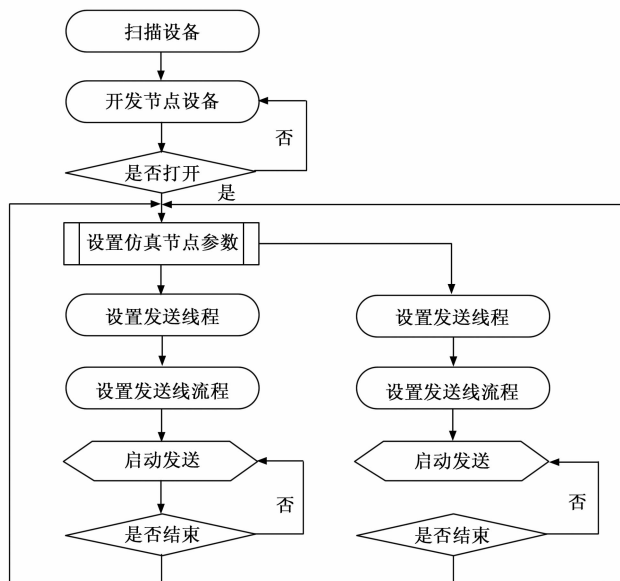


图 13 软件实现流程图

4 节点功能验证

验证环境搭建包括 SpaceWire 仿真节点卡、PCIE-光纤收发卡^[19]、工业控制计算机以及 SpaceWire 节点仿真测试软件。工控机安装 SpaceWire 节点仿真测试软件，实现对仿真节点的数据模拟以及控制。连接关系为：PCIE-光纤卡插入工控机的 PCIE 插槽内；仿真节点卡的 SpaceWire 接口通过专用串行线缆点对点相互连接；SpaceWire 节点卡的光纤接口通过光纤模块和光纤线缆和 PCIE-光纤卡的光纤接口连接。

本文的测试内容包括：链路 LINK 的测试、仿真节点卡数据收发测试，以及监控数据的测试。其中链路 LINK 的测试用于验证 LINK 的状态机的功能是否正确；仿真节点卡数据收发测试用于验证 LINK 状态机、

发送逻辑以及接收逻辑是否正确；监控数据的测试用于验证监控逻辑的功能是否正确。

本文以 2 通道 SpaceWire 节点作为测试对象，首先验证链路 LIKN 功能：第一步启动工控机，并启动仿真测试软件，新建测试用例，将 SpaceWire 仿真节点卡，按照上文介绍的连接方式连接好后对仿真节点卡上电，会发现仿真节点卡的 link 指示灯点亮，然后打开测试软件的监控显示界面，可以看出两个仿真节点之间已经 link 上，并且链接速率为 200 Mbps，链路速率稳定，没有重连现象。

其次验证两个仿真节点的数据收发功能：通过仿真测试软件新建数据收发的测试用例，设置仿真节点的发送数据大小为 1 Mb，设置仿真节点的发送速率为 200 Mbps。当两个节点之间建立正常的通信链接，并且链接速率为 200 Mbps，如图 14 所示。在 200 Mb/s 的通信速率下，启动仿真节点的收发。测试结果表明，当上位机下发数据包大小为 1 Mb 时，SpaceWire 链路总线有效载荷达到 18 Mb/s，已经接近极限链路速率值 200 Mb/s。同时在 SpaceWire info 窗口可以看出监控信息，包括链路速率、信用值、时间码信息、数据计数、流控计数、NULL 计数、帧计数、帧错误计数、重连计数及奇偶校验错误等。

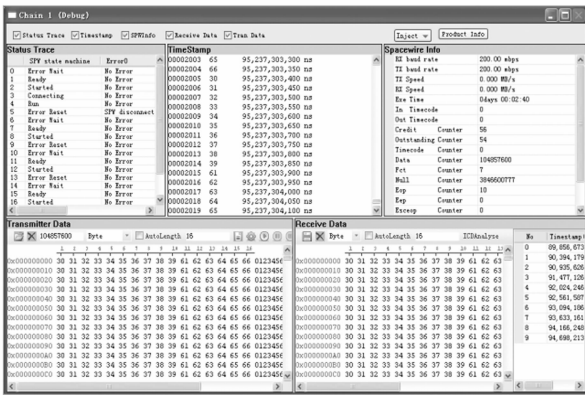


图 14 测试结果图

由以上测试结果可以看出本文通过 IP 核实现 SpaceWire 链路通信以及通过光纤方式实现仿真节点分布式部署完全可以满足仿真节点正常工作。

5 结束语

实验结果表明，本仿真节点设计完全可以支持两通道的 SpaceWire 节点仿真测试功能，并且节点在 200 Mbps 链路传输速率下数据收发传输稳定可靠，本文的设计符合 SpaceWire 联盟的 ECSS-E-ST-50-12C 设计规范，能够做到替换国外相同功能的芯片，大大降低了使用成本。同时该设计提供了总线链路监控功能，方便在仿真测试验证时进行问题定位以及性能分析。并且

该设计借助光纤传输的优点，实现了远距离分布式部署特殊应用需要。后期可在本设计的基础上增加 RMAP^[20] 通信协议的支持，能够更好地满足仿真节点的多功能应用场景。

参考文献：

[1] 乔立岩, 陈利彬, 彭喜元. 基于 IP 核的 SpaceWire-PCI 通信卡设计 [J]. 电子测量与仪器学报, 2010, 10 (24): 918-923.

[2] 赵光权, 毛宁元, 梁 军, 等. 面向 RMAP 协议的 SpaceWire 节点 IP 核设计 [J]. 计算机测量与控制, 2012, 20 (12): 3325-3328.

[3] Telecommunications Industry Association. Electrical characteristics of low voltage differential signaling (LVDS) interface circuits [S]. U. S. A, 2001: 5-9.

[4] 王 娜. SpaceWire 总线节点接口的设计 [D]. 哈尔滨: 哈尔滨工业大学, 2007.

[5] ANALOG DEVICE. Dual-Channel LVDS Isolators ADN4650/ADN4651/ADN4652 data sheet [S]. U. S. A, 2019: 17-22.

[6] XILINX. Spartan-6 FPGA Data Sheet: DC and switching characteristics [S]. U. S. A, 2011: 3-5.

[7] XILINX. Spartan-6 FPGA GTP transceiver signal integrity simulation kit user guide [S]. U. S. A, 2010: 7-9.

[8] XILINX. LogiCORETM IP 光纤 8B/10B v5.2 [S]. U. S. A, 2010: 3-7.

[9] ANALOG DEVICE. LTM4644/LTM4644-1 Quad DC/DC uModule regulator with configurable 4A Output Array [S]. U. S. A, 2018: 1-15.

[10] MORNSUN. DC/DC 模块电源 WRAS&. WRBS 系列 [S]. 广州: MORNSUN, 2022: 4-5.

[11] 李 政. 基于 SpaceWire 协议的 IP 核设计与 UVM 验证 [D]. 长沙: 湖南大学, 2019.

[12] 庄洪毅, 伊小素, 张 昊. 基于 SpaceWire 时间码备份路由实现与分析 [J]. 电子测量技术, 2016, 39 (11): 151-156.

[13] 柳 萌, 安军社, 史毅龙, 等. SpaceWire 高速总线节点控制器的设计与实现 [J]. 微电子技术, 2018, 44 (11): 1-4.

[14] 柳 萌, 安军社, 周昌义. SpaceWire 高速串行总线低信号速率建立链路的研究 [J]. 电子设计工程, 2018, 26 (18): 140-144.

[15] ECSS-E-ST-50-12C. SpaceWire-links, nodes, routers and networks [S]. European Cooperation for Space Standardization (ECSS). 2008: 124-128.

[16] 赵云富, 吴一帆, 孙 强, 等. SpaceWire 总线的流量控制机制研究 [J]. 微电子学与计算机, 2016, 33 (11): 1-5.

(下转第 329 页)