

基于 PCIe 接口的千兆 AFDX TAP 卡设计与实现

田 园¹, 李玉发¹, 蒲 凯¹, 刘李力²

(1. 中国航空计算技术研究所, 西安 710068;

2. 陕西华达科技股份有限公司, 西安 710065)

摘要:近年来, 大通流量使得机载网络对带宽的需求逐渐增长, 目前的 10/100 Mbps 传输速率已经不能满足需求, 故千兆 AFDX 技术研究急需开展; 基于千兆 AFDX 网络, 对 TAP 卡设计和实现进行了研究; 通过系统设计和详细设计对设计过程中的问题进行分析, 提出了一种基于 FPGA 的 TAP 卡设计方法; 模拟不同的网络情况对设计进行了验证; 结果表明该设计可以正确反映网络细节, 能够给网络级调试带来了极大的便利。

关键词: 千兆 AFDX; TAP; PCIe

Design and Implementation of PCIe Interface Card Based on Gigabit AFDX TAP

Tian Yuan¹, Li Yufa¹, Pu Kai¹, Liu Lili²

(1. China Aeronautical Computing Technique Research Institute, Xi'an 710119, China;

2. Shaanxi Huada Polytron Technologies Inc, Xi'an 710065, China)

Abstract: In recent years, the mass communication makes the airborne network bandwidth on demand is increasing, the 10/100Mbps transmission rate is already unable to meet the demand, so the Gigabit AFDX technology research is urgently needed. The design and implementation of TAP based on the research of Gigabit AFDX network. Through analyzing the design process of system design and detailed design analysis, put forward a design method of FPGA based TAP card. The verification of the design to simulate different network environment. The results show that the design can accurately reflect the details of the network, can bring great convenience to the network level debugging.

Keywords: Gigabit AFDX; TAP; PCIe

0 引言

传统机载网络有 ARINC429、1553B 总线, 因点对点或共享总线通信方式具有可靠性高的优点, 但通信网络传输速率只有 100 kb/s 和 1 Mb/s, 已经远远不能满足通讯速率的要求。航空全双工交换式以太网 (Avionics Full Duplex Switched Ethernet, AFDX) 作为一个确定性网络, 是标准以太网的一个子集, 传输速率 10/100 Mbps, 具有技术成熟、高可靠性、可扩充性等特点^[1]。随着航空电子技术的飞速发展, 对机载网络的数据总线通信速率要求更高, 千兆 AFDX 已经不能满足要求, 因此千兆 AFDX 网络及其相关技术的研究也已经展开。

随着机载网络的日益复杂, 网络的测试技术也变得日益重要, 在航空电子系统调试时, 通常会出现各种问题, 因此就需要一种测试设备, 在不中断网络的情况下, 获得网络上的数据。TAP 卡是一种串联在两个被监测设备之间的设备, 能够在不中断网络通讯的情况下监测两个被监测设备之间网络上的数据^[2]。当前国内使用的 AFDX 网络 TAP 卡多依赖进口 AIM 和 Techsat 两家公司, 成本较高。

本文着重研究了前兆 AFDX 网络 TAP 卡研究, 提出了一种基于 PCIe 接口的千兆 TAP 卡设计方法。从系统硬件设计、

详细设计等角度进行了详细阐述, 并通过专业测试卡对设计进行了详细验证。

1 TAP 卡概述

AFDX 网络由给主机提供网络收发功能的端系统 (end system)、交换机 (switch) 及传输链路组成, 如图 1 所示。从外界看, AFDX 网络是一个封闭的黑盒子。在进行系统间的调试时, 由于网络的庞大且复杂, 对网络的一些细节问题造成的网络故障, 以及监控网络性能、排除网络故障、分析网络流量等一系列问题。可以通过交换机的捕获口来监控网络上某一链路的传输数据, 但其功能具有一定的局限性。一是交换机的捕获口非常有限, 通常只有 1-2 个, 不能同时监视更多的链路。其次针对某个捕获口, 如果想要更改其所监控的链路, 往往需要更换交换机的配置信息, 给使用交换机的其他用户造成潜在的危险。

TAP 卡在网络中所处位置见图 1。通过将 TAP 模块串联至交换机与端系统之间, 既不影响交换机与端系统间的通信, 也可以对其间的通信进行监测。这种方式可以在不改变网络的情况下, 仅用将 TAP 卡串联至想要监控的链路上即可。

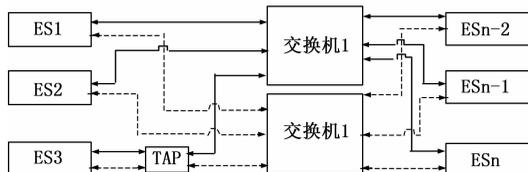


图 1 AFDX 网络结构

收稿日期:2014-03-08; 修回日期:2014-04-17。

基金项目:航空科学基金(20101931005)。

作者简介:田 园(1985-),男,陕西富平人,工程师,硕士,主要从事机载网络技术方向的研究。

因此 TAP 在设计中需要满足以下功能: 能够对网络中链路消息进行捕获及监控, 将通过 TAP 模块的数据能够被捕获并上报给主机, 并且能够反映数据帧之间在时间上的关系; 能够对收到的数据及时地转发, 不影响原有数据帧在时间上的先后关系, 并且将误差缩小到网络允许的范围。

2 详细设计

随着技术的发展, 在当前产品中, 使用大规模的现场可编程门阵列 (FPGA) 的设计越来越多, 且 FPGA 是整个设计的核心。本设计采用一片 Xilinx 公司生产的 Virtex 5 系列的 FPGA 实现其主要功能, 外围电路包括电源电路、时钟电路、复位电路、PROM 电路、双端口存储器电路、物理层接口电路。FPGA 实现的功能包括 PPC440、内部存储器、数据转发、数据上报、PCIe 接口、MAC 层接口模块、计时器模块。

FPGA 芯片功能的实现采用 PowerPC440 (PPC440) 构架, PPC440 是一种高性能的 RISC 处理器, 由 Xilinx 的 FPGA 的内部硬核实现, 可提供 PLB 总线及 AXI 总线, 本设计中选用较为常用的 PLB 总线实现, FPGA 设计框图见图 2。数据帧通过收发器, 进入高速串行 GTX 收发器, 通过串并转换后进入千兆以太网 MAC 核。数据转发模块将收到的数据复制为 2 份同时写入数据上报缓冲区和数据转发缓冲区, 同时也将需要置的标识复制为 2 份同时写入数据上报标识和数据转发标识。然后数据转发模块读取数据转发标识 FIFO 来读取数据转发缓冲区的内容进行发送操作。当数据上报标识 FIFO 非空时, 主机通过 PCIe 接口读取数据上报标识 FIFO 内的标识, 来读取数据上报缓冲区的内容完成上报主机的操作。

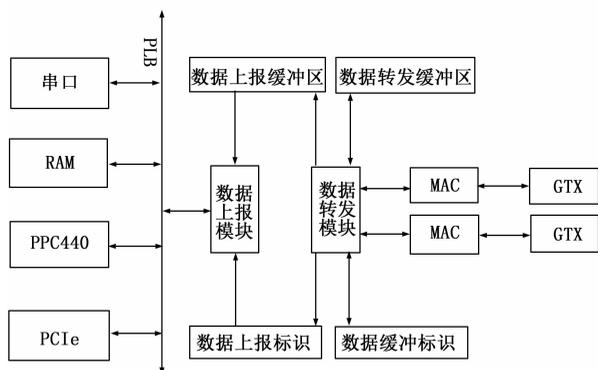


图 2 FPGA 设计框图

2.1 接口设计

千兆 AFDX 以太网物理层 (PHY 层) 和介质层 (MAC 层) 主要完成数据帧的接收和发送, 其实现构成如图 3 所示, 其中 RocketIO 收发器和光收发器两部分属于物理层, MAC 核属于介质层, SGMII 为 MAC 层与物理层之间的接口。

MAC 核和 SGMII 采用 Xilinx 公司 Virtex-5 系列 FPGA 的三速嵌入式以太网 MAC 核 (Embedded Tri-Mode Ethernet MAC) 实现。该 IP 核集成了 10/100/1 000 Mb/s 以太网 MAC 核, 符合 802.3-2002 规范, 支持全双工的 10/100/1 000 Mb/s, 半双工的 10/100 Mb/s。支持两路以太网 MAC (EMAC0/EMAC1), 支持多种可配置的物理层接口, 包括 MII、GMII、RGMII、SGMII 和 1000BASE-X PCS/PMA 接口^[3]。本设计使用该以太网 MAC 核 (Embedded Tri-Mode Ethernet MAC) 的千兆速度模式, 与物理层通过 SGMII 接口连接。SGMII 为串行 GMII 接口, 当使用 RocketIO GTX 收发器时, 它可将并行的 GMII/MII 接口转换为

串行接口, 能够显著地减少 I/O 引脚数目。

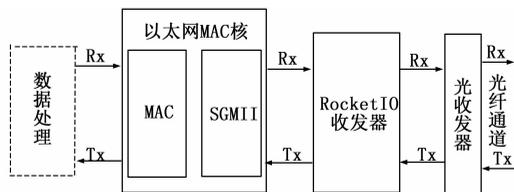


图 3 接口结构框图

RocketIO 收发器使用 Xilinx 公司的 RocketIO GTX 收发器核。该核是一种嵌入式高速串行收发器, 支持多种速率, 能够以 150 Mb/s~6.5 Gb/s 的速度运行, 支持 SATA、Aurora、千兆以太网、OBSAI、PCIE 等多种串行协议, 内部带有 50 Ω 或 75 Ω 的匹配电阻, 可以在高频环境下避免数据间的串扰。该硬核支持速率为 1.25 Gb/s 的 1000Base-X 千兆以太网。RocketIO 收发器主要包括 PMA 和 PCS 两个部分, PMA 包含串并转换器 (SERDES)、发送接收缓冲区、时钟发生和时钟恢复电路。PCS 包含 8B/10B 编解码和 CRC 校验, 集成了通道绑定和时钟修正的弹性缓冲。

光收发器使用 finisar 的 SFP (Small Form-factor Pluggable) 光收发器。这种收发器有支持热插拔, 最高支持 2.125 Gb/s 的双向数据连接等优点, 支持三种应用模式: 1.25 Gb/s 的以太网、双速 1.063/2.125 Gb/s 的 FC 通信及无线模式。设计中使用 1.25 Gb/s 的 1000Base-X 以太网模式。

2.2 数据收发设计

千兆以太网 MAC 核的用户接口负责提供给用户物理层接收到的数据和由用户提供给物理层需要发送的数据。数据收发过程需要使用 FPGA 内部双端口 RAM 和标记 FIFO 作为辅助。

当 MAC 核收到帧并向上层提交时, 逻辑通过接收状态机将数据缓存至 FPGA 内部的双端口 RAM 中, 在整个接收过程完成之后将帧在双端口 RAM 中的起始地址和长度写到标记 FIFO 中。数据转发模块和数据上报模块通过读标记 FIFO 获取索引值, 并相应地从双端口 RAM 中取收到的帧数据。接收状态机包含 IDLE、REV、REV_END 和 REV_TAG 4 个状态, 其状态迁移见图 4。

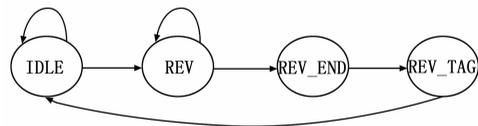


图 4 接收状态机

接收状态机说明:

IDLE: 接收计数归零, 记录写接收双端口 RAM 地址, 若接收标记 FIFO 不满, 并且接收数据有效位为高情况下, 进入下一状态 REV;

REV: 如果接收数据有效位为高, 接收计数计数器自增 1, 依据表 1 填充 32 位寄存器 frame_data, 其中 clientemac0rxid 为 MAC 核用户端输出的接收数据。当接收计数 [1..0] = '11' 时, 将 frame_data 的内容写入到接收双端口 RAM 中。当接收计数 [1..0] = '00' 时, 将写接收双端口 RAM 地址值增 1。如果接收数据有效位变低, 进入下一状态 REV_END; REV_END: 记录当前接收计数值为帧长度, 进入 REV_TAG; REV_TAG: 将 IDLE 状态下记录的写双端口 RAM 首地址和

表 1 接收数据填充表

| 计数 | 操作 |
|----|--|
| 00 | frame_data[31..24] = clientemac0rxid[7..0] |
| 01 | frame_data[23..16] = clientemac0rxid[7..0] |
| 10 | frame_data[15..8] = clientemac0rxid[7..0] |
| 11 | frame_data[7..0] = clientemac0rxid[7..0] |

REV_END状态保存的帧长度组成 32 位数据写入标记 FIFO 中。计算下一个预备接收帧的首地址，如果已经是最后一个缓冲区，则归零，否则加上一个缓冲区大小。进入 IDLE 状态。

当用户需要通过 MAC 核发送数据时，逻辑通过接收状态机将发送缓冲区的数据整合成为 MAC 核用户端的时序，从而将数据发送至物理层。整个发送过程需要发送缓冲区及缓冲区标识作为辅助。数据转发模块通过读标记 FIFO 获取索引值，从相应发送缓冲区中取得数据，通过发送状态机发送至 MAC 核用户端。发送状态机包含 IDLE、GET_LENG、TRA_WAIT、TRA 和 TRA_END 五个状态，其状态迁移见图 5。

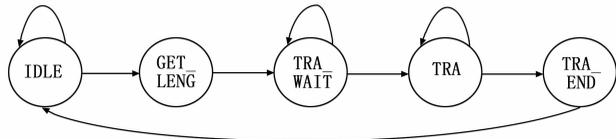


图 5 发送状态机

发送状态机说明如下：

IDLE：若标记 FIFO 的不空，进入下一状态 GET_LENG；

GET_LENG：读标记 FIFO，获得发送帧长度及发送缓冲区首地址，读发送缓冲区首地址，进入下一状态 TRA_WAIT；

TRA_WAIT：将读出内容赋值暂存于 frame_data 中。clientemac0txd 为传递给千兆 MAC 核的用户端的发送数据。将 frame_data 高 8 位赋值给 clientemac0txd [7..0]，置发送数据有效位置高，检测用户端 MAC 核的 ACK 信号是否为高，为高跳转至下一状态 TRA，为低继续保持在状态 TRA_WAIT；

TRA：将 frame_data 分 8 位依据表 2 中所示依次赋值给 clientemac0txd。发送计数增 1，当发送计数低两位为“01”时将读取发送缓冲区中的下一地址，放出读信号和读地址，在发送计数低两位为“11”时，将读出的数据保存到 frame_data 中，当发送计数与发送帧长度-1 相等时，进入 TRA_END；

表 2 发送数据填充表

| 计数 | 操作 |
|----|---|
| 00 | clientemac0txd[7..0] = frame_data[31..24] |
| 01 | clientemac0txd[7..0] = frame_data[23..16] |
| 10 | clientemac0txd[7..0] = frame_data[15..8] |
| 11 | clientemac0txd[7..0] = frame_data[7..0] |

TRA_END：发送最后一个字节给 clientemac0txd，并且置发送数据有效位置低，发送过程结束。进入 IDLE 状态。

2.3 数据转发设计

数据转发模块的作用是将千兆 MAC 核接收端收到的帧数据缓存至接收缓冲区，并在延迟最小的情况下，将保存在缓冲区的数据组帧发送至千兆 MAC 核发送端。通常的方法是将整个帧接收完成后通过计算得出帧长度，依据帧长度再启动发送控制流程。在接收过程中如果发现帧内容、目的地址、源地址、SN 号、CRC 校验码等字段存在错误可以立即终止当前帧的接收，释放缓冲区的内容，可以有效的控制发送端输出帧的

正确性和减小缓冲区的浪费。该方法的缺点是如果需要转发一连串的长帧，就会对数据帧之间的 BAG 和 JITTER 产生较大的影响。以最大允许帧长 1518 字节为例，从千兆 MAC 核传输来的数据为 8 bit，假设缓冲区数据位宽为 32 bit，那么接收一个单元需要 4 个时钟周期，发送 32 bit 同样需要 4 个时钟周期。从接收一个 1518 字节的帧到这个帧发送出去共需 $(1518 * 8/8/4) * 2 = 759$ 个时钟周期。千兆 MAC 核时钟为 125 MHz，一个时钟周期为 8 ns，总共需要 $759 * 8 = 6072$ ns。在千兆 AFDX 网络中，每个虚链路都受到带宽分配间隔 (BAG、bandwidth allocation Gap)、抖动 (jitter) 规划控制。上述计算中转发一个最长帧需要大于 0.006 ms 的时间，且这种效果会累加，累积造成的后果必将使某些帧通过转发因 bag 或 jitter 过大被过滤。

因此应该在接收到一定的数据后即可开启发送。这样产生的转发延迟为亚微秒级，没有累积效应，对毫秒级别的 bag 和微秒级别的 jitter 都不会产生影响。该方法存在的缺点是不能发现链路上数据帧的错误，并且当某个帧长度超过一个缓冲区大小，如果超出的部分没有及时发送出去而被新的内容覆盖掉就会出现错误。

2.4 数据上报设计

数据上报的具体过程是在数据上报缓冲区和数据上报标识 FIFO 的写操作完成时，对主机处理器报一个中断，当主机接收到该中断后，读取数据上报标识 FIFO 的内容。数据上报标识 FIFO 的内容含有数据上报帧长度和数据上报缓冲区存放起始地址，主机依据长度对数据上报缓冲区从起始地址开始的内容进行读取完成数据上报的过程。

主机与数据缓冲区通过 PCIe 总线进行通讯，PCIe 是一种串行总线，可进行点对点传输、双向传输和数据分通道传输，单向传输带宽最高可达 2.5 Gbps。PCIe 支持点到点互连，可极大地降低系统硬件平台设计的复杂性和难度。V5 系列 FPGA 支持 1~4 个 PCIe 硬核，单个硬核最大可以支持 8 通道的通信速率，可满足与计算机主机进行通讯的要求。因此，本设计使用 Xilinx 公司的 IP 核 Endpoint Block Plus for PCI Express。

2.5 时钟设计

该设计中需要的外部时钟源共 3 个，分别为来自于主机端的 PCIe 时钟、RocketIO GTX 电接口时钟、PPC440 工作时钟。

总线时钟由主机端提供，通过专用的 PCIe 连接器后经过 0.1 μf 的滤波电容后供给相应的 GTX 接口。RocketIO GTX 电接口需要 212.5 MHz 的差分时钟，通过差分晶振来提供 FC 工作所需的 212.5 MHz 时钟。为了提高 FPGA 输入时钟的品质，进行电容滤波后贴近相应的 FPGA 放置。采用单端晶振提供 40 MHz 系统时钟。40 MHz 时钟在 FPGA 内部，经 PLL (Phase Locked Loop) 后，分别生成两种频率的时钟：PPC440_CLK, 250 MHz，用于 PPC440 处理器工作频率；PLB_CLK, 125 MHz，用于 PLB 总线工作频率。

3 功能验证

功能验证需要对两方面的功能进行验证，一是验证 TAP 卡能否按要求对数据进行转发，二是验证 TAP 卡能够按照要求将收到的数据进行数据上报。使用成熟的千兆 AFDX 测试卡 APX-GNET-4 作为数据发送及接收的工具，利用 TAP 卡进行数据测试。

将测试卡 A 口设置为发送模式，通过工具 PBA.pro 组织一个千兆 AFDX 帧序列，可以包含一些特定错误以便观察 TAP 能否转发。将测试卡 B 口设置为接收模式。将 A 口的帧序列发出，观察 B 口接收到的帧序列及主机端通过 PCIe 接口收到的数据与 A 口发出的序列数据一致性、时间关系，结果见表 3。

(下转第 3023 页)

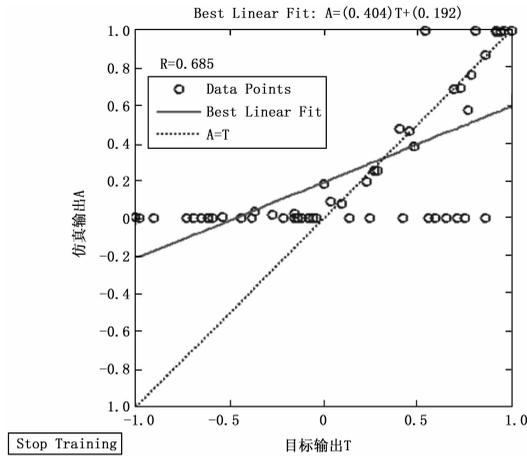


图 4 BP 仿真输出与目标数据的回归分析

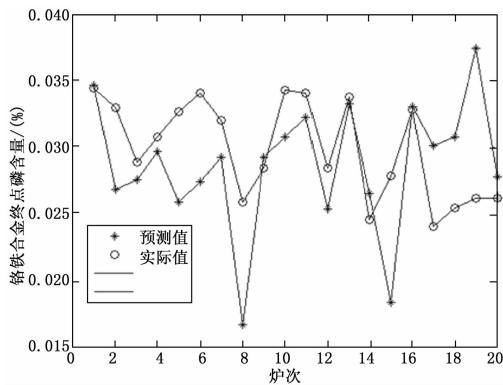


图 5 终点磷含量预报值与实际值比较

4 结论

1) 样本数据要充足, 数量不足则网络训练强度不够。另

外, 网络模型训练前要进行数据归一化处理, 剔除不可靠数据, 否则会影响模型的预报精度。

2) 网络模型的输入变量应该选择重要因素, 输入节点数目不宜过多, 否则会增加网络结构的复杂性, 导致预报误差增大。

3) RBF 神经网络只有一个隐层, 隐层激发函数使用指数衰减的非线性函数对输入输出映射局部接近, 在训练过程中可以避免陷入局部极小值, 数值逼近要达到相同的精度, RBF 神经网络比 BP 神经网络所需要的参数少。

4) RBF 神经网络相对于 BP 神经网络结构简单, 易于学习, 收敛速度快。

5) 本模型对转炉冶炼铬铁合金终点磷含量进行预报, 误差±0.003%范围内, 命中率达 85.7%, 满足生产要求。转炉冶炼中低碳铬铁过程中, 整个脱磷过程各反应区域的限制性环节主要是磷在渣中的传质过程和磷在铁水中的传质过程。

为了加快脱磷速率, 最大限度地降低终点磷含量, 在实际生产过程中, 脱磷粉剂的喷入速度要快, 从而缩短总喷粉时间, 石灰新鲜且充分溶入铁水中, 熔池均混时间要长。可以根据预报值动态调整石灰的添加量, 将终点磷含量控制在规定限度内。减少回炉次数有利于提高生产效率, 节约成本, 降低能耗。

参考文献:

[1] 谢书明, 陶 钧, 柴天佑. 转炉炼钢终点磷的智能预报 [J]. 控制理论与应用, 2003, 20 (4): 555.
 [2] 柴 杰, 江青茵, 曹志凯. RBF 神经网络的函数逼近能力及其算法 [J]. 模式识别与人工智能, 2002, 15 (13): 300.
 [3] 彭 泓, 王 聪, 王 斌. 基于再生核 RBF 神经网络的瓦斯突出预报系统 [J]. 计算机测量与控制, 2012, 20 (6): 1465.
 [4] 张晓斌, 韩永光. 转炉生产中低碳铬铁和中碳锰铁的技术研究与实现 [J]. 铁合金, 2008, (1): 16.
 [5] 孙国会, 武拥军, 梁连科. 含铬铁水氧化脱磷的研究 [J]. 材料与冶金学报, 2002, 1 (3): 182.

(上接第 3010 页)

表 3 功能验证结果

| 测试项 | 数据一致性 | 时间关系 |
|--------|-----------------|---------|
| 超短帧 | B 口及主机数据与发送数据一致 | 延迟 48ns |
| 超长帧 | B 口及主机数据与发送数据一致 | 延迟 48ns |
| 帧头错帧 | B 口及主机数据与发送数据一致 | 延迟 48ns |
| CRC 错帧 | B 口及主机数据与发送数据一致 | 延迟 48ns |

通过上述结果表明 TAP 卡能够对数据进行正确转发, 转发延迟 48 ns。PC 机端软件通过 PCIe 接口能够收到 TAP 卡上报的数据帧与 A 口发出的序列数据一致, 时间关系相同, 验证了 TAP 卡能够按照要求将收到的数据进行数据上报。

4 结论

本文对在千兆 AFDX 网络的基础上, 提出了一种千兆

AFDX 的 TAP 卡设计, 通过对其进行测试验证, 证明本设计原理正确, 能够满足千兆 AFDX 网络 TAP 测试卡地需求。随着网络系统的日益庞杂, TAP 卡能够对网络中的端系统及交换机使用中的一些细节问题造成的网络故障进行迅速定位, 给网络级调试带来了极大的便利。

参考文献:

[1] Airlines Electronic Engineering Committee Aircraft Data Network Part 7 Avionics Full Duplex Switched Ethernet (afdx) Network [S]. 2005.
 [2] 杨 峰, 田 泽. 基于 USB 接口的 AFDX 网络 TAP 卡设计与实现 [J]. 测控技术, 2013, 32 (5): 77-81.
 [3] 张志平, 张亚棣, 王红春, 等. 千兆 AFDX 网络测试卡设计与实现 [J]. 计算机测量与控制, 2013, 21 (10): 2618-2620.