

DOI:10.19651/j.cnki.emt.2005455

千兆以太网 MAC 控制器软核设计

文 丰 韩雨龙

(中北大学 电子测试技术国家重点实验室 太原 030051)

摘要:为解决大容量存储设备数据传输速率低、长距离传输的稳定性差以及现有千兆以太网硬核 MAC 控制器移植性低,可操控性差和在特定结构功能下资源浪费等问题,提出 FPGA 与物理层芯片相结合方法。通过对 MAC 层组成结构和工作原理的理解,提出 FPGA 与物理层芯片相结合方法,来完成数据高速传输。由于使用 FPGA 实现所以 MAC 控制器具有可控性强,移植性强,资源用量小和适应性强等特点。在自主研发平台基础上对其验证:表明该 MAC 控制器具有可行性和可控性。

关键词:千兆以太网;现场可编程门阵列;MAC 控制器;帧格式;流量控制

中图分类号: TN914 文献标识码: A 国家标准学科分类代码: 510.1050

Soft core design of gigabit ethernet MAC controller

Wen Feng Han Yulong

(National Key Laboratory for Electronic Measurement Technology, North University of China, Taiyuan 030051, China)

Abstract: In order to solve the problems of low data transmission rate of large-capacity storage devices, long-distance transmission stability, low portability of existing gigabit ethernet hard-core MAC controllers, poor maneuverability, and waste of resources under specific structure functions, FPGA and physical layer chip combination method. Through the understanding of the structure and working principle of the MAC layer, a method of combining FPGA and physical layer chip is proposed to complete high-speed data transmission. Due to the FPGA implementation, the MAC controller has the characteristics of strong controllability, strong portability, small resource usage and strong adaptability. It is verified on the basis of a self-developed platform: it shows that the MAC controller is feasible and controllable.

Keywords: gigabit ethernet; field programmable gate array; MAC controller; frame format; flow control

0 引言

随着大面积、高帧频图像传感器越来越多地应用在图像采集存储系统中,图像采集存储系统对数据传输速率提出了越来越高的要求。因此,合适的传输媒介成为存储系统与 PC 端测试系统重要桥梁。常用的传输媒介有 RS232,RS485,LVDS 和以太网,其中千兆以太网传输速率是最快的^[1]。

目前 MAC 控制器实现方式主要由两种方式,一是将 PHY 层和 MAC 层集合到一个芯片,成为具有完整功能的网卡,主要应用于嵌入式方面,例如文献[2]使用 WIZnet 公司的 W5300 以太网芯片,其芯片集成 MAC 层与 PHY 层并支持 TCP/IP 协议栈,虽然给用户使用方面带来极大便捷,但传输协议支持仅限制于该芯片,而且在特定功能下会造成资源浪费,而且灵活性大大折扣。二是 MAC 控制

器作为一个独立 IP 核存在,文献[3]基于 FPGA 设计了 MAC 控制器软核,相比于集成网络芯片,其具有更高的可移植性和可控性。通过上述分析比较,系统将采用 PHY 层芯片配合 MAC 控制器软核的方式完成千兆以太网的设计,由于千兆以太网传输速率非常快,容易造成瞬时过载导致数据丢失,所以在设计 MAC 控制器软核时,增加流量控制功能。

1 整体设计

如图 1 所示,系统主要由 FPGA,PHY 层 88E1111,外部 PROM 以及 RJ45 组成^[4]。PHY 芯片通过 1000Base-T 的形式接到 RJ45,FPGA 与 PHY 层芯片可以通过 GMII 或者 RGMII 接口连接。系统采用 GMII 接口作为 FPGA 与 PHY 层的通信接口,通过 FPGA 的 GPIO 实现。MAC 控制器功能实现由 FPGA 来实现,MAC 控制器主要完成

收稿日期:2020-11-29

• 150 •

对 PHY 层芯片的驱动及数据传输^[5]。

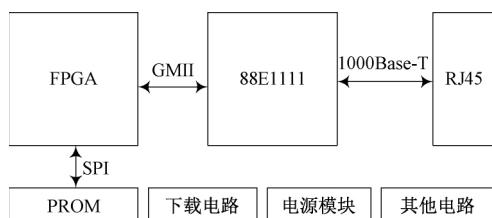


图 1 千兆以太网硬件电路结构

2 千兆以太网 MAC 控制器

MAC 介质访问控制子层主要完成数据解析和发送、出错通知以及可以选择的流量控制等功能^[6]。IEEE802.3 标准规定以太网帧格式如图 2 所示,包括前导码(Preamble)、帧起始定界符(SFD)、目的地址、源地址、长度/类型、数据、填充字段(PAD)和帧校验(FCS)^[7]。如图 3 所示,整个系统分为发送缓存模块、接收缓存模块、发送控制模块、接收控制模块、流量控制模块和 MII 管理配置模块^[8]。

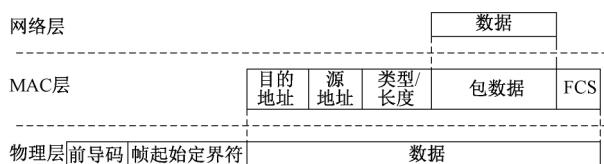


图 2 以太网各层帧格式

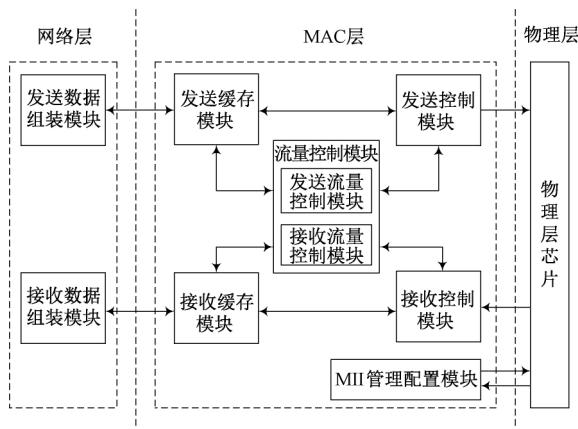


图 3 MAC 控制器系统结构

2.1 发送控制模块

如图 4 所示,发送状态机通过响应上层的命令与 PHY 层建立通信并通过控制数据整合模块完成数据的发送,同时将当前的通信状态及时反馈于上层,如果在这个过程中出现一些错误,Tx_en 有效并通知 PHY 层。数据整合模块负责从上层获取数据帧,并为其添加 Preamble、SFD、PAD 及 FCS,然后将数据用 GMII 接口发送给 PHY 层。

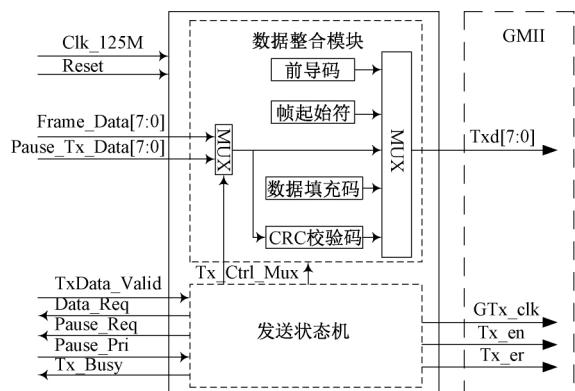


图 4 发送控制模块

如图 5 所示,发送控制模块由七个状态组成。当发送缓存模块要发送数据时,会等发送控制模块 Tx_Busy 信号处于无效状态^[9],才会置位并维持 TxData_Valid 信号。IDLE 空闲状态,用于监测信号 TxData_Valid,当 TxData_Valid 信号有效且 Pause_Pri 为无效,则跳转 Preamble,连续 7 个时钟周期发送 0x55^[10],接着跳转 SFD,发送 1 个时钟周期的 0xD5,并且向发送缓存模块发出请求信号 Data_Req。发送缓存模块在检测到 Data_Req 有效后,复位 Data_Req 并从缓存中取出帧数据交给发送控制模块发送。如果有效数据不足 46 Byte,进入 PAD 进行补全,接着进入 FCS;若有效数据帧长在 46 Byte 到 1 500 Byte 直接进入 FCS^[11]。FCS(帧校验序列)采用 32 位 CRC 校验,对目的 MAC 地址字段到数据字段进行校验,校验完成之后进入 IDLE,等待下一帧数据的发送。Tx_Ctrl_Mux 信号用于切换数据的来源,一方面来源于普通用户数据,另一方面来源于流量控制 Pause 帧。

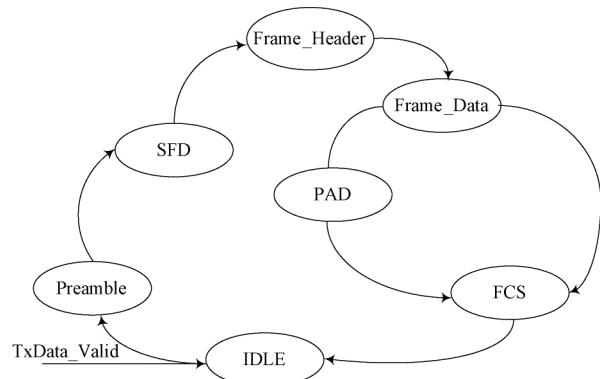


图 5 发送控制状态机

2.2 接收控制模块

如图 6 所示,数据分解模块将收到的包数据拆解为 Preamble、SFD 和有效数据,将校验合格的有效数据存入内部缓存。接受状态机主要责任是配合数据分解模块并通过判断有效数据中的类型决定是将有效数据传输到上层的

接收缓存模块还是接收流量控制模块。

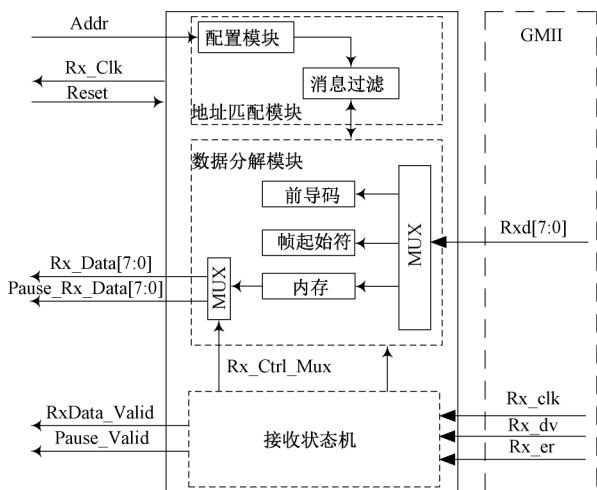


图 6 接收控制模块

如图 7 所示,接收控制模块由 8 个状态组成。IDLE 空闲状态,当监测到来自物理层芯片的 Rx_dv 信号有效时,依次跳转到 Preamble 和 SFD 状态。按照 IEEE802.3 协议要求,Preamble 长度为 7 Byte,之后为 1 Byte SFD。如果在 Preamble 之后还未出现 SFD,则 MAC 会丢弃该帧;如果监测到合法的 SFD,将跳转到 Des_MAC_Addr_Matching 状态,接收以太网数据并向内部缓冲存储且使能数据校验,在此过程中也将进行地址匹配。如果匹配失败,放弃接收剩余帧并将内部缓冲清空;地址匹配成功,进入 Check 状态。在 Check 状态下判断是普通帧还是 PAUSE 帧,如果是普通帧则将 Rx_Ctrl_Mux 信号置 1,选通接收缓存模块的通路;否则 Rx_Ctrl_Mux 信号置 0,选通接收流量控制模块的通路。经过一段时间数据的存储之后,进入 Check 状态。在 Check 状态下将 FCS 与 CRC 比较,如果不一致则将之前的数据清除;校验一致的情况下结合 Rx_Ctrl_Mux 信号向上层接收缓存或者接收流量控制模块上传数据并同时置位 RxData_Valid 或者 Pause_Valid,进入 IDLE 状态等待接收下一帧数据。

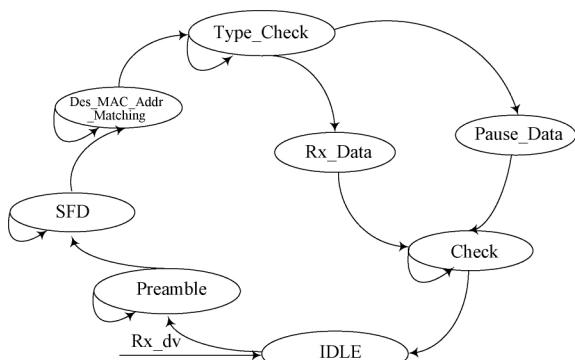


图 7 接受控制状态机

2.3 流量控制模块

千兆以太网相比于 10/100 Mbps 的传输速率来说,速度越来越快会导致瞬时过载发生接受缓冲溢出数据的情况,最终会严重影响网络性能。所以,IEEE802.3 协议提出流量控制结构来解决上述问题。流量控制结构基本原理是通过一种简单的停/等式方式实现^[12]。当接受方的接受缓冲区达到临界值时,会给发送方发出 PAUSE 帧,PAUSE 帧和普通数据帧类似,有着属于自己的帧标识,并且含有暂停时长数据。发送方在成功接受到 PAUSE 帧之后,会解析出暂停时长数据并对自己的发送模块进行对应时长的失能,失能过后将继续进行发送,这样就很好解决瞬时过载问题^[13]。如图 8 所示,当接受缓冲模块达到临界值时,Rx_Buff_Full 信号有效发送流量控制和发送控制配合完成 PAUSE 帧发送;接受控制模块和接受流量控制模块完成 PAUSE 帧的识别,校验及暂停时长提取,通过 Pause_Block 信号有效,阻止发送缓冲模块继续发送数据^[14]。

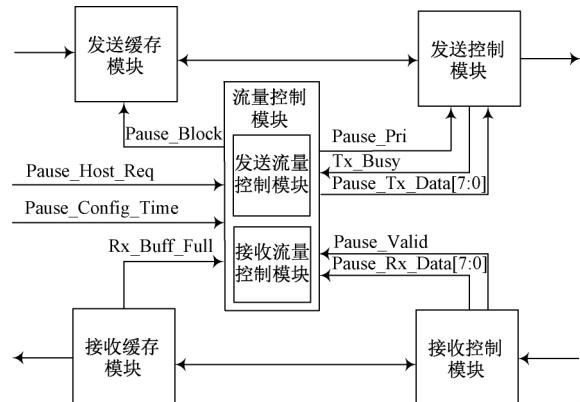


图 8 流量控制模块

1) 发送流量控制模块

由于 PAUSE 帧的作用是防止瞬时过载,所以 PAUSE 帧相比于普通帧具有一定的特殊性,在两者都要进行发送时,PAUSE 帧的优先级理应高于普通帧,但是在普通数据发送时,PAUSE 帧打断其会造成数据的丢失,所以就要求 PAUSE 帧要发送时,先置位 Pause_Pri 信号,等待普通帧发送完毕之后,在进行抢占,同时要求接受缓冲模块的临界值设置可以冗余大于两包的普通数据。发送控制模块中的 Tx_Ctrl_Mux 信号有效为 PAUSE 选通,预示着抢占成功,接着发送的数据将为 PAUSE 帧,普通帧将处于等待状态。

如图 9 所示,发送流量控制模块由 5 个状态组成, IDLE 为空闲状态,主要用于监测 Rx_Buff_Full 信号。如果 Rx_Buff_Full 信号有效则进 Wait_For_Send 状态;在 Wait_For_Send 状态下,将 Pause_Pri 信号置 1 并通过监测忙闲信号 Tx_Busy 的状态来判断当前普通帧是否发送完毕,若此信号无效则跳转至 Start_Send 状态,否则保持在本状态。在 Start_Send 下,发送流量控制将 Pause_Start_

Frame 置位,发送控制监测到 Pause_Start_Frame 信号有效,则开始发送 Preamble 和 SFD,发送完毕之后,将 Pause_Req 置位,发送流量模块监测 Pause_Req 请求信号有效,进入 Sending 状态。Sending 状态负责完成 Pause 帧的发送,Pause 帧如表 1 所示,Pause_Time 字段对应着需要暂停的时长,由外部信号 Pause_Config_Time 进行初始化配置。当 Pause 帧发送完毕之后,状态机进入 Send_Done 状态,在此状态下,也将同样使用倒计时的方法对接受缓冲区的空间进行判断,这将会有 3 种情况发生:(1)倒计时结束,接受缓冲区依然不足,将继续进入 Wait_For_Send 状态,进行 Pause 发送;(2)倒计时未结束,但接受缓冲区空间足够,也将进入 Wait_For_Send 状态,发送暂停时长为 0 的 Pause 帧,来使发送方继续进入发送状态;(3)倒计时结束

表 1 PAUSE 帧数据格式

目的 MAC 地址 (6 Byte)	源 MAC 地址 (6 Byte)	Type (2 Byte)	Opcode (2 Byte)	Pause_Time (64 bit)	PAD	FCS
0x* * * * *	0x* * * * *	0x8808	0x0001	0xFF... 0xFF	0x00... 0x00	0x* * * * *

2) 接收流量控制模块

如图 10 所示,接收流量控制模块主要作用是读取 Pause 帧中暂停时长参数并对发送缓冲模块进行阻塞。首先接受控制模块,在未确定是 Pause 帧还是普通帧的前提下,会先把数据存在内部缓冲,且在此过程中,如果发生地址匹配失败的情况,会把内部缓冲清空。当确定为 Pause 帧,接受控制模块会读取 Pause 帧。同时,置位 Pause_valid,接受流量控制模块收到 Pause_valid,开始接受 Pause 数据,并获取暂停时长参数。

接收流量控制模块由 6 个状态组成, IDLE 为空闲状态, 在此状态下等待 Pause_Valid 有效, 当其有效后进入 Pause_Filter 状态。在 Pause_Filter 状态下, 读取 Pause 帧的暂停时长数据, 读取完成之后, 进入 Pause_Time_Load 状态。Pause_Time_Load 状态下, 将上一状态读取的暂停时长数据装入寄存器并跳转 Pause_Wait_Time 状态, 开始进行倒计时计数, 且将 Pause_Block 置为有效; 当倒计数溢出时, 跳转 Pause_Time_Exit 状态, Pause_Block 信号无效, 对发送缓冲模块的阻塞进行解除。

2.4 MII 管理配置模块

串行管理接口(serial management interface),也被称为 MII 管理接口(MII management interface),包括 MDC 和 MDIO 两条信号线。MII 管理配置模块为以太网接口芯片中 MAC 层对 PHY 器件的控制管理模块,主要完成 PHY 层工作速率、接口选择以及工作模式的配置。如图 11 所示,MDIO 控制负责完成 PHY 层芯片的配置与检测逻辑,MDIO 接口负责完成读写时序。

3 MAC 控制器软核验证

MAC 控制器软核的验证主要以 PC 端和 FPGA 端数

并且缓冲区空间充足,将进入 IDLE 状态。

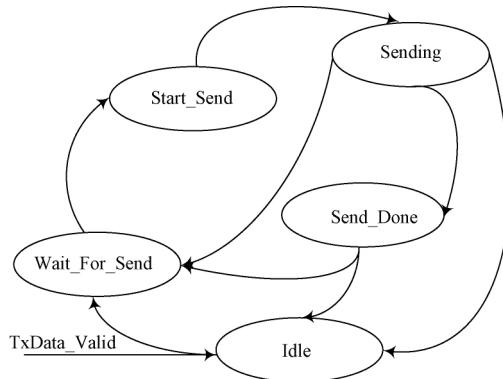


图 9 发送流量控制模块状态机

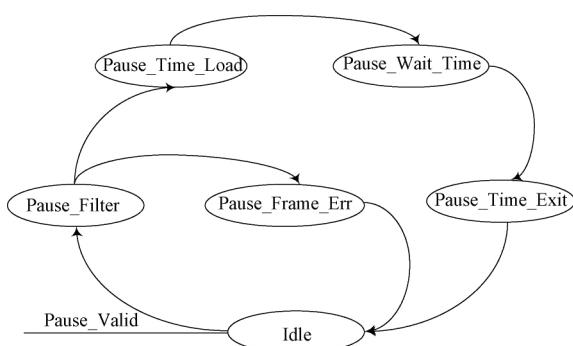


图 10 接收流量控制模块状态机

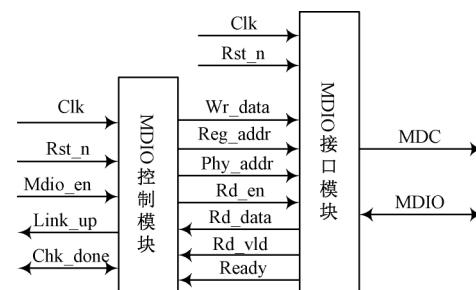


图 11 MII 管理配置模块

据传输透明化为基础,通过数据分析来完成对 MAC 控制软核性能测试。系统测试平台如图 12 所示,通过网线将 FPGA 端与 PC 端相连进行功能及性能测试,FPGA 端采用 Xilinx XC7A200T 作为主控芯片,88E1111 作为外围 PHY 层芯片,为了方便后期数据分析,当 FPGA 接收到命令之后,以自加数(含有帧计数)作为一帧有效数据发送 UDP/IP 包,便于后期 PC 端对发生丢包丢帧丢数现象进

行分析, FPGA 端所封装的 IP 数据包长为 1 000 Byte, 在 MAC 层为 IP 数据包添加 8 字节帧头, 12 Byte MAC 地址, 2 字节帧类型和 4 字节帧校验序列封装成标准以太网数据帧, 标准以太网数据帧长 1 026 Byte; PC 端使用上位机软件发送命令, 完成对 FPGA 端数据接受, 如图 13 所示, 一帧数据分为递增数区、帧计数区和帧尾, 通过判断帧计数是否连续来判断数据是否存在丢帧丢包现象; 同时 PC 端使用 Wireshark 软件对以太网数据进行实时抓取, 由于网卡接受数据首先去除 8 字节帧头, 再进行 4 字节帧校验, 只接受帧校验无误的数据包, 所以实际抓取数据为 1 014 Byte。如图 14 所示, FPGA 端 MAC 地址为 00:0a:35:01:fe:c0, PC 端 MAC 地址为 30:9c:23:0a:ca:26, 与实际情况相一致, 同时可以看出接受到的数据为 1 000 Byte 的有效数据。如图 15 所示, 接受数据 2.04 GB, 丢帧个数为 0, 上位机对接

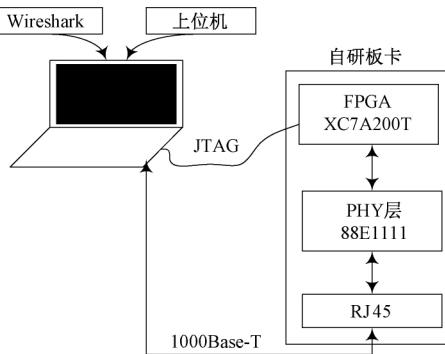


图 12 系统测试平台

EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	F0	96	C6	EB	90
EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	F0	96	C7	EB	90
EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	F0	96	C6	EB	90
EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	F0	96	C9	EB	90
EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	F0	96	CA	EB	90
EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	F0	96	CB	EB	90
EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	F0	96	D3	EB	90
EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	F0	96	CC	EB	90
EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	F0	96	CD	EB	90
EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	F0	96	CF	EB	90
EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	F0	96	D0	EB	90
EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	F0	96	D1	EB	90
EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	F0	96	D2	EB	90
EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	F0	96	D3	EB	90
EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	F0	96	D4	EB	90
EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	F0	96	D5	EB	90
EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	F0	96	D6	EB	90
EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	F0	96	D7	EB	90
EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	F0	96	D8	EB	90
EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	F0	96	D9	EB	90
EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	F0	96	DA	EB	90
EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	F0	96	DB	EB	90
EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	F0	96	DC	EB	90
EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	F0	96	DD	EB	90
EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	F0	96	DE	EB	90
EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	F0	96	DF	EB	90

图 13 上位机接受数据文件

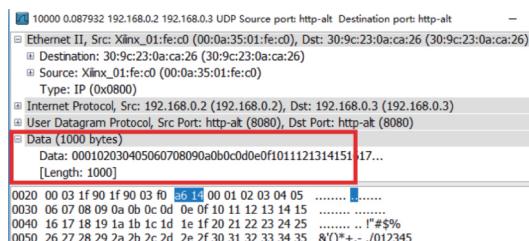


图 14 Wireshark 抓包结果

数据校验分析结果
-----数据文件: c:\20200923153052477.dat 2.04 GB
-----数据校验全部正确。
-----帧计数校验全部正确。
-----首帧计数: 0x1
末帧计数: 0x81E0
共有帧个数: 8552928
-----丢帧检测丢帧个数: 0
丢帧检测数据误码率: 0.0%

有效数据占比: 100.00%

图 15 帧校验报告

受到的有效数据以二进制文件进行保存, 之后通过判断帧计数是否连续确定是否存在丢包情况。经过一定长度时间测试之后, 发现平均丢包率为十万分之一。

通过 Wireshark 观察其所使用的时间如图 16 所示, 从软件界面可以看出接收 1 000 000 包数据(每包 1 000 Byte)总共耗时 8.799 832 s, 所以我们可以算出以太网传输速率为

$$V_{eth} = \frac{1000 \times 1000000 \times 8}{8.799832} \approx 910 \text{ Mbit/s}$$

999996	8.799831	192.168.0.2	192.168.0.3	UDP
999997	8.799831	192.168.0.2	192.168.0.3	UDP
999998	8.799831	192.168.0.2	192.168.0.3	UDP
000000	8.799832	192.168.0.2	192.168.0.3	UDP
1000000	8.799832	192.168.0.2	192.168.0.3	UDP
1000001	8.799833	192.168.0.2	192.168.0.3	UDP
1000002	8.799833	192.168.0.2	192.168.0.3	UDP

图 16 1 000 000 帧传输时间抓取图

4 结 论

本文根据对 MAC 层组成结构和工作原理的深入理解, 设计和实现了基于 FPGA 千兆以太网 MAC 控制器, 采用 FPGA+PHY 方法实现以太网千兆的传输速率^[15]。经测试, 数据传输速度基本满足千兆以太网要求且系统稳定可靠丢包率低。MAC 控制器高度透明化, 而且功能专一针对性强, 既减少不必要的逻辑资源的浪费, 也大大提高工作效率。

参考文献

- [1] 王琦. 基于千兆以太网的自定义协议分析及接口 IP 核设计[D]. 太原: 中北大学, 2019.
- [2] 乔立岩, 梁宇, 赵浩然, 等. 基于 W5300 的以太网接口设计[J]. 电子测量技术, 2012, 35(7): 129-132.
- [3] 朱保琨, 刘广文. 千兆以太网通信端口 FPGA 设计与实现[J]. 计算机工程与设计, 2016, 37(9): 2292-2298.
- [4] 朱习松, 赵霖, 卢礼兵. 基于 FPGA 的千兆以太网硬件系统实现[J]. 电子质量, 2017(12): 21-24.
- [5] 李振钢, 李晓风, 谭海波, 等. 基于 FPGA 的机房环境监测及数据传输系统设计[J]. 电子测量技术, 2020, 43(5): 168-174.

- [6] 张曼曼. 空天信息网络中 QoS 保障的可靠传输策略研究[D]. 哈尔滨:哈尔滨工业大学,2015.
- [7] 袁春改. 万兆 FCoE 监控卡硬件的设计与实现[D]. 成都:电子科技大学,2014.
- [8] 曾明. 一种基于 DMA 的千兆以太网 MAC 控制器的设计[J]. 中国集成电路,2018,27(5):33-37.
- [9] 郑利华. X-DSP 中千兆以太网 MAC 控制器的设计与实现[D]. 长沙:国防科学技术大学,2014.
- [10] 郑天宇, 尹达一. 千兆 UDP/IP 协议栈 FPGA 实现及其在高速成像系统中的应用[J]. 现代电子技术,2018, 41(18):1-5,11.
- [11] 王立莹, 王俊芳, 曹琨, 等. VLAN 及在千兆以太网 MAC 中的实现[J]. 无线电工程,2013,43(1):1-3.
- [12] 于强. 基于 E1 传输的多路网桥交换机的关键技术研
究[D]. 杭州:浙江大学,2012.
- [13] 耐志伟. 千兆以太网 MAC 控制器 IP 软核的设计[D]. 成都:电子科技大学,2010.
- [14] 丁世勇, 谭文文, 李桂英. 基于 FPGA 的以太网 MAC 控制器的设计与实现[J]. 电子设计工程, 2011, 19(21):163-165,169.
- [15] 靳咪. 基于 DMA 的双路千兆以太网 MAC 的设计与验证[D]. 西安:西安电子科技大学,2015.

作者简介

文丰(通信作者),博士,教授,主要研究方向为测量技术、微系统集成技术、传感器测量技术。

E-mail:nucwenfeng@163.com

韩雨龙,硕士,主要研究方向为数据采集技术。

E-mail:392636720@qq.com