

文章编号: 2095-2163(2023)05-0064-06

中图分类号: TP393

文献标志码: A

基于 FPGA 的 IEEE 1588 时钟同步系统的设计与实现

徐卓汀, 商艳娟, 王成群

(浙江理工大学 信息学院, 杭州 310018)

摘要: IEEE 1588 精密时钟协议用于分布式系统中各节点上独立时钟的同步, 纯软件实现的 IEEE 1588 精密时钟协议的时间戳只能标记在数据链路层, 受网络传输延迟以及网络抖动的影响, 时钟同步精度一般只能达到微秒级。本文设计了一种基于 FPGA 的 IEEE 1588 协议的时钟同步系统, 采用 EP4CE6E22C8 和 RTL8201CP 芯片, 通过编程实现 IEEE 1588 报文识别、时钟偏移补偿算法等。该方法进一步降低了网络传输延迟的不确定性, 提高了时间戳捕获的精度。通过实验对主从时钟同步精度的一致性进行测试, 满足设计要求。

关键词: IEEE 1588 精密时钟协议; FPGA; 时间戳; 时钟同步

Design and implementation of IEEE 1588 protocol based on FPGA

XU Zhuoting, SHANG Yanjuan, WANG Chengqun

(School of Information, Zhejiang Sci-Tech University, Hangzhou 310018, China)

[Abstract] IEEE 1588 precision clock protocol is used to synchronize the independent clocks on each node in the distributed system. The timestamp of IEEE 1588 precision clock protocol implemented by pure software can only be marked on the data link layer. Due to the impact of network transmission delay and network jitter, the clock synchronization accuracy can only reach the microsecond level. This paper designs a clock synchronization system based on IEEE 1588 protocol based on FPGA. EP4CE6E22C8 and RTL8201CP chips are used to realize IEEE 1588 message recognition and clock offset compensation algorithm through programming. This method further reduces the uncertainty of network transmission delay and improves the accuracy of timestamp acquisition. The experiment meets the design requirements by testing the consistency of master-slave clock synchronization accuracy.

[Key words] IEEE 1588 precision clock protocol; FPGA; time stamp; clock synchronization

0 引言

随着分布式网络应用范围及应用规模的不断增大, 同步系统内各分散节点时钟的一致性变得越来越重要, 精确的时间同步技术在分布式系统中占据着越来越重要的地位^[1]。目前, 时钟同步的主要方法包括 IRIG-B (Inter Range Instrumentation Group-B) 码时钟同步、网络时间协议 (NTP)、简单网络时钟同步 (SNTP) 和全球定位系统 (GPS) 等^[2]。传统的 NTP 只能提供毫秒级的时间精度, 不能满足现代工业以太网的高精度时钟需求。SNTP 协议是 NTP 的简化版本, 采用了简化的同步算法, 提高了性能, 但只能满足对时间精度要求不高的系统。GPS 可以

提供微秒级的时间精度, 但其需要 GPS 接收机这样的特殊设备, 不仅成本比较高, 而且不易实现^[3]。IEEE 1588 精密时间协议的出现一定程度上弥补了一些时钟同步方式的不足^[4]。IEEE 1588 协议基于以太网, 不需要建立新的时钟网络, 主从时钟通过交换时钟信息来进行同步^[5]。通过软件实现的 IEEE 1588 协议可以提供微秒级别的时间精度^[6]; 通过硬件实现的 IEEE 1588 协议可以提供纳秒级别的时间精度^[7]。同时, 硬件实现的 IEEE 1588 协议对资源要求低, 不仅可以通过支持 IEEE 1588 协议的专用芯片实现, 也可以在 FPGA 芯片上实现, 实现成本低, 适用于各种分布式系统^[8]。

本文在详细分析 IEEE 1588 协议基本内容的基

基金项目: 浙江省重点研发计划(2021C01047); 流程工业综合自动化国家重点实验室开放课题基金资助项目(2021-KF-21-06, 2020-KF-21-09)。

作者简介: 徐卓汀(1998-), 男, 硕士研究生, 主要研究方向: 嵌入式与物联网技术; 商艳娟(1997-), 女, 硕士研究生, 主要研究方向: 物联网、工业互联网; 王成群(1981-), 男, 博士, 讲师, 主要研究方向: 工业互联网、AI 及其在行业中的应用。

通讯作者: 王成群 Email: cqwang@zstu.edu.cn

收稿日期: 2022-06-14

础上,通过软硬件结合的方式,在FPGA芯片上实现IEEE 1588报文的检测和时间戳的捕获。实验验证结果表明该系统能够正常工作,能满足现代工业以太网的高精度时钟同步需求。

1 IEEE 1588 时钟同步原理

IEEE 1588 精确时间协议在同步过程中,主时钟和从时钟之间相互交换带有精确时间戳和预估时间戳的报文^[9]。各个从时钟得到这些时间戳信息后,计算与主时钟的链路传输延迟和时钟偏差,进而调整从时钟,达到与主时钟同步的目的。IEEE 1588 协议的主从同步过程如图 1 所示^[10]。

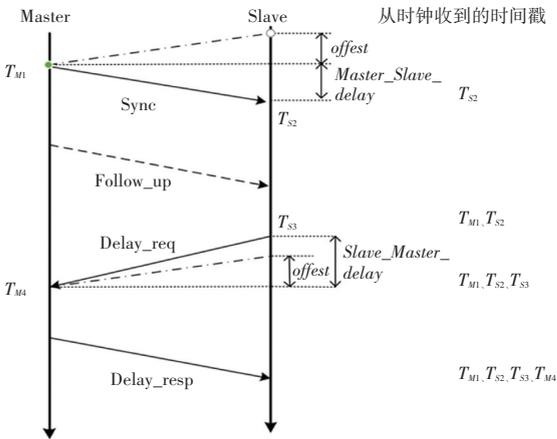


图 1 IEEE 1588 协议的主从同步过程

Fig. 1 Master and slave synchronization process of IEEE 1588

IEEE 1588 协议的同步原理是延时响应机制,通过 PTP(Precision Time Protocol) 报文承载时间戳来完成,PTP 报文分为 4 种:Sync 同步报文、Follow_up 跟随报文、Delay_req 延迟请求报文、Delay_resp 延迟请求响应报文^[11]。

一次时钟同步过程:

(1) Sync 报文周期性从主时钟向从时钟以广播的形式发出,主时钟在发送结束那一刻记录 Sync 报文离开主时钟的精确发送时间 T_{M1} ;

(2) 主时钟将精确时间戳 T_{M1} 封装进 Follow_up 报文中,发送给从时钟;

(3) 从时钟记录 Sync 报文到达从时钟的精确时间点 T_{S2} ;

(4) 从时钟发送 Delay_req 报文,并在发送结束那一刻记录下精确发送时间 T_{S3} ;

(5) 主时钟记录 Delay_req 报文到达主时钟的精确到达时间 T_{M4} ;

(6) 主时钟发送携带精确时间戳信息 T_{M4} 的 Delay_resp 报文到从时钟。

经过一次时钟同步后,从时钟寄存 4 个精确时间戳信息。利用 4 个时间戳可以计算链路延时和时钟偏移。

设报文由主时钟到从时钟的链路延迟,式(1):

$$Master_delay = T_{M1} - T_{S2} \quad (1)$$

此时主时钟与从时钟之间存在的时间偏差 offset1,式(2):

$$offset1 = T_{S2} - T_{M1} - Master_delay \quad (2)$$

报文由从时钟到主时钟的链路延迟,式(3):

$$Slave_delay = T_{S3} - T_{M4} \quad (3)$$

此时从时钟与主时钟之间存在的时间偏差 offset2,式(4):

$$offset2 = T_{M4} - T_{S3} - Slave_delay \quad (4)$$

因为 IEEE 1588 协议假设链路延迟是对称的,式(5):

$$Slave_delay = Master_delay \quad (5)$$

那么平均链路延迟 mean_delay,式(6):

$$mean_delay = \frac{(T_{M4} - T_{M1}) - (T_{S3} - T_{S2})}{2} \quad (6)$$

此时主从时钟偏差 offset,式(7):

$$offset = (T_{S2} - T_{M1}) - mean_delay \quad (7)$$

进一步计算得到 offset,式(8):

$$offset = \frac{(T_{S2} - T_{M1}) + (T_{S3} - T_{M4})}{2} \quad (8)$$

对从时钟进行补偿来达到同步于主时钟的目的。

2 系统软硬件设计

2.1 时间戳标记位置

IEEE 1588 协议的报文在交换消息过程中包含着各时钟的信息,所以时钟信息的提取对于提高同步精度有着十分重要的作用。时间的标记位置有 3 种方案,如图 2 所示。

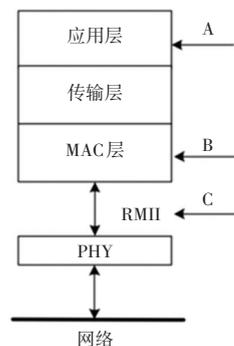


图 2 时间戳位置标记

Fig. 2 Time stamp position marking

方案 A: 在应用层进行时间戳的标记, 虽然实现方法简单, 但由于协议栈的抖动和延迟是不确定的, 会导致较大误差, 使用软件实现的精度只能达到毫秒级。

方案 B: 在介质访问控制层 (MAC), 避免了协议栈的抖动和延迟, 提高时间戳的标记精度, 能够使用软件实现, 时间精度达到微秒级^[12]。

方案 C: 在 MAC 层和物理层 (PHY) 之间的简化媒体独立接口 (RMII) 处标记时间戳, 通过软硬结合的方法实现。目前有两种方法, 一种是使用支持 IEEE 1588 协议的专用 PHY 硬件芯片, 如 DP83640, 可以在物理层检测 IEEE 1588 报文, 并将时间戳信息填充到 IEEE 1588 报文中; 第二种方法是在对原有硬件不做改变的情况下, 在 FPGA 芯片中识别 IEEE 1588 报文以及捕获时间戳^[13]。

时间戳的标记越接近于实际网络, 抖动和延迟就越小, 计算出的时钟偏差和链路延迟更接近于真实值, 所以在本文的设计中, 采用了方案 C, 因为方案 C 的第二种方法不需要专用的 PHY 芯片, 而且 FPGA 处理速度足够快, 实现方法更简单。

2.2 系统硬件搭建

系统硬件采用 Altera Cyclone IV 系列中的 EP4CE6E22C8 FPGA 芯片以及 RTL8201CP 接口 PHY 芯片, 分别作为系统的主控模块以及网络通信模块。

RTL8201CP PHY 芯片采用 RMII 模式下的以太网连接方式, 是简化的媒体独立接口, 减少了以太网通信所需的引脚数, 将引脚数从 16 个减少为 7 个, 有着更简单的连接结构。

Altera Cyclone IV 系列 FPGA 芯片作为主控模块, 自身能够在物理层和 PHY 之间的 RMII 接口处进行时间标记, 相对于其他如 STM32F407 系列 CPU 芯片只能在 MAC 数据链路层标记, 有着更高的时间戳精度。主控模块主要实现接受发送 PTP 报文, 以及时钟同步等功能。

2.3 系统软件设计

本文系统的软件结构主要分为协议层、应用层。协议层完成报文的解析以及封装过程, 应用层完成实时时钟以及精确时钟同步功能。其系统结构图如图 3 所示。

协议层分为 UDP/IP 协议、EtherCAT 协议以及 PTP 协议, EtherCAT 协议有两种实现方式: 一种是利用标准 IEEE802.3 以太网帧实现, 一种是放入 UDP/IP 协议中实现, 本文采用第二种实现方式。

系统的主时钟将 PTP 报文封装在 EtherCAT 中, 再将 EtherCAT 报文封装在 UDP 报文中, 最后通过网络传输到从时钟。从时钟接收到报文后, 解析报文, 进行时间戳的寄存和提取。

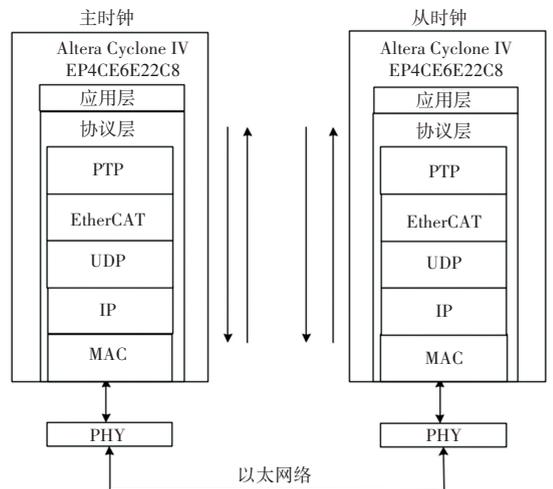


图 3 系统结构图

Fig. 3 System structure diagram

时钟同步系统上电后, 通过 PC 机, 以 UDP 协议的形式发送指令报文, 初始化各主从时钟的计数器, 并设定与 PC 机通信的为主时钟, 与主时钟通过网线相连的为从时钟。初始化各主从时钟后, PC 机周期性发送 Sync 报文, 进入时钟同步阶段。

如果本地时钟为主时钟状态, 其主时钟流程图如图 4 所示。主时钟接受来自 PC 机的报文, 识别并转发 Sync 报文, 对 RMII 接口的 TX_EN 信号进行锁存, 当 TX_EN 信号由高电平变为低电平时, 寄存这个周期的时钟计数器, 接着发送 Follow_up 报文, Follow_up 报文的 64 位精确时间戳为上一个寄存的时钟计数器的时间, 发送结束后, 等待接收下一条报文。接收到报文时, RMII 接口的 RX_EN 信号由低变高, 寄存该周期的时钟计数器, 识别接受的报文, 如果识别为其他报文, 丢弃该时间戳, 并回到等待接受 Delay_req 报文状态, 若为 Delay_req 报文, 保留该时间戳, 并读取来自从时钟的端口号, 接着构造 Delay_resp 报文, Delay_resp 报文的 64 位精确时间戳即为上一次寄存的时钟计数器。主时钟进入等待接受 PC 机报文状态。

如果本地时钟为从时钟, 从时钟流程图如图 5 所示。时钟计数器初始化后, 接受到报文时, RMII 接口的 RX_EN 信号由低变高, 寄存器寄存该周期的时钟计数器的值, 识别该报文是否是 Sync 报文, 若不是, 则丢弃寄存的时间戳, 并返回等待报文接受状态; 如果是, 继续接受并解析 Follow_up 报文, 读

取报文中的时间戳,并开始构造 Delay_req 报文。当发送 Delay_req 报文结束时,即 RMII 接口处的 TX_EN 信号由高变低,寄存该时刻的时钟计数器的值,并与接受 Follow_up 报文同样的过程接受 Delay_resp 报文,并读取报文中的精确时间戳。从时钟得到来自物理层和 PHY 之间的 RMII 接口处的 4 个精确时间戳,进入链路延迟以及时钟偏差计算阶段,并对从时钟进行补偿,得到同步的从时钟进入下一次时钟补偿阶段。

现,其频率补偿过程如图 6 所示。

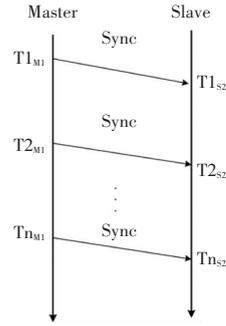


图 6 频率补偿过程

Fig. 6 Frequency compensation process

在时钟同步过程中,当系统进行第一次时钟同步过程中时,主站记录下第一次发送 Sync 同步报文的时间戳 $T1_{M1}$,从站记录下第一次收到同步报文的时间戳 $T1_{S2}$ 。同理,当系统进行第二次时钟同步时,记录下时间戳 $T2_{M1}$ 和 $T2_{S2}$ 。则主站的发送时间间隔 MCC,式(9):

$$MCC = T2_{M1} - T1_{M1} \quad (9)$$

同理,从站接收到的时间间隔 SCC,式(10):

$$SCC = T2_{S2} - T1_{S2} \quad (10)$$

主从站的时间偏差 MSC,式(11):

$$MSC = |MCC - SCC| \quad (11)$$

主从之间的时间比率,式(12):

$$\gamma = SCC/MSC \quad (12)$$

在每次时钟同步过程中,从时钟持续补偿计算出的时钟偏移量;在补偿过程中,判断偏移量的最高位是否为 1,若是表示从时钟快于主时钟,偏移量的值应取反加 1,补偿进从时钟中,反之,直接补偿从时钟;对于时间频率的补偿,从系统的第二次时钟同步阶段开始,判断 MSC 的值,若最高位为 1,说明主时钟频率高于从时钟频率,则从时钟的纳秒(ns)计数器每增加 γ 值时减 1,反之,主时钟频率低于从时钟频率,纳秒(ns)计数器每增加 γ 值时加 1。

3 时钟同步测试

为了验证 IEEE 1588 时钟同步系统的可行性,将 IEEE 1588 协议以及本地时钟补偿算法应用于基于 FPGA 的 EtherCAT 协议栈中,采用低延时、高带宽、确定性的现场总线结构,该协议栈具有区分 IEEE 1588 报文和普通报文的功能,主站能解析来自 PC 机的报文,并将报文内容转发至从站,从站能够解析各报文命令,执行时间戳提取、初始化操作、时钟同步以及同步输出等功能。

本系统以主站的本地晶振作为授时源,通过主

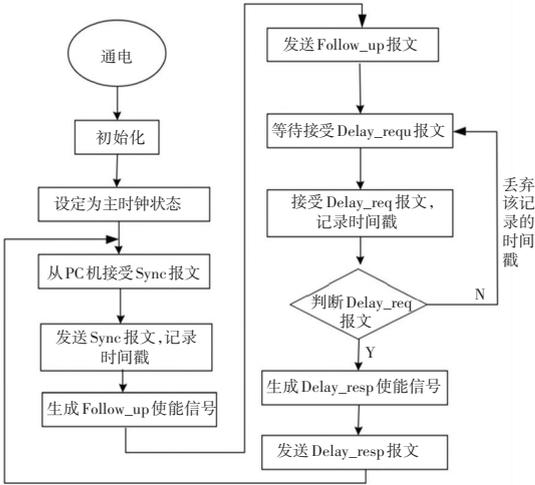


图 4 主时钟流程图

Fig. 4 Master clock state machine process

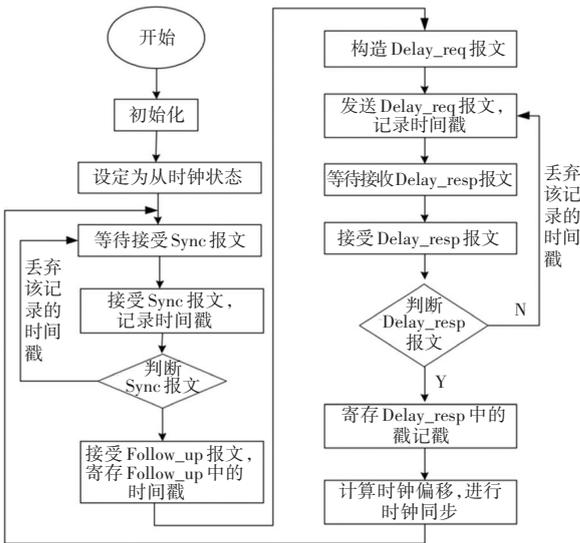


图 5 从时钟流程图

Fig. 5 Slave clock state machine process

2.4 本地时钟补偿算法

本地时钟补偿采用动态的频率补偿算法来实

从站直连的方式进行时钟同步性能测试,设置的同步时间间隔为 1 s,时钟同步过程分为两步:

(1)系统启动,发送初始化指令报文进行各主从站计数器清零操作;

(2)周期性发送 Sync 报文,进行时钟同步。

3.1 一主一从直连同步测试

系统初始化后周期性发送 Sync 同步报文进行时钟同步,其中主从站某一次输出 1 s 脉冲图如图 7 所示,横坐标每一格线代表 20 ns,该波形表示主从站的同步输出偏差为 0 ns。

200 次主从同步,其时间偏差结果如图 8 所示。

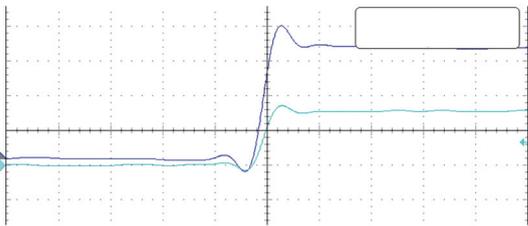


图 7 主从站某一次输出 1 s 脉冲

Fig. 7 Primary and secondary station output 1 s pulse diagram at one time

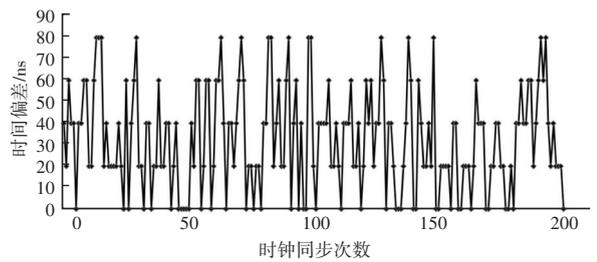


图 8 时间偏差结果

Fig. 8 Time deviation result

其中时间偏差为一个绝对值量。由图 8 可知,主从时钟最大时钟偏差为 80 ns,时钟偏差基本分布在 ± 80 ns 之间,能够满足现代工业以太网对时间精度的需求。

3.2 一主二从直连同步测试

系统初始化,进行 200 次同步测试,其主从同步结果如图 9 所示。

其中,ID1 时钟表示从站 1 与主站的时钟偏差,ID2 表示从站 2 与主站的时钟偏差。ID1-ID2 时钟差值(LCK)为 ID1 时钟与 ID2 时钟的差值,ID1-ID2 时钟差值(ns)为 ID1-ID2 时钟差值(LCK)乘 20 ns 所得。

No.	MainSendCnt	MainRecvCnt	循环周期(ck)	循环周期(ns)	ID1时钟(CLK)	ID2时钟(CLK)	ID1-ID2时钟差值(CLK)	ID1-ID2时钟差值(ns)	ID1Recv	ID2Recv
13	756522	757264	741	14820	3	5	-2.0	-40.0	bb	bb
14	745191	745934	742	14840	0	3	-3.0	-60.0	bb	bb
15	1085156	1085899	742	14840	0	3	-3.0	-60.0	bb	bb
16	1082521	1083266	744	14880	0	3	-3.0	-60.0	bb	bb
17	1029255	1029998	742	14840	2	4	-2.0	-40.0	bb	bb
18	942289	943032	742	14840	0	0	0.0	0.0	bb	bb
19	990774	991517	742	14840	0	1	-1.0	-20.0	bb	bb
20	773241	773983	741	14820	2	0	2.0	40.0	bb	bb
21	1098202	1098946	743	14860	1	1	0.0	0.0	bb	bb
22	946596	947338	741	14820	1	0	1.0	20.0	bb	bb
23	1006986	1007731	744	14880	1	2	-1.0	-20.0	bb	bb
24	1083971	1084716	744	14880	0	2	-2.0	-40.0	bb	bb
25	1141161	1141904	742	14840	1	1	0.0	0.0	bb	bb
26	955315	956058	742	14840	0	0	0.0	0.0	bb	bb
27	881160	881904	743	14860	2	1	1.0	20.0	bb	bb
28	987146	987890	743	14860	3	1	2.0	40.0	bb	bb
29	1083883	1084629	745	14900	2	1	1.0	20.0	bb	bb
30	1090452	1091194	741	14820	1	2	-1.0	-20.0	bb	bb
31	1038121	1038866	744	14880	1	3	-2.0	-40.0	bb	bb
32	994674	995418	743	14860	3	1	2.0	40.0	bb	bb
33	1021410	1022151	740	14800	3	2	1.0	20.0	bb	bb
34	1112475	1113217	741	14820	1	0	1.0	20.0	bb	bb
35	837204	837948	743	14860	2	3	-1.0	-20.0	bb	bb
36	986580	987325	744	14880	4	4	0.0	0.0	bb	bb
37	1032797	1033539	741	14820	6	4	2.0	40.0	bb	bb
38	1082737	1083480	742	14840	4	1	3.0	60.0	bb	bb
39	1005477	1006219	741	14820	1	2	-1.0	-20.0	bb	bb
40	1019029	1019774	744	14880	5	2	3.0	60.0	bb	bb
41	946085	946826	740	14800	6	3	3.0	60.0	bb	bb
42	851749	852495	745	14900	3	3	0.0	0.0	bb	bb
43	807373	808116	742	14840	3	2	1.0	20.0	bb	bb
44	1069780	1070523	742	14840	5	3	2.0	40.0	bb	bb
45	1012076	1012819	742	14840	1	1	0.0	0.0	bb	bb
46	1304495	1305238	742	14840	1	1	0.0	0.0	bb	bb

图 9 主从同步结果

Fig. 9 Master slave synchronization result

主站与从站 1 的时钟偏差图和主站与从站 2 的时钟偏差图如图 10 和图 11 所示。经过 200 次测试,从站 1 与主时钟的最大时钟偏差为 100 ns,时钟偏差基本分布在 ± 100 ns;从站 2 与主时钟的最大偏

差为 100 ns,时钟偏差基本分布在 ± 100 ns。相对一主一从的同步测试,由于其加大了链路延时的不对称性以及从站内部的处理延时,增大了时钟偏差,导致同步精度下降。

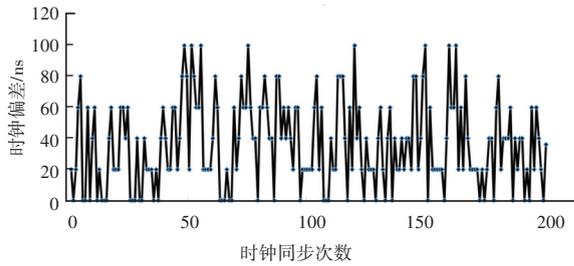


图 10 主站与从站 1 的时钟偏差

Fig. 10 Clock deviation between master station and slave station 1

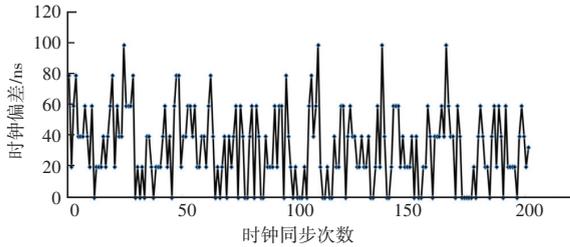


图 11 主站与从站 2 的时钟偏差

Fig. 11 Clock deviation between master station and slave station 2

4 结束语

本文详细分析了 IEEE 1588 协议,并将 IEEE 1588 协议以及本地时钟补偿算法应用于分布式时钟同步系统中,在一主一从的分布式时钟同步基础上,拓展到了一主两从的分布式时钟同步系统。实验证明该方案的同步精度能够达到 ns 级别,同步的时钟设备越少,时钟偏差越小。同步一个时钟设备和两个时钟设备,主从时钟偏差基本分布在 ± 100 ns 之间,能够满足现代工业以太网对高精度时钟的需求,简化了硬件电路设计,不需要专用的 PHY 芯片,时钟戳提取和报文识别都在 FPGA 芯片内部完成,降低开发难度和成本。

参考文献

- [1] 韩一德. 基于 IEEE 1588 协议的高精度时钟同步系统研究与实现[D]. 太原: 中北大学, 2021.
- [2] 谢华. 基于 GPS 信号和 NTP 协议的本地时间同步网络[J]. 工业控制计算机, 2010, 23(6): 75-76.
- [3] 何一航. IEEE 1588 高精度网络时钟同步研究与实现[D]. 武汉: 华中科技大学, 2011.
- [4] EIDSON J C, FISCHER M, WHITE J. IEEE-1588™ Standard for a precision clock synchronization protocol for networked measurement and control systems[C]//Proceedings of the 34th Annual Precise Time and Time Interval Systems and Applications Meeting. 2002: 243-254.
- [5] ZHAO B, WANG N. The implementation of IEEE 1588 clock synchronization system based on FPGA[C]//Fifth International Conference on Intelligent Control and Information Processing. IEEE, 2014: 216-220.
- [6] 刘宝明,冯振乾. 时间敏感网络关键技术与应用研究[J]. 工业控制计算机, 2021, 34(11): 1-4.
- [7] 周峰,刘迎潮,安笑蕊. 嵌入式 IEEE 1588 精确时钟同步实现[J]. 电气技术, 2013(4): 44-47.
- [8] 李培基,李卫,朱祥维,等. 网络时间同步协议综述[J]. 计算机工程与应用, 2019, 55(3): 30-38.
- [9] 王创,郑宾. 交互式多模型 IEEE 1588 时钟同步算法[J]. 自动化技术与应用, 2019, 38(4): 13-17.
- [10] 王贤德,贾连兴. 基于 DTS 和 IEEE1588v2 协议的网络跳变同步技术研究[J]. 装甲兵工程学院学报, 2017, 31(4): 87-92.
- [11] MOHAMMAD J H and THOMAS K and HOWARD S. A Recursive Method for Clock Synchronization in Asymmetric PacketBased Networks [J]. IEEE/ACM Transactions on Networking (TON), 2016, 24(4): 2332-2342.
- [12] KYRIAKAKIS E, SPARSØ J, SCHOEBERL M. Hardware assisted clock synchronization with the IEEE 1588-2008 precision time protocol [C]//Proceedings of the 26th International Conference on Real-Time Networks and Systems. 2018: 51-60.
- [13] QI M, WANG X, YANG Z. Design and implementation of IEEE 1588 time synchronization messages timestamping based on FPGA [C]//2011 4th International Conference on Electric Utility Deregulation and Restructuring and Power Technologies (DRPT). IEEE, 2011: 1566-1570.
- [14] 王攀杰,郭绍忠,侯明,等. 激活函数的对比测试与分析[J]. 信息工程大学学报, 2021, 22(5): 551-557.
- [15] HU J, LI S, SUN G, et al. Squeeze-and-Excitation Networks [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 42(8): 2011-2023.
- [16] LI J, JIN K, ZHOU D L, et al. Attention mechanism-based CNN for facial expression recognition [J]. Neurocomputing, 2020, 411: 340-350.
- [17] LUCEY P, COHN J F, KANADE T, et al. The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression [C]// Computer Vision & Pattern Recognition Workshops. IEEE, 2010: 174-183.
- [18] LIU M, SHAN S, WANG R, et al. Learning Expressionlets on Spatio-temporal Manifold for Dynamic Facial Expression Recognition [C]// 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2014: 94-102.
- [19] DONG N, ZHAO L, WU C H, et al. Inception v3 based cervical cell classification combined with artificially extracted features [J]. Applied Soft Computing Journal, 2020, 93: 151-171.
- [20] Shaheed Kashif, Mao Aihua, Qureshi Imran, et al. DS-CNN: A pre-trained Xception model based on depth-wise separable convolutional neural network for finger vein recognition [J]. Expert Systems With Applications, 2022: 191-199.
- [21] LI X, WU W, LI T, et al. Face Liveness Detection Based on Parallel CNN [C]//Journal of Physics: Conference Series. IOP Publishing, 2020, 1549(4): 042069.
- [22] DING H, ZHOU S K, CHELLAPPA R. Facenet2expnet: Regularizing a deep face recognition net for expression recognition [C]//2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017). IEEE, 2017: 118-126.

(上接第 63 页)