

时间触发以太网确定性调度设计与研究

芦伟, 朱纪洪, 杨佳利, 于少霖

(清华大学 计算机科学与技术系, 北京 100084)

摘要: 随着航空航天领域电子系统向分布式发展, 对机载网络高带宽、低延迟、高可靠性的要求越来越严格。传统的 CAN、ARINC429 等总线正在逐渐被 AFDX、TTP、TTE 等新一代航空总线技术代替。针对 TTE 交换式网络, 设计了一种新的兼容时间触发流量和标准的事件触发流量的 TTE 以太网交换机方案, 保证了时间触发流量的延时是可控的, 并且固定在一定的范围内; 经 Wireshark、Iperf3 等软件测试检验了交换机的交换速率、抖动和丢包率; 经 ModelSim 软件仿真, 并在 FPGA 硬件实验板上通过嵌入式逻辑分析仪和示波器检验了时间触发流量传输的可靠性、延迟范围和时间同步精度, 根据同步周期的不同使同步精度达到了 μs 级、ns 级。

关键词: 时间触发以太网; 实时; 同步

Time-triggered Ethernet Deterministic Scheduling Design and Research

Lu Wei, Zhu Jihong, Yang Jiali, Yu Shaolin

(Tsinghua University, Beijing 100084, China)

Abstract: With the distributed development of electronic systems in the aerospace industry, the requirements for high bandwidth, low latency and high reliability of airborne networks are becoming more and more strict. Traditional Controller Area Network (CAN), ARINC429 and other buses are gradually being replaced by Avionics Full Duplex switched ethernet (AFDX), Time-triggered Protocol (TTP), Time triggered Ethernet (TTE), which are a new generation of aviation bus technology. For the TTE switched network, a new TTE Ethernet switch solution compatible with time-triggered traffic and standard event-triggered traffic is designed. It guarantees that the time-triggered traffic delay is controllable and fixed in a certain range. The switching rate, jitter and packet loss rate of the switch were verified by software tests such as Wireshark and Iperf3. The simulation was performed by ModelSim, and the reliability, delay range and time synchronization accuracy of the time-triggered traffic transmission was verified by embedded logic analyzer and oscilloscope on the FPGA hardware experiment board. The synchronization accuracy is achieved in the μs and ns levels according to the synchronization period.

Keywords: time triggered ethernet; real time; synchronize

0 引言

现在航空电子实时关键系统的架构核心是综合模块化航空电子系统, 未来航空电子系统将向分布式综合模块化航空电子系统发展^[1], 需要大量实时关键数据需要传输。在分布式实时关键系统中有大量的节点, 且通过网络通信, 实时数据传输的可靠性和延时直接影响系统的性能。由于分布式的规模放大了网络通信的不确定性^[2], 所以使网络保持一个高效稳定的状态是提高系统性能的有效途径。针对这个问题, 国外现在提出了 AS6802^[3] 时间触发以太网 (Time triggered Ethernet, TTE) 标准以保证网络的高通信速率、高实时性、高可靠性。时间触发以太网是在以太网的基础上进行实时适应性改造, 在同一网络上无缝连接原有的不同类型的应用业务。同时, 时间触发以太网是介质访问控制 (Media Access Control, MAC) 层服务质量的增强, 对于使用异步标准以太网的应用是透明的, 在分布式

应用间提供确定的, 同步的, 无阻塞的通信并且不被任何异步的标准以太网流量负载影响, 是现在国际上最新的一项新型总线技术。国外的 TTTech 公司已经有相应的开发产品^[4], 已有成功的应用案例^[5] 如美国的 Orion 载人飞船采用基于 1000BASE-CX 物理层和双冗余配置的 TTE 网络综合互连方案。国内现主要是预研和开发阶段。

TTE 系统主要包括时间同步算法, TTE 交换机和 TTE 网络控制器等关键部分。针对嵌入式环境中单交换机网络, 设计了一种采用 FPGA 来实现混合流量实时交换的方案并进行了物理实验测试。文章主要分为五个部分。第一部分介绍 TTE 系统整体概况; 第二部分介绍 TTE 交换机设计的整体结构; 第三部分介绍实现同步的方法; 第四部分进行试验测量验证; 第五部分进行总结和问题分析。

1 TTE 系统的基本组成

整个时间触发以太网系统如图 1 所示。

整个网络中时间触发以太网交换机是核心关键部件, TTE 交换机与 TTE 交换机之间可以单链路或双链路冗余互连; 同时还包括了 TTE 网络控制器、标准以太网控制器和终端, 终端可以接入交换机上不同端口做备份。针对单交换机与各终端星型互连的情况下, 设计的 TTE 交换机简易

收稿日期: 2018-12-29; 修回日期: 2019-01-18。

基金项目: 国家自然科学基金项目 (61701276; 61603210; 61603240)。

作者简介: 芦伟 (1990-), 男, 河北兴隆人, 硕士研究生, 主要从事时间触发以太网方向的研究。

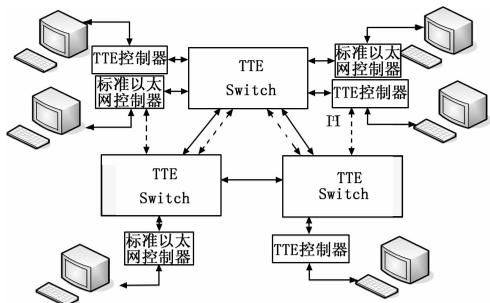


图1 时间触发以太网示意图

结构框图如图2所示。



图2 交换机基础结构示意图

一个时间触发交换机主要分为三层,分别是时间同步层、时间触发(Time-Triggered, TT)交换层、事件触发(Event-Triggered, ET)交换层^[6]。应具有以下功能^[7]:

- 1) 支持时间同步协议。
- 2) 区分 TT 数据帧和 ET 数据帧,并实现 TT 数据帧对 ET 数据帧的抢占。
- 3) TT 数据帧的无存储转发以及对 ET 数据帧按着标准以太网要求的转发。
- 4) 实现对被抢占的 ET 数据帧的重传。

时间同步层所发送的时间同步协议帧采用标准的以太网帧格式,在类型字段中定义一个特定的值(在研发实验中定义的测试值为16位16进制数0x0889)供TTE网络控制器解析。与定义时间同步协议帧一样,在标准以太网帧中类型字段定义另一个特定值(在研发实验中定义的测试值为16位16进制数0x0888)表示此数据帧是TT数据,这样一来通过对接收到的数据帧的类型字段进行判断就能区别出这是时间同步帧,还是TT数据帧,亦或是ET数据帧。

2 交换机硬件设计方案

交换机各功能模块如图3所示。

此设计主要由4个端口和中央的调度模块组成。其中每个端口完成解析数据包、查询目的MAC以及收发功能,每接收到一个ET数据帧,就用一个32位接收描述符与其相匹配,对于接收到的TT数据帧,不占用描述符。同时将时间同步协议帧的发送功能集成到了端口中,并未集成到调度模块,降低调度模块设计的复杂度。

2.1 接收模块

状态机如图4所示。

其中主要分为5个接收状态。初始复位之后先进入不接收状态,没有接收使能时自然过渡到空闲状态。

- 1) 空闲状态:当没有接收使能时或者接收使能信号消失时要回到空闲状态,或者在此阶段接收到了接收使能信

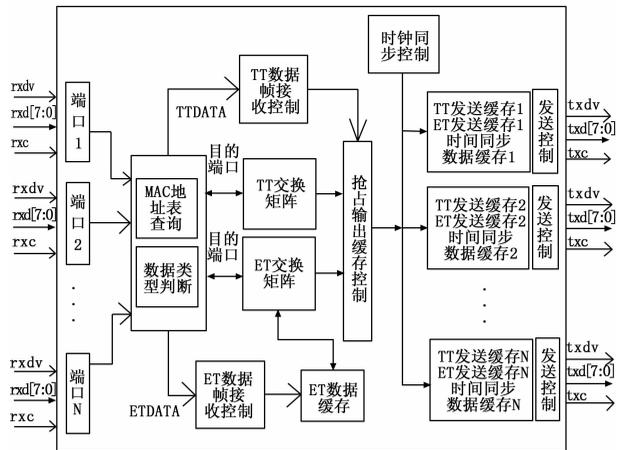


图3 交换机功能模块划分

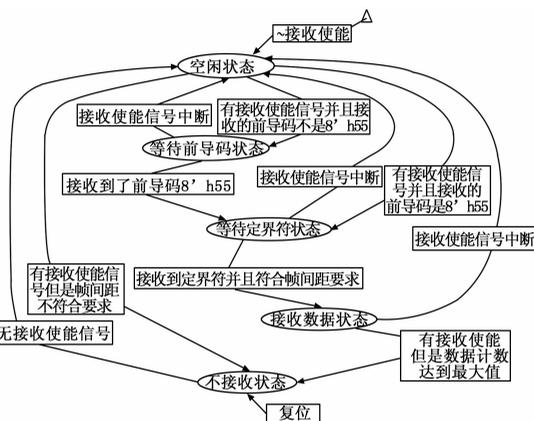


图4 接收模块状态机

号但是帧间距不符合要求就要进入不接收状态;

- 2) 等待前导码状态:当收到了接收使能信号,但是到来的数据不是前导码8位16进制数0x55时,停留在此状态,若是收到了前导码8位16进制数0x55则进入等待定界符状态;

- 3) 等待定界状态:在收到了前导码8位16进制数0x55后,等待定界符8位16进制数0xd5的到来,当定界符到来时,且满足帧间距要求,就进入接收数据状态;

- 4) 接收数据状态:在这个阶段要进行数据的接收,数据接收完后回到空闲状态,若是在接收阶段出现了问题就进入不接收状态。

- 5) 不接收状态:当有接收使能时,出现了错误的情况就一直保持在接收状态,直到接收使能消失时才回到空闲状态。

2.2 调度模块中状态机

调度模块中状态机如图5所示。

2.2.1 对于ET数据流量

调度模块和发送模块协调工作,根据各端口存储的接收描述符的数量,对ET数据帧按着轮询方式^[8]进行调度转发,对于被抢占的ET的调度待TT数据帧转发完后进行重新调度。ET数据帧输出时采用随机存取存储器(Random access memory, RAM)缓存,当需要对ET数据帧进行重

使用。同时下一数据帧到来时接着从此校验错误的的数据帧起始地址开始存储, 直接将原来错误的的数据帧覆盖。调度程序每次查询各接收端口的接收描述符是否为空。若不为空就读取接收描述符的信息, 调度程序根据接收描述符的信息将 N 口的接收到的 ET 数据帧从接收缓冲中读出写入到【M】端口的 ET 发送缓存 ET_fifo。

2) 如果此数据帧是 TT 数据帧, 由调度程序直接将此 TT 数据帧调度到【M】端口的 TT_fifo, 并不写入接收缓存, 也不分配接收描述符。

3) 当定时时间到时, 发送时间同步帧程序会将时间同步数据帧写入到各个端口的 TTime_fifo, 其中也包括【M】端口。

4) 最终由【M】端口根据自己端口中 3 种发送缓存中是否有数据需要发送, 进行按优先级的处理, 然后再将数据输出。

3 时间同步方法

在分布式系统中, 各个物理分散、彼此相对独立的节点各自维护其本地的时钟, 各时钟晶振的频率稳定性、运行环境不一样, 导致即使某一时刻所有的时钟是一致的, 一段时间后这些时钟仍然是不一致。若没有一个统一的全局时钟, 就无法保证系统中任意多个事件间的先后关系, 更无法做到在同一时刻同步解决相同的问题, 因此需要时间同步。

时间同步方法按同步作用路径上可分为主从同步和互同步两大类。主从同步不存在由从节点到主节点的同步作用路径, 如: 泛洪时间同步协议 (The flooding time synchronization protocol, FTSP^[9])、网络时间协议 (Network Time Protocol, NTP^[10])、IEEE1588^[11] 等, 优点是算法稳定性好; 缺点是对主节点精度可靠性要求高、需要维护网络拓扑, 如建立同步树等。互同步任何两个节点间都可能存在同步作用路径, 如: 参考广播时间同步 (Reference Broadcast Time Synchronization, RBS^[12])、成对广播同步 (Pairwise broadcast synchronization, PBS^[13])、共识时钟同步 (Consensus clock synchronization, CCS^[14]) 等, 其优点是容错性较好、可参考信息较多; 缺点是系统收敛情况较复杂, 通信、计算开销较高。

在 TTE AS6802^[2] 中采用的是互同步策略, 指定了两步式同步方法, 如图 8 所示: 在第一步中, 同步主机向压缩主机发送同步协议控制帧, 压缩主机由这些协议控制帧的

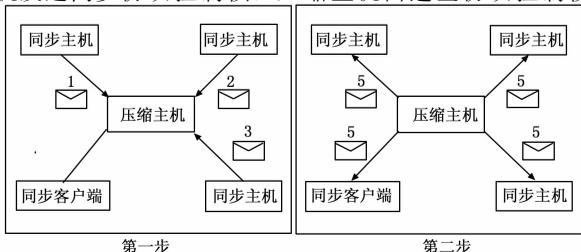


图 8 两步式同步过程

相对到达时间计算出平均值, 然后发送包含时间校正信息的新的协议控制帧作为响应, 反馈给同步主机和同步客户端。同时采用了透明时钟的机制, 容忍了同步协议控制帧在收发过程中的交换延迟, 使得其适用于多交换网络。

而针对本实验中所采用的单交换网络的情况, 设计采用主从式一步同步方法, 如图 9 所示。由星型结构中的交换机每隔固定时间向各终端节点发送含有同步时间戳的时间同步数据帧, 各终端节点解析出时间戳来修正自己的本地时钟, 此方式的时间同步数据帧传输延迟的不确定性更小, 时延更短。

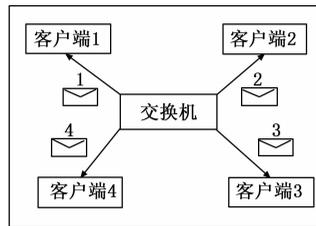


图 9 一步式同步过程

设计的时间同步协议帧的格式如图 10 所示。协议帧采用网络互连协议 (Internet Protocol, IP) 帧头, 用户数据报协议 (User Datagram Protocol, UDP) 帧头, 共 72 字节长 (含 8 字节前导码)。在 UDP 数据字段开头的 2 个字节没有用, 而是从第 3 个字节起, 开始存储 8 个字节的时间戳。



图 10 时间同步数据帧格式

设计的时间戳格式如图 11 所示。

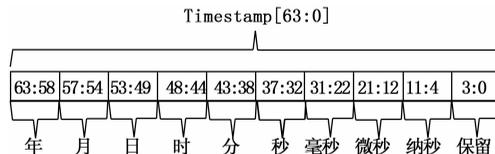


图 11 时间戳格式

时间戳共 64 位, 最后 4 位保留。时间戳以系统时钟周期为一次计数周期, 每隔一个计数周期, 时间戳中对应系统周期的计数值加 1。此系统中 sysclk 为 200 Mhz, 也就是 5 ns 记 1 次数, 时间戳中对应纳秒的计数值加 1, 当计数到 199 时, 对应微秒的计数值加 1, 纳秒计数值归 0, 以此类推。当需要进位时向上进位加 1, 当每个字段计数到最大值时归 0。年字段只设置了 6 位, 是一个偏移值, 最大计数到 63, 实际应用时另设置一个初始值, 初始值加上年字段的偏移值就代表当前的年份。

如图 12 所示, 时间戳记录的是 Txc 时钟域的时间同步数据帧有效信号 Txdv 第一次出现在物理链路上, 经 sysclk 打两拍后同步到 sysclk 时钟域的时刻 Ttimestamp:

$$Ttimestamp = Tsend + t1 + Ts + Ts \quad (1)$$

式中, Ttimestamp 指时间戳时间, Tsend 指物理线路发出时刻, t1 指的是使能信号出现到 sysclk 第一次采集到使能

信号之间的时间差, T_s 指 *sysclk* 的时钟周期。在图中可以看到同步协议帧中所存储的时间戳是随时间同步数据帧的发出随打入, 所以从发送时间同步数据帧的定时时刻起, 到物理链路上发出同步帧之间的等待时间也会记录到时间戳中, 不用再另外进行时间补偿。

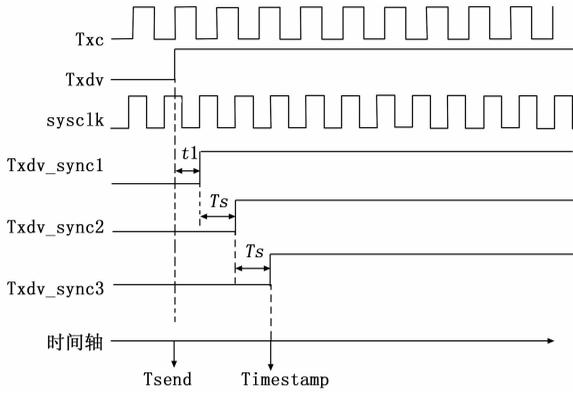


图 12 时间戳打入时刻仿真图

在接收端采用同样的策略, 接收端完全接收完 8 个字节时间戳时 (也就是接收数据字节计数从 0 到 59 时, 此时共 60 个字节) 产生 *Rxc* 时钟域的 *change* 信号, 由 *sysclk* 打两拍同步到 *sysclk* 时钟域上, 然后接收端再根据时间戳的内容修正本地时间, 如图 13 所示。

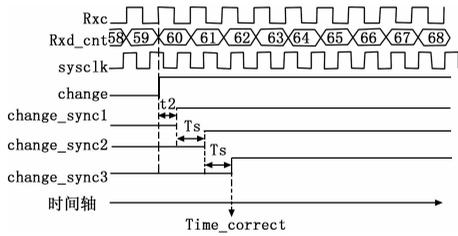


图 13 接收端解析时间戳

接收端修正本地时间时, 不能直接使用时间戳修正。因为从时间同步数据帧出现在物理链路上, 到 *change* 信号同步到 *sysclk* 时钟域上之间的时间差为 T_delay :

$$T_delay = T_trans + T60 + t2 + T_s + T_s \quad (2)$$

式中, T_trans 指数据帧在链路上传播时间, $T60$ 指接收端接收 60 个字节数据的时间, $t2$ 指 *change* 信号出现到 *sysclk* 第一次采集到 *change* 信号之间时间差, T_s 指 *sysclk* 的时钟周期。由于发送端和接收端的 *sysclk* 都是采用 200 MHz, 所以时间戳接收端相对于时间戳发出端, 本地时间修正为 $Time_correct$:

$$Time_correct =$$

$$Timestamp + (t2 - t1) + T60 + T_trans \quad (3)$$

式中, $(t2 - t1)$ 在 $(-5 \text{ ns}, +5 \text{ ns})$ 之间, T_trans 很小暂时忽略。而接收端网口 *phy* 时钟是 125 MHz, 8 ns 一周, 而接收端以 200 MHz 主时钟计数, 所以将 $T60$ 折算成需要补偿的时间戳计数值正好是 $60 \times 8/5 = 96$, 是个整数, 便于程序设计。这也是为什么要将时间戳存储在数据帧的第 53 字节到第 60 字节之间。所以最终本地时间的计数值修正为:

$$Time_correct = timestamp + 96 \pm 1 \quad (4)$$

取平均值:

$$Time_correct \approx timestamp + 96 \quad (5)$$

4 试验测量和分析

实验测量是基于 ALINX7101 实验板。该板使用的 FPGA 型号为 XC7A100T-2FGG484I, 属 Xilinx 公司 Artix-7 系列的产品, 速度等级为 2, 温度等级为工业级。有 4 个千兆以太网接口。

4.1 事件触发流量速率、抖动、丢包率测量

先取 1 个节点当交换机, 将写好的 verilog^[15] 程序烧入, 将 4 个端口分别连上 4 台测试电脑 PC1 至 PC4, 设置 IP 地址分别为 192.168.1.1 至 192.168.1.4, 掩码为 255.255.255.0。因为是局域网内测试, 暂时不需要设置网关。在每台 PC 上开起 Iperf3 测试软件, 每台 PC 既当服务端又当客户端, 服务端测试端口为 12345, 测试时间是 10s, 测试软件的发包速率是 1 Gbit/s, 发送 UDP 格式的数据包。经测试可查看网络速率, 抖动, 丢包率等情况。如图 14 所示。

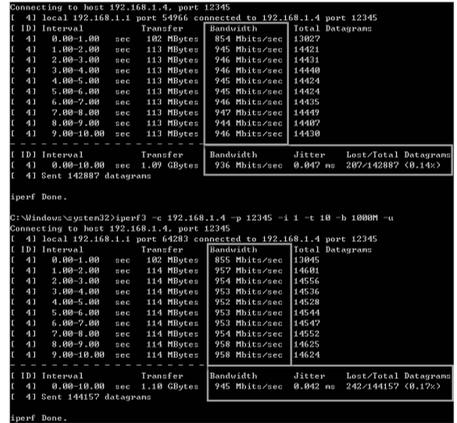


图 14 测试网络传输速度、抖动、丢包率

测试结果显示网络速率平均值为 940 Mbit/s 左右, 抖动平均值为 0.04 ms 左右, 丢包率为 0.15% 左右。

但由于是软件测试, 测试结果只作为大概一个性能的估计, 其中的测试数据包的丢包率测试并不是十分准确。采取用硬件测试的方法, 将 4 号端口连接的 IP 地址为 192.168.1.4 的台式机换成一个发固定数量测试数据帧的 FPGA 开发板, 测试数据目的 MAC 为 1 号台式机网卡 MAC, 帧间距在 1Gbit/s 的速率下为标准的 96 ns。在 1 号台式机上打开 wireshark 抓包软件, 在其中端口的 interface details 中的 statistics 标签下, 可以看到该机网卡上收到包的统计信息, 测试开发板每次固定发 430 000 个数据包。在未开始发送测试数据前, 如图 6 中右侧两列所示, 只有 1 号台式机向外发送数据, 并未有接收数据统计; 当开始发送测试数据时, 此时其他端口无数据流量不引入对 1 号口统计接收数据包个数的影响, 所统计的数据包个数完全来自于 4 号端口, 如图 15 中所示, 并未发生丢包。

4.2 时间触发流量延迟测试

在 modelsim 中编写 testbench, 指定 1 端口一直在发送

Statistics	
Transmit OK	252
Transmit Error	0
Receive OK	430000
Receive Error	0
Receive but no Buffer	0
Directed bytes transmitted w/o errors	0
Directed packets transmitted w/o errors	0
Multicast bytes transmitted w/o errors	16180
Multicast packets transmitted w/o errors	154
Broadcast bytes transmitted w/o errors	10553
Broadcast packets transmitted w/o errors	98
Directed bytes received w/o errors	444620000
Directed packets received w/o errors	430000
Multicast bytes received w/o errors	0
Multicast packets received w/o errors	0
Broadcast bytes received w/o errors	0
Broadcast packets received w/o errors	0
Packets received with CRC or FCS errors	0
Packets queued for transmission	0

图 15 FPGA 测试板测试丢包情况

广播包, 2 端口先发送 2 个目的地址为 3 端口的标准以太网数据帧, 后发送 2 个目的地址为 4 端口的时间触发数据帧, 指定抓取交换机调度模块的相关信号, 可以观察到 2 端口的时间触发流量抢占了 4 端口对标准流量的输出; 时间触发流量从进入到输出之间有抢占的情况延迟值大概是 366.5ns, 如图 16 所示。无抢占直接转发时延迟大概是 262.5ns, 如图 17 所示。

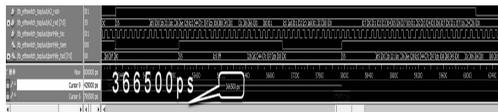


图 16 仿真 TT 帧有抢占时延迟

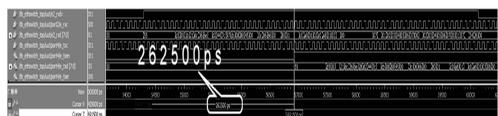


图 17 仿真 TT 帧无抢占时延迟

使用 FPGA 的内嵌式逻辑分析仪, 抓取物理信号, 有抢占时结果如图 18 所示, 看到当收到类型为 TT 数据帧的时候, 抢占对类型为 ET 数据帧的传输, ET 数据帧有效信号中断, 等待标准帧间距后发送 TT 数据帧。有抢占时 TT 数据帧在交换机上从接收到发出的延迟大概是 $(1808 - 1761) * 8 = 376 \text{ ns}$ 。无抢占时如图 19 所示, 当收到类型为 TT 数据帧的时候, 输出链路上无数据发送时, TT 数据帧在交换机上的传输延迟大概是 $(1762 - 1729) * 8 \text{ ns} = 264 \text{ ns}$ 。

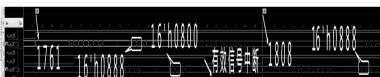


图 18 ILA 抓取 TT 帧有抢占时延迟



图 19 ILA 抓取 TT 帧无抢占时延迟

4.3 时间同步效果实验测量

为了观测同步效果, 将 PC 换成 7101FPGA 板卡, 向板卡中烧入可以解析时间同步帧的程序, 同时在程序中设定

每 1 ms 通过一个 I/O 口输出一个持续时间 200 us 高电平, 将 2 个板卡上的 I/O 口连接到一台示波器上, 观察上下 2 个高电平之间的时间差。如图 20 所示。通过对时间差的计算就可以算出同步精度是多少。

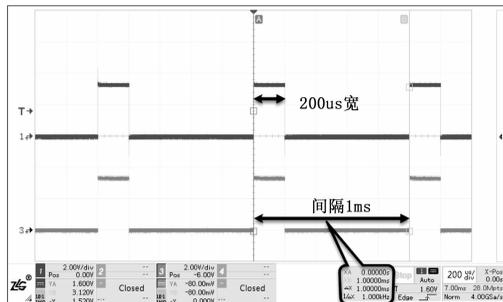


图 20 定时输出高电平方波

4.3.1 1 s 同步一次

以 1 s 为周期进行同步时, 1 s 钟内会有 1 000 次高电平出现, 系统刚上电测试时, 脉冲之间最小相差 7 ns 左右, 如图 21 所示; 最大相差 582 ns 左右, 如图 22 所示。但随着系统运行时间增加, 脉冲之间最大相差会增长到 600 ns 左右。

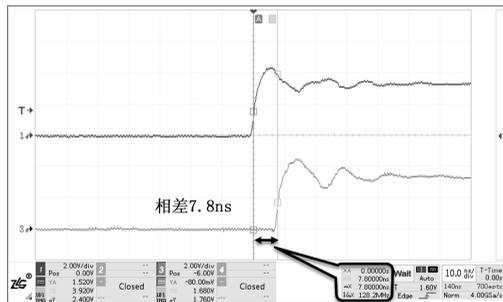


图 21 1 s 同步一次时最小同步时间差

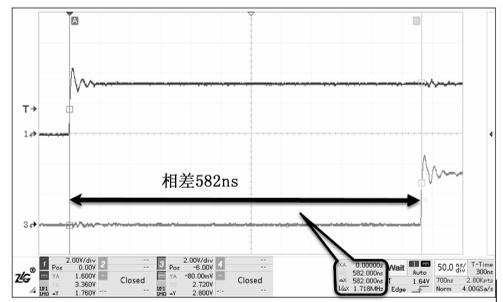


图 22 1 s 同步一次时最大同步时间差

4.3.2 10 ms 同步一次

若是以 10 ms 为周期进行同步时, 系统刚上电测试时, 脉冲之间同步的时间差最小相差 200 ps 左右, 如图 23 所示。最大相差 24 ns 左右, 不会超过 25 ns, 如图 24 所示; 但随着系统运行时间增加, 芯片温度升高, 最大相差会增长到 40 ns 左右。

4.3.3 20 ms 同步一次

若是以 20 ms 为周期进行同步时, 系统刚上电测试时, 脉冲之间时间差最小相差 200 ps 左右, 如图 25 所示; 最大相差 28 ns 左右, 不会超过 30 ns, 如图 26 所示。但随着系

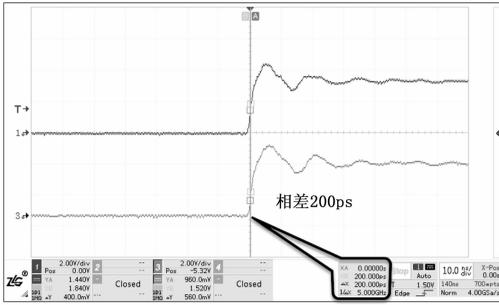


图 23 10 ms 同步一次时最小同步时间差

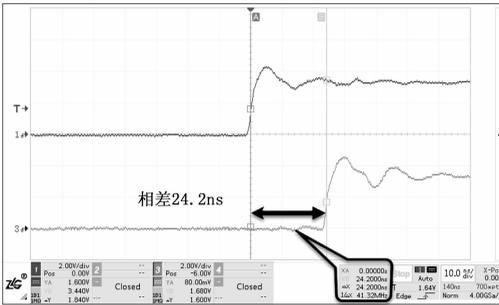


图 24 10 ms 同步一次时最大同步时间差

统运行时间增加, 最大相差会增长到 45 ns 左右。

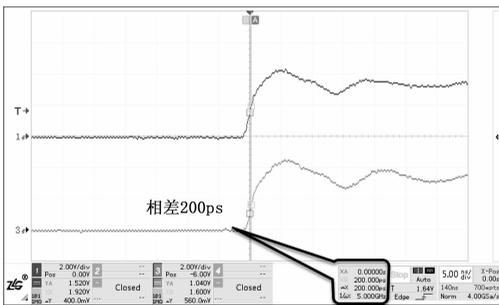


图 25 20 ms 同步一次时最小同步时间差

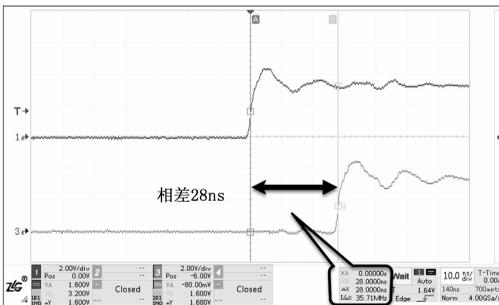


图 26 20 ms 同步一次时最大同步时间差

5 总结和问题分析

此交换机设计实验性的融合了 TT 数据和 ET 数据的无缝连接转发, 并实现了 TT 数据转发的固定延时的保证, 同步时间精度也达到了 μs 级、ns 级。针对于 TT 周期性数据的转发并没有采用将离线调度表存储到交换机上的策略, 而是采用的直接基于抢占的方式, 需要判断出类型域后才能进行抢占, 这样做相对于基于离线调度提前预置好输入与输出端口之间的连接线路而言, 造成了一定延时的增加,

但降低了交换机设计的复杂度, 同时增加了对于转发突发性 TT 数据帧的灵活性。也带了一些挑战, 比如是否能提前预知将要到来的是 TT 数据帧以及 TT 帧将要去往的目的端口, 那么就可以像基于离线调度表一样, 真正做到几乎是无延迟的转发; 针对于突发性 TT 数据帧的转发, 现在采用的避免冲突的策略是在突发性 TT 数据流量前向交换机查询输出端口的 TT 发送缓存是否可用, 如何更好的解决避免两个来自不同端口的 TT 流量同时突发性的发往同一个端口, 又是一个问题。

参考文献:

- [1] [奥] 罗兰·沃尔夫格 (Roland Wolfig). 综合化模块化航空电子系统的分布式平台: 对未来航空电子系统及其认证需求的见解 [M]. 北京: 航空工业出版社, 2015.
- [2] 张凤登. 分布式实时系统 [M]. 北京: 科学出版社, 2014.
- [3] Aerospace Standard SAE AS6802 [Z]. 2011
- [4] Plankensteiner M. TTEthernet: A Powerful Network Solution for All Purposes: [J]. Delft University of Technology, 2010.
- [5] 贾琪明, 李峭, 熊华钢. TTE 网络混合关键性通信的仿真与性能分析 [J]. 电光与控制, 2015, 22 (9): 7-10.
- [6] 高鹏飞. 时间触发以太网交换机设计 [D]. 西安: 西安电子科技大学, 2014.
- [7] Steinhammer K, Grillinger P, Ademaj A, et al. A time-triggered ethernet (TTE) switch [C]. Proceedings of the conference on Design, automation and test in Europe; Proceedings. European Design and Automation Association, 2006: 794-799.
- [8] Afridi S, Ahmad J, Yameen M, et al. The Quantitative Analysis of Round Robin Matching Scheduling Algorithms for VOQ Packet Switch Architecture [J]. 2012, 3 (4): 709-712.
- [9] Maróti M, Kusy B, Simon G, et al. The flooding time synchronization protocol [A]. in Proceedings of the 2nd international conference on Embedded networked sensor systems [C]. ACM, 2004, pp. 39-49.
- [10] Mills D L. Internet time synchronization: the network time protocol [J]. IEEE Transactions on communications, 1991, 39 (10): 1482-1493.
- [11] 袁振华, 董秀军, 刘朝英. 基于 IEEE1588 的时钟同步技术及其应用 [J]. 计算机测量与控制, 2006, 14 (12): 1726-1728.
- [12] Elson J, Girod L, Estrin D. Fine-grained network time synchronization using reference broadcasts [J]. ACM SIGOPS Operating Systems Review, 2002, 36 (SI): 147-163.
- [13] Noh K L, Serpedin E, Qaraqe K. A New Approach for Time Synchronization in Wireless Sensor Networks: Pairwise Broadcast Synchronization [J]. IEEE Transactions on Wireless Communications, 2008, 7 (9): 3318-3322.
- [14] Maggs M K, O'keefe S G, Thiel D V. Consensus clock synchronization for wireless sensor networks [J]. IEEE sensors Journal, 2012, 12 (6): 2269-2277.
- [15] 何 宾. EDA 原理及 Verilog 实现 [M]. 北京: 清华大学出版社, 2010.