

# 基于 ARM9200 体系的 IEEE 1588 硬件实现

戴 辉<sup>1</sup>, 涂 岸<sup>2</sup>

(1. 贵州大学 继续教育学院, 贵州 贵阳 550002;

2. 武汉大学 电子信息学院, 湖北 武汉 430072)

**摘 要:** 随着通信技术的发展, 通信系统对网络之间的同步精度越来越高, 传统的 GPS 方式以及 NTP 同步越来越不能满足系统的发展。2004 年, 安捷伦公司提出 PTP(IEEE 1588)同步方式, 该方式能达到次纳秒级精度, 可满足大型通信网络要求。本文主要分析 IEEE 1588 的同步机制, 并最终以太 AT91RM9200 为核心实现 IEEE 1588 同步系统。

**关键词:** IEEE 1588; ARM9200; DP83640; 嵌入式系统

中图分类号: TP393.04

文献标识码: A

文章编号: 1674-7720(2010)13-0063-03

## Hardware implementation of IEEE 1588 based on ARM9200 system

DAI Hui<sup>1</sup>, TU An<sup>2</sup>

(1. School of Continuing Education, Guizhou University, Guiyang 550002, China;

2. School of Electronic Information, Wuhan University, Wuhan 430072, China)

**Abstract:** With the development of communication technologies, the communication systems are getting higher requests for the precision of network synchronization. In 2004, Agilent Technologies introduced PTP (IEEE 1588) synchronization, which can achieve sub-nanosecond accuracy and meet the requirements of large communication networks. This article analyzes the IEEE 1588 synchronization mechanisms, and using AT91RM9200 as the core technology to realize IEEE 1588 synchronous system.

**Key words:** IEEE 1588; ARM9200; DP83640; embedded system

在通信领域,“同步”概念是指频率的同步,即网络各个节点的时钟频率和相位同步,其误差应符合相关标准。目前,在通信网中,频率和相位同步问题已经基本解决,而时间的同步还没有得到很好的解决。

在早期,通信系统负载较低,对时间的同步精度也较低,一般的 GPS 和 NTP 方式的同步<sup>[2]</sup>就可以满足 ms 级别同步要求,但是随着通信系统的发展,特别是 3G 技术的发展,传统的同步方式越来越不能满足同步要求。2004 年,安捷伦公司提出的 IEEE 1588<sup>[3-4]</sup>同步系统具有里程碑意义,它标志着网络同步开始进入次纳秒精度领域。而 2006 年,国家半导体推出的 DP83640 芯片将 IEEE 1588 时间戳方案在硬件中首次实现,使次纳秒级别网络同步精度实现变为可能。

### 1 IEEE 1588 实现原理

IEEE 1588,即精密时间协议 PTP(Precision Time Protocol)由安捷伦于 2004 年提出,是为了克服以太网实时

性不足而规定的一种对时机制。精密时间协议可以估算出数据包在网络上的往返延迟,并可独立地估算设备时钟偏差。网络时间协议的工作原理说明如下<sup>[5-7]</sup>:

(1)现场设备 A 发送一个 PTP 包给现场设备 B,并记录下数据包离开 A 的时间,该时间戳为  $t_1$ 。

(2)当此 PTP 包到达现场设备 B 时,现场设备 B 加上自己接收到该数据包的时间戳,该时间戳为  $T_{S1}$ 。

(3)现场设备 A 发送一个 Follow Up 消息给现场设备 B,通知 B 时间  $t_1$ 。

(4)当此 PTP 包离开现场设备 B 时,并记录下数据包离开 B 的时间  $t_2$ 。

(5)当现场设备 A 接收到该响应包时,加上一个新的时间戳,该时间戳为  $T_{m2}$ ,整个同步过程如图 1 所示。

(6)现场设备 A 再发送 Follow Up 消息给现场设备 B,通知 B 时间  $T_{m2}$ 。

至此,现场设备 B 已拥有足够的信息来计算 2 个重

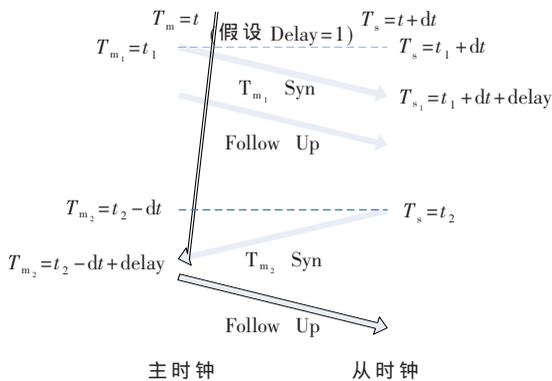


图1 PTP时间同步示意图

要的参数:一是 PTP 数据包来回一个周期的时延,二是现场设备 A 和现场设备 B 的时钟差。于是现场设备 B 能够设定自己的时钟与现场设备 A 同步。

可以根据下面公式算出现场设备 A 与现场设备 B 之间的时间差和延时:

$$\text{时间差: } dt = [(T_{s1} - t_1) - (T_{m2} - t_2)] / 2$$

$$\text{延时: } Delay = [(T_{s1} - t_1) + (T_{m2} - t_2)] / 2$$

## 2 系统硬件电路设计

### 2.1 支持 PTP 协议嵌入式接口板设计

根据 PTP 协议框架,支持 PTP 协议的嵌入式接口板硬件框图如图 2 所示。



图2 基于 DP83640 的接口板硬件框图

DP83640 高精度 PHYTER 收发器芯片内置高精度 IEEE 1588 时钟,并设有由硬件执行的时间标记功能,可为接收及发送的信息包印上时间标记。DP83640 最大的特色在于将 IEEE 1588 高精度时间协议(PTP)功能设于芯片的物理层之内,因此可以在最靠近网线的位置记录时间标记,而且分辨率达 8 ns,不但可为系统提供最准确的高精度控制功能,也确保采集回来的数据出现最少的抖动。

DP83640 预载 4 个时钟更新算法,用户可以利用任何一个算法调节电路板上的 PTP 时钟,而且偏差率保证不超过 10 亿分之一(即 1ppb<sup>[8]</sup>的准确度)。

DP83640 也可与部分符合 IEEE 1588 规定的处理器搭配一起使用,为测试及测量仪表等设备提供更准确的时钟同步功能。其系统连接后时钟关系图如图 3 所示。

### 2.2 具体电路设计

系统实际设计电路主要包含如下部分:核心处理器 ARM9200;存储电路,其中包含了 SDRAM 和 ROM 两部分的电路设计;网络接口电路,其中使用 DP83640 通过

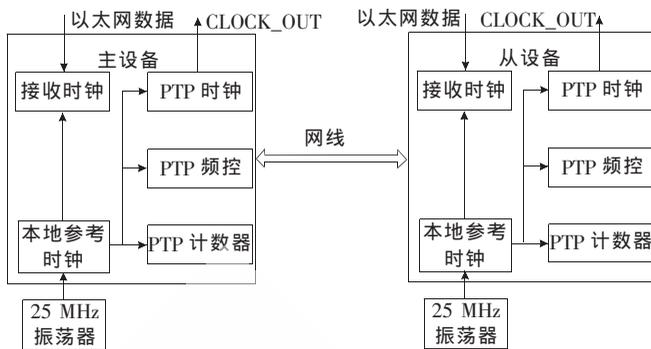


图3 主从系统时钟关系图

RMI 接口总线与 ARM9 处理器相连,把 PTP 报文的时间戳信息和当前时钟读数发送给 ARM9 处理器,同时接收来自 ARM9 处理器的时钟调整值来调整本地时钟。DP83640 通过 RMI 总线来获取以太网报文比特流而不是直接从物理层获取,从而实现 10 Mb/s 和 100 Mb/s 两种速率的自适应处理。

## 3 系统软件设计

系统软件结构如图 4 所示。

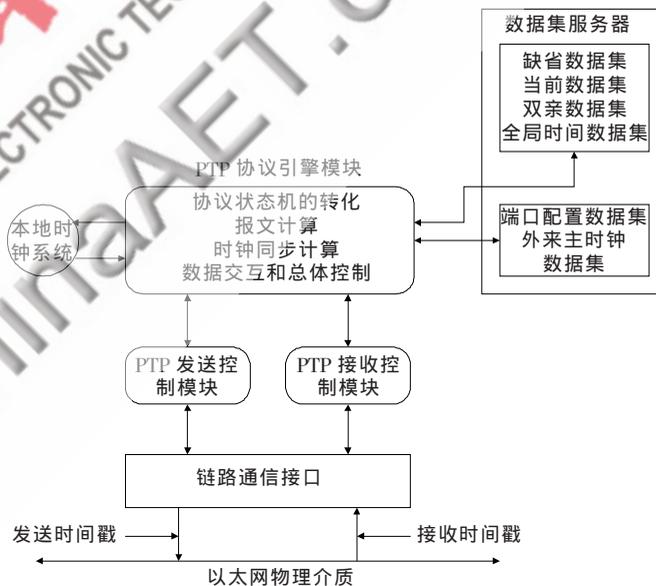


图4 系统的体系结构示意图

### 3.1 PTP 协议状态机模块

PTP 协议状态机主要负责管理设备上电时的初始化状态、出现异常情况下的故障状态、通常情况下的侦听状态、主时钟状态、从时钟状态和未校准状态,以及各状态之间的转化。

### 3.2 最佳主时钟选择模块

最佳主时钟算法模块主要用于选择本地网络中的最佳时钟作为主时钟,同时决定本地时钟应处的状态。

实现最佳主时钟的方法包括两个步骤:(1)从本地时钟系统通信端口中选出一个质量最好的同步报文;(2)根据上述结果,比较此时的主时钟和本地时钟属性,决

## 网络与通信 Network and Communication

定本地设备 PTP 协议引擎所处的状态。步骤(1)采用数据集比较算法模块来实现,步骤(2)采用状态决断算法模块来实现。该模块定义了 STATE\_DECISION\_ALGORITHM() 函数来进行状态的决断和数据集的更新。在缺省数据集的层数是 1 或 2 的前提下,调用 DATASET\_COMPARISON\_ALGORITHM 模块来判定本地时钟缺省数据集  $D_0$  和较好时钟  $E_{r_{best}}$  的数据集哪个更好。

通过比较,如果是缺省数据集  $D_0$  更好,则根据状态决断代码 M1,调用函数 CreateMasterClock() 根据缺省数据集  $D_0$  的属性值来构建主时钟,并且将  $D_0$  状态更改为 PTP\_MASTER,这代表  $D_0$  是目前最好的主时钟;否则,调用函数 CreateSlaveClock(),根据  $E_{r_{best}}$  的属性来构建从时钟, $E_{r_{best}}$  的状态则是 PTP\_SLAVE。

在主时钟的层数大于 2 的情况下,调用 DATASET\_COMPARISON\_ALGORITHM 模块,以对本地时钟缺省数据集  $D_0$  和最好时钟  $E_{r_{best}}$  进行判定。通过比较,如果是缺省数据集  $D_0$  更好,则调用 CreateMasterClock(),根据状态决断代码 M2,根据缺省数据集  $D_0$  的属性值来构建主时钟,并且更改  $D_0$  状态为 PTP\_MASTER;否则,将调用 DATASET\_COMPARISON\_ALGORITHM 模块将  $E_{l_{best}}$  与  $E_{r_{best}}$  进行比较,如果比较结果相同,根据状态决断代码 S1,根据  $E_{l_{best}}$  的属性来构建从时钟,更改  $E_{l_{best}}$  状态成 PTP\_SLAVE。如果经过比较  $E_{best}$  有更好的属性值,则根据状态决断代码 M3,根据  $E_{r_{best}}$  的属性来构建从时钟,并设置它的状态为 PTP\_SLAVE;否则,根据状态决断代码 P2,根据  $E_{best}$  的属性来构建主时钟,改变它的状态为 PTP\_MASTER。

最佳主时钟选择模块示意图如图 5 所示。

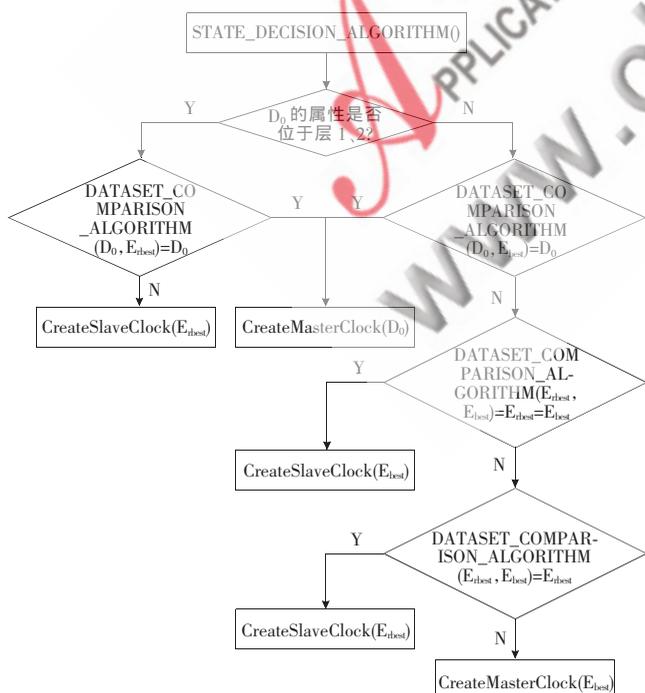


图 5 最佳主时钟选择

## 3.3 时钟调整模块

本地时钟同步算法是 PTP 协议中十分关键的部分,主要用来调整本地时间,与主时钟的基准时间保持一致。主时钟被选择出来后,就开始调用同步报文发送函数 PTP\_Send\_PTP\_SYNC\_MESSAGE(),构造同步报文并对报文初始化,记录发送出去的时间戳,再调用 udp\_send() 将报文打包并加入发送缓冲区进行发送。

调用跟随报文发送函数 PTP\_Send\_PTP\_FOLLOW\_MESSAGE(),构造跟随报文并对报文初始化,记录发送出去的时间戳,然后调用 udp\_send() 将报文打包并加入发送缓冲区进行发送。

从时钟 udp\_rcv() 设置接收缓冲区分别先后接收了同步报文和跟随报文,记录下接收的时间打上时间戳,再调用函数 PTP\_RECV\_PROCESS() 判断接收的是哪种报文,如果是同步报文,则调用函数 PTP\_RECV\_SYNC(),然后对报文解包,取出报文从主时钟发送过来的时间。如果是跟随报文,则调用函数 PTP\_RECV\_FOLLOWUP() 对报文解包,取出报文从主时钟发送过来的时间。

在这个函数最后,调用了 PTP\_Send\_Delay\_Req() 函数,它构造延迟请求报文并对报文初始化,记录发送出去的时间戳,然后调用 udp\_send() 将报文打包并加入发送缓冲区进行发送;而主时钟的缓冲区一旦有报文,则主时钟立即调用 udp\_rcv() 记录下接收的时间打上时间戳,再调用函数 PTP\_RECV\_PROCESS() 判断接收的是哪种报文,如果是延迟请求报文,则函数 PTP\_RECV\_DELAY\_REQ() 立即调用 PTP\_Send\_PTP\_REQ\_RESP(), 从时钟根据收到的 4 个报文收发时间,计算出主从偏差,对自己的时间进行校正。

整个程序流程如图 6 所示。

## 4 系统测试结果与结论

系统测试将两片 DP83640 连接到 1 Hz 信号输出,利用 DP83640 捕获上升沿时间戳实现,在 Linux 系统中,利用 MII 总线读取 DP83640 时间戳寄存器,数据结果如表 1 所示。

表 1 系统测试结果

记录时间序列	现场设备 A	现场设备 B	同步精度/ns
1	498 055 s 290 857 000 ns	498 055 s 290 857 010 ns	10
2	498 056 s 290 853 000 ns	498 056 s 290 853 012 ns	12
3	498 057 s 209 841 010 ns	498 057 s 209 841 021 ns	11
4	498 058 s 290 867 100 ns	498 058 s 290 867 115 ns	15
5	498 059 s 290 898 160 ns	498 059 s 290 898 174 ns	14
6	498 060 s 291 166 375 ns	498 060 s 291 166 390 ns	15
7	498 061 s 290 838 255 ns	498 061 s 290 838 270 ns	15
8	498 062 s 291 221 260 ns	498 062 s 291 221 278 ns	18

实验结果,表明利用 DP83640 达到了 10 ns 级别的同步精度,完成了系统的设计目标。

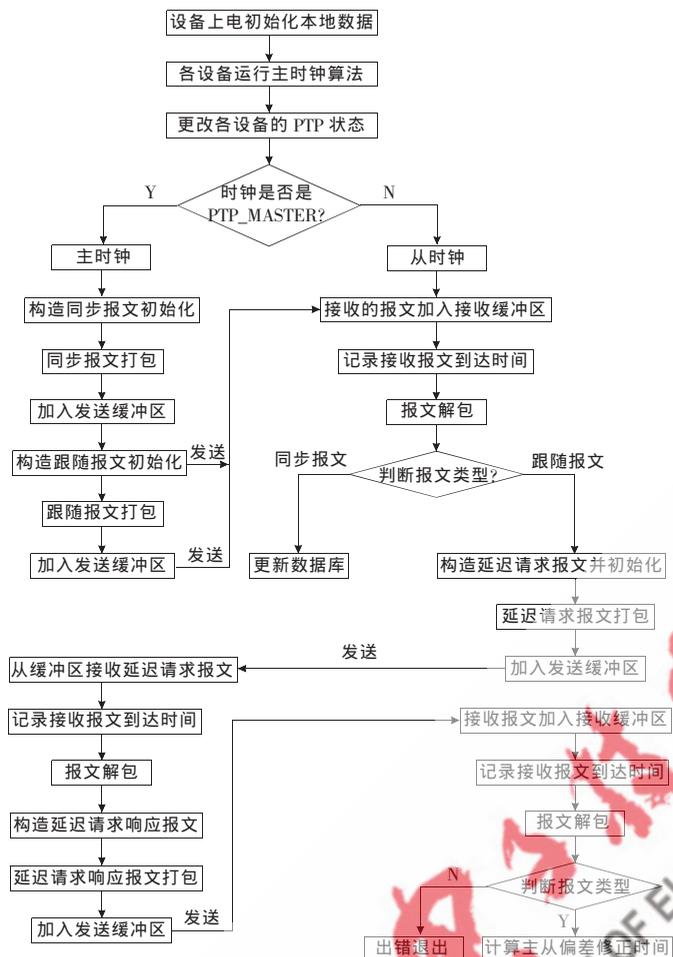


图 6 系统的体系结构示意图

IEEE 1588 提供的时间同步在理想条件下,网络完全

对称),能够达到极高的同步精度,为通信系统同步建设提供了更完美的解决方案。

参考文献

[1] 秦云川,徐大令,李彤.测量仪器总线技术的发展与现状[J].中国仪器仪表,2005(8):48-52.

[2] Ji Ling.LXI-2000 年代的新一代仪器系统总线[J].国外电子测量技术,2005(2):1-3.

[3] TC9 -Technical Committee on Sensor Technology, 1588 IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems [C]. 2002.

[4] EIDSON J C. IEEE 1588: Applications in Measurement, Control and Communication. 2005.

[5] 李鉴,孙宝成,周雷.一种新的局域网时间同步方法[J].计算机与数字工程,2005,33(9):161-164.

[6] 王建珍.计算机网络时间同步技术应用研究[J].山西电子技术,2005(4):7-8.

[7] 安荣亮.局域网时间同步原理及设计[J].电光系统,2005(2):47-49.

[8] Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI. <http://www.networksorcery.com/enp/rfc/rfc2030.txt>, 2009.

(收稿日期:2010-01-07)

作者简介:

戴辉,男,1964年生,硕士,主要研究方向:计算机应用的研究和开发。

涂岸,男,1986年生,硕士研究生,主要研究方向:自动化、通信协议。