

## 基于 IEEE 1588 的时钟同步技术在分布式系统中的应用

李聪, 高丽

(西安工业大学 电子信息工程学院, 陕西 西安 710032)

摘要:为实现分布式系统高精度同步数据采集及实时控制,提出一种基于 IEEE 1588 协议的分布式系统时钟同步方法。通过分析影响同步精度的因素,采用 FPGA 设计时间戳生成器,并且采用晶振频率补偿时钟解决时间戳的精确获取和从时钟相对主时钟的频率纠偏等问题。

关键词: IEEE 1588; 分布式系统; 时间戳; FPGA; 时钟同步

中图分类号: TH86

文献标识码: A

文章编号:1674-6236(2009)12-0054-03

### Application of distributed system based on IEEE 1588 clock synchronization

LI Cong, GAO Li

(School of Electronic Information Engineering, Xi'an Technological University, Xi'an 710032, China)

**Abstract:**In order to realize high accuracy synchronous data acquisition and control in distributed system, a distributed system clock synchronization method based on IEEE1588 protocol is proposed. Through analyzing the factors which affects the synchronized precision, using field programmable gate array (FPGA) designs timestamp generator oscillator frequency compensation solves the problems of timestamp precision acquisition and error correction from relative master time frequency.

**Key words:** IEEE 1588; distributed system; timestamp; FPGA; clock synchronization

## 1 引言

以太网技术的高速发展为分布式系统在工业领域的广泛应用创造了条件。以太网具有成本低、可靠性高、传输速度快、通用性强、开放性好、发展潜力大等优点。但以太网也存在网络的不确定性、实时性不强等缺陷。随着系统日益庞大复杂,分布化程度越来越高,对时钟同步的要求也逐步提高,虽然随着带宽的不断提高以及采用星形网络拓扑结构等减少冲突可能的技术的应用,以太网的不确定性、实时性有一定程度的提高,但由于以太网自身冲突检测的载波帧听多路访问(CSMA/CD)机制<sup>[1-3]</sup>,设备层和 I/O 层的数据采集与传输问题,以及 TCP 及 UDP 上的误差检测及翻译障碍等,以太网的实时性问题并未得到根本解决,仍不能满足精确定时的要求。

网络测量和控制系统的精密时钟同步协议标准 IEEE 1588 适用于以太网、CAN 总线和 PROFIBUS 等。IEEE 1588 的基本功能是使分布式网络内所有从时钟与主时钟保持同步,该标准定义一种精确时间协议 PTP (Precision Time Protocol),用于对标准以太网或其他采用多播技术的分布式总线系统中的传感器、执行器以及其他终端设备中的时钟进行微秒级同步。早期的网络时间协议(NTP)只有软件,而 PTP 协议同时使用硬件和软件,从而获得更精确的定时同步。PTP 针对

相对本地化、网络化的系统,子网或内部组件相对稳定的环境,特别适合于分布式系统在工业自动化方面的应用<sup>[4]</sup>。

## 2 时钟同步原理

理论上,为每个系统节点配备 GPS 模块可得到精确的时间,这样系统中所有节点的时钟就和标准时间一致,从而达到同步;但分布式系统中节点数量大, GPS 设备成本高、能耗大,且结构复杂,抗屏蔽性差,且出于安全性考虑排除大规模使用 GPS 的时钟同步方式。而基于以太网的 IEEE 1588 是通过在分布式系统各节点间交换时间报文的方法达到整个系统的时间同步<sup>[5]</sup>。IEEE 1588 通过交换报文来确定主时钟(Master)和从时钟(Slave)之间的时间偏移及报文传输的网络延迟。图 1 为 IEEE 1588 原理图。

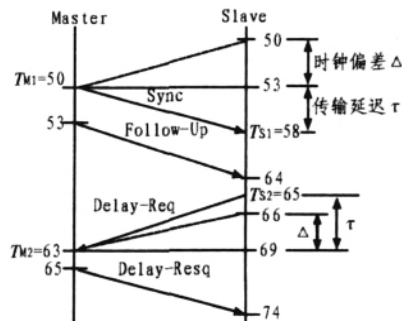


图 1 IEEE 1588 原理图

由图 1 可知,主时钟的节点按照定义的间隔时间(缺省是 2 s)周期性地向网络上所有从时钟节点发送“同步报文”(Sync),同时主时钟节点记录同步报文实际发送的时间戳,并在随后的“跟进报文”(Follow-up)中传送该精确时间戳  $T_{M1}$ 。这样,对传递和接收的测量与标准时间戳的传播可以分开。网络上所有其他从时钟节点收到上述报文后,记录同步报文的接收时间戳  $T_{S1}$ ,  $T_{S1}$  和  $T_{M1}$  的偏差中不但包含主从时钟的时间偏差,还包含未知的报文传输延迟,分布式系统中每个节点在网络中所处位置、布线方式、布线长度以及目前网络技术中的固有问题,也将造成测控数据在传输过程中的不同延迟。因此需要进一步测量并消除该传输延迟。与偏移测量不同,延迟测量是不规则进行的,从时钟节点按照定义的间隔时间(缺省值是 4~60 s 之间的随机值)向主时钟节点发送一个“延迟请求”(Delay Request)报文,同时记录该报文的实际发送时间,作为精确的发送时间戳  $T_{S2}$ ,而主时钟接收到该报文时也记下接收时刻的精确时间戳  $T_{M2}$ ,并将该时间戳在随后的“延迟响应”(Delay Response)报文中发送给相应的从时钟节点。假设网络延迟是对称的,则有以下两式:

$$T_{S1} - T_{M1} = \Delta + \tau \quad (1)$$

$$T_{M2} - T_{S2} = \tau - \Delta \quad (2)$$

式中,  $\tau$  为报文的传输延迟,  $\Delta$  为从时钟相对主时钟的偏差。

由式(1)和式(2)可求得:

$$\Delta = [(T_{S1} - T_{M1}) - (T_{M2} - T_{S2})] / 2 \quad (3)$$

$$\tau = [(T_{S1} - T_{M1}) + (T_{M2} - T_{S2})] / 2 \quad (4)$$

根据偏差  $\Delta$ , 调整从时钟, 实现定时。

### 3 影响同步精度的因素

(1)网络对称性 从图 1 看出,分布式系统为计算其网络延迟和时钟偏移所进行的 4 次测量都是建立在报文的传输延迟(Delay)在传输方向上是相同的基础上。但这只是理想的情况,在大网络负载的情况下,Delay 在传输方向上的差异会越来越大,造成同步精度的下降。但由于工业上的分布式系统多为相对本地化、局域网的系统,子网或内部组件相对稳定的环境,因此可忽略 Delay 在传输方向上的差异。

(2)网络拓扑结构 系统测量时,点对点的传输能提供最高的精度,网络的拓扑结构也会对同步精度造成影响,当采用 Hub 连接时,网络延迟抖动为 300~400 ns,当采用交换机时,由于存储-转发机制的影响,网络延迟会随着网络负载的变化而变化,因此具有较大的抖动值,使得上述网络传输延迟测量方法精度大大降低。为此,PTP 采用边界时钟(Boundary Clock)的方法解决这一问题,如图 2 所示。

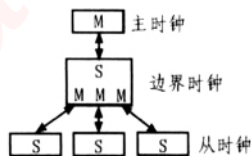


图 2 边界时钟的应用

在交换机内部与主时钟连接的端口可以看成从时钟端

口,在接收主时钟发送的同步报文后调整自己的本地时钟,然后用调整后的时钟去同步所有与交换机主时钟连接的从时钟设备,这样带有拓扑结构的连接就变成简单的点对点连接,从而有效去除交换机所造成的延迟抖动。

(3)晶振的频率 由于产生时钟的不同晶体本身的速度是不同的,而且晶体容易受温度等影响,产生晶振漂移,引起误差增大,因此需时从时钟的晶振频率进行补偿。使从时钟的晶振频率与主时钟保持一致。

### 4 时间戳的生成

基于以上方法,可有效消除主、从时钟偏差和报文在网络中的传输延迟,从而实现分布式系统的时钟同步。同步信息检测和时间戳生成方法不同,时钟同步精度也不一样,IEEE 802.3 中规定以太网帧的基本结构:前导码、帧起始定界符、目的 MAC、源 MAC、长度、数据和帧校验序列。其中,前导码由 7 个‘1 0’交替的 8 位字节组成,用于信号同步;而帧起始定界符包含 6 位交替的‘1 0’及末位的 2 个 1,末位的 2 个 1 通知接收端,跟在后面的是帧的实际字段,表示一帧开始。在以太网中,IEEE 1588 所定义的各种时钟报文(Sync 报文等)均以 UDP/IP 多播包形式发送的,报文的时标生成点位于帧起始定界符最后一位,如图 3 所示。

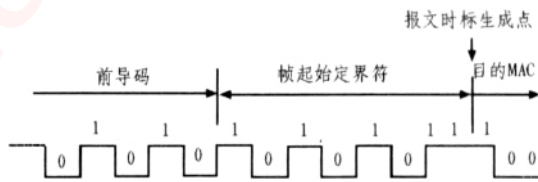


图 3 报文时标生成点

IEEE 1588 的同步原理决定了时钟同步的精度主要取决于时间戳的精度。时间戳的加盖有 3 种方法:

(1)硬件加盖方式 时间戳主要加盖在 MAC 层和 PHY 层之间的 MII(media independent interface)层,这也是最精确的加盖方式,需用硬件电路实现。

(2)软件加盖方式 时间戳加盖在网络的驱动层,在网络接口的驱动程序中实现,精度略低。

(3)加盖在应用层 因为这种方式无需修改其他系统软件,所以协议栈的操作延迟和负载都能对同步精度造成影响,因此同步精度最低。无论在网络驱动层还是在 MII 产生的时间戳都必须返回到 PTP 应用层并经由系统内核处理,然后才发送到 PTP 终端,由此可以看出时间戳的生成决定了 PTP 协议的运行。只有获得精确的时间戳才能发挥 IEEE 1588 所能达到的时钟同步精度。

以太网自身的 CSMA/CD 机制、上一层通信(如 TCP 及 UDP)上进行的误差检测及翻译障碍等,都占用以太网的时间。因此,用软件方式产生时间戳,必须优化加盖时间戳的驱动程序以提高系统内核处理带有时间戳报文的运算效率,同时优化报文发送前在网卡内部的排序,减小程序运行和网络负载对收发报文的影响,从而降低报文发送过程中的软件延

迟和抖动。

在以太网中，也可以在物理层放置硬件电路加盖时间戳，这样可以避免协议栈上部较大的时间抖动，消除报文传输中的网络延迟。由于主从时钟不同步的原因除了网络延迟外还有时钟偏差，可以采用频率可调时钟来校正从时钟相对主时钟的时钟偏差，根据从时钟处得出的自身与主时钟的偏差计算出相应的频率补偿值，从而控制时钟计数器的数值达到与主时钟的同步。硬件电路从物理层获得每个发送和接收报文的比特流，并记录时间戳信息，判断其是否为 IEEE 1588 相关协议报文，如果是相关报文，则把时间戳信息传送给上层软件，否则丢弃该报文的时间戳信息。对于 100 M 以太网，由于采用 4B/5B 编码和 Scrambler 技术，只能在物理层与数据链路层之间的 MII 层加盖时间戳信息。IEEE 1588 时钟同步实现过程如图 4 所示。

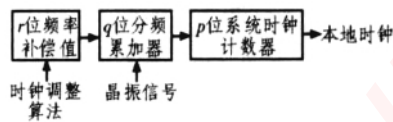


图 6 频率补偿时钟结构图

如果发生溢出，则在下一个晶振周期，时钟计数器增加一个增量，这个增量就是整个频率补偿时钟的分辨率；如果没有溢出，则在下一个晶振周期，时钟计数器保持原值。计数器的当前值即为调整后的本地时钟。因此可以通过改变频率补偿值来改变时钟计数器数值增加的频率，实现对晶振频率的补偿。 $q$  和  $r$  越大，频率补偿值的改变对累加器发生溢出的频率的影响越小，对晶振频率的补偿越精细。本系统选择  $r=32, q=32, p=64$ 。

使用 FPGA 不但能够获得最精确的时间戳，而且能够采用数字电路的方法精确补偿从时钟的晶振频率。FPGA 的控制器实现时钟调节算法，并由得到的时间戳和本地时钟相比较，得到频率补偿值，从而控制时钟计数器数值，补偿晶振的频率。而 CPU 则主要负责驱动 MAC 控制器完成相关的通讯功能。

## 6 结束语<sup>[7]</sup>

采用 IEEE 1588 精确时钟同步协议的以太网时钟同步技术，解决了通用以太网实时性和同步性差的问题，为基于多播技术的标准以太网的实时应用提供了有效的解决方案。IEEE 1588 精确时钟同步协议的制定满足网络化分布式系统的需求，利用 IEEE 1588 时钟同步技术可以在不增加网络负荷的情况下，实现整个分布式系统的高精度时钟同步，从而可以有效解决分布式系统的实时性问题，进而改善和提高系统的同步精度。

参考文献：

- [1] 王 骥, 张 玲. 分布式数据采集系统中的时钟同步[J]. 单片机与嵌入式系统应用, 2008(4): 31-33.
- [2] 袁振华, 董秀军, 刘朝英. 基于 IEEE 1588 的时钟同步技术及其应用[J]. 计算机测量与控制, 2006, 14(12): 1 726-1 728.
- [3] 张 妍, 孙鹤旭, 林涛宁, 等. IEEE 1588 在实时工业以太网中的应用[J]. 微计算机信息, 2005, 21(9): 18-21.
- [4] 桂本恒, 冯冬芹, 褚 健. IEEE1588 的高精度时间同步算法的分析与实现[J]. 工业仪表与自动化装置, 2006, 36(4): 20-23.
- [5] 戴宝峰, 崔少辉, 王 岩. 基于 IEEE 1588 协议的时间戳的生成与分析[J]. 仪表技术, 2007, 36(7): 15-17.
- [6] 黄云水. IEEE 1588 精密时钟同步分析[J]. 国外电子测量技术, 2005, 24(9): 9-12.
- [7] 林云标, 吴成柯, 李云松, 等. ICS1523 型高性能可编程行同步时钟发生器的原理及应用[J]. 电子设计工程, 2006, 14(5):

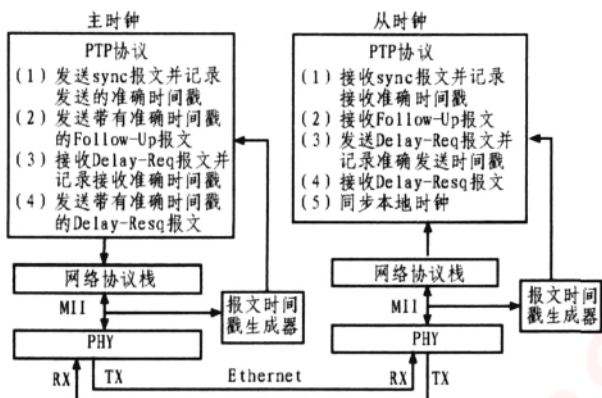


图 4 IEEE 1588 时钟同步的实现过程

## 5 基于 FPGA 的硬件同步方案

因为采用硬件电路获取时间戳，从而获得更高的同步精度，这里采用 FPGA 实现时间戳的获取和从时钟相对主时钟的频率纠偏<sup>[9]</sup>。分布式系统中节点时钟同步模块的内部结构如图 5 所示。

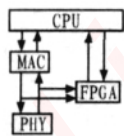


图 5 同步模块内部结构图

每个节点包含 CPU、以太网媒体访问控制器(MAC)、以太网物理层收发器(PHY)、FPGA 等 4 种主要器件。物理层收发器和 MAC 控制器通过标准的 MII 接口相连，CPU 通过总线与 MAC 控制器和 FPGA 相连。物理层收发器、MAC 控制器和 CPU 构成通讯协议栈完成数据包发送和接收。

为了使从时钟的晶振频率与主时钟保持一致，FPGA 中需要有一个频率可调的时钟以实现晶振纠偏，如图 6 所示，由  $r$  位频率补偿值寄存器、 $q$  位分频累加器、 $p$  位系统时钟计数器组成，一个普通晶体振荡信号输入 FPGA 模块。在每个晶振周期， $r$  位频率补偿值寄存器内保存的频率补偿值累加到  $q$  位分频累加器中，与累加器中的值相加。

# 嵌入式资源免费下载

## 总线协议:

1. [基于 PCIe 驱动程序的数据传输卡 DMA 传输](#)
2. [基于 PCIe 总线协议的设备驱动开发](#)
3. [CANopen 协议介绍](#)
4. [基于 PXI 总线 RS422 数据通信卡 WDM 驱动程序设计](#)
5. [FPGA 实现 PCIe 总线 DMA 设计](#)
6. [PCI Express 协议实现与验证](#)
7. [VPX 总线技术及其实现](#)
8. [基于 Xilinx FPGA 的 PCIE 接口实现](#)
9. [基于 PCI 总线的 GPS 授时卡设计](#)
10. [基于 CPCI 标准的 6U 信号处理平台的设计](#)
11. [USB30 电路保护](#)
12. [USB30 协议分析与框架设计](#)
13. [USB 30 中的 CRC 校验原理及实现](#)
14. [基于 CPLD 的 UART 设计](#)
15. [IPMI 在 VPX 系统中的应用与设计](#)
16. [基于 CPCI 总线的 PMC 载板设计](#)
17. [基于 VPX 总线的工件台运动控制系统研究与开发](#)
18. [PCI Express 流控机制的研究与实现](#)
19. [UART16C554 的设计](#)
20. [基于 VPX 的高性能计算机设计](#)
21. [基于 CAN 总线技术的嵌入式网关设计](#)
22. [Visual C 串行通讯控件使用方法与技巧的研究](#)
23. [IEEE1588 精密时钟同步关键技术研究](#)
24. [GPS 信号发生器射频模块的一种实现方案](#)
25. [基于 CPCI 接口的视频采集卡的设计](#)
26. [基于 VPX 的 3U 信号处理平台的设计](#)
27. [基于 PCI Express 总线 1394b 网络传输系统 WDM 驱动设计](#)
28. [AT89C52 单片机与 ARINC429 航空总线接口设计](#)
29. [基于 CPCI 总线多 DSP 系统的高速主机接口设计](#)
30. [总线协议中的 CRC 及其在 SATA 通信技术中的应用](#)
31. [基于 FPGA 的 SATA 硬盘加解密控制器设计](#)
32. [Modbus 协议在串口通讯中的研究及应用](#)
33. [高可用的磁盘阵列 Cache 的设计和实现](#)
34. [RAID 阵列中高速 Cache 管理的优化](#)



35. [一种新的基于 RAID 的 CACHE 技术研究与实现](#)
36. [基于 PCIE-104 总线的高速数据接口设计](#)
37. [基于 VPX 标准的 RapidIO 交换和 Flash 存储模块设计](#)
38. [北斗卫星系统在海洋工程中的应用](#)
39. [北斗卫星系统在远洋船舶上应用的研究](#)
40. [基于 CPCI 总线的红外实时信号处理系统](#)
41. [硬件实现 RAID 与软件实现 RAID 的比较](#)
42. [基于 PCI Express 总线系统的热插拔设计](#)
43. [基于 RAID5 的磁盘阵列 Cache 的研究与实现](#)
44. [基于 PCI 总线的 MPEG2 码流播放卡驱动程序开发](#)
45. [基于磁盘阵列引擎的 RAID5 小写性能优化](#)
46. [基于 IEEE1588 的时钟同步技术研究](#)
47. [基于 Davinci 平台的 SD 卡读写优化](#)
48. [基于 PCI 总线的图像处理及传输系统的设计](#)
49. [串口和以太网通信技术在油液在线监测系统中的应用](#)
50. [USB30 数据传输协议分析及实现](#)
51. [IEEE 1588 协议在工业以太网中的实现](#)
52. [基于 USB30 的设备自定义请求实现方法](#)
53. [IEEE1588 协议在网络测控系统中的应用](#)
54. [USB30 物理层中弹性缓冲的设计与实现](#)
55. [USB30 的高速信息传输瓶颈研究](#)
56. [基于 IPv6 的 UDP 通信的实现](#)
57. [一种基于 IPv6 的流媒体传送方案研究与实现](#)
58. [基于 IPv4-IPv6 双栈的 MODBUS-TCP 协议实现](#)
59. [RS485CAN 网关设计与实现](#)
60. [MVB 周期信息的实时调度](#)
61. [RS485 和 PROFINET 网关设计](#)
62. [基于 IPv6 的 Socket 通信的实现](#)
63. [MVB 网络重复器的设计](#)
64. [一种新型 MVB 通信板的探究](#)
65. [具有 MVB 接口的输入输出设备的分析](#)
66. [基于 STM32 的 GSM 模块综合应用](#)
67. [基于 ARM7 的 MVB CAN 网关设计](#)
68. [机车车辆的 MVB CAN 总线网关设计](#)
69. [智能变电站冗余网络中 IEEE1588 协议的应用](#)
70. [CAN 总线的浅析 CANopen 协议](#)
71. [基于 CANopen 协议实现多电机系统实时控制](#)
72. [以太网时钟同步协议的研究](#)

VxWorks:

1. [基于 VxWorks 的多任务程序设计](#)
2. [基于 VxWorks 的数据采集存储装置设计](#)
3. [Flash 文件系统分析及其在 VxWorks 中的实现](#)
4. [VxWorks 多任务编程中的异常研究](#)
5. [VxWorks 应用技巧两例](#)
6. [一种基于 VxWorks 的飞行仿真实时管理系统](#)
7. [在 VxWorks 系统中使用 TrueType 字库](#)
8. [基于 FreeType 的 VxWorks 中文显示方案](#)
9. [基于 Tilcon 的 VxWorks 简单动画开发](#)
10. [基于 Tilcon 的某武器显控系统界面设计](#)
11. [基于 Tilcon 的综合导航信息处理装置界面设计](#)
12. [VxWorks 的内存配置和管理](#)
13. [基于 VxWorks 系统的 PCI 配置与应用](#)
14. [基于 MPC8270 的 VxWorks BSP 的移植](#)
15. [Bootrom 功能改进经验谈](#)
16. [基于 VxWorks 嵌入式系统的中文平台研究与实现](#)
17. [VxBus 的 A429 接口驱动](#)
18. [基于 VxBus 和 MPC8569E 千兆网驱动开发和实现](#)
19. [一种基于 vxBus 的 PPC 与 FPGA 高速互联的驱动设计方法](#)
20. [基于 VxBus 的设备驱动开发](#)
21. [基于 VxBus 的驱动程序架构分析](#)
22. [基于 VxBus 的高速数据采集卡驱动程序开发](#)
23. [Vxworks 下的冗余 CAN 通讯模块设计](#)
24. [WindML 工业平台下开发 S1d13506 驱动及显示功能的实现](#)
25. [WindML 中 Mesa 的应用](#)
26. [VxWorks 下图形用户界面开发中双缓冲技术应用](#)
27. [VxWorks 上的一种 GUI 系统的设计与实现](#)
28. [VxWorks 环境下 socket 的实现](#)
29. [VxWorks 的 WindML 图形界面程序的框架分析](#)
30. [VxWorks 实时操作系统及其在 PC104 下以太网编程的应用](#)
31. [实时操作系统任务调度策略的研究与设计](#)
32. [军事指挥系统中 VxWorks 下汉字显示技术](#)
33. [基于 VxWorks 实时控制系统中文交互界面开发平台](#)
34. [基于 VxWorks 操作系统的 WindML 图形操控界面实现方法](#)
35. [基于 GPU FPGA 芯片原型的 VxWorks 下驱动软件开发](#)
36. [VxWorks 下的多串口卡设计](#)
37. [VxWorks 内存管理机制的研究](#)
38. [T9 输入法在 Tilcon 下的实现](#)
39. [基于 VxWorks 的 WindML 图形界面开发方法](#)
40. [基于 Tilcon 的 IO 控制板可视化测试软件的设计和实现](#)

41. [基于 VxWorks 的通信服务器实时多任务软件设计](#)
42. [基于 VXWORKS 的 RS485MVB 网关的设计与实现](#)
43. [实时操作系统 VxWorks 在微机保护中的应用](#)
44. [基于 VxWorks 的多任务程序设计及通信管理](#)
45. [基于 Tilcon 的 VxWorks 图形界面开发技术](#)
46. [嵌入式图形系统 Tilcon 及应用研究](#)
47. [基于 VxWorks 的数据采集与重演软件的图形界面的设计与实现](#)
48. [基于嵌入式的 Tilcon 用户图形界面设计与开发](#)
49. [基于 Tilcon 的交互式多页面的设计](#)
50. [基于 Tilcon 的嵌入式系统人机界面开发技术](#)
51. [基于 Tilcon 的指控系统多任务人机交互软件设计](#)
52. [基于 Tilcon 航海标绘台界面设计](#)
53. [基于 Tornado 和 Tilcon 的嵌入式 GIS 图形编辑软件的开发](#)

## Linux:

1. [Linux 程序设计第三版及源代码](#)
2. [NAND FLASH 文件系统的设计与实现](#)
3. [多通道串行通信设备的 Linux 驱动程序实现](#)
4. [Zsh 开发指南-数组](#)
5. [常用 GDB 命令中文速览](#)
6. [嵌入式 C 进阶之道](#)
7. [Linux 串口编程实例](#)
8. [基于 Yocto Project 的嵌入式应用设计](#)
9. [Android 应用的反编译](#)
10. [基于 Android 行为的加密应用系统研究](#)
11. [嵌入式 Linux 系统移植步步通](#)
12. [嵌入式 C++语言精华文章集锦](#)
13. [基于 Linux 的高性能服务器端的设计与研究](#)
14. [S3C6410 移植 Android 内核](#)
15. [Android 开发指南中文版](#)
16. [图解 Linux 操作系统架构设计与实现原理（第二版）](#)
17. [如何在 Ubuntu 和 Linux Mint 下轻松升级 Linux 内核](#)
18. [Android 简单 mp3 播放器源码](#)
19. [嵌入式 Linux 系统实时性的研究](#)
20. [Android 嵌入式系统架构及内核浅析](#)
21. [基于嵌入式 Linux 操作系统内核实时性的改进方法研究](#)
22. [Linux TCP IP 协议详解](#)
23. [Linux 桌面环境下内存去重技术的研究与实现](#)

24. [掌握 Android 7.0 新增特性 Quick Settings](#)
25. [Android 应用逆向分析方法研究](#)
26. [Android 操作系统的课程教学](#)
27. [Android 智能手机操作系统的研究](#)
28. [Android 英文朗读功能的实现](#)
29. [基于 Yocto 订制嵌入式 Linux 发行版](#)
30. [基于嵌入式 Linux 的网络设备驱动设计与实现](#)
31. [如何高效学习嵌入式](#)
32. [基于 Android 平台的 GPS 定位系统的设计与实现](#)
33. [LINUX ARM 下的 USB 驱动开发](#)
34. [Linux 下基于 I2C 协议的 RTC 驱动开发](#)
35. [嵌入式下 Linux 系统设备驱动程序的开发](#)
36. [基于嵌入式 Linux 的 SD 卡驱动程序的设计与实现](#)
37. [Linux 系统中进程调度策略](#)
38. [嵌入式 Linux 实时性方法](#)
39. [基于实时 Linux 计算机联锁系统实时性分析与改进](#)
40. [基于嵌入式 Linux 下的 USB30 驱动程序开发方法研究](#)
41. [Android 手机应用开发之音乐资源播放器](#)
42. [Linux 下以太网的 IPv6 隧道技术的实现](#)
43. [Research and design of mobile learning platform based on Android](#)
44. [基于 linux 和 Qt 的串口通信调试器调的设计及应用](#)
45. [在 Linux 平台上基于 QT 的动态图像采集系统的设计](#)
46. [基于 Android 平台的医护查房系统的研究与设计](#)
47. [基于 Android 平台的软件自动化监控工具的设计开发](#)
48. [基于 Android 的视频软硬解码及渲染的对比研究与实现](#)
49. [基于 Android 移动设备的加速度传感器技术研究](#)
50. [基于 Android 系统振动测试仪研究](#)
51. [基于缓存竞争优化的 Linux 进程调度策略](#)

## Windows CE:

1. [Windows CE.NET 下 YAFFS 文件系统 NAND Flash 驱动程序设计](#)
2. [Windows CE 的 CAN 总线驱动程序设计](#)
3. [基于 Windows CE.NET 的 ADC 驱动程序实现与应用的研究](#)
4. [基于 Windows CE.NET 平台的串行通信实现](#)
5. [基于 Windows CE.NET 下的 GPRS 模块的研究与开发](#)
6. [win2k 下 NTFS 分区用 ntldr 加载进 dos 源代码](#)
7. [Windows 下的 USB 设备驱动程序开发](#)
8. [WinCE 的大容量程控数据传输解决方案设计](#)



9. [WinCE6.0 安装开发详解](#)
10. [DOS 下仿 Windows 的自带计算器程序 C 源码](#)
11. [G726 局域网语音通话程序和源代码](#)
12. [WinCE 主板加载第三方驱动程序的方法](#)
13. [WinCE 下的注册表编辑程序和源代码](#)
14. [WinCE 串口通信源代码](#)
15. [WINCE 的 SD 卡程序\[可实现读写的源码\]](#)
16. [基于 WinCE 的 BootLoader 研究](#)
17. [Windows CE 环境下无线网卡的自动安装](#)
18. [基于 Windows CE 的可视电话的研究与实现](#)
19. [基于 WinCE 的嵌入式图像采集系统设计](#)
20. [基于 ARM 与 WinCE 的掌纹鉴别系统](#)
21. [DCOM 协议在网络冗余环境下的应用](#)
22. [Windows XP Embedded 在变电站通信管理机中的应用](#)
23. [XPE 在多功能显控台上的开发与应用](#)
24. [基于 Windows XP Embedded 的 LKJ2000 仿真系统设计与实现](#)
25. [虚拟仪器的 Windows XP Embedded 操作系统开发](#)
26. [基于 EVC 的嵌入式导航电子地图设计](#)
27. [基于 XPEmbedded 的警务区 SMS 指挥平台的设计与实现](#)
28. [基于 XPE 的数字残币兑换工具开发](#)

## PowerPC:

1. [Freescale MPC8536 开发板原理图](#)
2. [基于 MPC8548E 的固件设计](#)
3. [基于 MPC8548E 的嵌入式数据处理系统设计](#)
4. [基于 PowerPC 嵌入式网络通信平台的实现](#)
5. [PowerPC 在车辆显控系统中的应用](#)
6. [基于 PowerPC 的单板计算机的设计](#)
7. [用 PowerPC860 实现 FPGA 配置](#)
8. [基于 MPC8247 嵌入式电力交换系统的设计与实现](#)
9. [基于设备树的 MPC8247 嵌入式 Linux 系统开发](#)
10. [基于 MPC8313E 嵌入式系统 UBoot 的移植](#)
11. [基于 PowerPC 处理器 SMP 系统的 UBoot 移植](#)
12. [基于 PowerPC 双核处理器嵌入式系统 UBoot 移植](#)
13. [基于 PowerPC 的雷达通用处理机设计](#)
14. [PowerPC 平台引导加载程序的移植](#)
15. [基于 PowerPC 嵌入式内核的多串口通信扩展设计](#)

16. [基于 PowerPC 的多网口系统抗干扰设计](#)
17. [基于 MPC860T 与 VxWorks 的图形界面设计](#)
18. [基于 MPC8260 处理器的 PPMC 系统](#)
19. [基于 PowerPC 的控制器研究与设计](#)
20. [基于 PowerPC 的模拟量输入接口扩展](#)
21. [基于 PowerPC 的车载通信系统设计](#)
22. [基于 PowerPC 的嵌入式系统中通用 IO 口的扩展方法](#)
23. [基于 PowerPC440GP 型微控制器的嵌入式系统设计与研究](#)
24. [基于双 PowerPC 7447A 处理器的嵌入式系统硬件设计](#)
25. [基于 PowerPC603e 通用处理模块的设计与实现](#)
26. [嵌入式微机 MPC555 驻留片内监控器的开发与实现](#)
27. [基于 PowerPC 和 DSP 的电能质量在线监测装置的研制](#)
28. [基于 PowerPC 架构多核处理器嵌入式系统硬件设计](#)
29. [基于 PowerPC 的多屏系统设计](#)
30. [基于 PowerPC 的嵌入式 SMP 系统设计](#)

## ARM:

1. [基于 DiskOnChip 2000 的驱动程序设计及应用](#)
2. [基于 ARM 体系的 PC-104 总线设计](#)
3. [基于 ARM 的嵌入式系统中断处理机制研究](#)
4. [设计 ARM 的中断处理](#)
5. [基于 ARM 的数据采集系统并行总线的驱动设计](#)
6. [S3C2410 下的 TFT LCD 驱动源码](#)
7. [STM32 SD 卡移植 FATFS 文件系统源码](#)
8. [STM32 ADC 多通道源码](#)
9. [ARM Linux 在 EP7312 上的移植](#)
10. [ARM 经典 300 问](#)
11. [基于 S5PV210 的频谱监测设备嵌入式系统设计与实现](#)
12. [Uboot 中 start.S 源码的指令级的详尽解析](#)
13. [基于 ARM9 的嵌入式 Zigbee 网关设计与实现](#)
14. [基于 S3C6410 处理器的嵌入式 Linux 系统移植](#)
15. [CortexA8 平台的  \$\mu\$ C-OS II 及 LwIP 协议栈的移植与实现](#)
16. [基于 ARM 的嵌入式 Linux 无线网卡设备驱动设计](#)
17. [ARM S3C2440 Linux ADC 驱动](#)
18. [ARM S3C2440 Linux 触摸屏驱动](#)
19. [Linux 和 Cortex-A8 的视频处理及数字微波传输系统设计](#)
20. [Nand Flash 启动模式下的 Uboot 移植](#)

21. [基于 ARM 处理器的 UART 设计](#)
22. [ARM CortexM3 处理器故障的分析与处理](#)
23. [ARM 微处理器启动和调试浅析](#)
24. [基于 ARM 系统下映像文件的执行与中断运行机制的实现](#)
25. [中断调用方式的 ARM 二次开发接口设计](#)
26. [ARM11 嵌入式系统 Linux 下 LCD 的驱动设计](#)
27. [Uboot 在 S3C2440 上的移植](#)
28. [基于 ARM11 的嵌入式无线视频终端的设计](#)
29. [基于 S3C6410 的 Uboot 分析与移植](#)
30. [基于 ARM 嵌入式系统的高保真无损音乐播放器设计](#)
31. [UBoot 在 Mini6410 上的移植](#)
32. [基于 ARM11 的嵌入式 Linux NAND FLASH 模拟 U 盘挂载分析与实现](#)
33. [基于 ARM11 的电源完整性分析](#)
34. [基于 ARM S3C6410 的 uboot 分析与移植](#)
35. [基于 S5PC100 移动视频监控终端的设计与实现](#)

## Hardware:

1. [DSP 电源的典型设计](#)
2. [高频脉冲电源设计](#)
3. [电源的综合保护设计](#)
4. [任意波形电源的设计](#)
5. [高速 PCB 信号完整性分析及应用](#)
6. [DM642 高速图像采集系统的电磁干扰设计](#)
7. [使用 COMExpress Nano 工控板实现 IP 调度设备](#)
8. [基于 COM Express 架构的数据记录仪的设计与实现](#)
9. [基于 COM Express 的信号系统逻辑运算单元设计](#)
10. [基于 COM Express 的回波预处理模块设计](#)
11. [基于 X86 平台的简单多任务内核的分析与实现](#)
12. [基于 UEFI Shell 的 PreOS Application 的开发与研究](#)
13. [基于 UEFI 固件的恶意代码防范技术研究](#)
14. [MIPS 架构计算机平台的支持固件研究](#)
15. [基于 UEFI 固件的攻击验证技术研究](#)
16. [基于 UEFI 的 Application 和 Driver 的分析与开发](#)
17. [基于 UEFI 的可信 BIOS 研究与实现](#)
18. [基于 UEFI 的国产计算机平台 BIOS 研究](#)
19. [基于 UEFI 的安全模块设计分析](#)
20. [基于 FPGA Nios II 的等精度频率计设计](#)
21. [基于 FPGA 的 SOPC 设计](#)

22. [基于 SOPC 基本信号产生器的设计与实现](#)
23. [基于龙芯平台的 PMON 研究与开发](#)
24. [基于 X86 平台的嵌入式 BIOS 可配置设计](#)
25. [基于龙芯 2F 架构的 PMON 分析与优化](#)
26. [CPU 与 GPU 之间接口电路的设计与实现](#)
27. [基于龙芯 1A 平台的 PMON 源码编译和启动分析](#)
28. [基于 PC104 工控机的嵌入式直流监控装置的设计](#)
29. [GPGPU 技术研究与实现](#)
30. [GPU 实现的高速 FIR 数字滤波算法](#)
31. [一种基于 CPUGPU 异构计算的混合编程模型](#)
32. [面向 OpenCL 模型的 GPU 性能优化](#)
33. [基于 GPU 的 FDTD 算法](#)
34. [基于 GPU 的瑕疵检测](#)
35. [基于 GPU 通用计算的分析与研究](#)
36. [面向 OpenCL 架构的 GPGPU 量化性能模型](#)
37. [基于 OpenCL 的图像积分图算法优化研究](#)
38. [基于 OpenCL 的均值平移算法在多个众核平台的性能优化研究](#)
39. [基于 OpenCL 的异构系统并行编程](#)
40. [嵌入式系统中热备份双机切换技术研究](#)

## Programming:

1. [计算机软件基础数据结构 - 算法](#)
2. [高级数据结构对算法的优化](#)
3. [零基础学算法](#)
4. [Linux 环境下基于 TCP 的 Socket 编程浅析](#)
5. [Linux 环境下基于 UDP 的 socket 编程浅析](#)
6. [基于 Socket 的网络编程技术及其实现](#)
7. [数据结构考题 - 第 1 章 绪论](#)
8. [数据结构考题 - 第 2 章 线性表](#)
9. [数据结构考题 - 第 2 章 线性表 - 答案](#)
10. [基于小波变换与偏微分方程的图像分解及边缘检测](#)
11. [基于图像能量的布匹瑕疵检测方法](#)
12. [基于 OpenCL 的拉普拉斯图像增强算法优化研究](#)
13. [异构平台上基于 OpenCL 的 FFT 实现与优化](#)



## FPGA / CPLD:

1. [一种基于并行处理器的快速车道线检测系统及 FPGA 实现](#)
2. [基于 FPGA 和 DSP 的 DBF 实现](#)
3. [高速浮点运算单元的 FPGA 实现](#)
4. [DLMS 算法的脉动阵结构设计及 FPGA 实现](#)
5. [一种基于 FPGA 的 3DES 加密算法实现](#)
- 6.