

基于 FPGA 的精确时钟同步方法

黄文君¹, 俞彬²

(1. 浙江大学 工业控制技术国家重点实验室, 浙江 杭州 310027; 2. 浙江大学 先进控制研究所, 浙江 杭州 310027)

摘要: 为实现分布式系统高精度同步数据采集和控制的实时性要求, 提出了一种基于工业以太网的分布式控制系统时钟硬件同步方法. 基于高速数字逻辑硬件方法解析 IEEE1588 时间同步协议, 采用硬件描述语言 (VHDL) 和现场可编程逻辑门阵列 (FPGA) 设计时间戳捕获、晶振频率补偿、时钟同步算法等模块, 为嵌入式实时控制系统构架高精度的硬件时钟同步方案, 该方法解决了传统的基于嵌入式软件的时钟同步方案中时间戳不稳定、同步精度低等问题. 对基于工业以太网的分布式控制系统进行了动态测试验证, 实际测试数据表明系统各节点达到了亚微秒级的时钟同步精度, 长期运行结果验证了系统同步精度的稳定性.

关键词: 精确时钟同步; 工业以太网; 现场可编程逻辑门阵列; 实时控制系统

中图分类号: TP393

文献标识码: A

文章编号: 1008-973X(2007)10-1697-04

Method for precise time synchronization based on FPGA

HUANG Wen-jun¹, YU Bin²

(1. State Key Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou 310027, China;

2. Institute of Advanced Process Control, Zhejiang University, Hangzhou 310027, China)

Abstract: A hardware-based precise time synchronization method for the distributed control system (DCS) based on industrial Ethernet was proposed to realize synchronous measurement and real-time control. Using high-speed digital circuits, e. g. field-programmable gate array (FPGA), hardware-based strategy was designed for implementing the time synchronization protocol defined in IEEE1588. Timestamp capture module, oscillator frequency compensation module, time synchronization algorithm and etc, were all coded with high-speed integrated circuit hardware description language (VHDL). Problems with the traditional time synchronization method based on embedded software were solved, such as uncertain time delay of timestamp and low time accuracy. A precise time synchronization solution for the embedded controller was provided successfully. Test and experiments in DCS showed high time accuracy and high stability, and time accuracy reached the sub-microsecond range.

Key words: precise time synchronization; industrial Ethernet; field-programmable gate array (FPGA); real-time control system

时钟同步是基于网络的分布式实时控制系统数据调度和任务的核心技术之一, 它提供一种确定事件发生时间的先后顺序的机制, 以保证节点发送和接收消息、控制、事故记录等在时间逻辑上是完全正

确、可信的. 传统的基于嵌入式软件的集成分布式系统的时间同步方案 (如 NTP) 很难达到亚微秒级的同步精度, 很难满足高实时性的同步数据采集、控制和分析的要求.

IEEE1588 标准定义了一个能够在测量和控制系统中实现精确的时钟同步的协议,它利用网络的通讯链路实现时钟同步,无需其他的同步线缆和额外投入,能达到较高的同步精度^[1]. IEEE1588 能够在所有支持多播的网络上实现,特别适合于以太网.在工业以太网架构分布式控制系统中,基于嵌入式软件的方案由于受到数据帧多级缓冲时延不稳定、晶振频率漂移、中断服务程序响应时间等因素限制很难达到微秒级的同步精度.针对这一情况,本文结合工业以太网在实时控制系统中的应用,采用 IEEE1588 同步算法设计了一种基于 FPGA 的硬件时钟同步方案,采用硬件描述语言(VHDL)实现了亚微秒级的精确同步,满足高实时性、高精度的同步数据采集、记录和分析的需求.

1 基于 FPGA 的时钟硬件同步方案

1.1 硬件电路实现时钟同步方法

分布式控制系统时钟是用计数器来实现的,判定 2 个时钟是否同步有 2 条标准:一是两者计数器的数值是否相等,二是两者计数器数值的增加速度是否相等.根据 IEEE1588,要使从时钟同步于主时钟,首先要确定从时钟与主时钟的偏差.这个偏差值是根据数据包发送时间戳和接收时间戳的偏差来计算的,它的精度受到 2 个因素的影响:一是网络通讯链路传输延迟时间的不确定性,二是时间戳本身的精度.在基于嵌入式软件的时钟同步方案中,时钟计数器由 CPU 自身的计数器来实现,时间戳的获取点位于整个软件系统的最底层,即中断层^[2]. CPU 发送数据包时,先将待发送数据写入网卡芯片的发送缓冲区中,然后向网卡发出发送命令,同时读取计数器值作为该数据包的发送时间戳,由以太网的介质访问特性可知,这个时间戳并不是该数据包发出的真正时间,而且这个时间戳与该数据包发出的真正时间的差值也是不确定的. CPU 通过响应网卡中断接收数据包,在中断处理程序中,读取计数器值作为该数据包的接收时间戳,但这个时间戳并不是接收到该数据包的第一个比特时的时间戳,所以还要减去数据包的长度(单位:比特)与传输每一个比特所需的时间之积,得到实际的接收时间戳,由于 CPU 的中断响应时间是不确定的,这个时间戳还不是接收该数据包的真正时间,其不确定性依 CPU 的速度和操作系统的不同而不同.若网卡在接收到与时钟同步相关的数据包时,网卡芯片的缓存中还有其他未被 CPU 处理的数据包,则上述接收时间

戳的精度又要大打折扣.随着网络传输速率的提升,在由直通型交换设备组建的以太网中,通讯链路的传输延迟时间的不确定性越来越小,这导致时间戳本身的精度对同步精度的影响明显加大.时间戳本身的精度是由时间戳的获取方式决定的,在上述的基于嵌入式软件的方案中,时间戳的获取点位于网络层和数据链路层之间,若能将时间戳的获取点下移至数据链路层和物理层之间,则时间戳的精度将大幅提高,但这种方法只有硬件电路才能实现.

仅仅是周期性的校正从时钟无法实现精确的时钟同步,即使从时钟和主时钟的偏差可以计算得非常精确,每次校正后二者偏差为零,由于驱动时钟计数器的晶振的频率并不完全一致,2 个标称频率相同的晶振的实际频率之间的微小差别会导致从时钟和主时钟的偏差随时间的变化逐渐增加,这种漂移在高精度同步的情况下是不能容许的.若要调节从时钟的晶振频率使得它与主时钟晶振频率相同,或者采用模拟电路,采用具有频率调节功能的时钟振荡源来驱动时钟计数器;或者采用数字电路,用硬件电路实现一个算法,其效果相当于采用频率不可调整的晶振和少量数字电路搭建一个等效的频率可调的时钟计数器.采用数字电路更为廉价和方便.从上述分析可以看出,采用数字电路硬件有助于大幅提升时钟同步精度,而且硬件电路的优势是基于嵌入式软件的方案无法达到的.本文设计了一种基于 FPGA 和硬件描述语言(VHDL)的时钟同步系统,发挥了硬件电路在获取时间戳和晶振频率纠偏方面的优势,实现了比基于嵌入式软件的方案更为精确的时钟同步.

1.2 基于 FPGA 的时钟同步方案设计

现场可编程逻辑门阵列(FPGA)采用硬件逻辑门阵列和语言编程方式(VHDL)实现信号的输入、输出及信息处理,具有 ns 级别的数据处理和运算能力.本文设计的基于 FPGA 的时钟同步系统的结构如图 1 所示,它包含一个主时钟节点,若干从时钟节点,其中主时钟节点上带有 GPS(global posi-

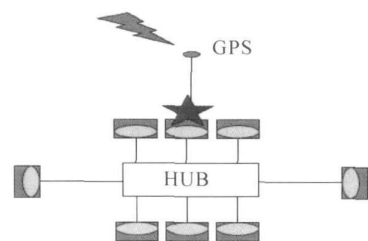


图 1 时钟同步系统示意图

Fig. 1 System architecture of synchronization system

tion system) 接收机. GPS 接收机发出的秒脉冲信号将主时钟同步于 UTC (universal time coordinated), 从时钟通过以太网的通讯链路同步于主时钟. 网络拓扑为星型, 以直通型交换机或集线器 (HUB) 作为交换设备.

系统中节点的内部结构如图 2 所示. 每个节点包含 CPU、以太网媒体访问控制 (media access control, MAC) 器、以太网物理层收发器、FPGA 等 4 种主要芯片. 物理层收发器和 MAC 控制器通过标准的 MII (media independent interface) 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, 完成数据包的发送和接收.

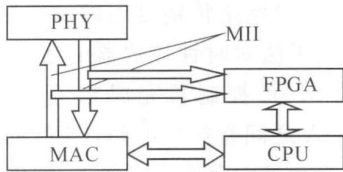


图 2 节点电路结构图

Fig. 2 Hardware architecture of clock node

FPGA 内部结构如图 3 所示, 它是整个同步系统的核心器件, 完成时间戳的获取和晶振的纠偏功能. FPGA 通过监听和分析 MII 接口上的信号来确定数据包的发送时间戳和接收时间戳. 根据 IEEE802.3 标准, MII 接口位于介质访问子层和物理层之间, 是一种简单、廉价、易于实现的电气接口, 它支持 10 和 100 Mbps 2 种传输速率, 它的存在使得同一种介质访问控制器能够对非屏蔽双绞线、屏蔽双绞线、光纤等多种物理传输介质进行访问和控制^[3].

基于 FPGA 的时间同步方案内部采用结构化的硬件描述语言设计了 MII 兼容的以太网发送/接收时间戳捕获器、晶振频率补偿器、时钟同步控制器、通用 CPU 接口、测试接口等模块, 其中时钟同步控制器实现了各个模块之间的数据高速交换、网络传输延迟滤波和基于 IEEE1588 的同步运算, 并将运算结果作用于其他模块以实现晶振频率补偿、

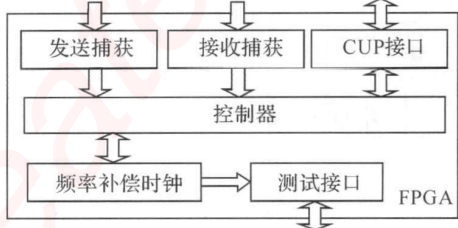
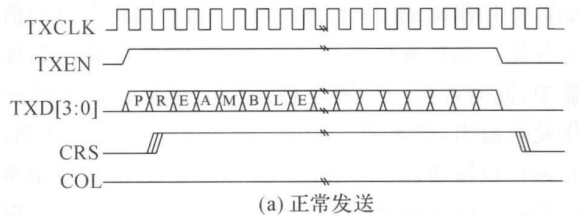


图 3 FPGA 模块结构图

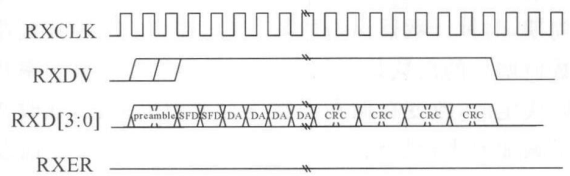
Fig. 3 Module architecture in FPGA

时钟同步等功能. 通用 CPU 接口为嵌入式实时控制系统提供高精度的硬件时钟访问接口.

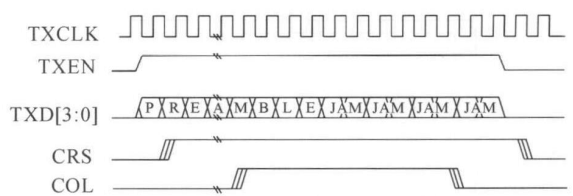
MII 接口共有 18 个信号, 它们分别是用于发送数据的 TXCLK、TXD [3:0]、TXEN、TXER, 用于接收数据的 RXCLK、RXD [3:0]、RXDV、RXER, 此外还有 CRS 和 COL, 以及用于管理的 MDIO 和 MDC. TXCLK 是本地晶振产生的时钟信号, 发送数据 TXD [3:0]、发送使能 TXEN 和发送差错 TXER 都同步于 TXCLK; RXCLK 是从接收到的信号中恢复出来的时钟信号, 接收数据 RXD [3:0]、接收有效 RXDV 和接收差错 RXER 都同步于 RXCLK; 载波监听 CRS 和碰撞检测 COL 既不同步于时钟 TXCLK, 也不同步于时钟 RXCLK. 图 4 (a)、(b) 和 (c) 分别是正常发送数据、正常接收数据和数据包发生碰撞时相关的 MII 接口信号的波形. MII 接口发送数据宽度是 4 位 (称为 Nibble), 以小端方式组装, 即每个字节的低 4 位为第一个 Nibble, 高 4 位为第二个 Nibble, 物理层电路发送数据时, 首先发送每个 Nibble 的最低位, 最后发送每个 Nibble 的最高位, 接收数据时也按这个规则来组装. 在发送端, 每次发送数据包时, 在向物理介质上发送数据位之前, 必须先发送 64 位的前导码, 用于接收端电路恢复时钟信号, 使得接收端的时钟信号频率与发送端相等. 前导码的比特序列为 1010101010101010, 共 62 位, 紧跟以 2 位 11 作为数据帧的起始定界符. 用 FPGA 监听 MII 接口中与发送相关的信号, 捕捉位于紧跟前



(a) 正常发送



(b) 正常接收



(c) 网络碰撞

图 4 MII 波形图

Fig. 4 MII waveform

导码之后的 2 个连续的高电平,即可准确地确定正在发送的数据包的发送时间戳.如果发生碰撞,该数据包需要重发.重发时还是要先发送前导码,于是可以获取一个新的准确的发送时间戳.由此可见,这种使用 FPGA 获取发送时间戳的方法消除了网络中随机发生的碰撞造成的影响,能大幅提高发送时间戳的获取精度.以 100 Mbps 以太网为例, TXCLK 为 25 MHz,发送时间戳的精度小于 40 ns. 当一个时槽的长度是 512 位,即 5.12 μ s 时,若网络上发生碰撞,则发送时间戳的精度必然不会好于 5.12 μ s. 在接收端,数据包的接收时间戳也按此方法获取,可以消除 CPU 的中断响应时间的不确定性造成的影响,同时也消除了数据包在网卡芯片的缓存中的延迟造成的影响.同样以 100 Mbps 以太网为例, RXCLK 为 25 MHz,接收时间戳的获取精度小于 40 ns. 在基于嵌入式软件的方案中,假设 CPU 为 ARM7TDMI 内核,运行于 50 MHz,在没有操作系统的情况下,其中断响应时间为 20 ~ 400 ns;如果嵌入式软件采用基于操作系统的消息处理,即使排除数据包在 MAC 控制器缓存中的延迟的影响,接收时间戳的响应时间精度也不会优于 1 μ s.

FPGA 中包含频率补偿时钟模块,它利用数字电路硬件代替了模拟电路,实现了晶振纠偏功能,克服了晶振漂移,使得普通的廉价晶振也能用于精确的时钟同步.频率补偿时钟的原理如图 5 所示,它由一个 p 位时钟计数器、一个 q 位累加器和一个 r 位加数寄存器构成.每个晶振周期,累加器将自身的值与存储在加数寄存器中的值相加,结果存储在累加器中,同时产生一个进位标志位表示这次的加法操作是否溢出,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.单位时间内时钟计数器数值增加的次数是由加数寄存器的值和晶振频率共同决定的,改变存储在加数寄存器中的加数,就改变了累加器进行加法操作时发生溢出的频率,从而改变了时钟计数器数值增加的频率,实现了对晶振频

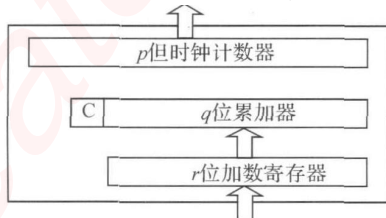


图 5 频率补偿时钟结构图

Fig. 5 Architecture of frequency compensation clock

率的补偿^[4]. q 和 r 越大,加数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.在本文设计的系统中,取 $p = 64$,前 32 位表示秒,后 32 位表示微秒, $q = r = 32$,对晶振频率补偿的步值小于 3×10^{-8} .

利用硬件描述语言, FPGA 完成了 2 个用一般 CPU 无法完成的工作.一是在 MII 接口处获取时间戳,二是用数字电路的方法实现对晶振频率的补偿,正是这 2 点对同步精度的提升起了决定性的影响. FPGA 中的控制器模块实现了时钟调节算法,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,即存储在频率补偿时钟中的加数寄存器中的值,从而达到了对时钟计数器的数值进行控制的目的.由于对网络传输延迟的滤波也在 FPGA 中完成, CPU 无法对时钟计数器的运行进行干预,只负责驱动 MAC 控制器完成相关的通讯功能. CPU 和 FPGA 之间有接口电路, CPU 对 FPGA 的访问等效于对片外内存的读写操作, CPU 可以通过这个接口完成本地时钟属性设定、时钟初值设定、时钟运行状态读取、某一事件的预定发生时间写入、已发生事件的时间戳读取等操作.

2 测试方法与应用验证

模拟基于工业以太网的分布式控制系统,搭建应用工业以太网的时钟同步测试系统,动态测试方式如图 6 所示.采用基于 FPGA 的时钟同步方案与嵌入式 CPU 接口模拟应用系统,另有第三方的测试电路模块与之配合.测试电路模块定期向 FPGA 发出一个测试脉冲,包括主时钟的 FPGA 和待测的若干从时钟的 FPGA,每个 FPGA 都有一个端口被连接到一根公共的铜导线上.这个同时传播到所有 FPGA 的测试脉冲被每个 FPGA 同步于自身的逻辑电路,然后用边沿触发器记录下当前的时钟计数器的数值.测试电路模块将这些数值读回,用串口发回测试计算机进行记录和分析.

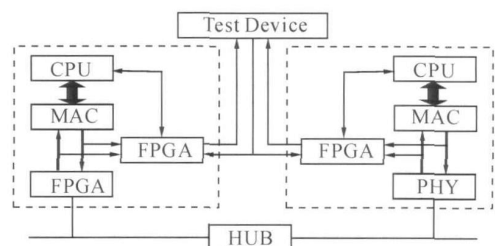


图 6 测试系统示意图

Fig. 6 Test embedded system

炉温的因素众多,可建立的模型方法也很多,但是要
实现基于模型的预测控制,数学模型必需包含3类
变量,即:初始变量、状态变量和控制变量,才能够达
到炉温 $w(Si)$ 的闭环控制.更进一步的模型控制研
究表明,成功的炉温预测控制方程必须建立机理模
型、推理模型和数理模型递阶式智能化集成的基础
上才能够实现,这些模型的互动将另文阐述.

参考文献(References):

[1] 刘祥官,刘芳.高炉炼铁过程优化与智能控制系统
[M].北京:冶金工业出版社,2003.
[2] 罗世华,刘祥官.高炉铁水含硅量的分形结构分析[J].
物理学报,2006,55(7):3343-3347.
LUO Shi-Hua, LIU Xiang-Guan. The fractal structure of
silicon content in molten iron in blast furnace [J]. *Acta
Physica Sinica*, 2006, 55(7): 3343-3347.
[3] 吴怀宇.时间序列分析与综合[M].武汉:武汉大学出
版社,2004.
[4] 郜传厚,周志敏,邵之江.高炉冶炼过程的混沌性解析
[J].物理学报,2005,54(04):1490-1494.
GAO Chuang-hou, ZHOU Zhi-min, SHAO Zhi-Jiang.
Chaotic analysis for blast furnace iron-making process
[J]. *Acta Physica Sinica*, 2005, 54(04):1490-1494.
[5] 王玉涛,严其艳,杨钢,等.高炉铁水含硅量的动态神经网
络多步预报[J].仪器仪表学报,2006,1:1448-1451.

WANG Yurtao, YAN Qiy-an, YANG Gang, et al.
Multi-step prediction of molten iron silicon content in
blast furnace using dynamic neural network [J]. *Chinese
Journal of Scientific Instrument*, 2006, 11:1448-1451.
[6] 王振龙.时间序列分析[M].北京:中国统计出版社,
2000.
[7] 石扬,张燕平,赵姝,等.基于商空间的气象时间序列数据
挖掘研究[J].计算机工程与应用,2007,42(1):201-203.
SHI Yang, ZHANG Yan-ping, ZHAO Shu, et al. Re-
search on meteorological time series data mining based
on quotient space [J]. *Computer Engineering and Appli-
cations*, 2007, 42(1):201-203.
[8] JIA Li, MENENTI M. Response of vegetation photo-
synthetic activity to net radiation and rainfall: a case
study on the tibetan plateau by means of fourier analysis
of MODIS FAPAR Time Series [J]. *Advances in Earth
Science*, 2006, 12:1254-1260.
[9] 徐东,刘志阳,徐奉臻.我国证券投资基金羊群行为的实
证分析(1999-2004)[J].哈尔滨工业大学学报,2006,
38:2132-2134.
XU Dong, LIU Zhi-yang, XU Feng-zhen. Empirical re-
search on herd behavior of security investment fund in
China based on LSV and TSP (1999-2004) [J]. *Journal
of Harbin Institute of Technology*, 2006, 38:2132-2134.
[10] 梅长林,周家良.实用统计方法[M].北京:科学出版
社,2002.

(上接第1700页)

本文设计的时钟同步系统运行在10 Mbps以
太网上,采用无数据帧缓冲的HUB互连各个设备,
共有1个主节点和10个从节点.在连续20 d的长
期测试中积累的测试数据显示,从时钟相对主时钟
的偏差的平均值小于 ± 35 ns,标准差小于100 ns,
峰值小于 ± 500 ns,这说明同步精度已经达到亚微
秒级.同步时间误差测试曲线见图7.图中: e 为同步
误差, n 为测试次数.

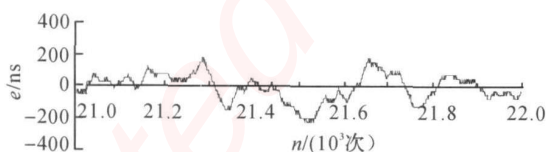


图7 时间同步测试曲线图

Fig.7 Test curve of time synchronization

3 结 语

本文采用FPGA硬件逻辑方法实现工业以太
网的高精度时间同步.通过实际应用验证,该方法具
有稳定性好、精度高等特点,采用硬件模块化结构可

以方便地与嵌入式实时控制系统集成应用.由于设
备所限,本文只完成了在10 Mbps以太网上的测
试,同步精度已达到亚微秒级.如果网络传输速率上
升到100 Mbps,预期可达到更高的同步精度.

参考文献(References):

[1] IEEE std 1588-2002, IEEE standard for a precision
clock synchronization protocol for networked measure-
ment and control systems [S]. New York: Institute of
Electrical and Electronic Engineers, 2002.
[2] EPA criterion for industrial control system, Digital com-
munication for measurement and control [S]. Beijing:
National Standard Committee, 2004.
[3] IEEE std 802.3-2002, Carrier sense multiple access with
collision detection (CSMA/CD) access method and
physical layer specification [S]. New York: Institute of
Electrical and Electronic Engineers, 2002.
[4] HARRIS K R, BALASUBRAMANIAN S, MOLDA-
VANSKY A. A frequency compensated clock for preci-
sion synchronization using IEEE 1588 and its applica-
tion to Ethernet [EB/OL]. 2003-11-12 [2007-01-20]. ht-
tp://iee1588.nist.gov

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. [基于磁盘异或引擎的 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 技术研究与发展](#)
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.