基于 FPGA NiosII 的等精度频率计设计 *

郝统关，程明
（郑州大学 信息工程学院，郑州 450001）

摘要：介绍了一种利用 FPGA 芯片设计的等精度频率计。对传统的等精度测量方法进行了改进，增加了测量脉冲宽度的功能，采用 SOPC 设计技术和基于 Nios II 嵌入式软核处理器的系统设计方案，通过在 FPGA 芯片上配置 NiosII 软核处理器进行数据运算处理，利用液晶显示器对测量的频率、周期、占空比进行实时显示，可读性好。整个系统在一片 FPGA 芯片上实现，系统测量精度高，实时性好，具有灵活的现场可更改性。

关键词：等精度；频率计；FPGA；SOPC；NiosII
中图分类号：TM935.13+3　　**文献标识码**：B　　**文章编号**：1001-1390(2009)02-0056-03

Design of Equal Precision Frequency Meter Based on FPGA NiosII

HAO Tong-guan, CHENG Ming
(Information Engineering College, Zhengzhou University, Zhengzhou 450001, China)

Abstract: An equal precision frequency meter designed with FPGA is introduced. It adopts Verilog Hardware Description Language to implement function module in frequency meter, and adds pulse width measurement on the base of traditional frequency measurement. SOPC designing technique and system designing plan based on Nios II soft core CPU are used in the design. It also adopts NiosII soft core CPU as data processing unit, uses LCD 1602 equipment to display frequency, periods and pulse in real-time. The whole system is in the realization of a FPGA chip. So it has a high-precision measurement, real-time and flexible change of scene.

Key words: equal precision measurement, frequency meter, SOPC, FPGA, NiosII

0 引　言

频率测量是电子测量技术中最基本的测量之一。传统测量频率的方法主要有直接测量法、分频测量法、测周法等，这些方法往往只适用于测量一段频率，当被测信号的频率发生变化时，测量的精度就会下降。因此，本文提出一种利用一片 FPGA 芯片来实现基于等精度原理的测量频率方法，在整个频率测量过程中都能达到相同的测量精度，而与被测信号的频率变化无关。系统利用 FPGA 的高速数据处理能力，实现对被测信号的测量计数，利用 Nios II 嵌入式软核处理器实现对频率、周期、脉冲宽度的计算及显示。

1 系统总体设计及工作原理

1.1 等精度测量原理

等精度测量原理图如图 1 所示。等精度测频技术的闸门时间不是固定的值，而是被测信号的整周期倍，即与被测信号同步，在计数允许时间内，同时对标准信号和被测信号进行计数，再通过数学公式推导得到被测信号的频率。由于门控信号是被测信号的整数倍，因此消除了对被测信号计数所产生的±1 误差，但是会产生对标准信号±1 的误差。因此，只要提高标准信号源频率，就可以大大提高测量精度，而且达到了在整个测量频段的等精度测量[1]。

1.2 系统整体硬件结构

系统总体硬件结构如图 2 所示。本系统选用

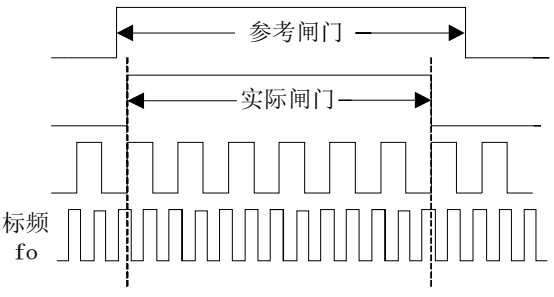

图1　等精度测量原理

Fig.1 Equal precision measuring principle

Altera 公司 Cyclone 系列型号为 EP1C6Q240C8 芯片，该芯片是一款高性价比的 FPGA 芯片，工作电压为 1.5V，其存储器密度可达 5980 个逻辑单元，包含 20 个 128× 36 位的 RAM 块，总的 RAM 空间达到 92160 位，可以实现 NiosII 嵌入式处理器，内嵌 2 个锁相环电路和一个用于连接 SDRAM 的特定双数据率接口；支持多种不同的 I/O 标准(包括 PCI 接口，以及串行设备接口等)[2]。在 FPGA 中定制了 NiosII CPU，用 Verilog HDL 语言编制了计数器控制模块，并利用 FPGA 的 IP 库资源生成了计数器模块，采用液晶和键盘来实现人机接口。

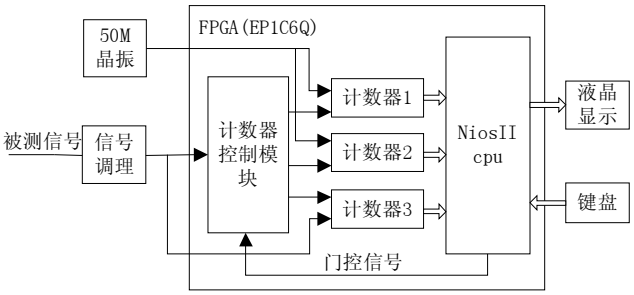

图2　系统硬件结构

Fig.2 System hardware structure

1.3　系统工作原理

被测信号经过信号调理电路变为被测脉冲信号，分为两路分别送至计数器控制模块和被测信号频率计数器(计数器 3)；计数器 1 用于标准频率计数，标频采用 50MHz 有源晶振提供；计数器 2 用于被测脉冲信号的脉宽计数。系统在 FPGA 内配置 NiosII 软核处理器来提供门控信号和进行数据处理。NiosII CPU 检测到启动按键后，根据被测信号频率的大小发出不同的门控信号，若被测信号频率小于等于 1Hz，则门控信号应大于 1 秒，本设计取 10 秒，因此，被测频率可测到 0.1Hz；若被测频率大于 1Hz，则门控信号取 1 秒即可。计数器模块在门控信号和被测脉冲信号控制下，使能标准频率计数器、被测信号计数器和脉宽计数器，当门控信号关闭(下降沿)后，NiosII CPU 分别读取这三个计数器的计数值进行运算处理得出被测信号的频率、周期和占空比，送至液晶(LCD1602)显示。

2　FPGA 模块设计

系统 FPGA 部分主要由计数器控制模块、计数器模块、锁存器、嵌入式 NiosII 软核处理器组成。整个 FPGA 核心模块如图 3 所示。

2.1　计数器控制模块

该模块主要用来同步被测信号。NiosII CPU 根据被测信号频率大小发出不同门控信号，门控信号启动

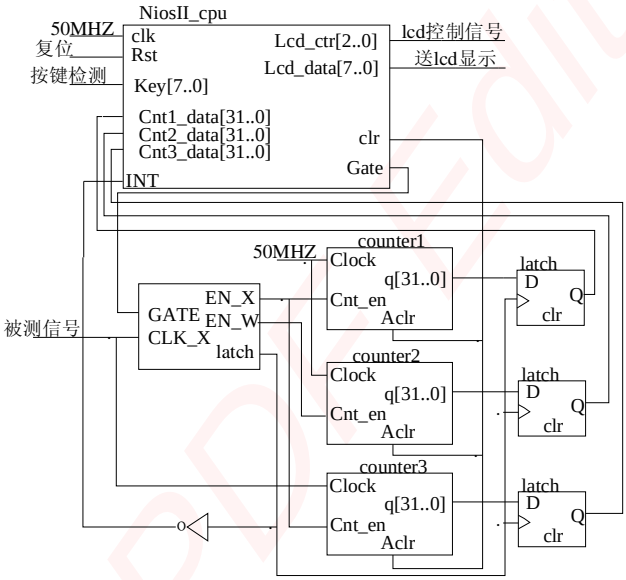

图3　FPGA整体结构

Fig.3 The overall structure of the FPGA

(上升沿)后，在被测信号的上升沿启动计数允许模块，允许计数器计数；门控信号关闭(下降沿)后，在被测信号的下一个上升沿关闭计数允许模块，停止计数，从而保证了门控信号是被测信号的整数倍，达到了等精度的目的，在停止计数的同时，发出锁存信号，一路锁存三个计数器的计数值，供 NiosII CPU读取处理，另一路取反后为 CPU提供查询信号，CPU根据此信号读取数据并处理显示。该模块是在 QuartusII 中用 Verilog HDL 硬件描述语言编写的[3]，有 GATE、CLK_x 这 2 个输入端；EN_x、EN_w、Latch，3 个输出端。GATE 为上一级给出的门控信号；CLK_x 为被测信号，EN_x 为给出的允许计数信号，在 EN_x 的高电平期间，对被测信号、标准信号同时计数，得到计数值 *cntx*，*cntb*；EN_w 高电平期间对标准信号进行计数，得到计数值 *cntw*；Latch 为锁存信号。由 *cntx* 和 *cntb* 可以计算出被测信号的频率和周期；由 *cnx*3和 *cntw* 可以计算出被测信号的脉冲宽度。具体理论分析如下：

设被测信号的频率为 F_x，标准信号频率为 F_b。则有：

$$cntx\times\frac{1}{F_x}=cntb\times\frac{1}{F_b}$$

由此得被测信号频率为：

$$F_x=\frac{cnts}{cntb}\times F_b$$

计数器的开关与被测信号是完全同步的，即在实际闸门中包含整数个被测信号的整周期，因而不存在对被测信号计数的± 1 误差。由上式对 *cntb* 求微分得：

$$dF_x=-\frac{cntx}{cntb^2}F_b\,dcntb$$

由上式可以看出，测量分辨率与被测信号频率的大小无关，仅与闸门时间及实际频率有关，即实现了被测频带内的等精度测量。闸门时间越长，时基频越高，测量分辨率就越高。被测信号周期为：$T_x=\frac{1}{F_x}$，占空比为：$H=\frac{cntw}{cntb}$。

2.2 计数器模块

该模块利用 QuartusII 中的 IP 资源生成。在配置过程中要注意计数器的宽度设置，本设计中配置了 3 个采用 32 位位宽的计数器，以保证计数值不溢出。

2.3 锁存器模块

该模块用于锁存计数器输出计数值，供 NiosII CPU 读取，进行处理显示。计数器模块在门控信号关闭（下降沿）的同时，停止计数，同时启动锁存模块，把测量的数据锁存起来，以便传输。

2.4 嵌入式NiosII软核处理器

设计所采用的处理器是 Altera 公司推出的第二代嵌入式 NiosII 软核处理器。NiosII 处理器是一个用户可配置的通用 32 位嵌入式精简指令集 CPU 和基于流水线设计的通用 RISC 微处理器，拥有五级流水线核指令与数据内存分开的哈佛结构。他包括三种软核 CPU：高性能软核、精简软核和标准软核。所有软核都是 100%代码兼容，设计者可根据系统需要选择 CPU 来调整嵌入式系统的性能及成本[4]。根据系统对 CPU 的性能要求，本设计选用了标准软核 CPU。

该模块是整个系统控制器的核心，它相当于单片机系统控制器中的 CPU，主要用来进行数据处理及显示。NiosII CPU 检测到外部中断信号时，分别读取计数器的计数值，然后处理后送液晶显示。

3 系统的软件设计

本系统软件是在 NiosII IDE 开发平台上利用软件集成开发工具 IDE 所提供的硬件抽象层（HAL）的函数支持下以 C 语言的形式编写完成[5]。系统的软件主流程图如图 4 所示。

系统运行开始，首先进行初始化，包括初始化液晶 LCD602，清零计数器。随后系统进入主循环状态，当检测到有按键时，启动测频，根据被测信号频率的大小发出不同的门控信号（GATE），若被测频率小于 1Hz，则发出 GATE 为 10 秒；若被测频率大于 1Hz，则 GATE 为 1 秒；门控信号上升沿启动控制计数器模块，计数器开始计数，当门控信号下降沿时，停止计数，同时发出锁存信号，当 NiosII CPU 检测到停止标志为低电平时，开始分别读取被测信号计数器、标准亲率计数器和测脉宽计数器的计数值，然后进行计算得出被测信号频率、周期和占空比，送液晶显示，随后程序跳至主循环。这样就实现连续测频功能。

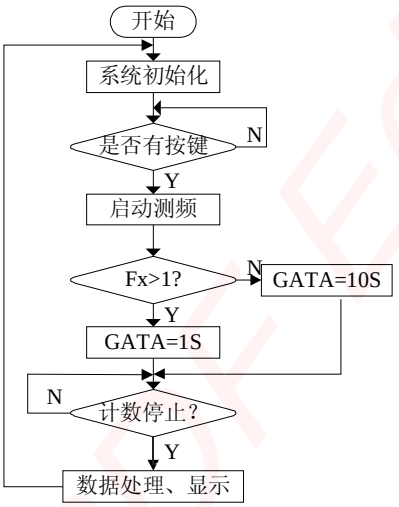

图4 系统软件主流程图

Fig.4 System software flow chart

4 结束语

该频率计利用 Quartus II 7.0 软件工作平台进行编译和综合仿真后，在自制的以 EP16Q240C8 为核心的开发板上进行了软硬件调试，功能全部正常，测试量程可自动切换，测量误差小于等于 0.1%。实验结果表明，此设计不仅具有设计功耗低、体积小、性能优越等特点，而且具有设计方式灵活、可裁剪、可扩充、可升级等优势，因此具有很好的应有前景和科研价值。

参 考 文 献

[1] 徐成,刘彦,李仁发,等.一种全同步数字频率测量方法的研究[J].电子技术应用,2004,38(12):43- 46.

[2] 徐光辉,程东旭,黄如.基于 FPGA 的嵌入式开发与应用[M].北京：电子工业出版社,2006.

[3] 王冠,俞一鸣.面向 CPLD/FPGA 的 Verilog 设计[M].北京：机械工业出版社,2007.

[4] 潘松,黄继业,曾毓.SOPC 实用教程[M].北京：清华大学出版社,2005.

[5] 郭书军,王玉花,葛纫秋.嵌入试处理器原理及应用——Nios 系统设计和 C 语言编程[M].北京：清华大学出版社,2004.

作者简介：

郝统关（1981-），男，汉族，河南焦作人，硕士研究生，电路与系统专业，主要研究方向为嵌入式系统。Email: htg3000@tom.com

程明(1949-)，男，汉族，河南郑州人，教授、硕士生导师，主要从事通信技术和计算机应用的研究。

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

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

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 CC++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

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

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

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μC-OSⅡ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

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

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