

基于 PCIE 104 总线的高速数据接口设计

杨子元^{1,2}, 包启亮², 王旭²

(1. 中国科学院 研究生院, 北京 100049; 2. 中国科学院 光电技术研究所, 四川 成都 610029)

摘要: PC/104 作为一种工业嵌入式的总线标准, 由于其小尺寸结构、低功耗, 以及软件通用性而被广泛用于航空航天、工业控制等领域。这里主要介绍了下一代总线技术 PCIE 总线在 PC/104 标准下的应用。阐述了数据接口卡的系统组成和各单元的功能, 以及驱动在 linux 操作系统下的实现原理。自行设计并实现了板卡原理图和 PCB, 同时简述了高速 PCB 仿真方面的知识, 并针对 PCIE 差分线进行了版级仿真。最后利用 PLX 公司提供的开发套件在 Linux 操作系统下完成了板卡驱动的开发和调试。

关键词: PCIE 总线; PC/104; 嵌入式; PCB 仿真

中图分类号: TN911-34

文献标识码: A

文章编号: 1004-373X(2011)14-0075-04

Design of High-speed Data Interface Based on PCIE/104 Bus

YANG Zi-yuan^{1,2}, BAO Qi-liang², WANG Xu²

(1. Graduate University of Chinese Academy of Sciences, Beijing 100049, China; 2. The Institute of Optics and Electronics, Chinese Academy of Sciences, Chengdu 610029, China)

Abstract: As a industry embeded bus, PC/104 is widely used in the fields of the aviation and industry control for the smart structures, low power, and compatibility with PC software. The application of the next generation bus technology (PCIE bus) which abides by the PC/104 specification is introduced. The system constitute of the data infterface card and the funtion of the each unit, as well as the principle of Linux drivers are elaborated. The board card principle diagram and PCB of the system were designed and relized independently. The knowledge in simulation of the high-speed PCB is expatiated. The broad simulation for the PCIE difference-trace was conducted. The driver for the board card was debugged in Linux with the help of PLX MDK.

Keywords: PCIE bus; PC/104; embeded bus standard; PCB simulation

0 引言

PCIE 总线是由 Intel 公司提出用来取代现行 PCI 总线的下一代总线技术, 被公认为未来总线的发展方向, 目前已经成功应用在了商业机上。相对于目前流行的 PCI 总线, 它具有如下特点:

采用了串行点对点模式, 数据总线从并行走回向串行无疑是一个趋势, 如现在流行的 SATA 总线、PCIE 总线和 rapidIO 总线都是串行总线。现在串行总线在速度上的优势可以说是毋庸置疑的。PCIE 总线在每个方向上都有 X1, X2, X4, X8, X16 或 X32 个信号对, 用户可根据不同的需求采用不同的配置。同时串行信号还具有管脚更少, 便于调试的优点。

PCIE 采用了基于数据包的协议来编码事物, 而不是 PCI 体系结构的总线周期。数据包被串行发送和接受, 并被字节拆分来通过物理链路。链路上实现的通道越多, 数据包发送的数据越快, 链路的带宽也越宽。同

时 PCIE 还定义了各种类型的数据包, 如存储器读/写请求、I/O 读/写请求、配置读/写请求、消息请求和完成数据包等^[1]。

PC/104 标准是一种嵌入式的总线标准, 具有功耗低, 尺寸小, 堆栈式结构的特点^[2]。随着目前各种应用数据传输量的增大, 现行的 PCI 总线在带宽方面已经略显疲态, 而新近瑞士逻辑提出的 PC/104 Express 标准, 使得 PCIE 总线技术被成功地应用在了 PC/104 标准的板卡上。本文采用了 PLX 公司的 PEX8311 桥接芯片, 完成了 PCIE 到局部总线接口的转换, 应用 PLX 公司提供的开发工具在 Linux 操作系统下实现了板卡的驱动应用。在软硬件两方面进行了设计, 完成了 PCIE/104 板卡的功能, 升级了系统的总线。

1 PCIE/104 高速信号接口卡的系统

PCIE/104 高速信号接口卡的系统设计方案如图 1 所示。

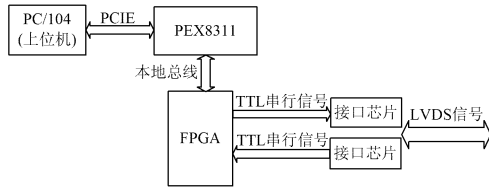


图1 系统总体框图

外部传感器采集到的数据通过 LVDS, RS 422 等电平方式,以固定的不同串口波特率传到 FPGA 内部集成的各自对应的双口 RAM 中。当写入了固定字节后,设定标志位 stage 为 1,同时发出中断信号 LINT # 给 PEX8311。PEX8311 产生了 assert_INTI 信息,并通过 PCIE 接口发给上位机。上位机保存好现在的任务后,通过 PCIE 开始发出存储器读命令给 PEX8311。PEX8311 获得命令后,向 FPGA 发出本地总线申请信号,FPGA 作为本地端的控制器将本地总线控制权交给 PEX8311。PEX8311 开始读数据,首先要进行的是读取标志位,在读取了标志位后,上位机就知道是哪几路串口信号需要读入。然后,把标志位清掉,这样相当于清掉了中断信号。接着,中断服务程序在进入到各个串口的 RAM 中,来读取相应的数据。在多路串口信号传输过来时,针对该系统,采用的是优先满足高速串口的原则,也就是采用高速串口的标志位作为中断信号,每次产生中断后查询其他标志位。数据到上位机后待处理。该系统目前实现的是两路串口,而多路串口原理相同。

1.1 硬件总体概述

主要包括以下几部分,PEX8311 桥接芯片负责完成本地总线和 PCIE 总线的相互装换;FPGA 选用了 spartan-3an 1400 K。这里在内部实现了 3 个模块,首先完成双口 RAM 模块;其次完成了多路串口数据的接受与协调模块;最后还要实现 PEX8311 芯片本地端控制器模块。

1.1.1 PEX8311 芯片介绍

PEX8311 是 PLX Technology 公司推出的一款专用于将 DSP, FPGA 等处理器总线接口升级为 PCIE 的桥接器件。利用 PEX8311 灵活的局部总线可以方便地连接多种存储器、缓存器及 FPGA, DSP 等逻辑芯片,使复杂的 PCIE 接口设计简单化。PEX8311 兼容 PCI Express 1.0 标准,其本地总线和寄存器与 PCI9056 兼容,能够提供完整的本地总线到 PCIE 的接口,包括地址转换、包生成与解码、信号中断支持及并/串转换等。

PEX8311 的特点如下:

- (1) 集成了单通道、全双工 2.5 Gb/s 传输的 PCI Express 端口;
- (2) 可配置局部总线宽度,支持 8 位、16 位和 32 位的总线方式;
- (3) 支持单路和多路总线操作模式;
- (4) 高性能的 DMA 数据传输,支持数据块模式、集散模式、循环队列管理模式和命令模式;
- (5) 支持端点和根复合模式;
- (6) 芯片小型封装,适合紧凑的电路板设计;
- (7) 芯片低功耗设计;
- (8) 3.3 V 的 I/O 并兼容 5 V 系统;
- (9) 启动配置的串行 E²PROM 支持(SPI 和 Microwire 接口);
- (10) 有 8 KB 的通配共享 RAM。

PEX8311 的内部结构如图 2 所示^[3]。

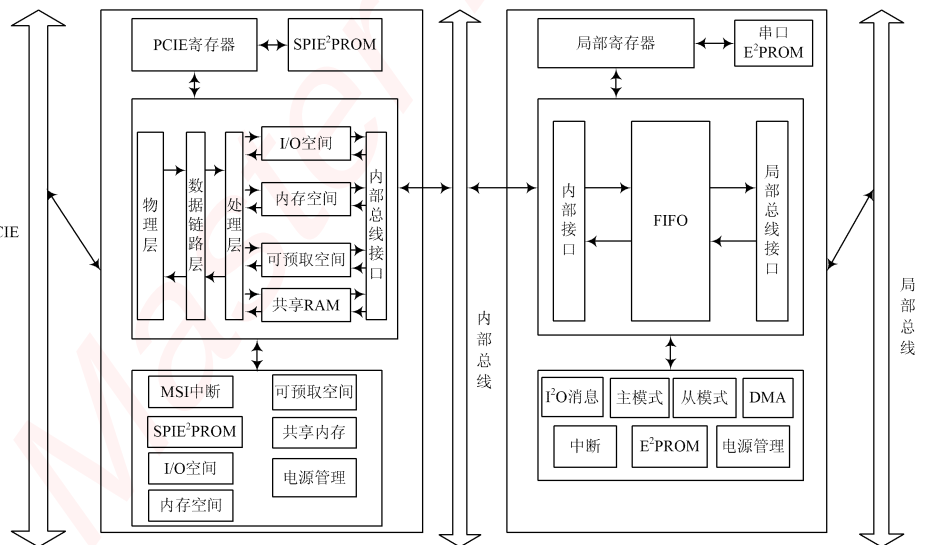


图2 PEX8311 内部结构

1.1.2 PCIE/104 总线标准介绍

PC/104 是最早由瑞士逻辑提出的一种工业总线标准,它由最早的 PC/104 总线发展到 PC/104⁺ 总线,直到目前的 PC/104 Express,分别对应 ISA 总线、PCI 总线和 PCIE 总线。由于它是一种堆栈型的嵌入式总线,所以将 PCIE 总线应用在这个标准上与普通的 PCIE 金手指有一些不同。为了满足 PC/104 的嵌入式堆栈结构,使其能够实现从板子上、下都可以连接,必须采用 PCIE Switch 芯片,这里使用的是 PERICOM 公司制造的 PI2PCIE2412 款 Switch 芯片。电路原理图如图 3 所示^[4]。

图 3 中接受、发送和差分时钟这 3 对差分信号线通过转接芯片变成了 6 对差分信号线。由主机板卡发出选择信号(cpu_dir),分别控制设备板卡是在主机板卡

的上面还是下面。同时采用 MAX6306 芯片来链接 PEX8311 的复位端和 PCIE 接插件的复位端,同时实现了手动复位和 LOCAL 端的复位。

1.1.3 PEX8311 的外围电路配置与 LOCAL 端接口的控制

PEX8311 有几个类别的模式选择,分别是根复合模式(Rootcomplex)与终端模式(Endpoint),根据需要这里选择了终端模式。此外 PEX8311 还区分 C 模式 J 模式,M 模式。这三种模式解释如下:C 模式的地址线与数据线是分开的,J 模式下地址线与数据线是复

用的,而 M 模式是针对了 Motor 的本地端 CPU 设置的。三种模式通过模式选择管脚选择,这里选择 C 模式作为工作模式。PEX8311 有 1.5 V,2.5 V,3.3 V 三种电源供电以及一个模拟电源 1.5VPLL。在配置寄存器方面,PEX8311 有两个配置寄存器,分别是采用 Spi-Compatible 接口的 PCIE 配置寄存器和采用了 Micro-Wire-Compatibel 接口的本地配置寄存器^[5]。采用了 Spartan-3an 1400 k 来提供控制接口与高速缓存,选择这款 FPGA 主要因为它是有内部 FLASH,不用再加外部的 E² PROM。

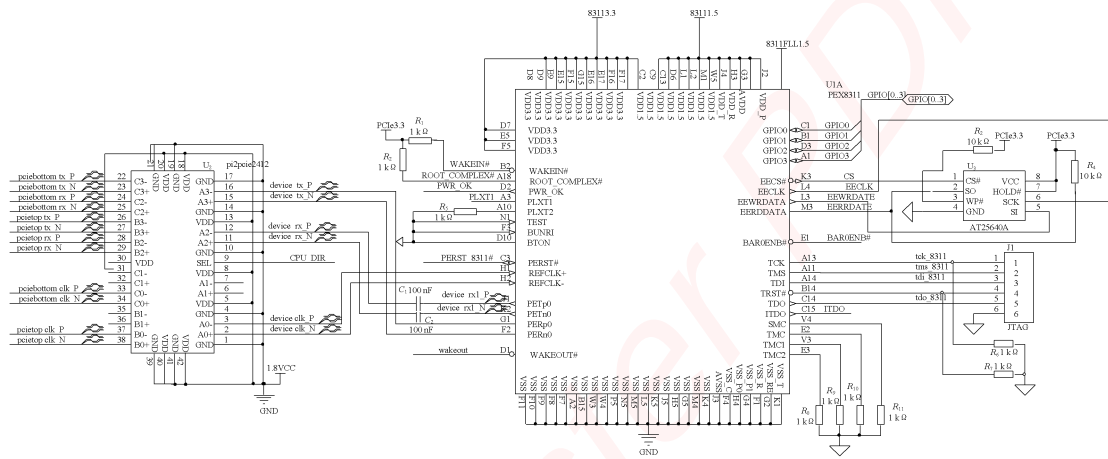


图 3 电路原理图 PCIE 口

由于 PC/104 是一种嵌入式的板卡,体积比较小,所以用这款 FPGA 是可以节省板上的空间。FPGA 内部程序的编写是关键之一。这里采用 Verilog,实现对 PEX8311 的控制,当上位机响应了中断后,对 PEX8311 发出读数信号。PEX8311 通过 L HOLD 申请控制本地总线,待收到 FPGA 发出的 L HOLDA 响应信号后获得本地总线的控制权,并立即启动 4 B 突发模式。FPGA 在收到有效的 LW/R 读信号和 ADS 地址选通信号后,发出 Ready 本地准备好应答信号。PEX8311 开始从双口中读取数据,传输最后一个数据时,PEX8311 发出 BLAST 信号,双口 RAM 使得读使能和输出使能无效。

1.2 系统软件部分的实现

该系统的软件部分是在 Linux 下实现的,在 Linux 中所有的设备都被看成文件来对待^[6]。在 Linux 内核中,设备驱动作为文件系统的一个模块存在。它向下负责与硬件系统的交互,向上通过一个通用的接口挂接到文件系统上面。从而和系统的内核链接起来。设备驱动为应用程序屏蔽了很多细节。使得应用程序对外设的操作就和操作普通的文件是一个样子的。利用 PLX 公司提供的开发工具,驱动的开发是比较方便的,本文系统中主要用到的是要编写一个中断服务程序:系统在

收到中断,保存现场。进入中断服务程序。首选读取标志位,再马上清中断,之后读取响应 RAM 中的数据。最后恢复现场,完成操作。

2 系统 PCB 设计和高速信号的完整性分析

由于 PCIE 的传输速率较高,单向速率达到^[7] 2.5 Gb/s。所以对板子的布线有严格的要求。叠层这里选用了 8 层 PUCB 板,有 4 个电源层,使得每个信号都能够屏蔽在电源层与地层中间。从而减少了信号的电磁辐射。对于 PCIE 的差分线部分:微带线要求差分线宽 5 mil,间距小于 12 mil,差分线间距离大于 20 mil,同时与地层间距为 3.5~5.5 mil。带状线要求线宽 4 mil,间距小于 11 mil,间距大于 20 mil。且对于收发差分线,差分线长差距不能大于 5 mil。这些都是为了能达到 PCIE 规范的要求,即单端阻抗 55 Ω ,差分阻抗 100 Ω (偏差 10%)^[8]而设置的。

经过 ploar 软件计算,将以上的数据输入进去,再加入 PCB 厂家提供的介电常数等参数,最后得出的阻抗完全满足设计需求。多层板的高速信号设计很有必要进行信号完整性仿真,应用厂家提供的 IBIS 模型,采用 Hyperlynx 对板子进行了本地端和时钟端的信号完整性仿真。并根据仿真对布线提供了约束条件。而针

对 PCIE 的高速差分端总线,因为在高速信号仿真方面 IBIS 模型还不够精确,所以,PLX 公司对其两对收发端口提供 SPice 模型,如图 4 所示^[9]。

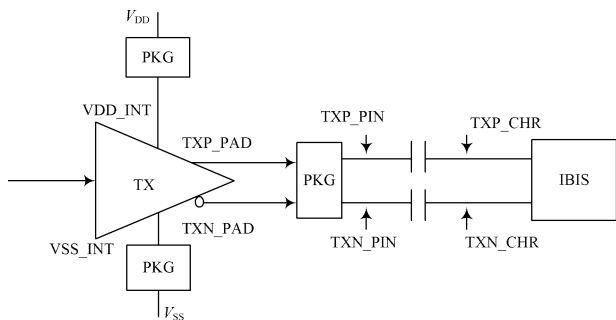


图 4 PCIE 差分接口 Spice 模型

将上述模型导入 HSpice 中,同时引入 PCIE Switch 芯片给出的 IBIS 模型与板子上的差分线 trace 的 rglc 模型。输出端的仿真效果图如下,可以看出差分信号的幅值是可以满足 PCIE 规范的电气要求的。

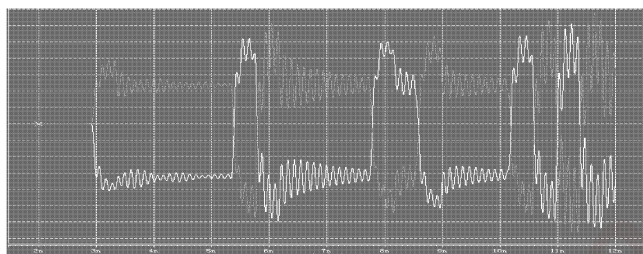


图 5 PCIE 差分线的板级仿真结果

3 结 语

PC/104 作为一种嵌入式总线标准已经被很多控

作者简介: 杨子元 男,1984 年出生,内蒙古包头人,硕士研究生。研究方向为高速信号的传输。

包启亮 男,博士,研究员。研究方向为总体光电系统控制。

王 旭 男,硕士,副研究员。研究方向为光电系统的电控设计。

(上接第 67 页)

[3] 安晓莉. AT89S51 单片机开发板的设计[J]. 电子设计工程, 2009, 17(9): 121-123.
[4] 王军琴. 基于 DSP 的单总线温度测量系统设计[J]. 工矿自动化, 2010(7): 35-37.
[5] 姚伟鹏,侯睿,肖军. 基于 AT89S 系列单片机实验板开发[J]. 西安航空技术高等专科学校学报, 2009, 27(5): 23-27.
[6] 邵贝贝. 单片机嵌入式应用的在线开发方法[M]. 北京:清华大学出版社, 2004.

制系统所采用,而 PCIE/104 接口的提出将未来最为流行的串行差分总线结构,引入到了这种嵌入式总线标准,从而为各种高速、高带宽的嵌入式系统提供了选择。目前该系统由于采用了 PCIE 总线,相比采用 PLX9054 实现的 PCI 接口具有明显的优势。

参 考 文 献

[1] 田玉敏,王崧,张波. PCI Express 系统体系结构[M]. 北京:电子工业出版社, 2005.
[2] Anon. PCI/104-express™ & PCIe/104™ specification [S/OL] [2008-03-24]. http://www.pc104.org/pci104_Express_specs.
[3] PLX. PEX8311 data book V0. 95 [M]. [S. l.]: PLX, 2007.
[4] PLX. PEX8311 RDK hardware reference manual version 0.9 [M]. [S. l.]: PLX, 2005.
[5] 马萍,唐卫华,李绪志. 基于 PCI Express 总线高速数据采集卡的设计与实现[D]. 北京:中国科学院研究生院, 2008.
[6] 魏永明,耿岳,钟书毅. Linux 设备驱动程序[M]. 北京:中国电力出版社, 2006.
[7] 许军,李玉山,贺占庄,等. PCI Express 总线技术研究[J]. 计算机工程与科学, 2006, 28(5): 141-143.
[8] PCI Special Interest Group. PCI express base specification revision 1. 0a [S]. [S. l.]: PCI Special Interest Group, 2006.
[9] 周俊峰. 基于 PCIE 总线数据采集系统硬件设计[D]. 成都:电子科技大学, 2009.
[10] 张海兵,李敏. Protel 电子设计实例与分析[M]. 北京:人民邮电出版社, 2005.

[7] 邵贝贝,宫辉. 嵌入式系统中的双核技术[M]. 北京:北京航空航天大学出版社, 2008.
[8] 李杰,曹宇,朱坚,等. 基于嵌入式 Linux 的矩阵键盘设计与实现[J]. 现代电子技术, 2006, 29(24): 81-83.
[9] 孙同景,陈桂友. Freescale 9s12 16 位单片机与嵌入式开发系统[M]. 北京:电子工业出版社, 2003.
[10] 楼然苗,李光飞. 单片机课程设计指导[M]. 北京:北京航空航天大学出版社, 2007.

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](#)

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](#)

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](#)

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](#)

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](#)

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](#)

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](#)