

基于 PCI 总线的图像处理及传输系统的设计

吴 静,马亚非

(中国空空导弹研究院 河南 洛阳 471009)

摘 要: PLX 公司的 PCI 9054 是一种通用的 PCI 总线接口芯片,可方便地进行软硬件开发实现 PCI 总线桥接。介绍了利用 PCI 9054 开发的图像处理及传输系统和配套的 WDM 驱动程序,以 FPGA 为系统硬件框架核心,从接口 LVDS 和 RS 422 读入图像数据及其他同步帧数据,经过处理后通过 PCI 接口芯片及 PCI 驱动程序进行与计算机系统的数据交换,从而实现基于 PCI 总线的高速图像传输处理。利用 PCI 9054 开发 PCI 高速数据传输系统具有较好的性价比。

关键词: PCI; PCI 9054; WDM; 设备驱动程序; FPGA; DMA

中图分类号: TP393.4

文献标识码: B

文章编号: 1004-373X(2007)18-117-04

Design of a Image Processing and Transferring System Based on PCI Bus

WU Jing, MA Yafei

(China Airborne Missile Academy, Luoyang, 471009, China)

Abstract: The PCI9054 is an universal PCI bus-chip developed by PLX Co., who has facility with bridging PCI bus by expediently hardware and software developing. Image processing and transferring system based on PCI9054 bus chip and the corresponding WDM drivers is briefly introduced. Hardware with FPGA centered, acquire the image data and the frame data from the LVDS and RS-422 interface, and exchange the processed data with computer system by using the PCI interface chip-set and the PCI device driver. Figure out the sketch of high speed image processing and transferring system. The application of PCI9054 in high speed data transferring system based on PCI bus is evidently appropriate.

Keywords: PCI; PCI9054; WDM; device driver; FPGA; DMA

1 引 言

PCI 总线是 Intel 公司推出的一种先进的高性能 32/64 位局部总线,可同时支持多组外围设备,不受制于处理器,数据吞吐量大(33 MHz 总线频率,32 位传输时峰值可高达 132 MB/s)。目前 PCI 是主流的计算机总线。

PCI 设备可分为主模式和从模式。主模式桥芯片可以进行 DMA 操作,而从模式只能收受读写操作。选择芯片不仅考虑性能和经济上的要求,而且还需要考虑硬件开发和驱动程序开发的难易。如果不提供足够的芯片说明和应用样例及开发工具,将大大增加开发难度和延长开周期。目前市场流行的 PCI 接口芯片见表 1。

通过表 1 的比较,结合项目的需求,本次设计中采用 PCI9054 接口芯片是较为理想的选择。

2 方案设计

根据 PCI 总线的特点,充分利用 PC 机软硬件资源,设计了基于 PCI 总线的高速数字图像采集卡作为远端实物与 PC 机数据通信的桥梁,实现将实时采集的同步串行数据流通过 PCI9054 接口控制芯片存储到计算机硬盘中,

用于实时显示或事后的处理分析。

表 1 各公司 PCI 接口芯片及相关特点

公司	芯片型号(模式)	芯片功能及特点	价格	开发技术支持
PLX	PCI 9052 (从) PCI 9054 (主)	型号众多,使用方便,性能好	较便宜	提供快速开发板 RDK 出售
CYPRESS	CY7C09449PV-AC(主)	内置 DPRAM,有效降低系统成本	便宜	技术文档
TI	PCI 2031 (从) PCI 1251 (主)	与 TI 的 DSP 可无缝连接	很便宜	提供评估模块
AMCC	AMCC 5920 (从) AMCC 5933 (主)	FIFO 接口适合于设计数据采集卡,性能好	较昂贵	提供评估板

2.1 硬件实现

数据先通过接口芯片进行电平转换,转换成 FPGA 的兼容电平,然后进入 FPGA 进行串并转换,之后送入存储器,再由 FPGA 从存储器中读出,通过 PCI 总线控制器,发送到 PCI 连接器,最后通过计算机的驱动程序和应用程序,把数据送到计算机的存储器内。原理框图见图 1。

串行数据的串并转换、缓存、处理和时序逻辑控制等功能的实现采用的 FPGA 是 Altera 公司的 Cyclone 系列 EP1C6。Cyclone FPGA 是目前市场上性价比最优且价格最低的 FPGA,器件基于成本优化的全铜 1.5 V SRAM 工

艺,Cyclone FPGA 支持各种单端 I/O 标准如 LVTTTL, LVCMOS,PCI 和 SSTL-2/3。

FPGA 的 I/O 端口多,可通过 VHDL 语言编程来自由支配,定义其接口功能,方便 PCB 版图设计时的布局布线,而且 FPGA 硬件的速度是 ns 级的,VHDL 程序内部对各功能模块的处理是按并行方式进行的,这样既很好地解决了采集信号通路多的问题,又能实时地、快速地传输处理高速数据流。同时,借助 EDA 工具软件 Quartus 直接进行代码编写、功能仿真和时序仿真,简单易行地完成硬件功能的验证、添加和修改。配置器件采用 EPCS1 JTAG 接口可以用来调试 FPGA,下载速度比较快,而且支持 Signal TAP。但是不能用来编程 EPCS1 芯片,故调试阶段采用 JTAG 模式。AS 接口主要是用来编程 EPCS1 芯片,同时也可以用来调试。具体过程是首先编程 EPCS1,然后通过 EPCS1 配置 FPGA,运行程序。

需要考虑的是 EPCS1 的编程次数是有限制的,虽然比 EPC 系列要多,但是太频繁的擦除和写入对芯片还是有一定影响的。所以,在调试结束后,程序固化的时候才使用 AS 方式。如果采用这种方式,必须采用 ByteBlaster 电缆才行。

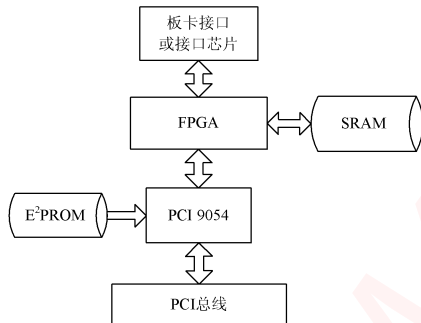


图1 PCI板卡硬件原理框图

存储器 SRAM 采用 ISSI 公司的 IS61LV6416,可提供 8 位、16 位输入、输出接口,最大存储容量为 64 k × 16 位。系统中使用 2 片 SRAM 作宽度扩展提供 32 位数据接口。

RS 422 接口芯片 MAX 1484 提供全双工、12 Mb/s, RS 422 收发器。单电源 +5 V 供电。

LVDS 接口芯片 MAX 9126 提供 4 路 LVDS 线接收器,内部端接。可接收 500 Mb/s (250 MHz) 数据。接收 4 路 LVDS 差分输入转换为 3.3 V CMOS 输出。内部集成并联终端电阻 (115 Ω),输入: 0.05 ~ 2.35 V。引脚与 DS90LV032A 兼容,支持 ANSI TIA/EIA-644 LVDS 标准。单电源 +3.3 V 供电。

E² PROM 存放厂家标志,设备标志及本地总线的基地址空间、I/O 空间、中断控制信号等信息。初始化时,系统自动将 E² PROM 的配置参数装入 PCI 配置寄存器,并根据本地总线对内存、I/O 端口和中断的需求统一划分,自动配置。本设计选用 Atmel 公司的串行 E² PROM

93LC56 作为 PCI 9054 的配置存储器。PCI 9054 有 4 根信号线用于与 E² PROM 的接口: EESK, EEDI, EEDO 和 EECS,串行 E² PROM 也有 4 个相应的引脚: CS, DI/DO, CLK, 其对应连接。

2.2 软件设计

为了保证系统的安全性和可移植性,Windows 操作系统对应用程序针对硬件的操作进行了限制,尤其 Windows 2000 和 Windows XP,不支持直接对系统的硬件资源的操作。因而在设计开发 PCI 设备时,需要开发相应的驱动程序来实现对 PCI 设备的操作,用户应用程序通过驱动程序来访问 PCI 设备,实现设计功能的原理框图见图 2。

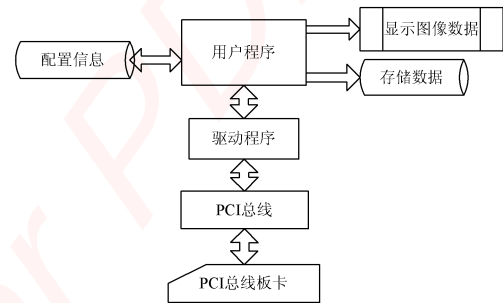


图2 PCI板卡系统软件原理框图

3 PCI 9054 的总线操作和工作模式

PCI 9054 具有较强的性能,符合 PCI v2.2 规范,集成了 2 个互相独立的 DMA 通道,每个通道都支持 Block DMA 和 Scatter / Gather DMA 模式,本地总线时钟可以与 PCI 时钟异步,最高为 50 MHz;本地总线可设为 8,16,32 位 3 种宽度,传输速率最高可达 132 MB/s;支持复用/非复用的 32 位地址/数据;本地总线有 3 种工作模式:M 模式、C 模式、J 模式,可利用模式选择引脚加以选择;M 模式用来与 Motorola 的 MPC 850 或 MPC 860 处理器进行无缝连接,C 模式中地址总线 and 数据总线为非复用的,J 模式则为复用的;提供 3 种物理总线接口:PCI,LOCAL,串行 EPROM(可选);PCI 9054 内部有 6 种可编程的 FIFO 存储器,以实现零等待突发传输以及本地总线和 PCI 总线间的异步操作,他们分别完成 PCI 发起读、写操作,PCI 目标读、写操作和 DMA 读、写操作。

根据系统实际工作情况,将本地总线设置为 C 模式,从而地址/数据复用的 PCI 总线转化成易于使用的、非复用的本地总线,工作频率为 40 MHz,本地总线宽度设置为 32 位。

本设计采用的 Block DMA 模式要求 PCI 主机或 Local 主机提供 PCI 和 Local 的起始地址、传输字节数、传输方向。主机设定 DMA 开始位启动数据传输,一旦传输完成,PCI 9054 设定 DMA“传输结束位”结束 DMA,如果中断允许位被使能,在传输结束时 PCI 9054 将向主机申请中断。在 DMA 传输中,PCI 9054 既是 PCI 总线的主控器

又是 Local 总线的主控器。

4 PCI 设备驱动程序的特点

PCI 总线是一种高性能、与 CPU 无关的 32/64 位地址数据复用的总线,他支持突发传输、即插即用、电源管理等功能。PCI 总线支持硬件资源动态自动配置,以支持即插即用。在 PCI 设备插入 PCI 插槽或上电后,PCI 总线配置机构自动根据 PCI 设备的要求实现配置。PCI 总线支持内存读写、I/O 端口读写、中断机制和 DMA 功能。由于这些硬件特点使 PCI 设备的 WDM 驱动程序的设计变得很复杂。在开发 WDM 驱动程序之前,还必须掌握 PCI 设备的需要分配的资源等配置信息以及 PCI 设备的功能和操作方法。

在 WDM 中,采用分层的驱动程序体系结构,总线驱动程序或类驱动程序在最底层直接与设备打交道,设备功能驱动程序在上层通过与低层驱动程序打交道,实现设备的功能,中间还可以有类过滤驱动程序或设备过滤驱动程序用于数据的过滤或转换。在 PCI 总线的驱动程序层中,其层次图如图 3 所示。

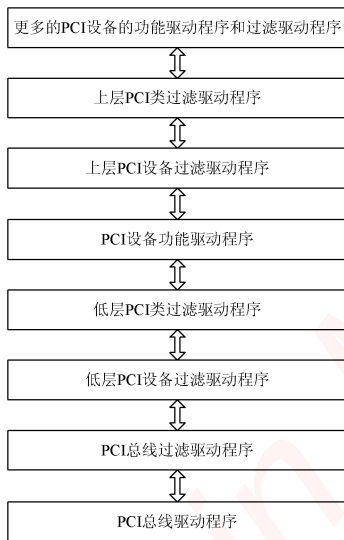


图 3 通用 PCI 总线的 WDM 驱动程序栈

5 WDM 驱动程序的设计

在 PCI 设备的 WDM 驱动程序中,一般是编写功能驱动程序。PCI 总线驱动程序由操作系统实现,过滤驱动程序一般在特殊的情况下需要编写。因此本文只讨论 PCI 设备功能驱动程序的设计。在 PCI 设备功能驱动程序中,需要处理 PCI 设备的内存、端口的读写、中断处理和 DMA 数据传输,实现 PCI 设备的功能,因此,PCI 设备功能驱动程序是很标准的 WDM 设备驱动程序。

PCI 设备驱动程序在框架上与其他类型的设备驱动程序基本相同,包括初始化、创建设备、卸载和删除设备、即插即用处理、分发例程处理、电源管理、WMI 等部分,限

于篇幅,在此只讨论 PCI 设备的特别之处。

(1) PCI 设备资源的获得

PCI 设备的硬件资源是由 PCI 配置机构动态分配的,由 PCI 设备实现 PCI 配置寄存器,提出需要分配的硬件资源,由 PCI 配置机构分配资源。驱动程序需要取得这些资源,才能操作硬件。因此,PCI 设备的硬件资源分配与管理是驱动程序中很重要的部分。硬件资源主要包括映射内存空间、I/O 空间、中断。当系统的 PNP 管理器在取得设备的资源后会向驱动程序发出 IRP_MN_START_DEVICE 的 IRP,在该 IRP 栈中包含了设备的资源信息。较为成熟的驱动程序都使用这种方法来获取 PCI 设备资源。

每个支持 PNP 功能的驱动程序,都应实现 IRP_MN_START_DEVICE 处理。在该 IRP 处理中应先交给低层驱动程序处理后,再根据 IRP 栈内内容进行资源分配。

(2) 内存读写

在 WDM 驱动程序中,对于硬件的内存映射一般需要用非分页内存,因为在一些运行在 DISPATCH_LEVEL 或更高的中断级例程中,禁止使用分页内存,比如在中断处理程序中就不可以使用分页内存。再者,使用非分页内存无需太多的转换,非常安全,效率也高。如果使用分页内存,系统就有可能将其调配到硬盘上,容易产出错误。但是,不能过多地使用非分页内存。

(3) I/O 读写

在 PC 上,I/O 空间是一个 64 KB 的寻址空间。I/O 端口的寻址方式与内存是不一样的。但是在 WDM 驱动程序中,对其处理与内存是一样的,将其看作寄存器,映射为设备内存。其映射方法和访问函数的用法与内存资源一样,只不过函数 XXX_REGISTER_XXX 改为 XXX_PORT_XXX。

(4) 中断的处理

在中断服务例程中,首先必须根据硬件信息来判断该中断是否是自己的设备发出的。这是因为 PCI 总线共享中断,系统在接收到中断后,顺序调用各个注册该中断资源的驱动程序的中断处理例程。如果有返回 TRUE 的例程,就代表该中断已处理,就不再调用其他例程。如果是返回 FALSE 的例程,则说明该中断没有处理,则继续调用其他的例程;如果返回错误,就会扰乱系统,造成系统崩溃。其框图如图 4 所示。

在中断服务例程中,相应的处理最好简洁快速,因为中断例程运行的级别很高,当有中断请求时,不但会打断应用程序的执行,而且会打断在硬件中断级以下的所有运行程序。在 WDM 中,提供了 DPC (Deferred Procedure Call) 例程,将在中断例程中耗时的但不需要立即处理的任务延时处理。比如,驱动程序接受应用程序的写 PCI 设备的数据,当写完后,硬件产生中断标志执行完毕,这时需要

结束该 IRP,就可以将结束 IRP 这个耗时的任务交给 DPC 完成。

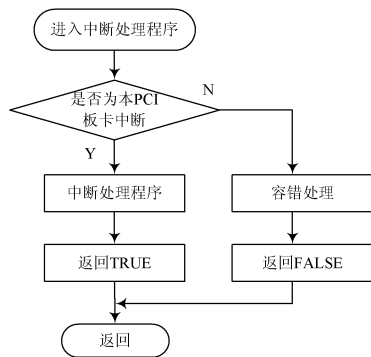


图 4 中断服务例程框图

(5) DMA 的处理

DMA 突发模式下,9054 作为 PCI 总线的主设备,同时也是 Local 总线的控制者,通过设置其 DMA 控制器内部的寄存器即可实现两总线之间的数据传送。

PCI 9054 的 DMA 传输过程可由以下几个步骤实现:

设置方式寄存器:设置 DMA 通道的传输方式,寄存器 DMAMODE0 或者 DMAMODE1 的位 9:0:表示块传输;1:表示散/聚传输。

设置 PCI 地址寄存器:设置 PCI 总线侧的地址空间。

设置 Local 地址寄存器:设置 Local 总线侧的地址空间。

设置传输计数寄存器:以字节为单位设置传输数据量。

设置描述寄存器:设置 DMA 传输的方向;在散/聚方式下,位 0 表示传输参数的加载地址,0:PCI 地址,1:

作者简介 吴静女,1980 年出生,陕西西安人,硕士研究生,助理工程师。

马亚非男,1978 年出生,河南洛阳人,本科,工程师。

(上接第 116 页)

5 结语

使用 MRA+LMBP 组合负荷预测模型进行负荷预测,对比于单纯的 LMBP 神经网络模型或其他传统方法,可以在减小负荷预测的平均误差的同时,在一定程度减小负荷峰值预测误差,从而提高了预测精度。具有较好的应用价值。

参考文献

- [1] Park D C, El-Sharkawi M A, Marks R J, et al. Electric Load Forecasting Using an Artificial Neural Network [J]. IEEE Trans. on Power Systems, 1991, 6(2): 442-449.
- [2] Otavio A S Carpinteiro, Agnaldo J R Reis, Alexandre P A da

作者简介 罗枚女,1969 年出生,陕西长安人,硕士,讲师。主要研究领域为智能控制。

Local 地址;位 1 表示传输链结束,0:未结束,1:结束;位 2 设置当前块传输结束后中断;位 3 指示 DMA 的传输方向,0:从 PCI 总线到 Local 总线,1:从 Local 总线到 PCI 总线;高 28 位[31:4]表示传输参数表的地址指针。

设置命令/状态寄存器:启动或停止 DMA 操作,并读此寄存器返回 DMA 状态。

6 结语

本文以 PCI 9054 为核心介绍了高速数字图像采集卡中各部分的硬软件设计方案。PCI 9054 以其强大的功能和简单的用户接口,为 PCI 总线接口的开发提供了一种简洁的方法,使用户只需要设计本地总线接口电路,即可实现与 PCI 总线的高速数据传输,使用起来非常方便。本设计经过测试数据能够高速传输且正确,利用 PCI 总线的高速特性实时传输和处理采集数据,有效地解决了数据传输和处理的实时性。

随着数字技术的发展,要求的数据传输速率将会越来越高,FPGA 技术和 PCI 总线将会更多地应用在数据传输的设计中,PCI 9054 有着较高的性价比,应用前景将会更加广阔。

参考文献

- [1] [美] Chris Cant. Windows WDM 设备驱动程序开发指南 [M]. 北京:机械工业出版社,2000.
- [2] Walter Oney Microsoft Press. Programming the Windows Model Driver. 1999.
- [3] Windows 2000 DD K Documents, Microsoft, 1999.
- [4] PCI9054 Data Book. Version 2. 1 (Plx Technology) 2000.

Silva. A Hierarchical Neural Model in Short-term Load Forecasting [J]. Applied Soft. Computing 2004, 4: 405-412.

[3] Srinivasan D. Parallel Neural Network - fuzzy Expert System Strategy for Short-term Load Forecasting System Implementation and Performance Evaluation [J]. IEEE Transactions on Power Systems, 1999, 14(3): 1 00-1 06.

[4] Martin T Hagan, Howard B Demuth, Mark H. Beale. Neural Network Design [M]. 戴葵,译.北京:机械工业出版社,2002.

[5] Mallat S G. A Theory for Multiresolution Signal Decomposition: The Wavelet Representation [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1989, 11(7): 674-693.

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](#)

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 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 定制嵌入式 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](#)

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 \$\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](#)

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](#)