

基于 COM Express 架构的数据记录仪的设计与实现

王 维^{1,2}, 蒋景宏^{1,2}, 刘 焱^{1,2}, 蔡惠智^{1,2}

(1.中国科学院声学研究所, 北京 100190;

2.北京中科海讯数字信号处理技术有限公司, 北京 100107)

摘 要: 为实现声纳数据的海量实时存储, 提出一种基于 COM Express 架构的数据记录仪的设计与实现方法。该记录仪在 COM Express 载板上集成了 ETXexpress 处理器模块、光纤数据采集子卡, 以及由 4 个 500 GB 硬盘组成的 RAID0 存储阵列。应用结果表明, 该记录仪可以达到 80 MB/s 以上的实时存储速率, 很好地胜任了声纳数据的存储任务。

关键词: COM Express; PCI Express 总线; 记录仪; 光纤数据采集

中图分类号: TP216+.2

文献标识码: A

文章编号: 0258-7998(2011)12-0029-04

The design and implementation of a data recorder based on COM Express architecture

Wang Wei^{1,2}, Jiang Jinghong^{1,2}, Liu Yao^{1,2}, Cai Huizhi^{1,2}

(1. Institute of Acoustic, Chinese Academy of Sciences, Beijing 100190, China;

2. Beijing Zhongkehaixun Digital Signal Processing Technology Co. Ltd, Beijing 100107, China)

Abstract: In order to implement the real-time mass data storage in the SONAR system, a method of data recorder's design and implementation which based on COM Express architecture is proposed. The recorder integrated ETXexpress CPU module, fabric data acquisition board and RAID0 storage array that composed by four 500 GB hard-disks on a COM Express carrier board. The results of application indicated that, the real-time storage velocity of the recorder can be above 80 MB/s, and it can be competent for task of SONAR data storage.

Key words: COM Express; PCI Express BUS; recorder; fabric data acquisition

在军事应用领域, 基于通用信号的处理平台, 通常需要实现高速大容量的数据存储。如何解决数据 I/O 和存储带宽瓶颈, 以满足雷达、声纳等系统对高速大容量数据存储的要求, 以及如何基于不同的架构平台, 开发通用、开放的存储模块, 是需要解决的一个问题。

现阶段声纳某型数据记录仪采用 CPCI 架构, 记录仪中的数据采集板卡与信号处理机之间的通信接口采用并行的 LVDS 连接线, 数据采集板卡通过 PCI 总线将接收到的数据提交至主控板进行存储。由于受并行连接线间串扰和 PCI 总线带宽的双重限制, 记录仪的性能、通用性和可扩展性差, 存储速率也不理想, 不能很好地满足未来声纳系统对实时数据存储的要求。因此, 基于新架构和高速串行总线的数据记录仪的研究变得十分迫切。

基于上述问题, 本文提出了一种基于 COM Express

架构的新型数据记录仪的设计与实现方法, 并且对其硬件重要组成部件——光纤采集子卡进行了详细讨论, 最后给出了测试结果。

1 记录仪系统结构与关键技术

1.1 记录仪系统结构

本设计从架构体系和总线结构两个方面进行考虑, 选择了德国控创公司的 ETXexpress-PC 模块作为处理器单元, 在 COM Express 载板上设计了 XMC-PCIE 规格的夹层卡作为光纤数据采集模块, 同时使用 4 个 500 GB 的磁盘组成 RAID0 存储阵列完成新型记录仪的设计。记录仪结构框图如图 1 所示。

控创公司的 ETXexpress-PC 模块使用 Intel 公司的 45 nm 工艺级酷睿 2 双核处理器以及 GS45 和 ICH9M 芯片组, 支持 SATA、USB、PCI Express、千兆以太网等高速

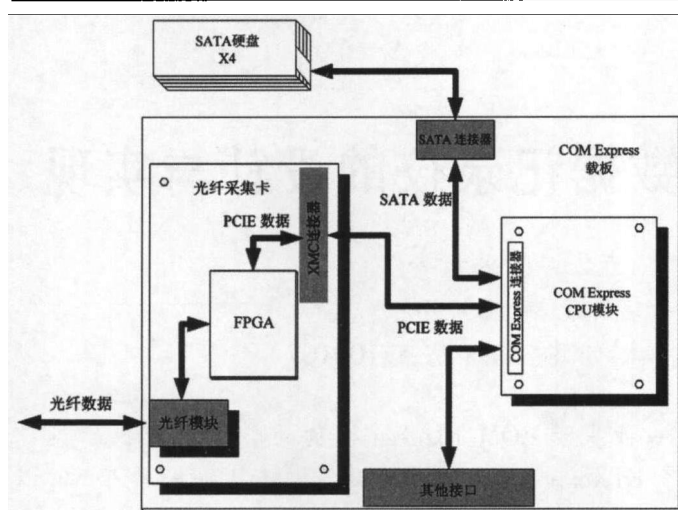


图1 记录仪结构框图

接口，并且模块都严格遵循 COM Express 规范，便于载板的升级，减少了重复设计带来的资源浪费^[1]。

为加速产品的开发，记录仪主要是在控创公司提供的一款 COM Express 评估板的基础上进行设计的。在有限的载板空间上，为了给光纤采集卡提供足够的安装空间，设计保留了原载板的 VGA、PS/2、RS232、USB、LAN 等必须的外围接口，去掉了 Express Card、Mini Express 等记录仪不需要却占用大量空间的接口。载板上没有给硬盘留出空间，是因为实际应用中为了方便硬盘的拆卸和更换，记录仪的机箱设计了专门的硬盘仓。SATA 硬盘是通过定制的加固 SATA 连接线与载板连接，所以只需要在载板的合适位置安装上定制的 SATA 卡座即可。

1.2 关键技术

COM(Computer On Module)Express 是国际工业电气协会(PICMG)定义的计算机模块标准，由几大嵌入式工业计算机厂商共同制定^[2]。COM Express 的处理器模块集成了 CPU 和南北桥芯片，所有的 I/O 功能都通过 AB 和 CD 两个 440 PIN 的双排连接器实现。其中，AB 连接器定义了 PCI Express、SATA、SDVO、千兆以太网以及 USB 等高速接口，CD 连接器定义了 PCI、IDE 等传统的并行总线接口。用户可以根据不同的应用场合选择不同 CPU 性能、功耗、I/O 能力的处理器模块，以及开发不同的功能模块。功能模块一般都设计成标准的接口，通过特定的连接器与载板相连。与传统的 CPCI 架构相比，COM Express 具有接口丰富、使用灵活、开发周期短的优势。

数字光纤通信是一种远程数据传输技术，具有高带宽、低延迟、高可靠性、传输距离长、抗干扰性强、技术成熟等优点^[3]。点到点的光纤通信速率可以达到 2 Gb/s~4 GB/s，在远程数据采集、远程实时监控的领域有着广泛的应用前景。本设计选用光纤来代替并行 LVDS 连接线作为记录仪数据采集接口，很好地克服了高速串扰和通信距离的问题。

传统的 32 bit/33 MHz PCI 总线的理论传输速率为

133 MB/s，远远达不到要求，成为了制约记录仪存储性能的主要因素。PCI Express 是第三代高性能 I/O 总线，在总线结构上由并行总线变为了串行总线，采用点到点的互连，大大提高了系统的传输带宽。并且其总线结构从地址空间、配置机制以及软件上均保持了与 PCI 总线的兼容^[4]。使用 PCIE 总线代替 PCI 总线作为数据采集卡与主控板之间的数据通路，不仅可以大幅提高存储的性能，还可以保持良好的向上兼容性。

磁盘阵列 (RAID) 是指将多台硬盘通过 RAID 控制器 (软件或硬件控制器) 组合成虚拟单台大容量硬盘的使用，通过资源冗余来提供各种服务质量。例如，将多个独立的磁盘组织成一个逻辑盘，提供更大的存储容量；通过数据分割、多通道并行来提高数据的 I/O 速率；通过保存冗余的数据、校验信息来提高存储系统的可靠性等。使用多磁盘的 RAID 技术将大大提高存储容量、存储速率以及可靠性。

2 光纤采集卡的设计

2.1 光纤采集卡概述

光纤采集卡是记录仪的核心部分。板卡是基于 Xilinx 公司的一款 Virtex5 系列 FPGA 芯片 XC5VLX50T 进行开发的，遵循 VITA42.3 XMC PCI Express 规范所定义的尺寸形状和机械结构，使用 Samtec 公司的标准 XMC 连接器 ASP-105885 与 COM Express 载板之间进行互连通信。板卡主要由光纤接口模块、DDR2 高速缓存模块、PCIE 接口模块、电源管理模块、时钟管理模块和复位电路组成，其系统框图如图 2 所示。

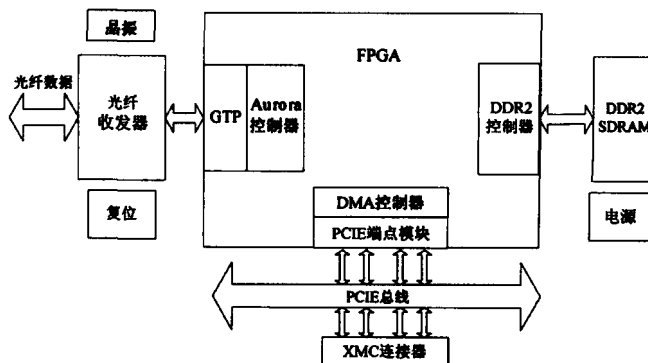


图2 光纤采集卡系统框图

2.2 光纤接口设计

光纤模块采用 FINISAR 公司的 FTLF1321 SFP 高速收发器。光纤采集的工作流程是：光纤收发器接收数据并进行光电转换，然后将串行的数据传输到 FPGA 的 Rocket IO 硬核进行串并转换，最后将并行的数据存储在 FPGA 的内部 FIFO 中。为了增加数据的可靠性、降低误码率，以及匹配收发双发的速率，需要引入流控、数据管理和编解码的机制，因此，设计使用了 Xilinx 公司免费提供的 Aurora 协议对光纤通道进行驱动和管理。Aurora 协议是一种简洁、轻量级、可裁剪的通信协议，使用非常

《电子技术应用》2011年第37卷第12期

灵活,占用系统资源少,并且可以扩展多路光纤通道。

2.3 DDR2 高速缓存设计

由于数据采集系统都是实时系统,所采集的数据持续地送往记录仪,而 Windows 操作系统的非实时性、内核调度以及多线程操作的不可预知性,导致可能出现数据丢失。因此,为了保证记录仪的可靠性并减轻处理器模块的缓存压力,应在光纤采集卡上设计容量大、快速存取能力的缓存装置。板卡选用了 2 片 Micron 公司的 MT47H128M16B 型号的 DDR2 SDRAM 进行高速缓存。该芯片采用 1.8 V 核电压,最高支持 667 MHz 时钟频率。本设计采用 Xilinx 公司的内存管理生成器 IP 核 MIG3.1 来开发 DDR2 控制器,完成 FPGA 对 DDR2 的控制。DDR2 控制器在管理数据高速缓存的同时,还需要管理与 PCIE 总线控制模块的 DMA 传输中断请求。

2.4 PCIE 总线控制逻辑设计

PCIE 总线控制逻辑是光纤采集卡的难点。由于 XC5VLX50T 器件内嵌了 PCIE endpoint block(端点模块),该 IP 核实现了 PCIE 总线的完整功能,因此可以选用 FPGA 的 PCIE endpoint block 硬核和 Rocket IO 硬核来实现 PCIE 协议的物理层、链路层和传输层的功能^[5]。PCIE 总线与处理器的交互可以通过 PIO 或 DMA 两种方式进行。PCIE 硬核本身只包含了 PIO 的功能模块,而 PIO 模式需要 CPU 的参与,因此会影响到 CPU 的运行效率。为满足数据实时高速的存储,自行设计了 DMA 控制器来完成 DMA 功能。PCIE 协议规定,PCIE 的事务可分为四种:存储器事务、IO 事务、配置事务和消息事务。事务的执行或者完成由发送和接收事务包(TLP)具体实现,也即 PCIE 端点与 CPU 之间通过 TLP 进行通信。本设计中主要需要使用的 TLP 包括:存储器读 TLP(MRd)、存储器写请求 TLP(MWr)和带数据的完成 TLP(CplD)。因此 PCIE 总线控制逻辑的重点是 PCIE 事务包的构建与解析、中断的管理以及 DMA 传输的控制。DMA 控制逻辑框图如图 3 所示。

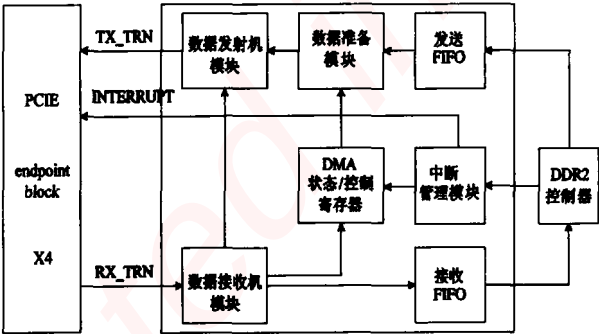


图 3 DMA 控制逻辑框图

图 3 中主要模块功能如下:

(1)DMA 状态/控制寄存器模块主要负责板卡状态的管理、DMA 传输的管理以及中断信号的部分管理。在例化 PCIE endpoint block 硬核时,向系统申请 128 B 的

BAR0 存储器空间来存放这些寄存器,主要的寄存器如表 1 所示。板卡驱动加载后,CPU 会读取 StutasReg 寄存器来察看板卡的状态,如果光纤链路或者 PCIE 链路出现故障,模块负责给出相应的故障指示信号。如果板卡状态正常,CPU 会向 CtrlReg 寄存器写入 DMA 传输的地址和长度信息,向 IntReg 寄存器写入相应中断的使能标志。

表 1 DMA 状态/控制寄存器

寄存器	偏移地址	用途
StutasReg	0x10	保存光纤、PCIE 链路建立情况
CtrlReg	0x20	控制 DMA 传输的启动和停止设置 DMA 传输的地址、长度
IntReg	0x30	设置各种中断的使能和状态

(2)数据发射机模块负责从数据准备模块读取数据,构建 MRd、MWr、CplD 事务包,并且发送到 endpoint block。数据准备模块所组织的数据来自于发送 FIFO 或者 DMA 状态/控制寄存器。

(3)数据接收模块负责从 endpoint block 接收并解析 MRd、MWr、CplD 事务包,将接收到的数据存储到接收 FIFO、DMA 状态/控制寄存器,或者通知数据发射机模块构建相应的 CplD 事务包。

(4)中断管理模块负责中断信号的产生、清除、复位,主要管理了四种类型的中断:DMA 读中断请求、DMA 读完成中断请求、DMA 写中断请求和 DMA 写完成中断请求。其中,DMA 中断读/写请求由 DDR2 控制器给出,用于通知 CPU 光纤采集卡中空余的缓冲空间/缓存的数据量足以达到一次读/写 DMA 传输的长度;DMA 读/写完成中断由中断管理模块给出,用于通知 CPU 此次读/写 DMA 传输已经完成。

一次写 DMA 数据传输的流程如下:(1)DDR2 缓存模块对光纤接收的数据进行缓存,当缓存的数据量达到设定的阈值以后向中断管理模块发出 DMA 写中断请求。中断管理模块将这个中断传递到 endpoint block,经 PCIE 链路去中断 CPU,同时写 IntReg 寄存器记录本次中断的类型。(2)CPU 收到中断后发起一个存储器读事务,读取 IntReg 寄存器的相应位。经过判断是 DMA 写中断请求以后,向 CtrlReg 寄存器发起一个存储器写事务来启动 DMA 传输。同时 CPU 向 IntReg 寄存器发起存储器写事务来关闭 DMA 写中断,打开 DMA 完成写中断。(3)DMA 启动以后,CPU 就不用参与数据的传输了,剩下的传输工作交给 DMA 控制器完成。数据的写 DMA 传输实质上就是 DMA 控制逻辑不断地向板卡驱动的 DMA 缓冲区发存储器写 TLP。每发送一个 TLP,DMA 的传输长度值就会减去上次 TLP 数据的长度,一直减到零为止。(4)当 DMA 传输结束以后,中断管理器会向 CPU 发送 DMA 写完成中断,同时再次写 IntReg 寄存器记录本次中断的类型。(5)CPU 收到并判断这个中断后,会通知应用程序读取收到的数据进行存盘,同时打开 DMA 写中断,关闭 DMA 写完成中断,等待下一次的写 DMA 传输。

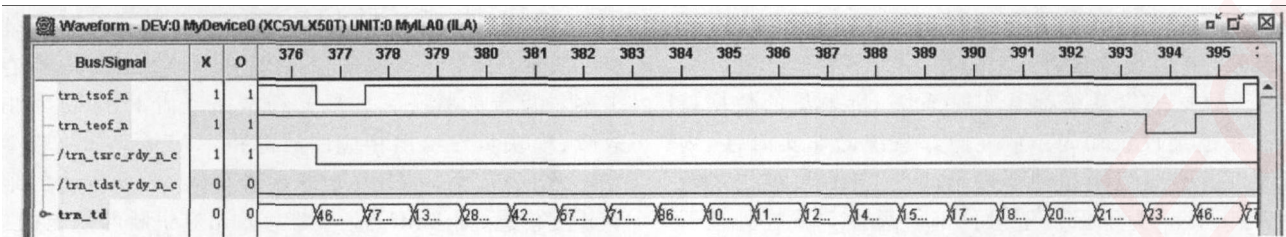


图 4 PCIE 时序图

3 系统调试与性能测试

为验证系统设计的 DMA 传输符合 Xilinx 的 PCI Express endpoint block IP 核所规定的时序以及 SATA 存储的性能,搭建了一个记录仪存储测试平台。该平台由信号处理机通过光纤向记录仪发送以 32 bit 字长递增的数据进行存储。FPGA 的硬件逻辑在 Xilinx 集成开发环境 ISE11.2 下进行,因此可以使用在线逻辑分析仪 ChipScope 捕捉到如图 4 所示的 DMA 传输时序图。

从图 4 可以看出,在 trn_sof_n 和 trn_eof_n 的两个低电平之间就是 PCIE 总线传输的 1 个 TLP, 完全符合 endpoint block IP 核的时序要求,验证了 DMA 的逻辑设计。

在完成了数据从采集卡 DMA 传输到驱动程序缓冲区的操作后,由应用程序完成从驱动中提取数据并且进行写磁盘的操作。最后通过 Matlab 程序对存盘数据的读取校验,验证了记录仪所记录数据的正确性。表 2 给出了在不同单次 DMA 长度和单次存盘长度下的平均存盘速率的影响。

表 2 磁盘存储测试结果 (MB/s)

单次存盘长度	单次 DMA 长度		
	512 KB	1 MB	2 MB
512 KB	50.1	54.5	58.9
1 MB	75.5	78.2	79.7
2 MB	81.6	84.7	86.8
16 MB	84.2	88.5	89.3

从表 2 可以看出,该记录仪的存储速率受单次存盘数据长度的影响较大,受单次 DMA 长度影响相对较小。其原因是:(1)PCIE 的数据传输速率远大于数据的存盘速率;(2)非实时操作系统软件开销较大,对相同大小的数据量进行操作的次数越多,所花费的时间就越长。因此,使用时应该尽量增大单次存盘的数据长度,以提高

记录仪的存储性能。

本文实现了一种基于 COM Express 架构的新型记录仪的设计,该设计主要完成了光纤数据的高速采集、与处理器模块之间的 DMA 传输以及对数据的高速存储。实际应用表明,该记录仪与传统的记录仪相比较,在稳定性和存储速率方面都得到了极大的提升,可以很好地胜任未来声纳数据的采集存储的需要。而且由于 COM Express 是一种开放的架构,提供了许多主流的高速接口,可以灵活地扩展其他的功能模块,给嵌入式设计也提供了一种新的思路。

参考文献

[1] Kontron Corporation.ETXexpress-PC user's guide.<http://www.kontron.com>, 2009.

[2] PEBLY B.COM Express:the next generation computer on module standard.Principal system engineer[M].Radisys Corporation, 2005.

[3] 代孝森,张晋宁,沈辉.基于 PCI Express 总线的光纤检测系统[J].现代雷达,2011(1).

[4] 马萍,唐卫华,李绪志.基于 PCI Express 总线高速采集卡的设计与实现[J].微计算机信息,2008(9).

[5] 刘凯,徐欣.基于 Vertex5 的 PCI-Express 的总线接口设计[J].现代电子技术,2008(5).

(收稿日期:2011-06-10)

作者简介:

王维,男,1986 年生,博士生,主要研究方向:高速嵌入式系统。

蒋景宏,男,1979 年生,博士后,主要研究方向:声纳/雷达系统和信号处理平台设计。

刘垚,男,1985 年生,博士生,主要研究方向:高速嵌入式系统。

(上接第 28 页)

参考文献

[1] 孙戈,郭小玄.基于 Hi3510 的 Linux 嵌入式视频服务器的设计[J].煤炭技术,2009,28(11):146-148.

[2] 周辉,叶桦,仰燕兰.基于 WebGIS 与车载移动视频的智能车辆监控系统[J].东南大学学报,2010,40:192-197.

[3] 李渊,于海勋.基于 Hi3510 的车载监控系统的设计[J].微计算机应用,2008,29(1):67-69.

[4] 深圳市海思半导体有限公司.Hi3511/Hi3512 硬件设计用户指南[z].2009.

[5] 深圳市海思半导体有限公司.Hi3511/Hi3512_linux 开发环境用户指南[z].2008.

[6] 毕厚杰.新一代视频压缩标准—H.264/AVC[M].北京:人民邮电出版社,2005.

(收稿日期:2011-06-01)

作者简介:

吴小飞,男,1983 年生,硕士研究生,主要研究方向:计算机测量与控制。

马永杰,男,1967 年生,博士,教授,主要研究方向:智能控制。

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](#)

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](#)

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](#)

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](#)

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](#)

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](#)

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](#)