

基于COM Express的信号系统 逻辑运算单元设计

■ 刘 贞（北京全路通信信号研究设计院有限公司，北京 100073）

摘要：设计一种适用于铁路信号系统的逻辑运算单元，采用标准的COM Express模块作为主控CPU，利用82583芯片外扩多路以太网通信接口，系统内外通信均使用高速以太网，极大改善了运算单元的运算性能和数据吞吐能力。操作系统选用符合IEC 61508 SIL 3标准的Vxworks Cert平台，系统软件和应用软件采用分层式结构设计，便于各部分软件独立升级或功能持续改进。

关键词：信号系统；逻辑运算单元；以太网

1 概述

随着现代铁路信号技术的发展，分属不同应用的地面控制系统种类愈来愈多样化，如联锁系统、列控中心、无线闭塞中心、临时限速服务器、列车自动监控中心等。而这些系统都需要一个可靠、安全的逻辑控制运算单元。

当前，各厂家在上述系统中使用的逻辑运算单元分别有^[1,2]：

- 1) 通用工控机组合；
- 2) 标准CPCI机笼组合；
- 3) 通用服务器组合；
- 4) 嵌入式载板组合。

而上述组合结构都不同程度的存在如下问题：

- 1) 工控机、服务器硬件主板更新升级太快，系统软件需要保持同步更新，不利于软件版本的稳定；
- 2) 工控机、服务器大多带有风扇，且对环境温度要求苛刻，后期维护工作量较大；
- 3) 数据通信大多采用并行总线，如ISA、PCI、VME等，数据吞吐能力有限；
- 4) 嵌入式板载模块运算能力有限。

2 逻辑运算单元工作原理

2.1 逻辑运算单元工作模型

为了描述问题简单，忽略外部设备后，将逻辑运算单元模型化抽象后如图1所示，信号系统的逻辑运算单元主要完成3个工作，分别为：数据输入、安全运算和结果输出。为了保证数据通信安全，数据输入必须采用安全通信协议，从数据的准确性和时效性两个维度对通信数据进行防范。经过运算后的结果数据经二取二安全比较后由输出通道输出。



刘贞，男，博士毕业于哈尔滨工业大学，工程师。主要研究方向为嵌入式系统设计，曾参与北京地铁8号线、全电子计算机联锁等项目。



图1 逻辑运算单元工作原理

2.2 逻辑运算单元的特点

作为信号系统的控制核心，运算单元有其自身特点，主要表现在：

安全性要求，必须满足故障导向安全的要求，系统结构上必须采用二取二或者三取二的冗余结构，用于消除软硬件单点故障引起的安全风险。

运算性能要求，信号系统的控制逻辑越来越复杂，同时又需要对安全通信协议进行编码和解码，故其 CPU 要达到一定的处理能力。

数据通信吞吐能力要求，地面控制系统管辖的设备越来越多，控制软件大多采用周期执行的方式，尽可能的需要在每个周期和其他设备通信，高速、可靠的通信通道必不可少。

可靠性、环境适应性要求，逻辑运算单元应具有较高的可靠性，可靠性的降低会显著提升安全性。另外，信号系统工作环境复杂，硬件系统应具备一定的抗噪性，能够在高温、低温、潮湿、振动等环境下正常工作。

2.3 通信接口

随着相关安全通信协议的不断完善，以太网标准设备可靠性的提升，铁路信号系统间和内部也越来越多的采用以太网通信。以太网组网方便，远距离传输也有成熟的工业产品，逻辑运算单元显然要具备高速可靠的以太网接口，便于系统功能拓展。

3 逻辑运算单元硬件架构设计

3.1 基于PC104的运算单元

PC104 标准产生较早，其外围配套电路设计简单，模块化扩展功能方便，目前在很多信号系统的运算单元中得到应用。但因其采用并行数据总线 ISA 或 PCI，致使其运算性能和数据通信吞吐能力有限。

3.2 COM Express技术

COM Express 是国际工业电气协会 (PICMG) 定义的计算机模块标准，是一种高度集成的单板计

算机，特别适合于执行自定义的工业计算机解决方案，适合于在标准的单板计算机因结构或由于缺乏扩展性而不适合时使用，在 COM Express 标准里抛弃了之前的低速度 PCI、IDE 信号，是一个以全新的高速计算机接口为主的标准，PCI Express，串行 ATA，千兆网口，与 PC104 标准相比，数据吞吐能力大为提升^[3]。

从架构上来讲，COM Express 可以理解为将一个 Highly Integrated CPU 及其最小系统做成一个 Module，并且这个 Module 可以通过 COM Express 定义的标准接口扩展一个用户定制的具有专门用途的主板 (Customized Carrier Board) 上，从而构成一个产品或者系统^[4]。

3.3 基于COM Express的运算单板

使用标准的 COM Express 模块配合少量外围电路，即可实现运算单板功能。如图 2 所示，采用 COM Express type6 标准模块，自带 6 路 PCI Express x1 接口，利用 Intel 的 82583 芯片外扩多路千兆以太网接口，另外 SATA 接口可直接扩展 Cfast 存储卡，用于装载操作系统文件及应用程序。由于模块 GPIO 数量有限，通过一片 FPGA 扩展后可用于读取系 ID、CPU ID 等配置信息，也可以用于输出 LED 灯指示信息。

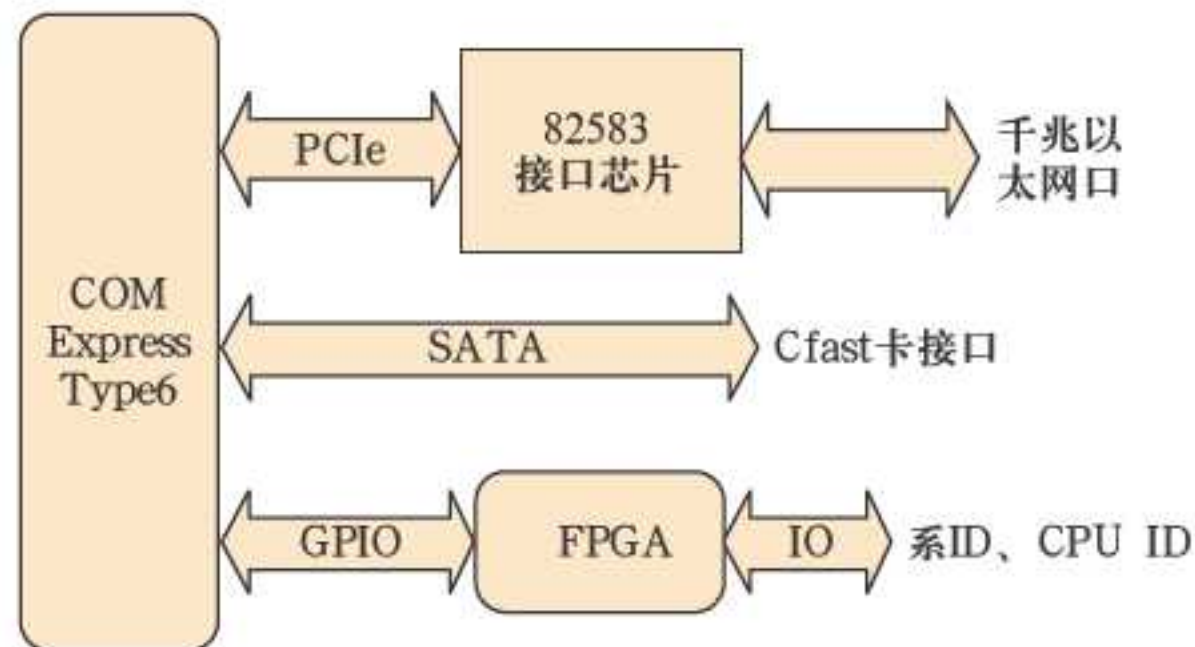


图2 基于COM Express的运算单元

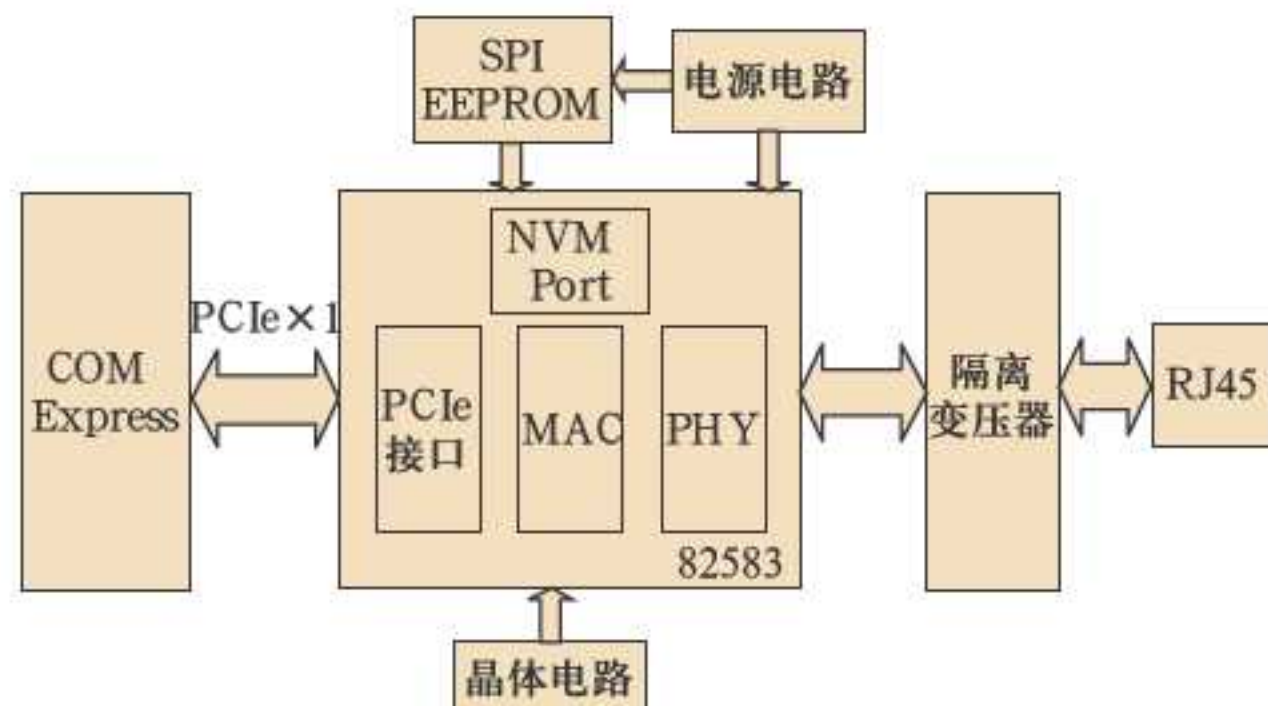


图3 以太网接口扩展电路

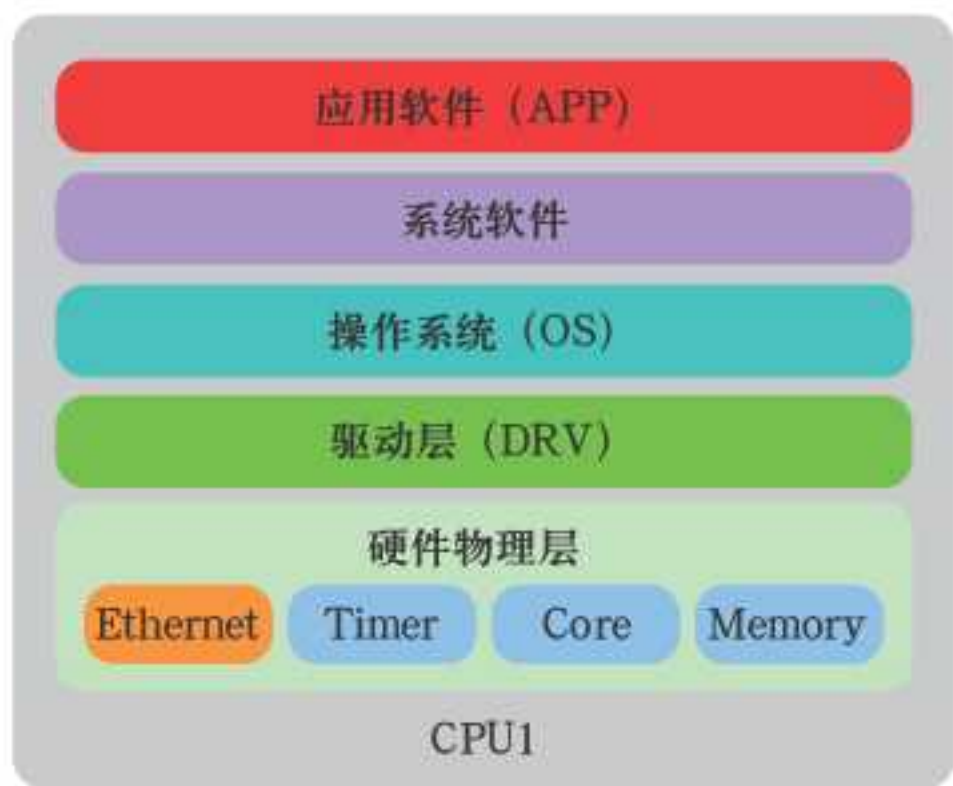


图6 系统工作模型

4.3 二乘二取二软件实现原理

逻辑运算单元 CPU 的工作主要是周期性的对以太网输入的数据进行安全逻辑处理，并将计算后的数据通过以太网对外输出。如图 7 所示，在每个逻辑运算周期的二取二计算主要有以下几个步骤：

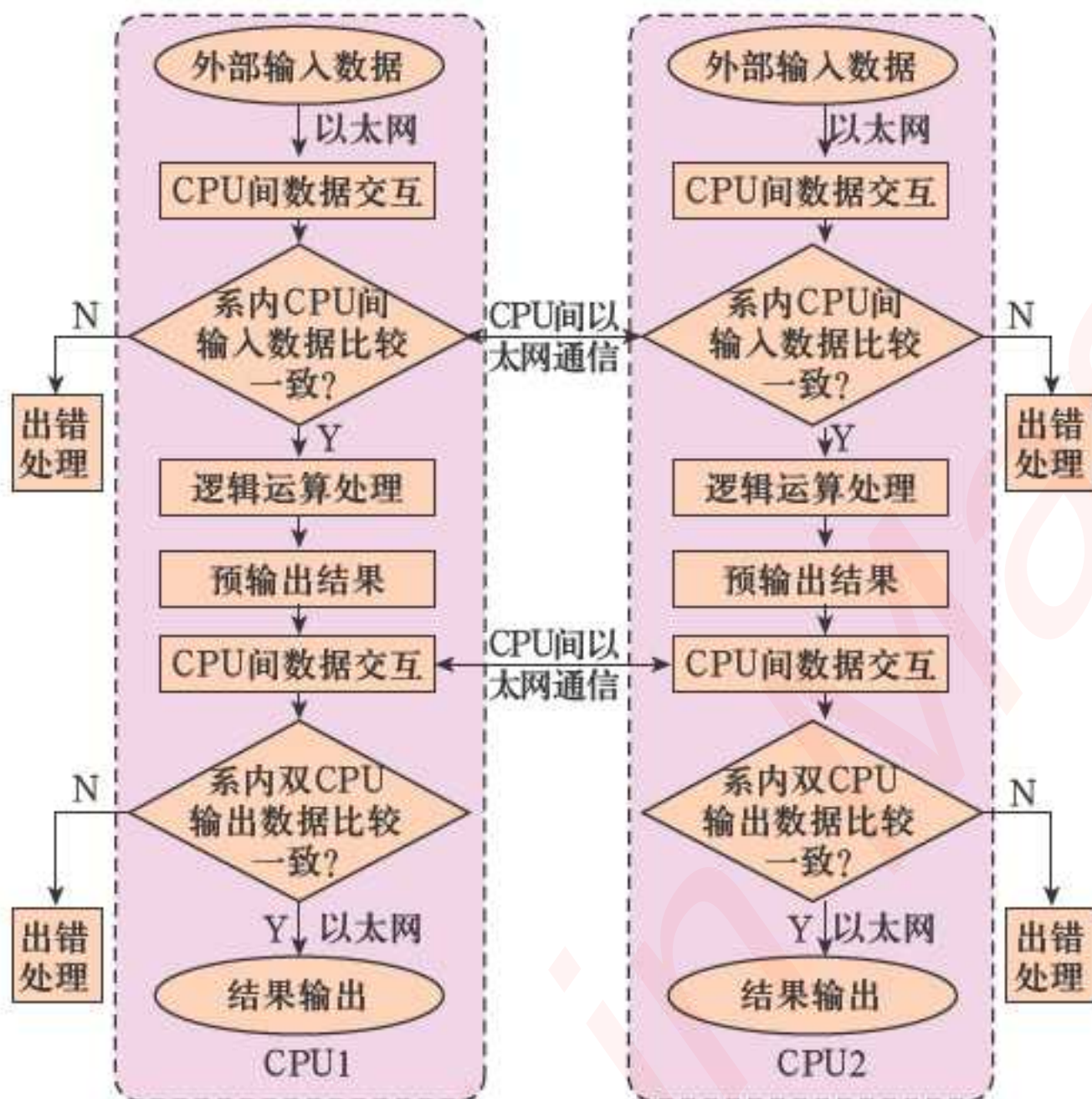


图7 系内双CPU二取二工作原理

- 1) 系内双 CPU 均通过各自对外以太网通道获得外部输入数据；
- 2) 系内双 CPU 彼此交换本周期接收的输入数据；
- 3) 对交换的数据进行比较；
- 4) 将比较一致的数据送入逻辑计算器，产生预输出；
- 5) CPU 间交换预输出结果数据；
- 6) 比较预输出数据；

7) 将比较一致的输出数据通过以太网再次发送。

4.4 系内、系间数据交互机制

从工程应用角度而言，开发人员希望应用软件系内、系间在单周期内可交换比较的数据愈多愈好，目前大多数信号系统运算单元采用固定容量的 DPRAM 进行系内或系间数据交换，除去配置和系统软件的数据交互开销，可供应用软件使用的容量非常有限。本文所述系统，将采用高速千兆以太网接口作为系内双 CPU 交互的通道，可在一定时期内满足应用开发人员不断增长的双 CPU 数据交换比较需求。

如图 8 所示，以系间数据交互为例描述逻辑单元安全冗余处理机制，在一个通信周期内的工作流程如下：

- 1) 主系 CPU1 将“数据包 A”发送至从系 CPU1；
- 2) 主系 CPU2 将“数据包 B”发送至从系 CPU2；
- 3) 从系 CPU1 将接收到的“数据包 A”直接复制发送至从系 CPU2；
- 4) 从系 CPU2 将接收到的“数据包 B”直接复制发送至从系 CPU1；

经过上述 4 个步骤后，从系的两个 CPU 均获得了“数据包 A+B”，由于系内 CPU 间通信采用直连的高速千兆以太网，其数据吞吐能力获得显著提升，通信延时亦随之减小，使其每个 CPU 都具备冗余通信的能力。

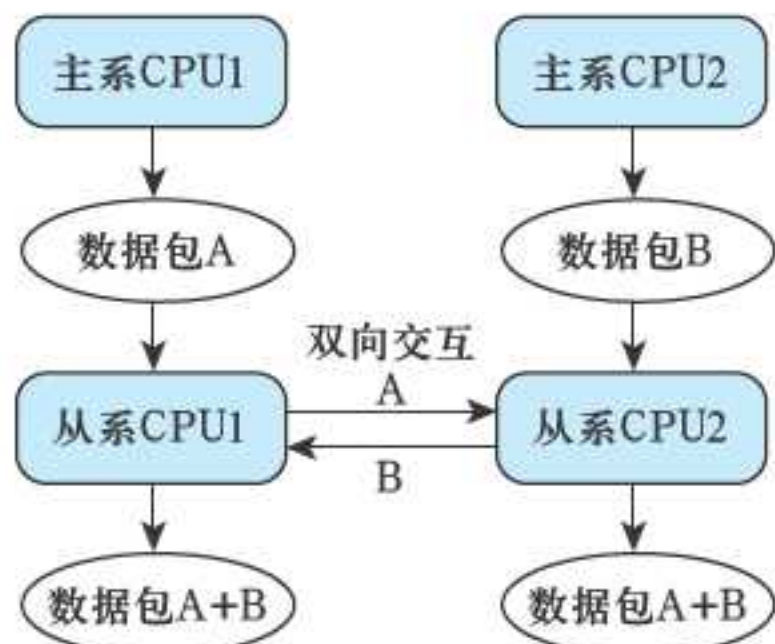


图8 系间数据交互原理

5 结论

本文设计了一种基于 COM Express 模块的逻辑运算单元，利用 PCIe 总线外扩以太网接口，极大的提升了运算单元的数据吞吐能力，有效降低 CPU 在通信方面的资源消耗，提供整体运算性能。

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](#)

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](#)
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](#)

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](#)
8. [基于 COM Express 架构的数据记录仪的设计与实现](#)