

## 总线协议中的 CRC 及其在 SATA 通信技术中的应用

郭红卫, 易茂祥, 李新力

(合肥工业大学 电子与应用物理学院, 安徽 合肥 230009)

**摘要** 分析了基于总线协议下的 CRC 校验关键技术的算法及实现原理。提出了一种适合总线协议生成多项式的 CRC 产生器与校验器的硬件电路实现方法。通过该方法, 依照 SATA 国际 I/O 标准, 编写的 CRC 产生器与校验器 Verilog 代码, 已通过 VCS 的仿真验证, 并成功集成于 SATA 总线, 实现了该总线的通信。该方案进行的 CRC 产生器和校验器设计, 具有可靠性高, 实用性广, 便于提高工程开发效率等优点。

**关键词** CRC 校验; 总线协议; Verilog; SATA

**中图分类号** TN915.04 **文献标识码** A **文章编号** 1007-7820(2010)11-094-03

### CRC in the Bus Protocol and its Application in SATA Communication Technology

Guo Hongwei, Yi Maoxiang, Li Xinli

(School of Electronic Science and Applied Physics, Hefei University of Technology, Hefei 230009, China)

**Abstract** This paper analyzes the algorithm for and implementation principle of CRC key technology. Implementation of the generative polynomial suited to the bus protocol CRC generator and CRC checker with a hardware circuit is proposed. By this method, the CRC generator and CRC checker verilog code we programmed by the SATA international standards are verified by the vcs simulator and integrated to the SATA bus successfully and SATA communication is therefore achieved. The scheme of the CRC generator and CRC checker has the advantages of high reliability and great practicality and that it can improve the rate of project development.

**Keywords** CRC; bus protocol; Verilog; SATA

在总线数据的传输中, 由于噪声以及 EMI 等因素的影响, 使被传输数据出现误码。为控制传输数据错误的产生, 常在信息传输通道中对数据进行编码、加扰等方案, 而在接收端则采用解码、解扰方式, 并对接收数据进行校验, 因 CRC 校验可靠性比较高, 所以在各种总线协议中得到了广泛应用。文中研究了总线协议下硬件电路的 CRC 实现方法, 将此实现方式应用在 SATA 总线通信中, 获得了较好的结果。

### 1 CRC 校验的原理

#### 1.1 总线协议中的 CRC

CRC (Cyclic Redundancy Check), 即循环冗余校验。CRC 校验的基本思想是校验和 (Check Sum) 加在帧尾, 使带校验和的帧的多项式能被生成多项式  $G(x)$  除尽, 收方接收时, 用  $G(x)$  去除它, 若有余

数, 则传输出错。在总线协议中, 生成多项式  $G(x)$  一般由发方和收方共同商定, 生成多项式的高位和低位必须全为 1, 并且生成多项式必须比传输信息对应的多项式短<sup>[1-2]</sup>。

#### 1.2 CRC 校验的计算法

(1) 设  $G(x)$  为  $r$  阶, 在帧的末尾加  $r$  个 0, 使帧为  $m+r$  位, 相应多项式为  $X_r M(x)$ 。

(2) 按模 2 除法用对应于  $G(x)$  的位串去除对应于  $X_r M(x)$  的位串。

(3) 按模 2 加法在对应于  $X_r M(x)$  的位串中加上余数, 结果就是要传送的带校验的和的多项式  $F(x)$ , 则  $F(x) = X_r M(x) + [X_r M(x) \text{ MOD } 2 G(x)]^{[3]}$ 。

#### 1.3 CRC 检验能力

(1) 将发送多项式定为  $F(x)$ 。

(2) 接收多项式定为  $F(x) + E(x)$ 。

(3) 按照 CRC 算法, 求得余数  $(F(x) + E(x)) / G(x) = 0 + \text{余数 } (E(x) / G(x))$ 。

(4) 若余数  $(E(x) / G(x)) = 0$ , 则错误不被发现, 否则, 错误可以发现。

## 2 CRC硬件电路的设计方法

### 2.1 CRC中的线性反馈移位寄存器

CRC校验器可以采用 LFSR (线性反馈移位寄存器)方法产生, 自主的 LFSR 具有二进制抽头 (tap)系数  $C_1, C_2, \dots, C_N$  这些系数决定了  $Y(N)$ 是否被反馈给寄存器的特定级。

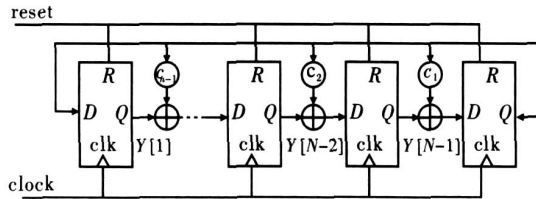


图 1 多级 LFSR

如果  $C_N - J + 1 = 1$ , 那么对于  $j = 2, \dots, N$ , 第  $j$  级的输入是由  $Y[j - 1]$  和  $Y[N]$  的异或形成, 否则第  $j$  级的输入就是  $j - 1$  级的输出, 即  $j - 1 - Y[j] - Y[j - 1]$ 。抽头系数向量决定了 LFSR 特征多项式的系数, 这些系数描述了它的周期性特征。

### 2.2 对于 CRC 中线性反馈移位寄存器的编码方法

依照 CRC 产生方案, 采用 Verilog 进行硬件设计。对于  $CRC\ Width = 8, Data\ Width = 4, Polynomial = X^8 + X^2 + X + 1$ , 根据抽头系数可以做出 CRC 多项式的 LFSR 实现图如图 2 所示。

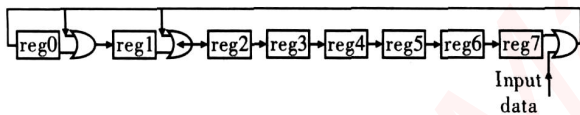


图 2  $X^8 + X^2 + X + 1$  的 lfsr 实现图

CRC 硬件电路的 LFSR 实现本质是串行转化为并行, 因为 Input data 是 4 位的, 当输入了一组 4 位 data 后, 随着现行反馈移位寄存器的数据移位, 原位于 reg3 位置的数据  $crc\_reg[3]$  被移位到 reg[7], 于是将  $crc\_reg[3]$  赋值给  $next\_crc[7]$ , 同理,  $crc\_reg[2]$  赋值给  $next\_reg[6]$ 。next\_crc[5] 是首先进入 LFSR 的数据  $data\_in[0]$  与原 reg7 位置上的数据  $crc\_reg[7]$  进行异或, 其结果传至 reg1 位置的输出端, 与 reg1 的数据  $crc\_reg[1]$  进行异或, 然后右移进入 reg2, 继续右移至 reg5 的位置, 得到  $next\_crc[5] = data\_in[0] \wedge crc\_reg[7] \wedge crc\_reg[1]$ 。

该算法的 CRC Verilog HDL 代码核心部分<sup>[4-6]</sup>

```
assign next_crc[0] = data_in[3] ^ crc_reg[4];
assign next_crc[1] = crc_reg[5] ^ data_in
```

```
[2] ^ d[3] ^ crc_reg[4];
assign next_crc[2] = crc_reg[5] ^ data_in[1] ^ crc_reg[6] ^ data_in[2] ^ data_in[3] ^ crc_reg[4];
assign next_crc[3] = crc_reg[5] ^ data_in[0] ^ data_in[1] ^ crc_reg[6] ^ data_in[2] ^ crc_reg[7];
assign next_crc[4] = data_in[0] ^ data_in[1] ^ crc_reg[6] ^ crc_reg[7] ^ crc_reg[0];
assign next_crc[5] = data_in[0] ^ crc_reg[7] ^ crc_reg[1];
assign next_crc[6] = crc_reg[2];
assign next_crc[7] = crc_reg[3]。
```

依照上述对 CRC 硬件电路实现方式的分析, 可根据不同总线协议的要求改换生成多项式的次数使得相应硬件电路的 Tap 系数发生变化, 并且输入数据的位宽也可以根据需要任意进行选择设计, 使得提出的 Verilog 编程方案可以得到更加广泛的应用, 达到实现更多总线协议下的同类 CRC 硬件电路的目的。

## 3 CRC 硬件电路在 SATA 通信技术中的应用

### 3.1 SATA 总线协议中的 CRC 传输

SATA (Serial ATA) 是硬盘接口之一, 采用串行方式进行数据通信的总线, 克服了很多并行数据传输存在的缺陷, 其目的主要在于提高通信数据的传输速度。SATA 总线主要由 4 个部分组成, 分别是物理层、链路层、传输层和应用层。其中物理层决定了数据的传输形式, 使数据以串行方式输出, 链路层是对数据进行编码解码, 并通过发送提取原语的方式提供握手机制。传输层主要负责传输数据命令。应用层负责执行命令以及发出命令。因 SATA 传输速度比以往的总线传输更加快捷, 并且抗电磁干扰能力要比并行数据的传输方式强, 所以已成为下一代硬盘首选的数据通信传输方式, 现在的 SATA 第三代国际标准已经将其速度定为了  $600\text{ MB} \cdot \text{s}^{-1}$ , 如此快的传输速度, 要降低信道的误码率, 保证数据能够正确的传输, 探错机制要十分可靠, 所以国际 SATA I/O 组织在通信协议中加入了 CRC 校验机制<sup>[7-9]</sup>。

在 SATA 总线协议中, CRC 的计算是针对所有被传输数据的, 即所有的非原语数据全部要进行 CRC 编码, 并将最后编码的结果插入到帧尾, 进行传输。

### 3.2 SATA 通信中的 CRC 握手机制

在 SATA 协议中, 数据传输前都需要应用种子对 CRC 进行初始化, 以 SATA 帧的传输为例, 在原语帧

头传输以前，所有主机和设备 CRC 应初始化为种子值 XX52325032 h。

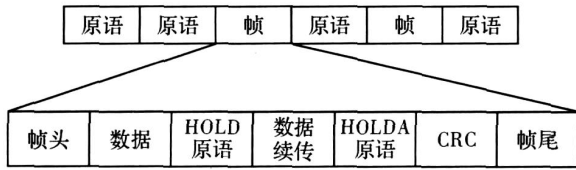


图 3 CRC在总线中的传输位置

在 SATA 总线中，主机与设备之间是相互的，在主机一边传输的数据，要进行 CRC 计算，将计算后的值插入到帧尾，然后传输出去，在设备另一边接收数据，则是应用相同的 CRC 算法对数据进行校验，然后报告校验结果。反之，设备传输数据，主机接收数据也是通过同样的方式进行 CRC 产生和校验。

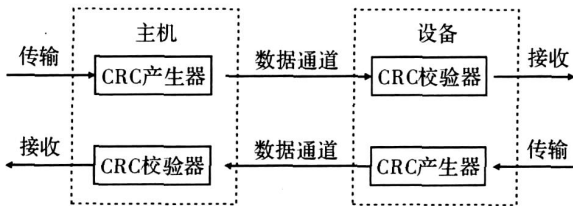


图 4 总线协议中的 CRC 校验的握手机制

### 3.3 SATA 通信协议中的 CRC 产生器

在 SATA 总线 CRC 产生器的设计，主要由 CRC Cal 模块和 CRC Insert 模块以及一个多路选择器组成。通过计算模块 (CRC Cal) 对总线中的有效传输数据进行 CRC 计算，CRC 插入 (CRC Insert) 机制对帧尾进行监控，控制多路选择器。当总线的高层提示传输数据结束 (EoF) 时，将会有 EoF 信号通知 CRC Insert 模块，此时 CRC Cal 模块将停止进行对数据的 CRC 计算，并且将计算好的 CRC 结果插入到传输的数据尾部，一起发送出去。

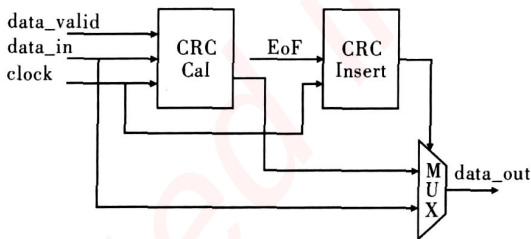


图 5 总线协议中的 CRC 产生器

### 3.4 SATA 通信协议中的 CRC 校验器

在 SATA 总线中，CRC 校验器是 CRC Cal 模块和 CRC err Arbiter 模块组成，由 CRC 计算模块 (CRC Cal) 依次对接收到的数据进行 CRC 计算，并通过 CRC err Arbiter 模块对计算结果实时监控，当接收数

据结束时，发出 CRC 校验报告，总线会根据报告结果来判断数据是否需要重新进行传输。

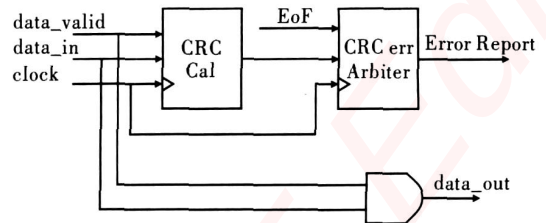


图 6 总线协议中的 CRC 校验器

### 3.5 SATA 通信中 CRC 硬件电路实现方法的应用

在 SATA 总线中，采用 CRC 校验，增强了数据在传送过程中侦错与除错方面的功能，在此总线中，CRC 校验过程并不复杂，只要接收端发现数据传输出现问题，就会自行将这些数据丢弃，然后发送握手原语要求发送端进行数据重发。SATA 国际 I/O 组织将 CRC 生成多项式定为  $G(x) = X^{32} + X^{26} + X^{23} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$ 。按照该要求，确定 Tap 系数，并采用位宽 32 bit 的数据传输方式，基于上述的 Verilog 实现方法，成功编写了 SATA 总线 CRC 校验器的代码，并应用于工程，其仿真结果与 SATA 要求完全符合。

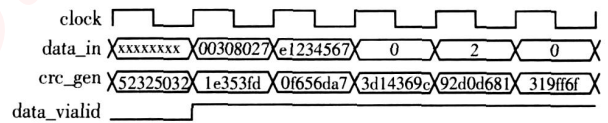


图 7 SATA 总线 CRC 校验波形

## 4 结束语

在总线协议控制中，CRC 起着重要作用，其领域已经涉及到通信等各种总线的数据传输当中，CRC 控错算法功能强大，已成为总线协议的重要环节。本文将 CRC 原理直接与总线协议相结合，通过对硬件电路的分析，提出的硬件电路实现方案可以应用于 SATA 总线通信。其实用性强、可靠性高，有助于解决电路设计中遇到的 CRC 设计问题。

### 参考文献

- [1] Robert J McEliece 信息论与编码理论 [M]. 2 版. 李斗, 殷悦, 罗燕, 译. 北京: 电子工业出版社, 2005.
- [2] 孙丽华. 信息论与纠错编码 [M]. 北京: 电子工业出版社, 2005.
- [3] 卢开澄, 卢华明. 编码理论与通信安全 [M]. 北京: 清华大学出版社, 2006.



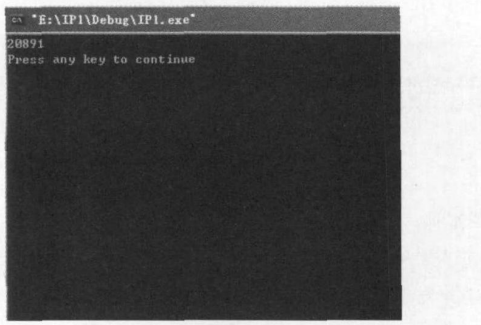


图 6 IP数据报检验结果

同理，当屏幕上运行的结果为 65 535 时，换算成二进制正好为 1111111111111111，这时即为一个合法的 IP 报头。

## 4 结束语

本文利用 IP 报头校验和实现了一种对 IP 协议的

(上接第 93 页)

了光滑海面上标准大气、蒸发波导、表面波导的传播损耗分布图，通过伪彩色图分析大气波导对电磁波的陷获作用，可知蒸发波导和表面波导是形成海上超视距传播的主要因素。这些为雷达在大气波导环境下的应用提供了有力的依据，在军事领域具有较高的应用价值。

## 参考文献

- [1] 焦林, 张永刚. 大气波导条件下雷达电磁盲区的预报研究 [J]. 西安电子科技大学学报: 自然科学版, 2007, 34(6): 989 - 994
- [2] Bean R, Dutton B E J. Radio Meteorology [M]. USA: Dover Publications, 1968

(上接第 96 页)

- [4] Paul Garrett 编码理论中的数学 [M]. 北京: 机械工业出版社, 2005.
- [5] 肖国镇, 卿斯汉. 编码理论 [M]. 北京: 国防工业出版社, 1993.
- [6] SATA - D Board Members Serial ATA Revision 2.6 [C]. SATA International Organization, 2007: 289 - 334.

分析，并且借助 VC++ 实现仿真，不仅验证了算法的可行性，还为以后的实际工作奠定了基础。

## 参考文献

- [1] 郭士秋. IP 协议体系 [M]. 北京: 电子工业出版社, 2002
- [2] 周逊. IPv6: 下一代互联网的核心 [M]. 北京: 电子工业出版社, 2003.
- [3] 蒋亮, 郭健. 下一代网络移动 IPv6 技术 [M]. 北京: 机械工业出版社, 2005.
- [4] Giuseppe Campobello. Parallel CRC realization [J]. IEEE Computer Society, 2003, 52(10): 1312 - 1319.
- [5] 熊家军, 陈钢, 张红军. ATM 承载 IP 的地址解析方法分析 [J]. 计算机工程, 2001, 27(2): 237 - 240.
- [6] Comer D E. Internetworking with TCP/IP [M]. 3 版. 北京: 清华大学出版社, 1998.
- [3] Hall M P M. 对流层传播与无线电通信 [M]. 梁卓英, 张忠治, 译. 北京: 国防工业出版社, 1984.
- [4] Kuttler R, Dockey G. Theoretical Description of Parabolic Approximation/Fourier Split - step Method of Representing Electromagnetic Propagation in the Troposphere [J]. Radio Science, 1991, 26(2): 381 - 393.
- [5] Barrios A E. A Terrain Parabolic Equation Model for Propagation in the Troposphere [J]. IEEE Trans Antennas Propagat, 1994, 42(1): 90 - 98.
- [6] Kuttler J R, Janaswamy R. Improved Fourier Transform Methods for Solving the Parabolic Wave Equation [J]. Radio Sci, 2002(37): 1 - 11.
- [7] Douglas J Smith. HDL Chip Design [M]. USA: Doone Publications, 1997.
- [8] Xilinx. Linear Feedback Shift Register V3.0 [Z]. USA: Xilinx Corporation, 2003.
- [9] Xilinx. Cyclic Redundancy Check [Z]. USA: Xilinx Corporation, 2009.

# 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](#)

## 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](#)

## 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](#)
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](#)

## 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](#)

## 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](#)



RT Embedded <http://www.kontronn.com>

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](#)