

USB3.0 协议分析与框架设计

索晓杰, 翟正军, 姜红梅

(西北工业大学 计算机学院, 西安 710072)

摘要: 随着科技的发展, 最高传输速度为 480Mbps 的 USB2.0 已不能满足数据传输的需求; 于是 5Gbps 的 USB3.0 应运而生; 文中在详细分析了 USB3.0 协议的基础上, 完成了 USB3.0 IP 核的框架设计; 该 IP 核框架符合协议要求, 实现了协议的基本功能; 该 IP 核架构有助于学习和理解 USB3.0 协议, 完成了 USB3.0 IP 核的 VHDL 代码实现阶段之前的协议分析与研究工作, 对后期工作有指导作用。

关键词: USB3.0; USB3.0 协议分析; IP 核框架

Analysis and Frame Design of USB3.0 Protocol

Suo Xiaojie, Zhai Zhengjun, Jiang Hongmei

(School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China)

Abstract: With the development of science and technology, USB2.0 that the highest transmission speed is 480Mbps has not been able to satisfy the demand for data transmission. So the 5Gbps USB3.0 emerges as the times require. In this paper, based on a detailed analysis of the USB3.0 protocol, completed the framework design of the USB3.0 IP core. The IP core framework compatible with the protocol requirements, Realized the basic functions of the protocol. This IP core framework contributes to the study and understanding of USB3.0 protocol, completed the protocol analysis and research work before the VHDL code realization stage of the USB3.0 IP core, has the guiding role in the late work.

Key words: USB3.0; analysis of USB3.0; frame of IP core

0 引言

USB 总线具有速度快、通用性好、扩展性强、功耗低、稳定、易开发等众多优点, 在工业及军事领域获得了广泛的应用。USB 从 1.0 (12Mbps) 发展到 2.0 (480Mbps), 推动了 USB 的普及^[1]。目前 USB 接口的产品和设备, 从键盘、鼠标、摄像头、打印机、U 盘到手机、扫描仪等等, 可以说无所不在。

随着科技的发展, 用户对 USB 的数据传输速度有了进一步的要求。2008 年 11 月 USB3.0 (5Gbps) 规范应运而生, 这将进一步推动 USB 的广泛使用。由于 USB3.0 产品尚未打开市场。因此到目前为止, 市场上 USB3.0 控制器还很少, USB3.0 协议的研究也处于起步阶段, 关于 USB3.0 IP 核开发方面的文章寥寥无几。鉴于这样的情况, 本文在充分分析 USB3.0 协议的基础上, 设计了 USB3.0 IP 核模块框架, 并介绍了各子模块的功能。

1 USB3.0 协议分析

USB3.0 是一种超高速数据传输协议, 协议分 11 章, 从物理接口、数据传输、包类型、包结构、链接命令结构、控制状态机、差错控制、电源管理等方面做了详细描述^[1]。USB3.0 包括超高速 (SuperSpeed) 和高速 (HighSpeed) 两部分, 其中高速部分就是我们熟知的 USB2.0, 最高传输速度

480Mbps, 因此 USB3.0 向下兼容 USB2.0 及更低版本; 超高速部分传输速度可达 5Gbps, 我们平时说的 USB3.0 指的就是该部分, 协议中重点讲述了该部分的实现规范。USB3.0 协议结构图如图 1 所示。

USB3.0 协议结构图中实现了 USB3.0 超高速和 USB2.0 高速两个模块。超高速协议模块包括物理层、链路层、协议层三部分; 高速协议模块包括 ULPI 接口和协议层两部分。两个协议模块共用端点缓冲区。在使用过程中, 由 Function (控制) 单元根据设备的使用环境决定哪个协议模块工作。

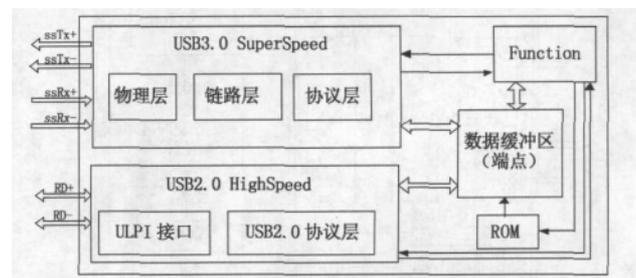


图 1 USB3.0 协议结构

由于超高速模块和高速模块在物理层、链路层、协议层的实现都有很大区别, 所以要实现完整的 USB3.0 协议, 就必须实现超高速和高速两部分。表 1 给出了高速 (USB2.0) 与超高速的区别。

超高速协议中的物理层、链路层、协议层是实现超高速模块的关键。

超高速的物理层定义了超高速总线的信号传输技术, 采用 8b/10b 编码。物理层的作用分为两部分, 一个是接收数据, 一个是发送数据。

收稿日期: 2012-02-27; 修回日期: 2012-04-28。

基金项目: 西北工业大学研究生创业种子基金资助 (Z2011141)。

作者简介: 索晓杰 (1986-), 男, 陕西人, 工学硕士, 主要从事测控与嵌入式技术方向的研究。

超高速的链路层的主要任务是维持链路连通性，保证数据的正确传输。链路层定义了包结构、链路命令结构、链路训练状态机 (LTSSM)，介绍了链路层控制和电源管理功能和链路差错恢复策略。

超高速的协议层是在链路层确保数据包正确传输的基础上，管理设备和主机之间的端到端数据流。协议层定义了包类型、包格式、设备与主机发送数据的应答规范、USB 数据传输的四种方式。

表 1 高速与超高速协议的区别

	超高速	高速
数据速率	5.0 Gb/s	480 Mb/s
接口线数目	9 根 (包含 2.0 的两根差分数据线)	4 根
信号特性	8b/10b 解码, AC 耦合, SSC (扩频时钟)	NRZI 解码, DC 耦合, 无 SSC
总线电源	150mV 的 un-configured 电源和 900mA 的 configured power	100mA 的 un-configured 和休眠状态的器件, 和 500mV 的 configured 器件
插拔/交换	异步事件处理	设备轮询
电源管理/链路控制	带有空闲、待机和休眠的优化的电源管理	有延时的进入和退出的 Port 级的休眠
电缆/接口	两对差分线, 全双工屏蔽式的双绞线	一对差分线, 半双工的非屏蔽的双绞线

2 SuperSpeed 协议 IP 核设计

2.1 SuperSpeed 协议 IP 核整体设计

由上节分析可知 USB3.0 协议非常复杂，包含超高速 (SuperSpeed) 和高速 (HighSpeed) 两个部分，由于 High-Speed 的实现技术已经十分成熟，所以 USB3.0 协议中重点介绍了 SuperSpeed 部分。本文主要完成 SuperSpeed 模块的 IP 核结构设计。图 2 是 SuperSpeed 协议 IP 核结构图。

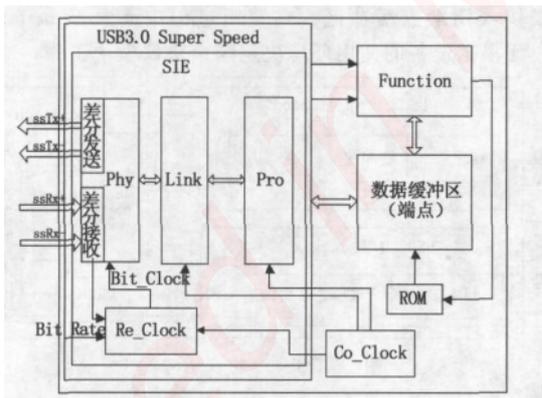


图 2 SuperSpeed IP 核结构

超高速协议模块图中，SIE 部分包含了物理层 Phy、链路层 Link、协议层 Pro。这三个模块主要完成包头、数据载荷、链接命令的接收和发送。控制模块 Function 根据 SIE 模块的工作状态及端点缓冲区的使用情况，控制 SIE 模块、缓冲区正确有序的工作。

2.2 物理层模块设计

物理层数据接收和发送流程如图 3 所示。

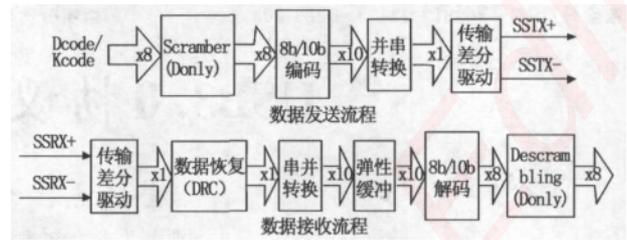


图 3 数据发送/接收流程

接收过程为：通过差分传输驱动和数据恢复 (DRC) 获取串行数据，然后经串并转换生成的 10 位数据通过 8b/10b 解码器转换成 8 位数据，再通过扰码解除模块对数据解扰，最后送给链路层处理。其中弹性缓冲区是为了解决串并转换和 8b/10b 解码之间的时钟匹配问题。

发送过程则显得比较简单，主要从链路层接收数据并通过扰码处理、8b/10b 编码、串并转换后经传输差分驱动发送到物理线路上。

8b/10b 编码是目前高速串行通信中经常用到的一种编码方式。直观的理解就是把 8bit 数据编码成 10bit 来传输。采用这种编码方法的根本目的是“直流平衡 (DC Balance)”^[2]。当高速串行流的逻辑 1 或逻辑 0 有多个位没有产生变化时，信号的转换就会因为电压位阶的关系而造成信号错误，直流平衡的最大好处便是可以克服以上问题。

串并转换模块转换成并行数据应为 10 位，方便与 8b/10b 解码/编码模块接口设计。

在接收流程中添加了弹性缓冲模块，串并转换传给弹性缓冲的数据与控制都工作在接收时钟域，而弹性缓冲传给 8b/10b 解码的数据和控制都必须工作在系统时钟域。因此弹性缓冲要把数据与控制同步到系统时钟域。弹性缓冲的作用是匹配系统时钟和接收时钟。弹性缓冲区主要是通过添加或删除 SKP ordered set 来实现时钟匹配的。

Scrambler 和 Descrambling 模块是成对出现的，可以通过使用信号选择其是否工作。

USB3.0 超高速协议采用全双工，在发送和接收流程中用到的编码、解码、串并转换、串并转换、扰码、解扰等子模块，使用 VerilogHDL 实现时要充分考虑并行性。

2.3 链路层模块设计

超高速 USB 定义了四种包类型：LMP (Link Management Packet)、TP (Transaction Packet)、DP (Data Packet)、ITP (Isochronous Timestamp Packet)。其中 LMP、TP、ITP 只有包头 HP (Header Packet)，DP 由 DPH (Data Packet Header) 和 DPP (Data Packet Payload) 两部分组成。链路层和协议层都需要处理这四类包。包头结构如图 4 所示。

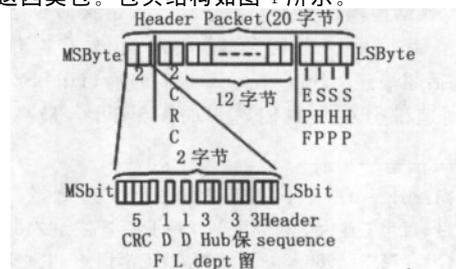


图 4 包头结构

超高速 USB 还定义了链接命令用来实现链路管理和确保数据正确传输。通信双方通过发送链接命令通知对方自己的工作状态、缓冲区可用空间大小、数据接收是否成功等信息。链接命令结构如图 5 所示。

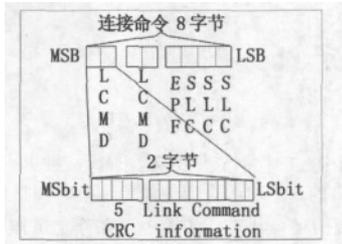


图 5 链接命令结构

链路层分为接收和发送两个子模块。接收模块的任务是接收四种 HP、DPP 和链接命令。接收 HP 时通过计算 CRC-5 和 CRC-16 以及包头有效长度判断是否接收到正确包头。控制模块根据接收到的包头信息来设置端点缓冲区、RX 包头序列号、返回相应的链接命令应答。接收到 DPP 后核对 CRC32，如果正确，送到端点缓冲区，如果错误，进入错误处理状态。链接命令接收成功后根据命令要求，控制模块完成相应处理。

链路层接收到包头和数据载荷时，通过核对 CRC 校验码和缓冲区可用大小以及包头序列号，来判断是否接收到正确的包。无论接收是否正确，控制模块都要作相应的处理，更新 RX 包头序列号、RX 数据缓冲区可用大小等信息，复位或使能其它定时器。接收模块流程如图 6 所示。

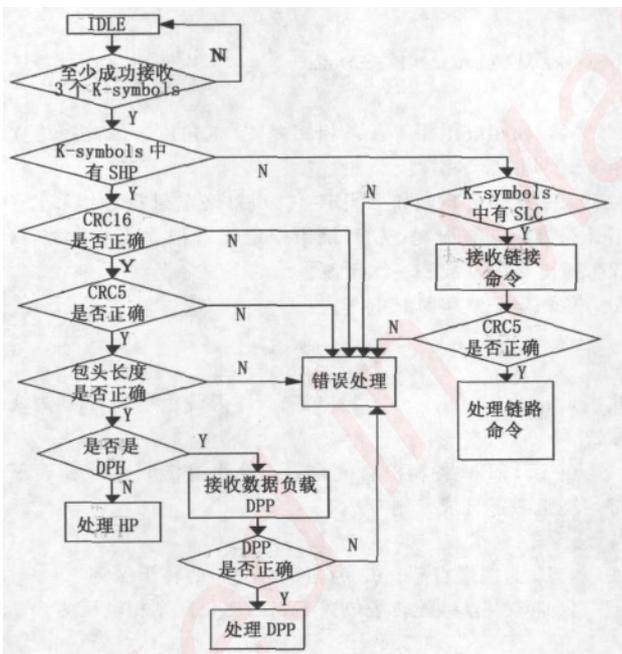


图 6 链路层接收模块

发送模块的任务是组包发送，LMP、TP、ITP 3 个包头只需要添加 4 个 K-symbols，DP 不但要处理包头，还要在数据载荷部分的开头和结尾各添加 4 个 K-symbols。如果协议层来的数据载荷部分大于 1024 字节，需要分开传输。

链路层还定义了链路训练状态机，主要是控制链路状态的转换。状态转换图、状态间的转换条件协议中描述的比较清楚，参照协议用 VerilogHDL 实现。

2.4 协议层模块设计

链路层定义的包头除去前四个 K-Symbols，剩余的 16 字节便在协议层定义了。协议层根据四类包的不同用途，详细定义了 16 字节中各字段的含义。其中 Revision、Type、CRC-16、Link Control Word 四个字段在四个包中是相同的。

Revision 字段占用 2 个二进制位，标记了使用的是 USB 的哪个版本。00b: USB3.0, 01b-11b: 保留。

Type 字段占用 3 个二进制位，标记了属于哪类包。000b: LMP, 001b: TP, 010b: DPP, 011b: ITP, 100b-111b: 保留。

CRC-16 字段占用 2 个字节，用来校验包头前 12 个字节。

Link Control Word 字段占用 2 个字节，其中包含 3 位包头序号，3 位预留位，Hub Depth 3 位，1 位 Delayed，1 位 Deferred，5 位 CRC-5。

除了上面说的四个字段外，每个包在剩余位定义了不同的字段，其作用与意义各不相同。协议层分为组包和解包两个模块。在解包过程中，协议层根据 Type 字段确定收到的是哪类包，然后按照协议依次解析其余各字段，并将包头解析结果送给控制模块，或将 DPP (Data Packet Payload) 送到对应端点缓冲区。如果解析中出错，则将错误汇报给控制模块，控制模块作相应处理。包解析状态转换图如图 7 所示。

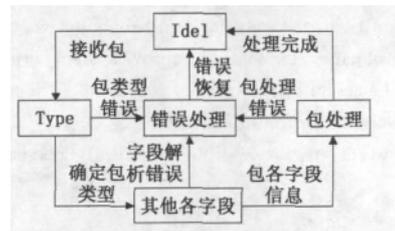


图 7 包解析状态转换图

组包过程则比较简单，根据控制模块的信息，生成相应的包，然后发给链路层。

协议层还详细描述了控制 (control) 传输、批量 (bulk) 传输、中断 (interrupt) 传输和同步 (isochronous) 传输四种传输方式。

2.5 功能模块设计

功能模块除了要执行 SIE 模块中四类包传输的控制及链路控制，还要在设备枚举时完成主机发来的请求。功能模块框图如图 8 所示。其中协议引擎部分完成四类包的传输控制及链路控制，枚举控制部分完成设备枚举的请求处理。

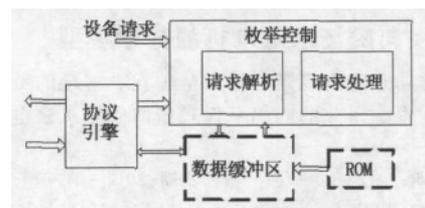


图 8 功能模块框图

枚举控制模块中的请求解析部分主要完成设备请求的解析工作。该模块接收到主机发来的设备请求后，按照命令中各字段的意义，解析出本次接收的是什么请求命令，然后将结果提

(下转第 2253 页)

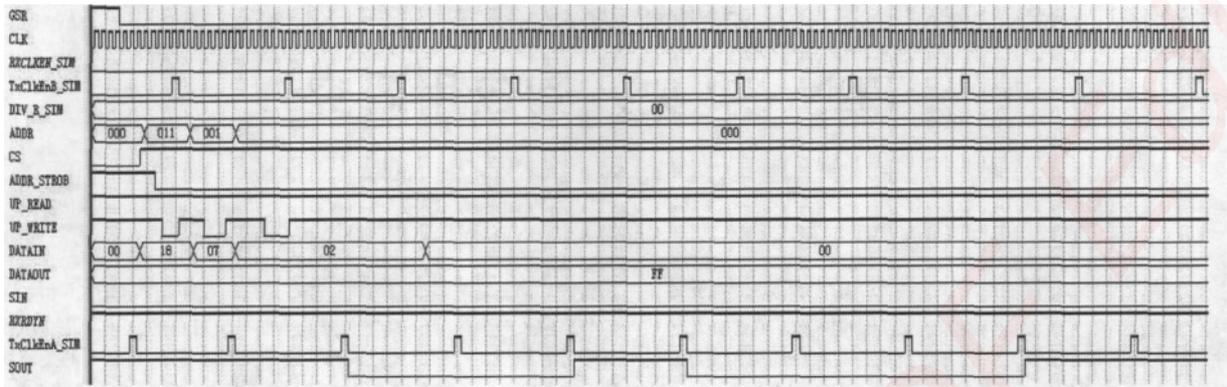


图 5 发送数据时 UART 功能仿真波形

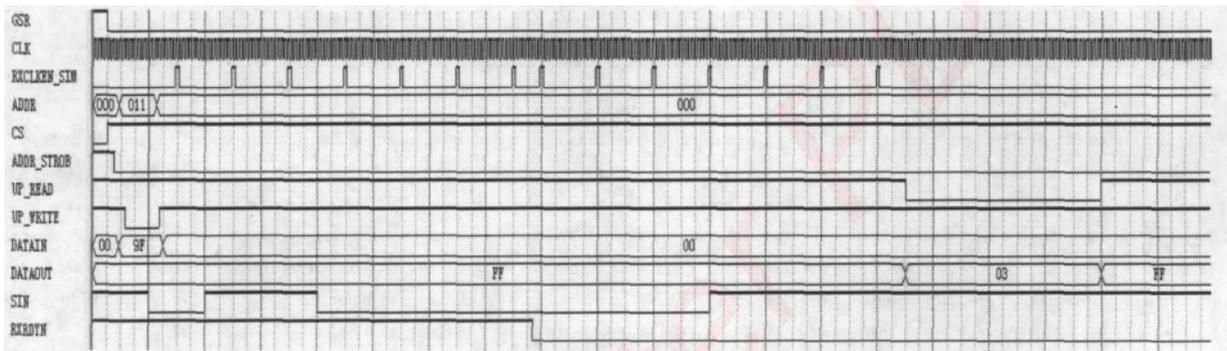


图 6 接收数据时 UART 功能仿真波形

至 CPU。

接收数据时 UART 仿真波形如图 6 所示。图 6 中将 9FH 写到 LCR，设置数据位 8 位，停止位 2 位，偶校验。有效起始位之后，从 SIN 输入 1100000011B，而从并行口送出的数据是 03H，与输入数据比较是一致的，显示接收数据功能仿真正确。

4 结束语

本论文基于工程应用中的实际需求和现代 FPGA 技术上的优势，为解决传统 UART 芯片中的不足之处所设计。设计的 UART 由于配置有 256 字节的先进先出堆栈，能够有效减小其对 CPU 资源的占用，提高 IP 核的性能。设计的 FIFO 大小可编程设定，便于设定容量，这样就突破了使用专用接口芯

片必须等待其产品的逐步升级的限制，有利于设计开发人员提升其设计系统的整体性能。所设计的 UART 可编程设置发送数据字间隔，使得性能不同的设备间可以顺畅地进行数据交换，并通过对字间隔的检测，可有效解决在接收数据时有效数据帧的检测问题。经过验证，显示本设计符合设计要求。

参考文献：

- [1] 高志, 黄生叶. 基于 FPGA 的通用高速串行互联协议设计 [J]. 计算机测量与控制, 2009, 17 (9): 1826-1827.
- [2] 王建平, 张宝剑, 王军涛. 通信原理 [M]. 北京: 人民邮电出版社, 2007.
- [3] 张凯, 林伟. VHDL 实例剖析 [M]. 北京: 国防工业出版社, 2004.

(上接第 2235 页)

交给请求处理模块。请求处理模块根据解析模块的结果，调用不同请求命令的解析流程，完成设备枚举。枚举过程与 USB2.0 基本一致，本文不再赘述。

2.6 数据缓冲区(端点)模块设计

数据缓冲区的作用是，用来暂存各个端点要发送或者接收到的数据。其中 0 端点属于控制端点，其余端点则可配置为数据传输端点及反馈端点。每个端点的输入 (RX) 输出 (TX) FIFO 都有字节的读写信号、数据位信号、空满信号等。

ROM 模块用来存放设备描述符、配置描述符、报表描述符等信息。

3 结论

本文主要分析 USB3.0 SuperSpeed 协议，并在此基础上设计了 USB3.0 IP 核的结构。设计的 IP 核符合协议的要求。设计的 IP 核没有用到 MCU，控制、枚举等任务由功能模块完成，这样就无需设计 SIE 模块与 MCU 之间的接口。控制信号

不通过 MCU 直接到功能模块，因此可以加快信号的传输速度，提高 IP 核的效率。本文设计的 IP 核结构图，对后期用硬件描述语言实现具有指导性的意义。对其他 USB3.0 的研究者有极高的参考价值。

参考文献：

- [1] HP, Intel, Microsoft, etc. Universal Serial Bus 3.0 Specification [S]. 2008, July 30.
- [2] 洪波, 金宁, 殷海兵. 一种新的 8B/10B 编码器的设计方法 [J]. 器件与应用, 2009, 32 (S2): 102-104.
- [3] 凌怀奇. USB2.0 设备控制器 IP 软核设计 [D]. 2008, 5.
- [4] 付华杰, 刘丽君, 等. 高速 USB IP 核的设计与开发 [J]. 微电子学与计算机, 2008, 25 (7): 127-129.
- [5] 许培培, 贾铂奇, 等. 一种通用并行 CRC 计算原理及其实现 [J]. 微计算机信息. 2010, 9-3 (26): 110-111.
- [6] 胡伟, 张新家, 李姜峰. 基于 FPGA 的 USB2.0 协议分析仪设计 [J]. 计算机测量与控制, 2008, 16 (9): 1353-1355.

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](#)

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](#)

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](#)

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\[可实现读写的源码\]](#)

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](#)

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](#)

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](#)