

基于 MPC8260 处理器的 PPMC 系统

南京大学电子系近代电子学实验室 (210093) 方承志 谈宜育 李 元
中兴南京研究所网络硬件开发部 (210012) 王爱东

摘 要：介绍了 MPC8260 处理器 (PowerPC 系列) 的基本特点, 以及一种基于 MPC8260、具备高度可扩展性的 PPMC 硬件实现和软件标准初始化流程。

关键词：嵌入式系统 PPMC PowerPC MPC8260 PCI

伴随网络时代的来临, 带宽急剧增长, 而新业务要求嵌入式系统具备很强的处理能力, 配备高带宽的外围总线。PCI (Peripheral Component Interconnect, 外围部件互连) 总线以其优越的性能, 成为高端嵌入式系统首选总线。以往嵌入式系统式样众多, 硬件模块可重用度低、可互替性差, 硬件平台的繁杂使得加载 RTOS (实时操作系统) 没有规则性可言, 极大地延长了研发周期。

鉴于以往嵌入式系统的缺点, 结合高端嵌入式系统的新要求, VSO (VITA Standard Organization) 在 IEEE 制定的 P1386.1 标准上, 提出了 PPMC (Processor PCI Mezzanine Card) 概念, 目的是确保各个厂家不同的 CPU 对外拥有相同的 PCI 接口。它对 PPMC 插座接口的电气特性、引脚信号、PPMC 和其 Carrier Board 的连接以及 PPMC 的尺寸作了严格规定, 力求在高端嵌入式系统中促进系统模块化。

本文介绍了一种以 MPC8260 处理器为基础的 PPMC 实现, 构建了一个微型计算机系统。该系统完全符合 VSO 对 PPMC 的要求, 在嵌入式 PCI 系统中可以即插即用, 并可应用于 CompactPCI 系统中。该 PPMC 系统处理能力优异 (内核最高处理能力可达 350MIPS), 带有多种通讯协议处理接口, 符合多种 RTOS (pSOS、VxWorks 等) 对其运行平台的要求, 且在硬件上满足 RTOS 的两种主要调试手段 (串口调试和以太网调试), 应用开发相当容易。此外, 该系统具备高度的现场可扩展性, 通过简单的软件配置, 可以方便地提供多种通讯协议的前端处理接口。

1 MPC8260 处理器简介

PowerPC 芯片是由 Motorola、Apple、IBM 三家公司联合研制的一种 RISC 处理器, 具备超强处理能力, 广泛出现在处理器的高端应用场合。Motorola 公司考虑到电信和网络市场的特点, 在 PowerPC 芯片的基础上, 推出了专为嵌入式系统设计的 MPC8260 芯片。该芯片内嵌一个 PowerPC 603e 内核, 有着与 PowerPC 完全相同的指令集合, 并且集成了若干通讯

协议处理单元, 是目前最先进的专为电信和网络市场设计的微处理器。MPC8260 芯片的功能框图以及外部引脚示意如图 1 所示。

由图 1 可知, MPC8260 主要包括三部分。第一部分是高性能的 PowerPC 603e 内核, 体系结构与其它 PowerPC 芯片相同, 运行频率为 100~266MHz。

第二部分是总线接口。在兼容 MPC860 局部总线的基础上, MPC8260 另外提供了高性能的 PowerPC 60x 总线。该总线是一种标准总线, 32 位地址线, 64 位数据线, 支持多 PowerPC 芯片互连。可以通过简单的寄存器配置来决定采用哪种总线。本系统选用 60x 总线, 总线工作频率为 66MHz。

第三部分是通信处理模块 CPM (Communication Processor Module)。该模块是 MPC8260 相对其它处理器的最大优势所在, 除了带有一个增强的 32 位 RISC 专用通信控制器外, 模块中还内嵌有 3 个 FCC (Fast Communication Controller)、2 个 MCC (Multiple Channel Controller)、4 个 SCC (Serial Channel Controller)、2 个 SMC (Serial Managements Controller)。所有这些控制器都可以用来处理特定的通讯协议。

MPC8260 芯片中包括四组通用 I/O 端口: PA、PB、PC、PD。这四组通用 I/O 端口与 CPM 有着密切关系。PA、PB、PC 包含 32 个管脚, PD 包含 28 个管脚。各组端

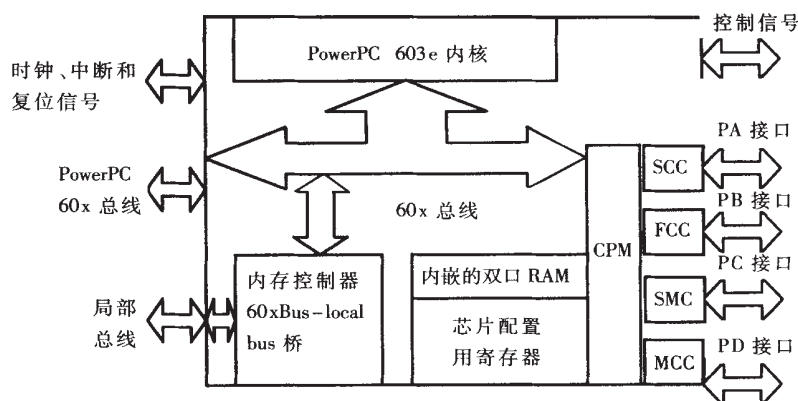


图 1 MPC8260 框图

口中的每一个管脚都可以通过软件配置外接不同的信号。例如,管脚 PC[2]可以配置为外接以太网 MII (Media-Independent Interface)接口的 $\overline{CD}$ 信号,也可配置为外接 ATM (Asynchronous transfer mode 异步传输模式)UTOPIA 接口的 TXD 信号,还可配置为一般的 I/O 信号。

PA、PB、PC、PD 四组通用 I/O 端口的灵活配置,结合内部 SCC、FCC、SMC 的可配置性,使得 CPM 的功能得以倍增。CPM 模块利用这四组通用 I/O 端口完成的通讯协议接口主要如下:

(1) 2 个 ATM UTOPIA 接口,可用来实现 2 个 155 Mbps ATM SAR。

(2) 8 个 TDM (时分复用线)接口,其中 2 个 TDM 接口可以无缝连接到 T3/E3,最高可以提供 256 个全双工 64kbps HDLC 通道。

(3) 3 个 MII 接口,可用来实现 3 个 10/100Mbps 自适应以太网接口。

(4) 4 个 RS-232 串口。

由此可知,MPC8260 内嵌了多种通讯处理模块,可以方便地为用户提供一个全新的系统解决方案来建立高端嵌入式系统。

2 基于 MPC8260 的 PPMC 系统

图 2 为基于 MPC8260 的 PPMC 系统模块框图。

注:篇幅所限,VSO 在 IEEE 标准上专门为 PPMC 定义的标准插座没有画出来,而图 2 中,除了连接到两个 RJ45 插座的信号外,其它黑色粗体箭头代表的信号都必须连到该插座上去。请参阅 VSO 相关手册。

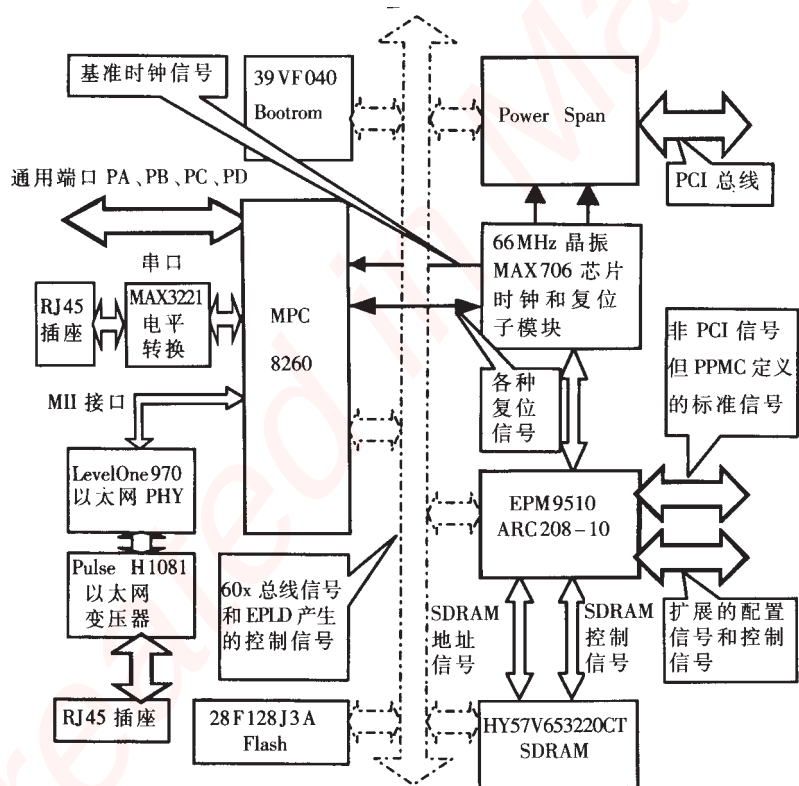


图 2 PPMC 框图

各子模块的解释如下:

(1) 10/100Mbps 以太网子模块接口和 RS-232 串口子模块。两者的实现都利用了 MPC8260 的通用 I/O 管脚。以太网子模块还包括外接的以太网 PHY 和以太网变压器。PHY 选用 LevelOne 公司的 970 系列芯片。变压器选用 Pulse H1081。以太网 MII 接口管脚分布如表 1 所示。

表 1 以太网 MII 接口管脚分布

MII 信号	MPC 8260 管脚	含义
TxD[0:3]	Port B B[4:7]	发送数据线
RxD[0:3]	Port B B[8:11]	接收数据线
CRS	Port B B[12]	开始发送
COL	Port B B[13]	发送冲突
TX_EN	Port B B[14]	可以发送
TX_ER	Port B B[15]	发送失败
RX_ER	Port B B[16]	接收到坏帧
RX_DV	Port B B[17]	接收到数据
TxCLK	Port C C[16]	发送时钟
RxCLK	Port C C[19]	接收时钟

RS-232 串口子模块包含一个电平转换器 MAX3221。此外,相对一般的 9 线 RS-232 接口而言,本系统中无论是数据接收还是发送,MPC8260 总是处于准备好的状态。因而 RS-232 使用 5 线接口就可以,串口管脚分布如表 2 所示。

表 2 串口管脚分布

信号	管脚	含义
TxD	Port D D[26]	发送数据线
RxD	Port D D[27]	接收数据线
$\overline{CD}$	Port D D[28]	检测到载波
RTS	Port C C[12]	请求发送
CTS	Port C C[13]	可以发送

(2) 时钟和复位子模块为系统提供时钟信号和复位信号。时钟子模块核心部件为高稳定性 66MHz 石英晶振。该晶振提供系统基准时钟。考虑到晶振直接带负载能力有限,时钟子模块中还包含 1 个零延时缓冲器。复位子模块核心为 MAX706 芯片,结合后面提到的 EPLD,可为系统各器件提供上电复位、硬复位、软复位等各种复位信号。

(3) Power Span 子模块为 PPMC 系统提供对外 PCI 接口,PCI 总线工作在 33MHz。

(4) EPLD 子模块。因为需要实现的控制逻辑较多,采用 Altera 公司的可编程逻辑芯片 EPM9510ARC208-10,可以为 SDRAM 产生时序信号、读取接口配置信号、汇总各个中断信号、汇总复位信号和时钟信号。

(5) SDRAM 子模块。选用 4 片现代公司的 HY57V653220CT SDRAM 芯片。

3 硬件实现的要点

就硬件实现而言,有几点需要注意:

(1)为以太网和串口分配管脚。如果仅仅是配置一个 MII 接口和一个 RS-232 接口,有多种实现方案。但必须考虑到系统功能的可扩展性,尽可能多地预留 TDM 接口和 ATM UTOPIA 接口。

2)PCI 桥接和总线仲裁

VSO 规定,PPMC 系统必须带有 PCI 接口。但 MPC8260 芯片本身不带 PCI 接口,因而需要外挂 PCI 桥接芯片 PowerSpan。

PowerSpan 是 Tundra 公司生产的 PCI 桥接芯片,总线频率范围为 25~66MHz,支持 PCI2.2 规范。PowerSpan 内部模块如图 3 所示。可以与 MPC8260、PowerPC 603e/740/750 等 PowerPC 系列芯片无缝接合。电气特性符合 VSO PPMC 标准。

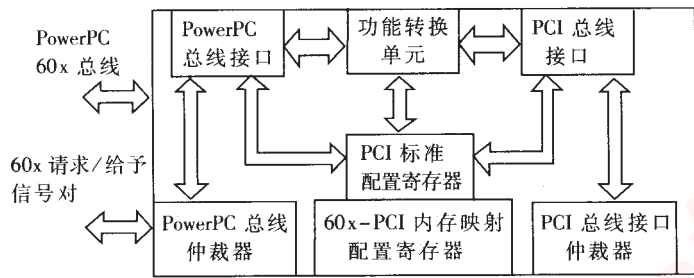


图 3 PowerSpan 内部模块示意图

实际应用时,PowerPC 60x 总线和 PCI 总线上都挂有多个器件,每个器件都可能需要获取总线的控制权以便发起总线操作,因此总线控制权的仲裁十分重要。

就 PCI 总线而言,采用 PowerSpan 内嵌的 PCI 总线仲裁器作为整个 PCI 总线的仲裁器,应该将 PPMC 卡标准插座上各组 GNT、REQ 信号连到 PowerSpan 相应管脚处。PowerPC 60x 总线的情况有点不同,MPC8260 芯片和 PowerSpan 芯片都自带 60x 总线仲裁器。从系统的扩展性考虑,采用 MPC8260 芯片自带的 60x 总线仲裁器。因而 PowerSpan 的 PB_BR[1]、PB_BG[1]、PB_DBG[1]等信号必须连到 MPC8260 芯片对应处。

3)片选信号和内存分配

MPC8260 60x 总线包含 32 位地址线,可以访问高达 4G 字节的内存。MPC8260 内嵌有一个内存分类控制器,该分类控制器提供了多个片选信号。在本系统中,地址空间分为 6 类子空间,如表 3 所示。

以 64 位数据线方式访问 PowerSpan 内部寄存器空间,以及通过 PowerSpan 映射的 PCI 空间。访问二者时,不需要片选信号。PowerSpan 自行判断 60x 地址是否指向自身,决定是否回应。

CS0 代表的 Bootrom 空间有着特殊含义。根据 MPC8260 的规定,复位时,MPC8260 自动读取 Bootrom 中

表 3 内存分配表

60x 内存类型和片选信号	地址范围	含义
PowerSpan Register Space	0X30000000~0X3000FFFF	4KB
MPC8260 Register Space	0XF0000000~0XF001FFFF	128KB
Bootrom (CS0) D[0:7]	0XFE000000~0XFE0FFFFF	1024KB
Flash (CS1) D[0:15]	0XE0000000~0XE0FFFFFF	16MB
SDRAM (CS2) D[0:63]	0X00000000~0X03FFFFFF	64MB
PCI Image n (n=1...8)	Maximum Block Size=2GB Minimum Block Size=64KB	

的指令运行启动程序。Bootrom 空间是唯一不需要额外配置、MPC8260 就可以读取的子模块。

相关配置寄存器包括:MPC8260 中的 BRn、ORn(n=0,1,2)寄存器、IMMR 寄存器以及 PowerSpan 相关设置中的 Slave Image 和 Target Image。

4 硬件实现的难点

保证外部 PCI 器件(没有位于 PPMC 上)通过 PowerSpan 高速访问 PPMC 系统所带 SDRAM 内存,是设计上一大难点。图 4 是外挂 PCI 器件时,内存存取示意图。

图 4 从左向右为 Slave Image 方向。MPC8260 通过设置 PowerSpan 的相关 Slave Image 寄存器,将外部 PCI 空间映射到 60x 空间后,PowerSpan 即成为透明桥。MPC8260 以 60x 总线地址访问外部 PCI 器件,PowerSpan 会自动进行有关转换工作。

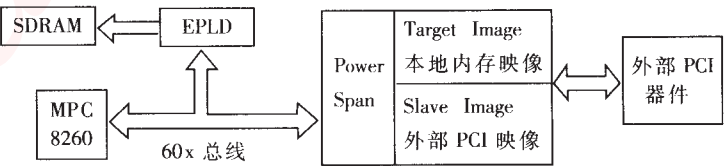


图 4 外挂 PCI 器件示意图

图 4 从右向左为 Target Image 方向。此时外部 PCI 器件要访问 PPMC 上的 SDRAM。通过设置 PowerSpan Target Image 中相关寄存器,将 60x 总线地址空间的一部分(也就是 PPMC 上 SDRAM 对应的 60x 地址空间中的一段)映射到外部 PCI 空间上,外部 PCI 器件就可以以 PCI 总线地址访问 PPMC 所带 SDRAM,PowerSpan 自动进行有关转换工作。

由此可知,MPC8260 和 PowerSpan 都可成为 60x 总线的 Master,都会访问 SDRAM。从效率考虑,必须允许二者可以猝发存取 SDRAM。

与读写 ROM 不同,读写 SDRAM 的时序关系很复杂,要考虑到多种因素,如 SDRAM 的刷新问题。遗憾的是,就 60x 总线信号的时序而言,无法直接与 SDRAM 连接。特别是对 PowerSpan,没有内嵌任何内存控制器,即使 MPC8260 提供了一些辅助控制信号,也无法直接连

接,因而外部须提供逻辑转换。就本 PPMC 系统而言,必须在 EPLD 内实现一个 SDRAM 控制器,这就是图 4 中引入 EPLD 的原因。

该控制器采用 Verilog HDL 实现,可为 SDRAM 提供一系列符合时序的控制信号。控制器就其本质而言,是一个状态机。随着外部 60x 总线输入信号的变化,该状态机在一系列不同状态之间转换,同时输出 SDRAM 控制信号,以及 60x 总线的回应信号。例如就读数据而言,隐去刷新等状态,状态机如图 5 所示。

图 6 是在触发方式下,2-beat 读取数据时,内存控制器产生的控制信号。此时,Page Closed、CAS latency 为 3 个时钟周期,Activate to Read interval 为 2 个时钟周期。 $\overline{RAS}$ 为控制器输出的行地址选择信号, $\overline{CAS}$ 为列地址选择信号, $\overline{DQM}$ 为字节选择信号。

5 初始化软件流程

本 PPMC 的初始化流程有着与 RTOS 中标准 BSP(Board Support Package)要求相同的次序,程序流程如图 7 所示。

系统初始启动时,MPC8260 自动读取 Bootrom 中的指令。Bootrom 中最初一段程序采用 PowerPC 汇编语言编写,该部分程序必须对系统进行基本的初始化工作。其中最重要的是初始化那些与 SDRAM 有关的寄存器,以便可以尽快使用 SDRAM,从而在其后可以使用 C 语言编写的程序进行后继初始化工作。

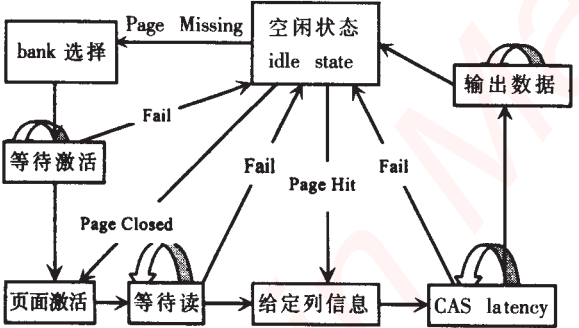


图 5 读数据状态机示意图

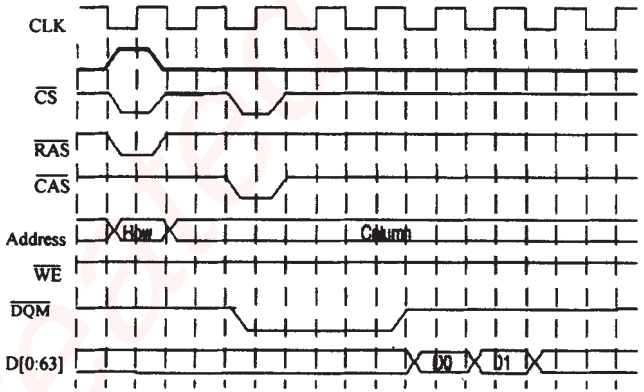


图 6 内存控制器产生的信号时序

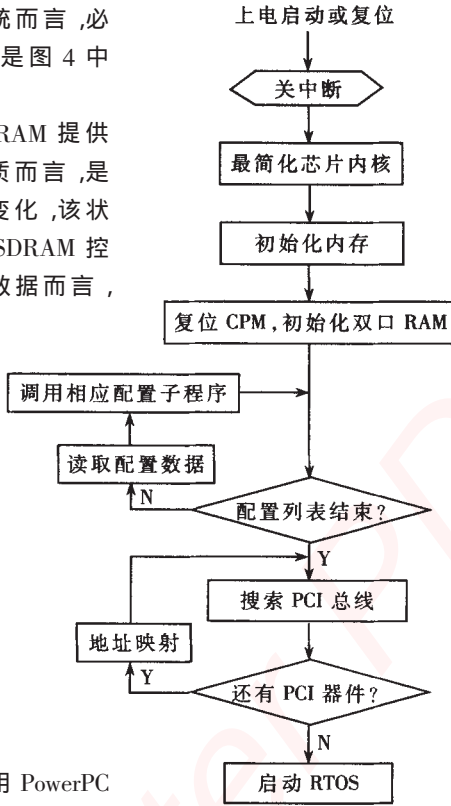


图 7 (a) 初始化流程图

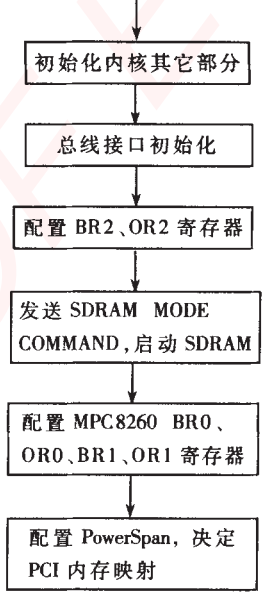


图 7 (b) 复位 CPM 内核具体流程图

本文构建了一个符合 VSO 和 IEEE 标准的 PPMC 系统,可以为嵌入式系统(特别是通讯用嵌入式系统)提供一个功能强大、高度模块化、具备高度可扩展性的微型计算机系统。

参考文献

- 1 IEEE P1368 标准
- 2 VSO PPMC 标准
- 3 Motorola Inc. MPC8260 User's Manual
- 4 Tundra Inc. PowerSpan User's Manual
- 5 Altera Inc. MAX9000 User's Manual

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](#)
48. [基于 PCI 总线的图像处理及传输系统的设计](#)
49. [串口和以太网通信技术在油液在线监测系统中的应用](#)

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](#)
30. [VxWorks 实时操作系统及其在 PC104 下以太网编程的应用](#)

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](#)
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 订制嵌入式 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](#)
36. [基于嵌入式 Linux 的 SD 卡驱动程序的设计与实现](#)
37. [Linux 系统中进程调度策略](#)

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](#)
20. [基于 FPGA Nios II 的等精度频率计设计](#)
21. [基于 FPGA 的 SOPC 设计](#)
22. [基于 SOPC 基本信号产生器的设计与实现](#)

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](#)