基于龙芯 2F 架构的 PMON 分析与优化

张菊莉，张君毅，孟小锁
(西安微电子技术研究所，陕西 西安　710065)

摘　要：基本输入/输出系统(BIOS)是现代计算机系统的重要组成部分，负责计算机系统的开机自检、板级初始化、加载操作系统内核以及基本 I/O 功能。以龙芯 2F CPU 架构为研究背景，在其基础上分析了 PMON 的系统结构与工作流程，通过添加网络驱动程序，最终实现通过 PMON 加载 Linux 和 VxWorks 操作系统。进行必要的功能完善与性能优化，并经过严格的测试以排除可能存在的深层次缺陷，实现了一个性能稳定、功能完善、高可靠性的能与国产龙芯 2F 处理器平台紧密搭配的 BIOS。

关键词：BIOS；CPU；龙芯 2F 处理器；PMON
中图分类号：TN919-34　　**文献标识码**：A　　**文章编号**：1004-373X(2011)02-0019-03

Analysis and Optimization of PMON Based on Loongson-2F CPU

ZHANG Ju-li，ZHANG Jun-yi，MENG Xiao-suo
((Xi'an Micro-Electronic Institute，Xi'an 710065，China))

Abstract：Basic input/output system (BIOS) is an important component of current computer systems，which controls computers' POST，board-level initialization，loading operating systems and basic I/O functions. Based on the research background of Loongson-2F CPU，the system structure and working flow of PMON are analyzed. The loading of Linux and Vxworks operating systems was realized by adding a network driver. The necessary function perfection and performance optimization were carried out. The reliable and stable BIOS which can match the Loongson-2F CPU platform made in China was achieved.

Keywords：BIOS；CPU；Loongson-2F CPU；PMON

龙芯 CPU 使用 PMON 作为基本输入输出系统(BIOS)。PMON 具有强大而丰富的功能，除基本的 I/O 功能外[1]，还包括硬件初始化与检测、操作系统引导和程序调试等功能[2]。PMON 早期的版本功能少且扩展性不好。现在龙芯 CPU 上使用的 PMON 添加了硬盘和其他文件系统的支持，以及显卡的支持等[3]。本文在深入分析 PMON 源码的前提下，添加了 82551 网卡驱动以便以网络下载加载操作系统内核，并对移植好的 PMON 系统进行了功能、性能和稳定性等方面的测试以适应某些重要领域对计算机系统高可靠性要求。通过专门的测试程序来实现此任务，并且尽量保证高的测试覆盖性。

1　基于龙芯 2F 架构的 PMON 分析

1.1　PMON 执行流程分析

在龙芯 2F CPU 上电之初，内存和内存控制器处于不确定状态，因此 CPU 开始执行的 BIOS 代码只能放在非易失性介质中。PMON 的二进制代码就存放于主板上一块 512 KB 的 FLASH 芯片上，其虚拟地址为 0xBFC00000，物理地址是 0x1FC00000[4]。

由于在 ROM FLASH 运行的速度比较慢，并且空间有限，不能随时更改 ROM 中的内容，就需要把引导程序搬到内存里运行。因此，PMON 的运行过程分为两个阶段[5]：第一阶段是在 ROM FLASH 中运行，主要进行基本硬件初始化，如：北桥、南桥、内存控制器、缓存和串口初始化等；第二阶段在内存中执行，主要完成环境变量和基本数据结构的初始化、PCI 总线扫描和设备初始化，显卡初始化、网络协议和设备初始化，并对搜索到的 PCI 总线上的设备进行驱动程序的加载与配置等，最后加载操作系统内核。

1.2　PMON 在 ROM 中的执行流程

龙芯 CPU 开始执行的指令将 CP0 控制寄存器的状态寄存器和原因寄存器清零，让 CPU 处于内核模式[4]。这是因为 MIPS CPU 启动时，必须定义足够的 CPU 控制寄存器状态，以使 CPU 能执行非缓存的指令。在对缓存初始化之前，不能访问缓存。非缓存的异常入口在 0xBFC00000，但此处没有足够的空间存放启动代码，于是设计了一个跳转。这个跳转可以测试 CPU 是否正常工作。如果硬件发生了某种严重的错误，可能会导致 CPU 严重的异常。如果 CPU 正常启动起来并跳转到了正确的位置并执行了预先设计的指令，就可以相信 CPU 正常，某些硬件正常。跳转之后，

PMON 进行一系列的初始化与测试工作,这个过程如图 1 所示。

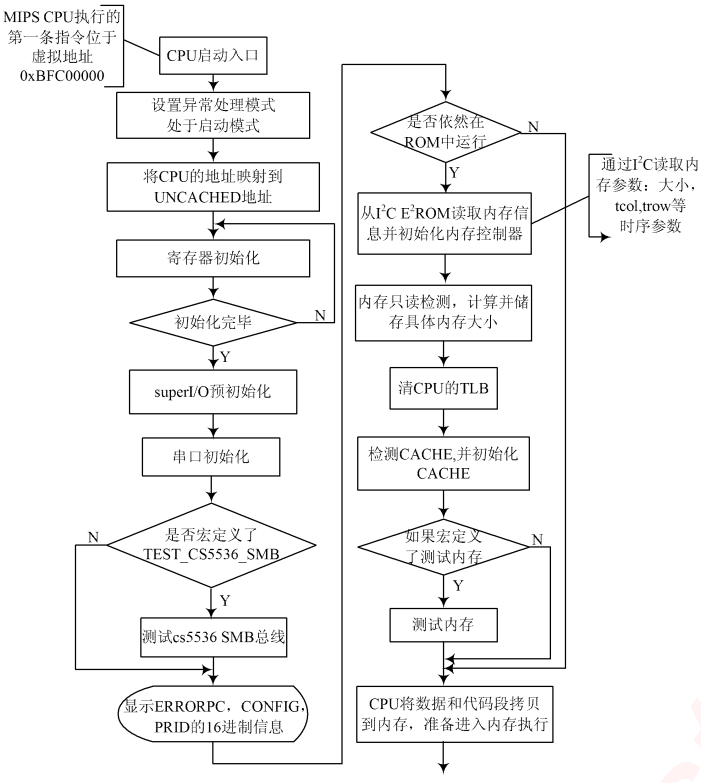

图 1　PMON 在 ROM 中的执行流程

1.3　PMON 在内存中的执行流程

PMON 在内存中的执行过程非常复杂,下面按照其执行流程分析主要的函数:

_init () :主要功能是顺序执行初始化。列表上的所有函数,建立必要的数据结构和编译环境。初始化列表上的函数主要可以分为 3 类:命令初始化函数,文件系统初始化函数,可执行文件初始化函数。

Evenint () :环境变量初始化函数,这些环境变量解释并执行不同的命令。它们对 PMON 来说是必需的,用户可以定义另外的变量来保存任何字符串,如文件名和命令符等。

Tgt_devinit () :主要功能是南桥初始化和 PCI 设备初始化。其中南桥初始化函数是 cs5536_init ()。PCI 设备初始化由函数_pci_businit () 完成,分为两步:先是北桥初始化,然后是设备初始化,包括对设备的扫描和设备所需资源的分配[6]。PCI 设备的初始化部分是 PMON 分析中的难点。

Init_net () :主要进行了如下的工作:虚拟内存初始化等;用_pci_deviceinit 函数完成了 PCI 配置空间的参数初始化,并且初始化显卡;网络系统的初始化包括网络设备的初始化和协议的配置等。

最后加载操作系统内核。这里会用到两个重要的命令 Load 和 g。Load 命令用于加载文件(内核)到内存,g 命令执行内核文件。加载了内核文件之后就可以通过 g 命令来执行内核。

2　基于龙芯 2F 架构的 PMON 修改与优化

2.1　添加 82551 网卡驱动

网卡是一个 PCI 设备,所以其驱动在 PCI 设备初始化时实现。对于 PCI 设备,有一个重要的数据结构 struct cfdata cfdata,这个数组是根据具体平台的配置文件生成的,以下是配置文件的相关部分:

```
mainbus0    at root
localbus0   at mainbus0
pcibr0 at mainbus0
pci *       at pcibr ?
fxp0        at pci ? dev ? function ?
rtk0        at pci ? dev ? function ?
ohci *      at pci ? dev ? function ?
usb *       at usbbus ?
pciide * at pci ? dev ? function ?
wd *        at pciide ? channel ? drive ?
ide_cd * at pciide ? channel ? drive ?
```

这个部分描述了设备之间的链接关系,cfdata 是这个关系的数组表示。另外还有一个 PV 数组定义一个设备的父设备,每个节点的父设备都是一个数组,在设备的 cfdata 结构中定义数组的开始。

上述配置文件中的 fxp0 表示的就是网卡设备。因为 fxp0 是 PCI 子设备,因此在查找 PCI 子设备时,其 4 个子设备:fxp0,rtk0,ohci,pciide 的驱动也会在此时加载。查找设备有一个函数:config_found,其会调用 config_serch,config_search 从静态设备树 cfdata 中查找当前设备的子设备,然后对设备调用 mapply 函数,进行设备的匹配,如果设备存在则会调用该设备的 ca_attach 函数来加载设备的驱动程序。这里找到网卡设备之后会调用 fxp_attach 函数。在设计的网卡驱动中,实现了操作系统将怎样通过网卡驱动来讲网络包发送出去,而网卡收到网络数据包之后怎样通过操作系统来做后续处理[7]。在 PMON 中,网卡中断通过查询来实现。fxp_attach 函数会调用 pci_intr_establish 将中断程序注册到查询列表 poll_list 上。网口要将数据包发送出去,就必须提供一个网络接口,以提供给发送函数。在这里,将网络数据包放在网络接口的 ifp →if_snd 队列中,然后启动 if_start 来开始发送。在网卡发送完一个包后,检查发送队列,如果有剩余则继续发送,直到发送完毕。然后通过函数 e100_poll 来检查是否收到包,并进行收包处理。先进行缓冲区的清除,使能接收队列,查询是否有数据可接收,有则开始接收。无则继续查询并发出信息。

2.2 编译 PMON

添加好了网卡驱动程序之后,需要对 PMON 进行重新配置和编译[2]。配置和编译的过程如下。主要用到几条命令:

```
Cd / usr/ src/ pmon/ zloader. cs5536
Make cfg  # 根据配置文件重新生成 makefile
Make tgt = rom  # 生成 rom bin 文件
Make tgt = ram  # 生成网络加载文件 garam
```

通过网络来更新 PMON 时使用下面的命令:load - f 0xbfc00000 - r tftp://192. 16. 12. 30 来进行 PMON 的在线烧写。并通过网络来加载操作系统:load tftp://192. 16. 12. 30/ VxWorks. 如果加载系统内核成功则验证了网卡驱动的有效。

at usbbus at pci dev function at pcibr at mainbus0 at mainbus0

3 系统测试

3.1 系统启动阶段网卡测试

由于启动阶段网卡只用于加载操作系统内核,对性能并没有太高要求。只需测试其功能即可。通过在启动过程中用另外一台主机对其进行 ping 操作来测试其功能。其界面如图 2 所示。

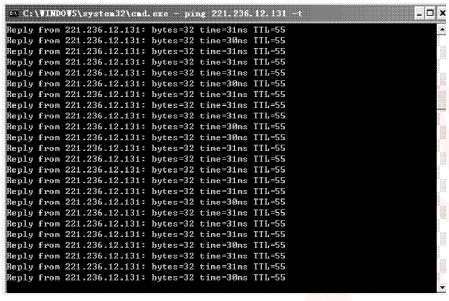

图 2 通过主机进行 ping 操作

3.2 系统启动时间测试

经过测试,在硬盘启动模式下,系统从加电到操作系统启动完毕耗时不超过 35 s,符合绝大部分应用需求。

3.3 500 次开关机测试

为测试移植后的系统 POST 功能的稳定性,进行了 500 次开关机实验。经验证,成功启动次数为 500,失败次数为 0,成功率 100 %。

3.4 200 h 老练测试

为测试 PMON 在系统运行时的稳定性,用专用的整机测试软件进行了 200 h 老练测试。经测试,系统运行正常,未出现故障,测试合格。

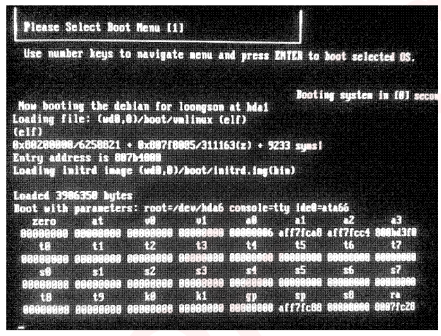

图 3 PMON 启动界面

4 结 语

为了适应 PMON 在国产计算机系统中的需求,对其进行了分析与优化。在分析了源码的基础上,对其进行了必要的修改与优化,增加了 82551 网卡的驱动等。为了验证优化后系统的稳定性,进行了一系列的测试验证,包括启动阶段的网络测试、系统启动时间测量、500 次开关机测试以及 200 h 老练测试等。经验证,修改后的 PMON 系统运行稳定,可以在多个重要领域中进行应用。

参 考 文 献

[1] 韩山秀,樊晓桠,张盛兵,等. BIOS 的设计与实现[J]. 微电子学与计算机,2005,22(11):113-115.

[2] 江苏中科龙梦科技有限公司. PMON 手册. Revision 0. 1[M]. 南京:江苏中科龙梦科技有限公司,2009.

[3] PCI Special Interest Group. PMON users manual [EB/ OL]. [1998-01-30]. http:// www. wanfangdata. com. cn.

[4] SWEETMAN D. MIP 处理器设计透视[M]. 赵俊良,张福新,陶品,译. 北京: 北京航空航天大学出版社,2005.

[5] 蔡军生. http://blog. csdn. net/ caimouse.

[6] Element. PCI local bus specification, version 2. 3 [EB/ OL]. [2007-10-19]. http:// www. pudn. com.

[7] Intel. 82551IT fast Ethernet PCI controller datasheet, revision 4. 0 [EB/ OL]. [2004-09- 03]. http:// download. intel. com.

[8] 中国科学院计算机技术研究所. Loongson2F 用户手册(1. 0 版)[M]. 北京:中国科学院计算机技术研究所,2009.

[9] 李雷,郑为民,刘金刚. 基于 PMON 的龙芯 2E 处理器 BIOS 优化设计[D]. 北京:首都师范大学,2008.

[10] 李雷,郑为民,刘金刚. 基于 PMON 的 BIOS 初始化 VGA 模拟器[J]. 计算机工程,2009,35(1):204-206.

作者简介: 张菊莉 女,1984 年出生,硕士。主要研究方向为计算机控制理论。
张君毅 男,1983 年出生,硕士。主要研究方向为嵌入式操作系统和实时嵌入式软件。
孟小锁 男,1961 年出生,研究员,硕士生导师。主要研究方向为实时嵌入式软件与网络集成式软件开发。

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. 基于磁盘异或引擎的 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
50. USB30 数据传输协议分析及实现
51. IEEE 1588 协议在工业以太网中的实现
52. 基于 USB30 的设备自定义请求实现方法
53. IEEE1588 协议在网络测控系统中的应用
54. USB30 物理层中弹性缓冲的设计与实现

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
31. 实时操作系统任务调度策略的研究与设计
32. 军事指挥系统中 VxWorks 下汉字显示技术
33. 基于 VxWorks 实时控制系统中文交互界面开发平台
34. 基于 VxWorks 操作系统的 WindML 图形操控界面实现方法

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
38. 嵌入式 Linux 实时性方法
39. 基于实时 Linux 计算机联锁系统实时性分析与改进
40. 基于嵌入式 Linux 下的 USB30 驱动程序开发方法研究

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
21. DCOM 协议在网络冗余环境下的应用

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
18. 基于 MPC8260 处理器的 PPMC 系统
19. 基于 PowerPC 的控制器研究与设计
20. 基于 PowerPC 的模拟量输入接口扩展

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μC-OSⅡ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
26. ARM11 嵌入式系统 Linux 下 LCD 的驱动设计
27. Uboot 在 S3C2440 上的移植
28. 基于 ARM11 的嵌入式无线视频终端的设计

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 NiosⅡ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
23. 基于龙芯平台的 PMON 研究与开发
24. 基于 X86 平台的嵌入式 BIOS 可配置设计

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