基于 X86 平台的嵌入式 BIOS 可配置设计

张　雁，熊庭刚，马　中

(武汉数字工程研究所产品研发部，武汉 430074)

摘　要：基于 X86 平台的嵌入式计算机得到了广泛应用。该文就如何有效提高嵌入式 BIOS 的配置速度，抛弃传统的 BIOS 开发的繁琐过程，缩短 BIOS 的开发周期，提出了一种不同于传统 BIOS 的新的设计方法——将 POST 过程的检测和初始化代码与初始化数据分离，以简化配置过程对核心代码的维护。实践证明，该方法使嵌入式 BIOS 的配置工作更加简单、高效。

关键词：嵌入式；BIOS；可配置；X86 平台；Chipset

Configurable Design of Embedded BIOS Based on X86 Platform

ZHANG Yan, XIONG Tinggang, MA Zhong

(Department of Products Development, Wuhan Digital Engineering Institute, Wuhan 430074)

【Abstract】Embedded computer based on X86 platform has been widely applied to many fields. There are a great amount of interest in speeding up the configuration of embedded BIOS, shortening the developing period, and discarding the old complicated BIOS configuration process. This paper brings forward a new design method of BIOS differing from traditional ones——to separate codes of test and initialization from the data of initialization, which simplifies the maintenance of core codes during the configuration. The practice shows that the new method is more simple and effective in the embedded BIOS configuration.

【Key words】 Embedded; BIOS; Configurable; X86 platform; Chipset

1 概述

嵌入式计算机系统随着其应用方式不同，具有不同的体系结构形式[1]。基于 X86 平台的嵌入式计算机，因其采用与标准 PC 相同的硬件结构、软件操作系统和软件开发平台，开发方便，应用程序设计资源丰富，因此在嵌入式系统中占据越来越大的份额，尤其在工业控制和军事领域得到了广泛的应用[2]。

不同于没有操作系统、功能单一的消费类电子产品，也不同于加载启动完全由 Boot Loader 引导加载程序来完成的某些嵌入式系统，基于 X86 平台的嵌入式计算机的结构由 PC 而来，其功能复杂，初始化硬件和引导操作系统仍然由 BIOS 来完成。但一个嵌入式系统的 BIOS 并不需要像通用 PC BIOS 那样具有那么多的灵活性，因为通常它仅需处理某种特定的硬件配置方案。所以，嵌入式 BIOS 是在传统 BIOS 的基础上，移除了许多嵌入式平台不必要的功能，为基于 X86 平台的嵌入式计算机系统所定制的基本输入输出系统。

在这一类嵌入式计算机的开发过程中，BIOS 的配置是一个非常重要的环节，决定了整个嵌入式产品的开发周期长短和上市时间。在采用 X86 架构的计算机系统中，造成 BIOS 差异的原因，除了 CPU 规格和系统配置不同外，构成系统的 Chipset 是最关键的因素[3]。即使是为同一代 CPU 系列所开发的 BIOS，由于每套 Chipset 在配置和功能上的不同，以及在寄存器、I/O 端口地址和 DRAM Bank 数量等方面的差异，其初始化的数据和方式都不尽相同。因此为不同嵌入式系统配置 BIOS 时，需要大量地修改 Chipset 和 SuperI/O 有关的源程序，甚至可能对整个程序的结构进行调整，这样不仅使得 BIOS 的开发与配置工作量相当大、耗费的时间非常长，而且容易出错。

如何才能抛弃传统的 BIOS 开发的繁琐过程，快速、有效地提高嵌入式产品 BIOS 的配置速度，缩短开发周期，尽量减少对核心代码的修改，避免因对程序修改而造成混乱和隐患，提高开发的效率，是当前基于 X86 平台的嵌入式系统开发急待解决的问题。本文提出一种不同于传统 BIOS 的新的设计方法——将 POST 过程的检测和初始化代码与初始化数据分离，以简化配置过程对核心代码的维护，实现嵌入式 BIOS 的灵活配置。

2 嵌入式系统的 BIOS 结构分析

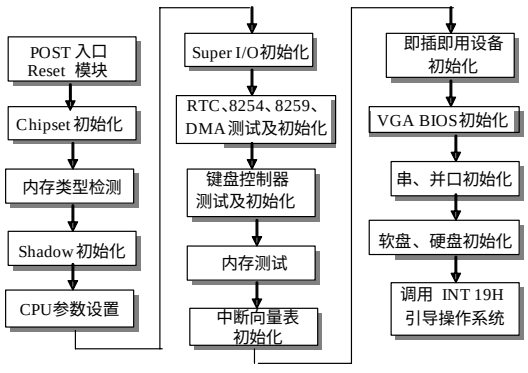

图 1 POST 工作流程

嵌入式 BIOS 与通用 BIOS 一样，其工作可以由两部分来概括：POST 和软件中断服务例程。软件中断服务例程是为 INT13H、INT15H 等软件中断提供服务的程序，这是一些通

用的程序，对于任何 PC 架构的系统来说，它们具有相同的接口和功能。POST 通常称为上电自检模块，它负责上电后对系统所有硬件和寄存器的检测和初始化工作，最后调用 INT 19H 装入引导设备的引导扇区，然后将控制权交给引导程序，引导操作系统[4,5]。其过程实际上是由一系列的检测工作和一些把特定的数据写入指定硬件寄存器的指令序列来组成的。其工作流程如图 1 所示。

该流程中除了 Chipset 和 SuperI/O 的初始化，以及少量有关 CPU 的参数设置外，其它过程都是对 PC 标准部件的操作，在对不同平台的 BIOS 进行配置时，对标准部件的初始化改动很少甚至不必修改，配置过程实际上就是针对 Chipset 和 SuperI/O 初始化部分的重新设计。

3 嵌入式 BIOS 可配置设计

3.1 总体方案

首先将 Chipset 和 SuperI/O 初始化信息进行分类，对不同的操作类型定义相应的数据结构，然后按定义将初始化数据格式化，根据 POST 的顺序，存放在专门的初始化数据表格区。每个不同应用的系统虽然其配置不同，采用不同的 Chipset 和 Super I/O，但是 POST 的执行顺序大同小异，所以在设计中将执行顺序固定。

每一组数据表格起始地址采用动态分配，通过初始化数据向量表来得到这组数据在内存中的绝对地址。初始化数据向量表中以固定次序存放按 POST 执行顺序进行初始化的数据表格的入口地址。

初始化数据表格区从 0FA00：100H 处开始，初始化数据顺序存放，每组表格以 0FFH 为结束标志。对应不同操作类型的初始化数据，由不同的初始化子程序进行初始化处理，它们之间的映射由初始化子程序向量表来完成。

通过初始化数据向量表和初始化子程序向量表，实现了 BIOS 的程序代码和数据的完全分离，检测和初始化代码在程序代码区，初始化数据在数据区，这样对于不同的配置，只需修改初始化数据向量表和初始化数据表格区，不必再对初始化代码进行修改。

按以上方案设计的嵌入式 BIOS 程序框架如图 2 所示。

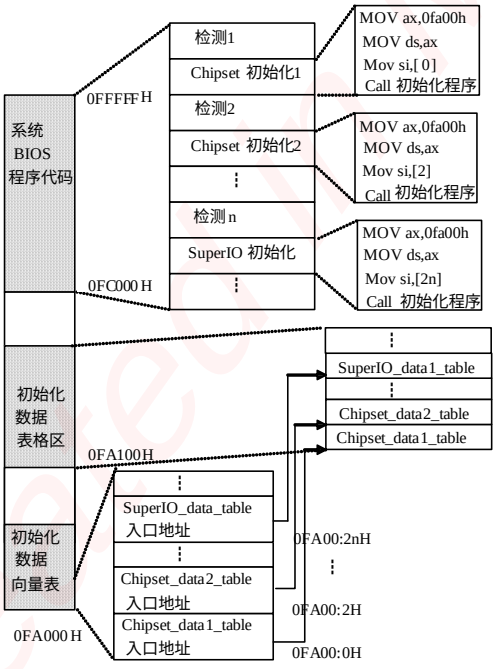

图 2 嵌入式 BIOS 程序框架

3.2 初始化信息分类

通过对 BIOS 核心代码的分析，将初始化信息进行抽象处理，可以按照数据和操作进行分类，其中，数据分为 3 种类型，操作分为 6 种类型，具体定义如下：

(1)数据类型：1)8 位方式；2)16 位方式；3)32 位方式。

(2)操作类型：1)I/O 直接地址全设置；2)I/O 直接地址按位设置；3)通过 I/O 索引地址全设置；4)通过 I/O 索引地址按位设置；5)存储器直接地址全设置；6)存储器直接地址按位设置。

3.3 数据结构定义

为便于处理，将数据类型和操作类型组合为一个字节，形成类型编码，而随后定义的所有数据结构都是由类型编码加上地址和数据组成。类型编码、数据类型、操作类型的定义分别见表 1、表 2、表 3。

表 1 类型编码定义

位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
0	0	数据类型		操作类型			

表 2 数据类型定义

位 5	位 4	描 述
0	0	数据为 8 位方式
0	1	数据为 16 位方式
1	0	数据为 32 位方式

表 3 操作类型定义

位 3	位 2	位 1	位 0	描 述
0	0	0	0	I/O 直接地址全设置
0	0	0	1	I/O 直接地址按位设置
0	0	1	0	通过 I/O 索引地址全设置
0	0	1	1	通过 I/O 索引地址按位设置
0	1	0	0	存储器直接地址全设置
0	1	0	1	存储器直接地址按位设置

(1)I/O 直接地址全设置

采用三元组结构定义该操作类型的数据：

（类型编码，口地址，设定值）

(2)I/O 直接地址按位设置

采用四元组结构定义该操作类型的数据：

（类型编码，口地址，屏蔽值，设定值）

(3)通过 I/O 索引地址全设置

采用五元组结构定义该操作类型的数据：

（类型编码，索引口，索引值，数据口，设定值）

(4)通过 I/O 索引地址按位设置

采用六元组结构定义该操作类型的数据：

（类型编码，索引口，索引值，数据口，屏蔽值，设定值）

(5)存储器直接地址全设置

采用三元组结构定义该操作类型的数据：

（类型编码，存储单元地址，设定值）

(6)存储器直接地址按位设置

采用四元组结构定义该操作类型的数据：

（类型编码，存储单元地址，屏蔽值，设定值）

通过 I/O 索引地址来寻址是指通过两个 I/O 地址来访问一组寄存器，这是目前 PC 机中用于扩展 I/O 的一种方法。因为在早期 PC 机中，IO 端口的寻址空间定义为 1KB，通过索引来寻址扩大了 I/O 的寻址空间。存储器空间比起 IO 空间大得多，所以目前不存在这个问题。

3.4 初始化子程序与类型编码之间的映射

BIOS 程序使用一个公用的初始化程序，按照类型编码，通过初始化子程序向量表的映射，调用相应的子程序处理初始化信息。数据类型和操作类型的组合共有 18 种可能，因此，

代码段要提供 18 种初始化子程序。

初始化子程序向量表的功能类似中断向量表，它在代码段进行定义。向量表占用 96 个单元，现在用到了 36 个单元，预留 60 个单元，一方面是为了简化向量表的查询，另一方面也是留待以后扩充新的子程序。初始化子程序的序号与类型编码一致。

4 配置实例与分析

在按以上方案完成嵌入式 BIOS 的程序设计后，结合实际工作，为我们自行设计的基于 X86 平台的某型号嵌入式计算机系统配置了 BIOS，验证了方案的正确性和可行性。

该计算机采用 ST 公司开发的 SoC（System on a Chip）ST486 Client。它集成了 CPU 和芯片组的所有功能部件。系统内存为 32MB DRAM，Super I/O 芯片采用 FDC37C782。它包含硬盘控制器、软盘控制器、DMA 控制器、中断控制器、系统实时时钟和键盘控制器，提供 2 个 RS-232 串行口，1 个并行口，1 个软盘接口，1 个鼠标接口和 1 个键盘接口。

通过芯片资料的分析，确定了需要处理的 Chipset 的初始化数据有 21 组，Super I/O 的初始化工作可以一次完成，因此其数据只有 1 组，这 22 组数据按数据结构定义格式化后，根据代码的初始化顺序存放在数据区。

以对 STPC_CACHE1(寄存器号 21H)的初始化为例，这是一个操作类型为通过 I/O 索引地址按位设置、数据类型为 8 位方式的初始化数据，其各项数据如表 4 所示。

表 4 STPC_CACHE1 初始化数据数据结构

类型编码	索引口	索引值	数据口	屏蔽值	设定值
00000011B	22H	STPC_CACHE1	23H	01H	0a0H

因此，它在数据区的六元组格式应该是：

03H,22H,21H,23H,01H,0a0H

根据类型编码 03H，调用 03 号子程序处理这个六元组数据。按照 POST 的流程，BIOS 以上述方式逐步完成对 Chipset 和 SuperI/O 所有寄存器的初始化。

从对硬件资料的分析开始，到初始化数据分组的规划和按数据结构格式确定初始化数据，用本文提出的设计方案配置嵌入式加固计算机系统的 BIOS 仅仅用了 3 个月。该 BIOS 满足加固计算机系统的需求，并支持 VxWorks、 MS-DOS、Windows 等操作系统。如果采用通用 BIOS 的改造方法，由于通用 BIOS 要满足各种硬件配置的需求，其功能强大，分支繁多，源码的分析工作就占据了大量的时间，而修改工作涉及代码和数据，同时，对芯片手册的研究，Chipset 和 SuperI/O 各寄存器的分析工作也是必不可少的，以同样的人力，为该型嵌入式加固计算机配置通用 BIOS，花费 1 年甚至更长的时间。而随着今后工作对该方案 BIOS 程序的日趋完善，BIOS 开发人员对 BIOS 工作流程的更加熟悉，嵌入式 BIOS 配置的速度将会进一步提高。

5 结束语

嵌入式 BIOS 是自主开发基于 X86 平台的嵌入式系统必备的基础软件，如何进行自主的 BIOS 设计，快速、灵活地为自行研制的嵌入式计算机配置最优化的 BIOS，确保 BIOS 的信息安全，提高嵌入式计算机产品的竞争力，是当前面临的迫切要求。本文提出了一种不同于传统 BIOS 设计思想的新的设计方法，完成了总体方案设计、数据结构的定义和初始化程序的设计，并在实际工作中成功地实现了对自主设计的嵌入式应用的加固计算机系统的嵌入式 BIOS 配置，该 BIOS 支持的嵌入式系统现已在某工程中成功应用。该方案使得 BIOS 开发人员面对不同的硬件平台，只需更改初始化数据区的数据，而不必修改初始化程序代码，减少了对核心代码的维护，使嵌入式 BIOS 的配置工作更加简单、高效。

参考文献

1 Ganssle, Jack G. The Art of Programming Embedded Systems[M]. Academic Press, 1992: 21-23.

2 陈章龙. 嵌入式系统的架构及其开发应用[C]. 嵌入式系统及其应用研讨会, 2002.

3 陈文钦. BIOS 研发技术剖析[M]. 北京: 清华大学出版社, 2001: 30-37.

4 Gilluwe F V. The Undocumented PC. A Programmer’s Guide to I/O, CPUs, and Fixed Memory Areas[M]. Addison Wesley, Longman, Inc., 2000.

5 Norton P, Aitken P, Wilton R. PC Programmer’s Bible[M]. Microsoft Press, 1993: 178-180.

(上接第 176 页)

参考文献

1 Microsoft. 全面掌握电子商务开发技术[M]. 北京: 清华大学, 2000-10.

2 邓仲华. 电子商务系统分析与设计[M]. 武汉：武汉大学出版社, 2003-08.

3 李凡长. 动态模糊逻辑引论[M]. 昆明：云南科技出版社, 2004-04.

4 李凡长, 朱维华. 动态模糊逻辑及其应用[M]. 昆明：云南科技出版社, 1996-04.

5 Stair R M, Reynolds G W. 信息系统原理[M]. 张 靖, 译. 北京 :机械工业出版社, 2003-06.

(上接第 196 页)

5 唐红卫, 桑 农, 曹治国, 等. ART-2 神经网络的研究与改进[J]. 红外与激光工程, 2004, 33(1): 101-106.

6 Ramchandran K. Wavelets, Subband Coding, and Best Bases[J]. Proceedings of IEEE, 1996, 84(4): 541-560.

7 谭志国. 灰色自适应谐振理论及其性能评估[D]. 南京: 南京航空航天大学, 2003: 16-17.

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

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

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

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
18. 基于 MPC8260 处理器的 PPMC 系统

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

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

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