

基于 GPU FPGA 芯片原型的 VxWorks 下驱动软件开发

马城城,田 泽,黎小玉

(中国航空计算技术研究所,陕西 西安 710119)

摘 要 为满足日益复杂的应用需求、减轻 CPU 日益繁重的图形处理任务,促使图形处理器 GPU 产生、应用和不断发展。驱动软件作为 GPU 的重要组成部分,与 GPU 硬件的契合程度直接影响整个图形系统性能的发挥,出于各种原因高端 GPU 配套的图形驱动软件对外不公开或价格昂贵,对图形应用系统的开发带来不便。文中基于自研 GPU 芯片 FPGA 原型图形系统,讲述了 VxWorks 下 GPU 驱动软件的设计与实现,该驱动软件为用户提供 3D 处理和 2D 处理接口。其中 3D 处理实现完整的 OpenGL1.3 基本库及 GLU、GLUT 辅助库 2D 处理使用 VxWorks 操作系统的 WindML 组件实现。较好实现了图形处理软件与硬件的配合,对自主 GPU 芯片应用开发意义重大。

关键词 VxWorks 图形处理器 OpenGL WindML

中图分类号 TP311

文献标识码 A

文章编号 1673-629X 2013 07-0084-03

doi 10.3969/j.issn.1673-629X.2013.07.021

Development of Driver Software for GPU Based on FPGA in VxWorks

MA Cheng-cheng, TIAN Ze, LI Xiao-yu

(China Aeronautical Computing Technique Research Institute, Xi'an 710119, China)

Abstract In order to meet the complicated application demand and reduce the increasingly graphic task on CPU, the Graphic Process Unit (GPU) has developed continually. The driver is an important part of GPU that affects the performance of whole system by cooperating with GPU hardware. It's difficult to create graphic applications on GPU because the driver is not opened for many reasons. It introduces the design and implementation of self-design GPU driver based on VxWorks. The driver offers 3D operation and 2D operation. The 3D operation achieves OpenGL1.3 kernel library, GLU library and GLUT library. 2D operation is realized by WindML in VxWorks. The driver does well in the cooperation between graphic hardware and graphic software. It provides a useful reference for application on GPU chip.

Key words VxWorks GPU OpenGL WindML

0 引 言

单一 CPU 很难满足复杂三维图形的绘制需求,基于专用的图形处理器来提高系统图形的绘制性能,可以减轻 CPU 负担,整体提升系统的性能^[1]。在这种趋势下,研究和设计适合嵌入式系统使用的图形处理器具有重要意义。图形处理器驱动软件是图形处理器的重要组成部分,是图形处理器性能发挥的关键,研究和设计图形处理器驱动软件对图形处理器的研究设计具有重要意义。

在 GPU 芯片开发过程中,开发基于 FPGA 芯片原型的驱动程序,在芯片原型阶段充分验证芯片功能,并尽早实现驱动软件的开发,为加速芯片应用推广打下基础。

文中将介绍基于自主研制的 GPU 芯片 FPGA 原型图形系统,以 VxWorks 为软件平台的 GPU 驱动软件的设计与实现。对 GPU 驱动软件的开发具有一定借鉴意义。

1 基于自主研制的 GPU 芯片 FPGA 原型图形系统概述

随着应用需求的不断发展,人们对于高质量的图形处理需求在不断增加,专门用于图形处理的 GPU,是为了减轻 CPU 日益繁重的图形处理任务,使得 CPU 更能专注于通用控制^[1]。GPU 每一次突破性发展,实质上都是将一部分原本属于 CPU 的处理功能从 CPU 剥离出去,在自己身上重新实现并且对处理性能予以进一步的优化提高。经过多年发展,图形处理系统形成现今 CPU+GPU 的处理结构。

基于自主研制 GPU 芯片 FPGA 原型的图形处理系统采用目前流行的 CPU+GPU 结构,其中自研 GPU 芯片 FPGA 原型实现完整的 OpenGL 1.3 基本库,CPU 采用 Motorola MPC8245 处理器。在图形应用程序的处理过程中,CPU 负责将要执行的图形应用程序翻译成 GPU 可处理的渲染序列,通过 PCI 总线传递给 GPU。如图 1 所示,GPU 接收到 CPU 传来的渲染序列,通过命令处理器将渲染序列分为 2D 和 3D 处理命令,然后分别经过 2D 处理器或者 3D 处理器处理,显示输出处理结果。

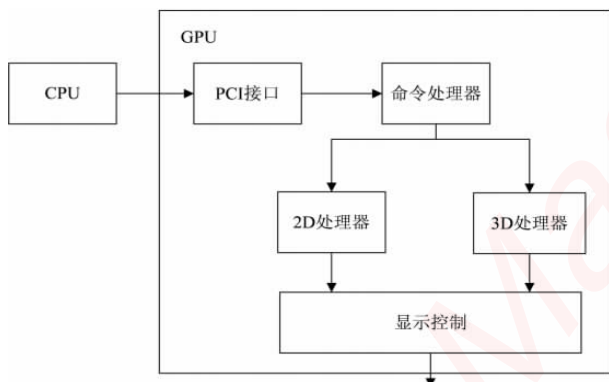


图 1 自研图形处理器系统

2 VxWorks 及图形驱动开发概述

VxWorks 是美国 WindRiver 公司开发的一款高性能、可裁减的嵌入式实时操作系统^[2]。它以良好的可靠性和卓越的实时性被广泛地应用于通信、航空、航天等高精尖技术及实时性要求极高的领域中,如卫星通信、飞机导航等。

Wind 媒体库 WindML—Wind Media Library 是嵌入式实时操作系统 VxWorks 提供的一个可裁减的多媒体组件^[3],该组件库支持基于嵌入式系统的多媒体应用程序。它定义了图形驱动、视频驱动、字体驱动、输入驱动和音频驱动^[4],为各种嵌入式系统提供基本的图形、视频和音频技术,并可以开发基于这些系统的设备驱动。它提供一套与硬件无关的逻辑 API 给应用层调用,而给底层提供不同硬件构架的驱动,使程序具

有很好的移植性和可扩展性。它由两部分组成 软件开发工具包 SDK 和驱动开发工具包 DDK^[5],如图 2 所示。

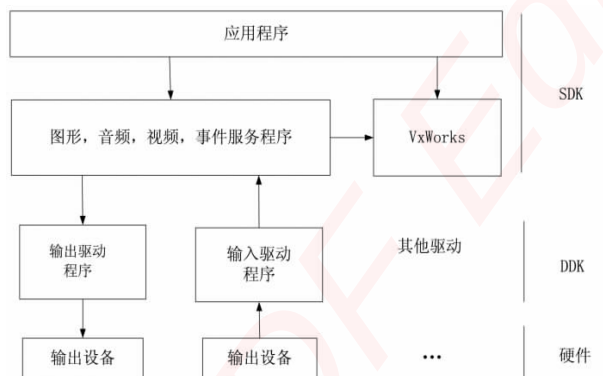


图 2 WindML 架构

SDK 组件用于开发应用程序。它为图形、输入处理、多媒体、字体和内存管理提供了一个广泛的 API^[6],允许开发者为不同硬件平台编写硬件独立的移植代码。DDK 组件是可扩展的、自定义的,适用于实现驱动程序。它为通用硬件配置提供一系列完整的参考驱动程序。

3 基于自主研制的 GPU 芯片 FPGA 原型的 VxWorks 下驱动软件开发

图形处理器驱动软件是为隐藏图形设备的底层实现^[7],给应用层软件提供标准图形接口的中间软件。如图 3 所示,它由三层组成 硬件驱动层,操作系统层,用户接口层^[8]。其中用户接口层包含窗口系统,2D 处理,3D 处理。

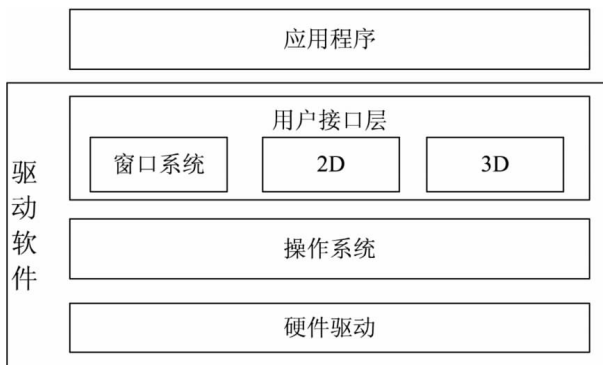


图 3 图形处理器驱动软件的架构

用户接口层为不同图形处理提供所需的接口,窗口系统是在屏幕某限定区域管理程序输入和输出的一组工具,主要负责管理窗口、事件、文本的输入、文本和图形的输出以及各种对象属性,使用用户生成的事件来调度。2D 处理,3D 处理负责处理来自用户的不同图形处理请求。

操作系统层负责管理存储空间,任务调度和 I/O 设备,并且使用调度算法和 I/O 封锁来调度,为上层应

用提供必须的软件环境。

硬件驱动层负责直接操作硬件,隐藏硬件的具体实现,为上层用户提供操作硬件的接口。

本驱动软件硬件驱动层实现 PCI 驱动、WindML DDK 定义的硬件操作接口等 操作系统层使用 Vx-Works 及 WindML 组件 用户接口层的窗口系统和 2D 处理通过封装 WindML SDK 接口实现 3D 处理实现 OpenGL1.3 基本库,GLU 实用库 及 GLUT 工具库。

其中 3D 处理实现的 OpenGL (Open Graphics Library 是 SGI 公司开发的一套计算机图形处理系统,是图形硬件的软件接口。其独立于硬件设备、窗口系统和操作系统,用于三维图像 二维亦可 绘制。OpenGL 不是一种编程语言,而是一种应用程序编程接口 API^[1]。其库函数包含 OpenGL 基本库 (GL)、OpenGL 实用库 (GLU)、OpenGL 工具库 (GLUT) 等。

图形应用程序通过本驱动软件处理时,首先根据调用接口功能的不同转入 2D 处理或 3D 处理,其中 2D 处理和窗口系统通过实现 WindML DDK 层接口实现硬件加速^[1] 3D 处理首先转化为 OpenGL 基本库,然后翻译为图形处理器可识别的操作指令,通过 PCI 总线传给硬件完成处理。图 4 所示为自研图形处理器驱动软件的模块设计。

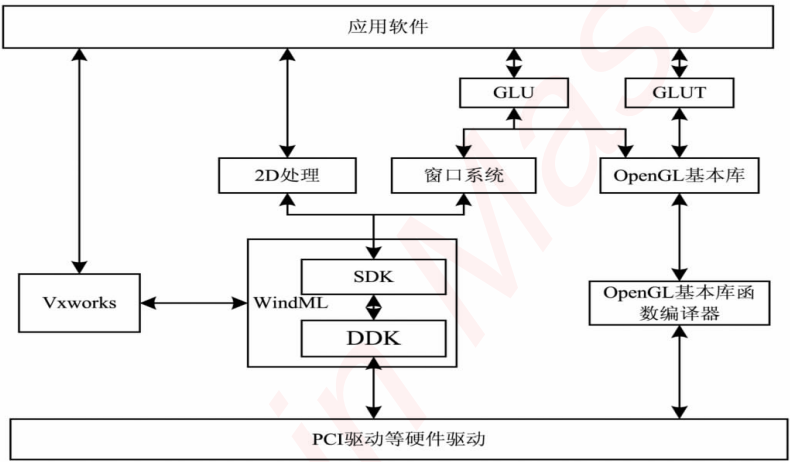


图 4 图形处理器驱动软件的设计

PCI 驱动程序实现链接 CPU 和 GPU 的 PCI 总线的初始化,为图形处理命令的传输,寄存器操作等硬件操作提供通道。

OpenGL 基本库函数编译器负责将 OpenGL 基本库函数编译成 GPU 可识别的操作命令序列。

WindML 的 SDK 层向上层应用提供窗口系统接口和 2D 处理接口,DDK 层实现 GPU 硬件的管理和控制。

GLU、GLUT 是将 OpenGL 基本库结合窗口系统封装成 OpenGL 实用库和 OpenGL 工具库。

下面主要针对图形处理器驱动软件实现的难点和

关键部分进行分析介绍。

OpenGL 基本库函数编译器是将 OpenGL 基本库函数编译成图形处理硬件能够运行的操作指令,该部分需对操作指令格式理解的基础上,对所有 OpenGL1.3 基本库函数进行编译。

自研图形系统操作指令格式如图 5 所示包含命令码,控制字和四个数据字。

命令码	控制字	数据1	数据2	数据3	数据4
[0:9]	[0:9]	[96:127]	[64:95]	[32:63]	[0:31]

图 5 自研图形系统操作指令格式

命令码段用来识别不同的 OpenGL 基本库接口,控制字段用来指定数据的组织和存储格式,数据段用来按控制字段指定的格式存储命令要使用的参数。

如 glColor3f (R,G,B) 的命令编码如图 6 所示 :

命令码	控制字	数据1	数据2	数据3	数据4
[0:9]	[0:9]	[96:127]	[64:95]	[32:63]	[0:31]
0001000000	0000000011	R	G	B	

图 6 glColor3f 编码格式

其中命令码用来指定 glColor3 函数,控制字定义如下 :

ctrl [0] 表示数据类型 :
3'b000,byte 3'b001,short
3'b010,int 3'b011,float3'b100,
double 3'b101,unsigned byte
3'b110,unsigned short 3'b111,unsigned int

数据段如图所示,R、G、B 参数分别存在数据 1、数据 2、数据 3 中。类似对所有 OpenGL1.3 基本库接口进行编译。

WindML 的移植是该软件实现的关键部分也是难点,该部分需要实现图形显示设备的创建函数,显示模式的选择,显示控制的初始化以及 2D 操作的硬件加速。硬件加速部分根据自身硬件情况选择实现 ugl_ugi_driver 结构体中所定义的接口^[1]。

```
typedef struct ugl_ugi_driver
{
/* Data Members */
UGL_MODE * pMode ;/* display mode */
UGL_PAGE * pPageZero ;/* First Page */
void * extension ;/* optional driver extensions */
/* General */
```


87.5% 高于没有进行 RBF 数据预处理的粗糙集挖掘预测精度 81.25%)。分析所得结果,由于 RBF 神经网络的预处理,使得传递给粗糙集进行挖掘的数据更加精确。通过这种方法可以大大降低数据中一些不可靠的数据对于数据挖掘的影响,使得挖掘的效果更加显著。尤其是在大型数据挖掘项目上,优化后的数据在降低错误率、提高精度方面有着积极的作用。

当然,随着对 RBF 神经网络与粗糙集的研究的深入,可以通过优化 RBF 神经网络的算法及粗糙集数据挖掘的算法提高 RBF 神经网络与粗糙集在数据挖掘中的准确性。

5 结束语

文中融合 RBF 神经网络与粗糙集理论等数据挖掘技术,利用 RBF 训练速度快,泛化能力强的优点,提出一种新的先由 RBF 神经网络优化数据,再传递给粗糙集进行数据挖掘的新技术。通过对比未经过 RBF 神经网络处理的数据挖掘结果,反应出 RBF 神经网络与粗糙集理论结合的算法的良好效果。

参考文献 :

[1] Chen M S, Han J, Yu P S. Data mining: an overview from a

(上接第 86 页)

```
UGL_STATUS *(info) struct ugl_ugi_driver * pDriver,
UGL_INFO_REQ infoRequest, void * info ) ;
UGL_STATUS *(destroy) struct ugl_ugi_driver * pDriver ) ;
.....
UGL_UGI_DRIVER ;
```

4 结束语

文中介绍了自主研制的基于 FPGA 的 GPU 芯片原型的驱动软件的设计实现,其主要的难点在于 WindML 的移植和使用,需要对 WindML SDK 层和 DDK 层接口进行分析结合已有驱动开发, GLUT 和 GLU 的封装实现需透彻理解其接口的定义,结合对 mesa3D 源码的分析实现。目前 3D 处理仅支持 OpenGL 1.3 接口,使用 WindML 提供的简单窗口系统,下一步可在已有的接口之上封装实现更高版本的 OpenGL 接口以及移植功能更加强大的窗口系统,以满足更多的应用需求。

参考文献 :

[1] 饶志恒. 图形处理器管线的研究与实现 [D]. 长沙 湖南大学, 2010.

database perspective [J]. IEEE Trans on Knowledge and Data Engineering, 1996, 8 (6): 866-883.

- [2] 何炎祥, 陈莘萌. Agent 和多 Agent 系统的设计与应用 [M]. 武汉 武汉大学出版社, 2001.
- [3] 李映颖, 朱立贵, 张德全, 等. 基于 BP 和 RBF 神经网络对试飞数据预处理比较研究 [J]. 计量与测试技术, 2009, 36 (2): 1-2.
- [4] 唐昌盛, 曲建岭. 基于 RBF 神经网络的飞参数据预处理 [J]. 计测技术, 2007, 27 (1): 11-16.
- [5] 汪小燕, 杨思春. 一种基于分辨矩阵的新的属性约简算法 [J]. 计算机技术与发展, 2008, 18 (2): 77-79.
- [6] 陈 贞. 基于属性权重的区分举证启发式约简算法 [J]. 莆田学院学报, 2007, 14 (1): 15-18.
- [7] Pawlak Z. Rough sets [J]. International Journal of Computer and Information Science, 1982, 11 (5): 341-356.
- [8] Pawlak Z. Rough sets: theoretical aspects of reasoning about data [M]. Boston: Kluwer Academic Publishers, 1991: 72-80.
- [9] 王国胤, 于 洪, 杨大春. 基于条件信息熵的决策表约简 [J]. 计算机学报, 2002, 25 (7): 759-766.
- [10] 丁宝桢, 桑 琳, 朱全英, 等. 基于信息熵的粗糙集属性约简及其应用 [J]. 计算机工程与应用, 2007, 43 (3): 245-248.
- [11] 王国胤. Rough 集理论与知识获取 [M]. 西安 西安交通大学出版社, 2001: 93-116.

- [12] 阙 恒. 嵌入式图形处理器设计 [D]. 南京 南京航空航天大学, 2007.
- [13] 周启平, 张 扬, 吴 琼. Vxworks 开发指南与 Tornado 使用手册 [M]. 北京 中国电力出版社, 2004.
- [14] 董英英, 王启峰. 基于 S3C2440 的 WindML 图形驱动设计 [J]. 现代电子技术, 2010 (16): 69-71.
- [15] WindML DDK Programmer's Guide, 3.0 [M]. America: Wind River Systems Inc., 2002.
- [16] 李海亮, 石鹏程. Vxworks 的 WindML 图形界面程序的框架分析 [J]. 工业控制计算机, 2007, 20 (1): 46-49.
- [17] WindML SDK Programmer's Guide, 3.0 [M]. America: Wind River Systems Inc., 2002.
- [18] 赵 俊, 张克环, 李仁发. 嵌入式通用图形加速芯片的研究与设计 [J]. 计算机工程与应用, 2008, 44 (26): 74-76.
- [19] Gamma E, Helm R, Johnson R, et al. Design Patterns: Elements of Reusable Object-oriented Software [M]. USA: Pearson Education, 1995.
- [20] 杨国东. 嵌入式图形处理器设计与实现 [D]. 济南 山东大学, 2010.
- [21] 张继伟. 基于 WindML 环境下的显卡驱动设计 [J]. 现代电子技术, 2010 (14): 78-80.
- [22] 姚宇峰, 邓志杰, 陈光武, 等. Vxworks 图形驱动研究 [J]. 单片机与嵌入式系统应用, 2007 (7): 26-28.

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](#)
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](#)
55. [USB30 的高速信息传输瓶颈研究](#)

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](#)
41. [Android 手机应用开发之音乐资源播放器](#)

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](#)
21. [基于 PowerPC 的车载通信系统设计](#)

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](#)
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](#)
29. [基于 S3C6410 的 Uboot 分析与移植](#)

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](#)

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](#)
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](#)
25. [基于龙芯 2F 架构的 PMON 分析与优化](#)

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](#)