

基于 PCI9656 控制芯片的高速网卡 DMA 设计

The DMA Designing of Fast NIC Based on PCI9656 Chip

(郑州解放军信息工程大学) 罗晓富 李永才 李欧 林白 赵桦

Luo, Xiaofu Li, Yongcai Li, Ou Lin, Bai Zhao, Hua

摘要: 本文介绍了 Windows 环境下 PCI 设备及 DMA 工作原理, 并在分析 PCI9656 控制芯片硬件特性和 DMA 工作原理的基础上通过实例详细描述了 DMA 的编程技术。

关键词: PCI 设备; DMA; 驱动程序

中图分类号: TP302.1 文献标识码: A

文章编号: 1008-0570(2005)04-0054-02

Abstract: This paper presents PCI device and the working principle of DMA under Windows. Based on the analysis of hardware characteristics of PCI9656 control chip, We describe DMA programming technology with representative examples.

KeyWord: PCI device ; DMA ; device driver

1 前言

目前, 基于 PCI 总线开发的设备已成为主流。总线是计算机各模块间进行信息传输的通道, 在实际的系统应用中, 经常需要进行高速的数据处理卡 (如语音通信、视频采集等) 的设计, 它要求较高的传输速率, 这依赖于计算机总线的结构。以前的数据处理卡大多是基于标准 ISA 总线的, 这种 16 位总线结构的最大缺点是传输速率太低 (5MB/S), 无法满足数据的实时高速传输。PCI 总线的出现解决了这个问题, 它是一种独立于处理器的本地总线, 具有 32 位和 64 位两种复用的地址数据通道, 一边与处理器和存储器总线接口, 另一边为外设扩展提供了高速通道。33.3MHz、32 位的 PCI 总线可以实现 132MB/S 的数据传输速率, 64 位的更是能达到 528MB/S。DMA 是一种高速的数据传输操作, 它允许在外部设备和存储器之间直接读写数据, 既不需要 CPU, 也不需要 CPU 的干预。整个数据传输操作在一个称为“DMA 控制器”的控制下进行, 使整个计算机系统效率大大提高。

论文主要针对我们开发的 Gibit 高速语音卡驱动程序的 DMA 编程部分进行了分析, 文中先简要介绍一下我们所采用的 PLX 公司的 PCI9656 芯片的硬件特性, 然后对 DMA 操作实现部分进行了详细的分析。

2 PCI9656 与 DMA

PCI9656 是一 64 位、66MHz PCI 总线主控 I/O 加速器, 主要用于 PowerQUICC(tm) 和通用 32 位、66MHz 局域总线设计。PCI 9656 采用 PLX 业界领先的数据流水线架构 (Data Pipe Architecture) 技术, 配备 DMA 引擎、可编程直接主控和直接从属数据传输模式以及 PCI 信息传输功能。符合 PCI v2.2 规范。零等待状态突发操作, PCI 总线突发传输速度高达 528 MB/S, 局域总线突发传输速度高达 264 MB/S。

DMA (直接内存访问) 是指一种高速的数据传输操作, 以计算机读入数据为例, DMA 是不通过 CPU 而直接写入数据, 它以块的方式传输数据, 只要主机设定本地和 PCI 端的起始地址, 传输字节数和传输方向, 再设置 DMA 传输方式就可以了。PCI9656

本地总线通过 PCI 总线与计算机存储器相连, PCI9656 以突发方式向存储器连续发送本地存储器的数据, 大大提高了数据传输速率。

PCI9656 提供两个独立的链接 DMA 通道, 每个通道有自己的双向 32 Qword (256 字节) 深层 FIFO。

下图是 PCI9656 PowerQUICC ComPactPCI 适配器的设计图, 由此可以看出 PCI9656 与 PCI 总线及本地总线, DMA 传输之间的关系。

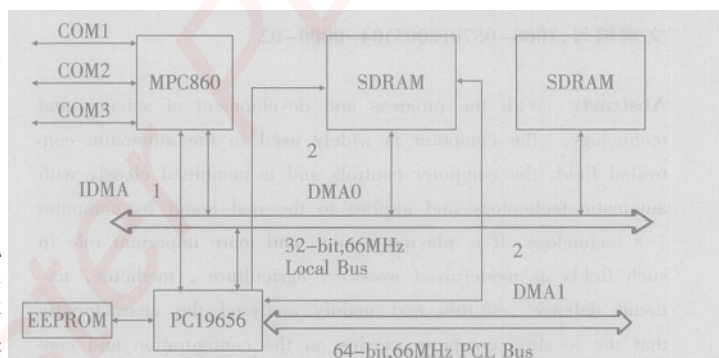


图 1 PCI9656 PowerQUICC ComPactPCI 适配器

3 DMA 传输策略

执行一个 DMA 传输需要考虑下面几个因素:

(1) 如果设备有总线主控能力, 那么它就有访问主存的必要电子部件, 因此我们只需要告诉它几个基本事实, 如从哪开始, 需要传输多少单位的数据, 是输入操作还是输出操作, 等等。你可以向硬件设计者咨询以得到细节部分, 否则你只能参考硬件级的说明文档。

(2) 一个有分散/聚集能力的设备可以在自身与不连续的物理内存区之间传输大块数据。设备的分散/聚集能力对软件十分有利, 它可以避免对具有连续页帧的大数据块的需求。页可以被简单地锁定在所在的物理内存, 然后把内存地址告诉设备。

(3) 如果设备不是总线主控设备, 那么你需要使用计算机主板上的系统 DMA 控制器。这种形式的 DMA 传输有时被称为副 DMA。与 ISA 总线连接的系统 DMA 控制器对所能访问的物理内存和一次传输的数据量会有些限制。EISA 总线的 DMA 控制器去掉了这些限制。在 Windows 2000 及以后系统中, 你不必知道硬件具体插入到哪种类型的总线, 因为系统自动参考这些不同的限定。

(4) 通常, DMA 操作将包括编程硬件映射寄存器或操作前后的数据复制。如果设备需要连续地读写数据, 你不会希望在每次 I/O 请求中都做这两步, 这将大大地降低处理速度, 在某些情况下是不能接受的。因此, 你应该分配一个公用缓冲区, 设备和驱动程序可以在任何时间同时访问这个缓冲区。

我们采用的 9656 芯片执行无分散/聚集的总线主控 DMA 操作, 并且利用了公用缓冲区方法。图二显示了一次传输过程。在 StartIo 例程中, 你通过请求适配器对象的所有权而启动了

DMA传输。当 I/O 管理器同意给你适配器对象的所有权后,它就分配一些映射寄存器为你临时使用,并回调你提供的 AdapterControl 例程。在你的 AdapterControl 例程中,你执行一个"传输映射"步骤来安排传输的第一阶段。如果映射寄存器不够,则需要多个传输阶段;你的设备必须能处理发生在两个阶段间的任何延迟。

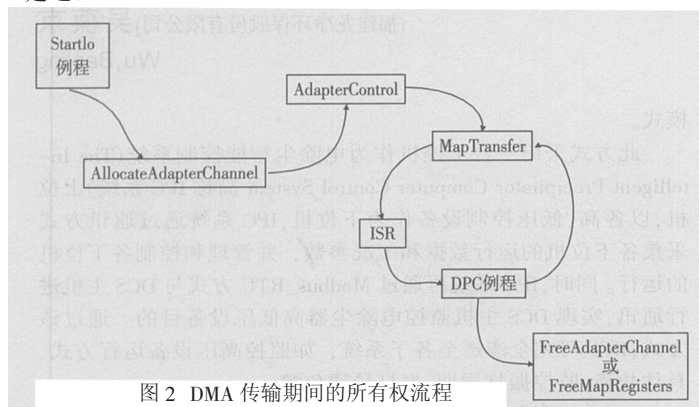


图2 DMA 传输期间的所有权流程

一旦 AdapterControl 例程初始化完第一阶段的映射寄存器,就可以通知设备开始操作。传输完成后设备将产生一个中断,于是系统调度一个 DPC。如果必要,DPC 例程将初始化下一阶段的传输,否则它将完成这个请求。在这个过程的某个时刻,你将释放映射寄存器和适配器对象。

4 执行 DMA 传输

大多数的 DMA 控制器有一个相似的构架。它的最简单的结构是,它有一个 DMA 缓冲区的开始地址和一个纪录传输的数据位字节数或者字数的计数寄存器,当这些寄存器被程序填满和设备开始的话,DMA 控制器开始按照它自己的方式传输数据,随着每次的传输,它增加存储器地址寄存器和减小计数寄存器,当计数寄存器到达 0,DMA 控制器产生一个中断,设备准备传输另一个数据。

在公共缓冲区 DMA 中,设备使用系统空间中的一个单一的非分页缓冲区,所有的 DMA 传输都使用这个缓冲区。硬件认为公共缓冲区是和驱动程序交换控制和状态信息的中介。这里说说公共缓冲区主 DMA 工作原理:

图3示,假定设备使用一个缓冲区作为命令缓冲区,另一个缓冲区作为返回的状态缓冲区。



图3 公共缓冲区信息交换的执行

下面将针对我们的项目中 PCI9656 的 DMA 传输部分几个主要例程作一简要分析。

4.1 分配和释放公共缓冲区

首先要做的就是分配公共缓冲区。公共缓冲区的存储器的物理空间必须是相邻的和在设备的 DMA 逻辑空间的范围内,使用 Adapter 对象的一个方法 AllocateCommonBuffer 函数可以保证这两个条件。

```
(*pdx->AdapterObject->DmaOperations->AllocateCommonBuffer)
```

```
(pdx->AdapterObject, <length>, &pdx->paCommonBuffer, FALSE);
```

除了分配公共缓冲区之外,AllocateCommonBuffer 函数也分配映射寄存器(如果需要的话)和为设备设置一个传输,加载需要的映射寄存器。这样,缓冲区就可以立即的和持续的使用。缓冲区保持可用直到 FreeCommonBuffer 函数被明确的调用,通常在 IRP_MN_STOP_DEVICE 的处理例程中。

```
(*pdx->AdapterObject->DmaOperations->FreeCommonBuffer)
(pdx->AdapterObject, <length>,
pdx->paCommonBuffer,
pdx->vaCommonBuffer, FALSE);
```

4.2 处理 IRP_MN_START_DEVICE

这个处理必须执行下列任务:

4.2.1 调用 IoGetDmaAdapter 函数来定位与设备相关联的 Adapter 对象。

```
pDevExt->pDmaAdapter = IoGetDmaAdapter( pDevObj, &dd,
&pDevExt->mapRegisterCount);
```

```
if (pDevExt->pDmaAdapter == NULL)
```

```
return
```

```
STATUS_INSUFFICIENT_RESOURCES;
```

```
else return STATUS_SUCCESS;
```

4.2.2. 使用 AllocateCommonBuffer 函数来得到驱动程序和设备都可以访问的相邻的非分页的存储空间。

4.2.3. 为后来的使用存储公共缓冲区的虚拟地址到设备 Extension。

4.3 Start I/O 例程

发送一个命令给设备,Start I/O 例程执行下列任务:

4.3.1 在公共缓冲区中创建一个命令结构,使用虚拟地址将它存储在设备 Extension 中。

```
PDEVICE_EXTENSION pdx= (PDEVICE_EXTENSION) fdo->
DeviceExtension;
```

```
PIO_STACK_LOCATION stack =IoGetCurrentIrpStackLocation
(Irp);
```

```
PMDL mdl = Irp->MdlAddress;
```

```
ULONG nbytes = MmGetMdlByteCount(mdl);
```

```
PVOID vaddr = MmGetMdlVirtualAddress(mdl);
```

4.3.2. 如果需要的话(CacheEnabled 参数被设置为 TRUE),就要调用 KeFlushIoBuffers 函数来强制将 CPU 的高速缓冲存储器中的数据输出到物理存储器中。

4.3.3 最后,Start I/O 例程设置在设备的控制寄存器中的一个标志,来通知设备有一个命令在等待。

```
if (! NT_SUCCESS(status))
```

```
{
```

```
IoReleaseRemoveLock(&pdx->RemoveLock, Irp);
```

```
CompleteRequest(Irp, status, 0);
```

```
StartNextPacket(&pdx->dqReadWrite, fdo);
```

```
}
```

为了回应这个通知标志,设备开始处理在命令缓冲区中的命令。

4.4 ISR

当设备完成命令缓冲区中的命令,就放置一个信息到状态缓冲区和产生一个中断。为了响应状态缓冲区中的信息,驱动程序的 ISR 做下列工作:

4.4.1. 复制状态缓冲区中的内容到设备 Extension 的各种各样的域。

4.4.2. 如果需要的话,ISR 在设备控制寄存器中设置另一个标志位来表示已经读取了状态信息。

4.4.3. 调用 IoRequestDpc 函数继续在较低的 IRQL 下处理这个请求。

```
if (pdx->busy) 见第 231 页)
```

用数学形态学的方法能够清晰、连续地检测到储粮害虫的特征边缘,为进一步对储粮害虫的特征提取打下了基础。

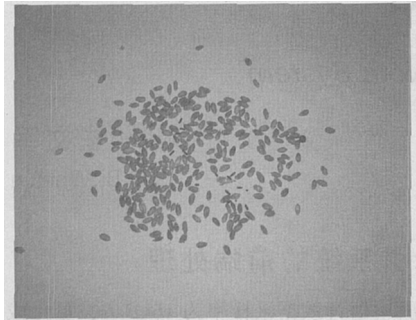


图1 原图像

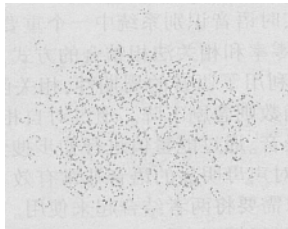


图2 采用 Soble 算子的边缘检测结果

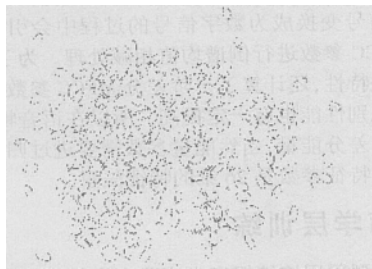


图3 采用 Robert 算子的边缘检测结果

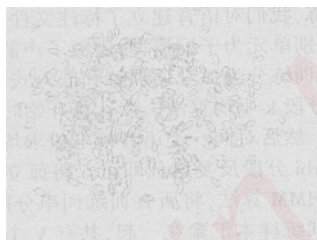


图4 采用数学形态学的边缘检测结果

4 结论

基于图像识别的在线粮虫检测是储粮虫情测报的发展趋势,粮虫图像的边缘检测算法是识别的关键。本文从数学形态学的基本思想出发,将其应用于储粮害虫图像的边缘检测,经算法研究表明,该算法是一个简单、实用、有效的边缘检测算法,为后续的图像特征提取、模式识别提供了良好的基础。

参考文献:

- [1]万拯群,当前我国科学保粮问题之我见[J],粮食储藏,1999,28(1):25-29。
- [2]Wilkin D R,Fleural-Lessard F,The detection of insects in grain using conventional samplings spears [C]. Proc.5th Int.Wkg.Conf. On Stored-Product Protection, Bordeaux, France,1990,1445-1450。
- [3]姚渭等.粮虫陷阱检测技术研究及应用,中国粮油学会首届学术年会论文集(储藏专业卷)[C].石家庄,2000:7-16。

detection of insect peats in cereal grains [C],Proc.Int.Conf.on Diffuse Spectroscopy,MD USA, 1992, 96-100.

[5]郭敏,尚志远.储粮害虫声信号的检测和应用[J],物理学和高新技术,2001,30(1):39-42.

[6]徐舫,邱道尹等.图像识别在粮仓害虫检测方面的应用研究[J].郑州工程学院学报,2001,22(1):78-81。

[7]崔屹.图像处理与分析——数学形态学方法及应用[M].北京:科学出版社,2000:15-28。

作者简介:周龙,男,1970年生,湖北省武汉市,博士,副教授,从事电气控制技术、模式识别、神经网络和灰色理论及其应用方面的研究工作。电话:(027)62639631 或 83913547,Email: zhou-long123@sohu.com

(430023 湖北武汉汉口常青花园武汉工业学院电气信息工程系)周 龙

(Wuhan Polytechnic University, Wuhan 430023, P. R. China) Zhou,Long (收稿日期:2005.1.4)

(接第 55 页)

```
{
    pdx->busy=FALSE;
    IoRequestDpc(pdx->DeviceObject, NULL, pdx);
}
```

4.5 处理 IRP_MN_STOP_DEVICE

当设备停止,驱动程序应该使设备不再活动。使用 FreeCommonBuffer 函数来释放分配的缓冲区。

5 结束语

毋庸置疑,微软的垄断地位使得开发适应于其应用系统的设备驱动程序变得越来越重要,而 PCI 设备及 DMA 传送是其中的主流。本文通过实例简要介绍了 Windows 操作系统中 PCI 设备的工作原理并详细描述了 PCI 设备中的 DMA 操作的驱动程序编程技术,对设备驱动程序的编写具有一定的参考价值。

参考文献

- [1]马鸣景,赵秋霞,曾光裕.高性能个人计算机硬件结构及接口。国防工业出版社,2001
- [2]Art Baker,Jerry Lozano.The Windows 2000 Device Driver Book. Second Edition, 2000
- [3]Walter Oney.Programming The Microsoft Windows Driver Model. Second Edition,2003
- [4]PCI 9656 Product Brief ,9656-SIL-PB-P1-1.0

作者简介:罗晓富(1975-),硕士生,男,汉,江西吉安,主要研究方向:网络安全与应用;电话:13383825926,E-mail:longriver2004@hotmail.com;李永才(1975-),本科,男,汉,河南南阳,研究员,主要研究方向:计算机网络与安全;李鸥(1963-),博士,男,汉,河南郑州,教授,主要研究方向:宽带网络应用和网络安全;林白(1979-),硕士生,女,汉,山东济南,主要研究方向:网络安全;赵桦(1979-),硕士生,女,汉,北京,主要研究方向:网络安全。

(450002 郑州解放军信息工程大学) 罗晓富 李永才 李欧 林白 赵桦

(College of Information Engineering, PLA ,ZhengZhou 450002) Luo,Xiaofu Li,Yongcai Li,Ou Lin,Bai Zhao,Hua 通信地址:

(450002 郑州 1001 信箱 825 分箱中国人民解放军信息工程学院通信工程系研究生队)罗晓富

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](#)
56. [基于 IPv6 的 UDP 通信的实现](#)
57. [一种基于 IPv6 的流媒体传送方案研究与实现](#)
58. [基于 IPv4-IPv6 双栈的 MODBUS-TCP 协议实现](#)
59. [RS485CAN 网关设计与实现](#)
60. [MVB 周期信息的实时调度](#)
61. [RS485 和 PROFINET 网关设计](#)
62. [基于 IPv6 的 Socket 通信的实现](#)
63. [MVB 网络重复器的设计](#)
64. [一种新型 MVB 通信板的探究](#)
65. [具有 MVB 接口的输入输出设备的分析](#)
66. [基于 STM32 的 GSM 模块综合应用](#)
67. [基于 ARM7 的 MVB CAN 网关设计](#)
68. [机车车辆的 MVB CAN 总线网关设计](#)
69. [智能变电站冗余网络中 IEEE1588 协议的应用](#)
70. [CAN 总线的浅析 CANopen 协议](#)
71. [基于 CANopen 协议实现多电机系统实时控制](#)
72. [以太网时钟同步协议的研究](#)
73. [基于 CANopen 的列车通信网络实现研究](#)
74. [基于 SJA1000 的 CAN 总线智能控制系统设计](#)
75. [基于 CANopen 的运动控制单元的设计](#)
76. [基于 STM32F107VC 的 IEEE 1588 精密时钟同步分析与实现](#)

77. [分布式控制系统精确时钟同步技术](#)
78. [基于 IEEE 1588 的时钟同步技术在分布式系统中应用](#)
79. [基于 SJA1000 的 CAN 总线通讯模块的实现](#)
80. [嵌入式设备的精确时钟同步技术的研究与实现](#)
81. [基于 SJA1000 的 CAN 网桥设计](#)
82. [基于 CAN 总线分布式温室监控系统的设计与实现](#)
83. [基于 DSP 的 CANopen 通讯协议的实现](#)

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](#)
35. [基于 GPU FPGA 芯片原型的 VxWorks 下驱动软件开发](#)
36. [VxWorks 下的多串口卡设计](#)
37. [VxWorks 内存管理机制的研究](#)
38. [T9 输入法在 Tilcon 下的实现](#)
39. [基于 VxWorks 的 WindML 图形界面开发方法](#)
40. [基于 Tilcon 的 IO 控制板可视化测试软件的设计和实现](#)
41. [基于 VxWorks 的通信服务器实时多任务软件设计](#)
42. [基于 VXWORKS 的 RS485MVB 网关的设计与实现](#)
43. [实时操作系统 VxWorks 在微机保护中的应用](#)
44. [基于 VxWorks 的多任务程序设计及通信管理](#)
45. [基于 Tilcon 的 VxWorks 图形界面开发技术](#)
46. [嵌入式图形系统 Tilcon 及应用研究](#)
47. [基于 VxWorks 的数据采集与重演软件的图形界面的设计与实现](#)
48. [基于嵌入式的 Tilcon 用户图形界面设计与开发](#)
49. [基于 Tilcon 的交互式多页面的设计](#)
50. [基于 Tilcon 的嵌入式系统人机界面开发技术](#)
51. [基于 Tilcon 的指控系统多任务人机交互软件设计](#)
52. [基于 Tilcon 航海标绘台界面设计](#)
53. [基于 Tornado 和 Tilcon 的嵌入式 GIS 图形编辑软件的开发](#)
54. [VxWorks 环境下内存文件系统的应用](#)
55. [VxWorks 下的多重定时器设计](#)
56. [Freescale 的 MPC8641D 的 VxWorks BSP](#)
57. [VxWorks 实验五\[时间片轮转调度\]](#)
58. [解决 VmWare 下下载大型工程.out 出现 WTX Error 0x100de 的问题](#)
- 59.

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](#)
42. [Linux 下以太网的 IPv6 隧道技术的实现](#)
43. [Research and design of mobile learning platform based on Android](#)
44. [基于 linux 和 Qt 的串口通信调试器调的设计及应用](#)
45. [在 Linux 平台上基于 QT 的动态图像采集系统的设计](#)
46. [基于 Android 平台的医护查房系统的研究与设计](#)
47. [基于 Android 平台的软件自动化监控工具的设计开发](#)

48. [基于 Android 的视频软硬解码及渲染的对比研究与实现](#)
49. [基于 Android 移动设备的加速度传感器技术研究](#)
50. [基于 Android 系统振动测试仪研究](#)
51. [基于缓存竞争优化的 Linux 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](#)
21. [DCOM 协议在网络冗余环境下的应用](#)
22. [Windows XP Embedded 在变电站通信管理机中的应用](#)
23. [XPE 在多功能显控台上的开发与应用](#)
24. [基于 Windows XP Embedded 的 LKJ2000 仿真系统设计与实现](#)
25. [虚拟仪器的 Windows XP Embedded 操作系统开发](#)
26. [基于 EVC 的嵌入式导航电子地图设计](#)
27. [基于 XPEmbedded 的警务区 SMS 指挥平台的设计与实现](#)
28. [基于 XPE 的数字残币兑换工具开发](#)
29. [Windows CENET 下 ADC 驱动开发设计](#)
30. [Windows CE 下 USB 设备流驱动开发与设计](#)
31. [Windows 驱动程序设计](#)

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](#)
22. [基于 PowerPC 的嵌入式系统中通用 I/O 口的扩展方法](#)
23. [基于 PowerPC440GP 型微控制器的嵌入式系统设计与研究](#)
24. [基于双 PowerPC 7447A 处理器的嵌入式系统硬件设计](#)
25. [基于 PowerPC603e 通用处理模块的设计与实现](#)
26. [嵌入式微机 MPC555 驻留片内监控器的开发与实现](#)
27. [基于 PowerPC 和 DSP 的电能质量在线监测装置的研制](#)
28. [基于 PowerPC 架构多核处理器嵌入式系统硬件设计](#)
29. [基于 PowerPC 的多屏系统设计](#)
30. [基于 PowerPC 的嵌入式 SMP 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](#)
30. [基于 ARM 嵌入式系统的高保真无损音乐播放器设计](#)
31. [UBoot 在 Mini6410 上的移植](#)
32. [基于 ARM11 的嵌入式 Linux NAND FLASH 模拟 U 盘挂载分析与实现](#)
33. [基于 ARM11 的电源完整性分析](#)
34. [基于 ARM S3C6410 的 uboot 分析与移植](#)
35. [基于 S5PC100 移动视频监控终端的设计与实现](#)

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](#)
26. [CPU 与 GPU 之间接口电路的设计与实现](#)
27. [基于龙芯 1A 平台的 PMON 源码编译和启动分析](#)
28. [基于 PC104 工控机的嵌入式直流监控装置的设计](#)
29. [GPGPU 技术研究与实现](#)
30. [GPU 实现的高速 FIR 数字滤波算法](#)
31. [一种基于 CPUGPU 异构计算的混合编程模型](#)
32. [面向 OpenCL 模型的 GPU 性能优化](#)
33. [基于 GPU 的 FDTD 算法](#)
34. [基于 GPU 的瑕疵检测](#)
35. [基于 GPU 通用计算的分析与研究](#)
36. [面向 OpenCL 架构的 GPGPU 量化性能模型](#)
37. [基于 OpenCL 的图像积分图算法优化研究](#)
38. [基于 OpenCL 的均值平移算法在多个众核平台的性能优化研究](#)
39. [基于 OpenCL 的异构系统并行编程](#)
40. [嵌入式系统中热备份双机切换技术研究](#)

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](#)
7. [数据结构考题 - 第 1 章 绪论](#)
8. [数据结构考题 - 第 2 章 线性表](#)
9. [数据结构考题 - 第 2 章 线性表 - 答案](#)
10. [基于小波变换与偏微分方程的图像分解及边缘检测](#)
11. [基于图像能量的布匹瑕疵检测方法](#)
12. [基于 OpenCL 的拉普拉斯图像增强算法优化研究](#)
13. [异构平台上基于 OpenCL 的 FFT 实现与优化](#)
14. [数据结构考题 - 第 4 章 串](#)
15. [数据结构考题 - 第 4 章 串答案](#)
- 16.

FPGA / CPLD:

1. [一种基于并行处理器的快速车道线检测系统及 FPGA 实现](#)
2. [基于 FPGA 和 DSP 的 DBF 实现](#)
3. [高速浮点运算单元的 FPGA 实现](#)
4. [DLMS 算法的脉动阵结构设计及 FPGA 实现](#)
5. [一种基于 FPGA 的 3DES 加密算法实现](#)
6. [可编程 FIR 滤波器的 FPGA 实现](#)
7. [基于 FPGA 的 AES 加密算法的高速实现](#)
8. [基于 FPGA 的精确时钟同步方法](#)
9. [应用分布式算法在 FPGA 平台实现 FIR 低通滤波器](#)
10. [流水线技术在用 FPGA 实现高速 DSP 运算中的应用](#)
11. [基于 FPGA 的 CAN 总线通信节点设计](#)
12. [基于 FPGA 的高速时钟数据恢复电路的实现](#)
13. [基于 FPGA 的高阶高速 FIR 滤波器设计与实现](#)
14. [基于 FPGA 高效实现 FIR 滤波器的研究](#)
- 15.

