

CPU 与 GPU 之间接口电路的设计与实现

石茉莉¹, 蒋 林², 刘有耀²

(1 西安邮电大学 计算机学院, 陕西 西安 710061; 2 西安邮电大学 电子工程学院, 陕西 西安 710061)

摘 要: 在构建 CPU(Central Process Unit, CPU) 与 GPU(Graphic Process Unit) 或者 CPU 与其它设备协同计算的过程中, 通过 PCI(Peripheral Component Interconnect) 总线将 GPU 等其他设备连接至 CPU, 承担并行计算的任务. 为了解决 PCI 接口芯片与 GPU 芯片之间的异步传输和时序匹配问题, 基于 PCI 总线规范与 GPU 芯片的时序规范, 采用跨时钟域信号的处理方法, 设计了一个 CPU 与 GPU 之间跨时钟域连接的时序匹配接口电路. 通过仿真, 验证了该电路的正确性. 结果表明, 该电路可工作在 252 MHz 频率下, 能够满足 GPU 与 CPU 间接口电路对速率和带宽的要求.

关键词: PCI 总线; GPU; CPU; 总线仲裁; 异构计算

中图分类号: TP302 **文献标识码:** A **文章编号:** 1000-7180(2013)11-0023-04

Design and Implementation of Interface Circuit Between CPU and GPU

SHI Mo-li¹, JIANG Lin², LIU You-yao²

(1 School of Computer Science, Xi'an University of Posts and Telecommunications, Xi'an 710061, China;

2 School of Electronic Engineering, Xi'an University of Posts and Telecommunications, Xi'an 710061, China)

Abstract: During constructing the Collaborative Computing between Central Process Unit and Graphic Process Unit or Central Process Unit and other device. Through the Peripheral Component Interconnect connects Graphic Process Unit and Central Process Unit, it's responsible for doing the parallel computing. Point at the asynchronous transmission and timing matched in the connection of Peripheral Component Interconnect IP core and the Graphic Process Unit, based on the standard of the Peripheral Component Interconnect and the Graphic Process Unit chip, use the method of processing asynchronous signals, this paper design an timing matched interface circuit between the Central Process Unit and Graphic Process Unit, which aims at the different clock systems and the timing matched of them. The simulation results prove that the interface circuit between Central Process Unit and Graphic Process Unit can works at 252 MHz frequency, it achieves the circuit demand, and realize high-speed data transmission between Graphic Process Unit and Central Process Unit.

Key words: Peripheral Component Interconnect; Graphic Process Unit; Central Process Unit; Bus Arbitration; heterogeneous computing

1 引言

近年来, 利用专用处理器作为通用处理器的协处理器, 组成异构并行系统, 已成为目前研究的热点之一. 构建 CPU-GPU 异构系统成为不可逆转的趋势^[1]. 在实际应用中, 基于 GPU+CPU 的协作处理方法体现了高效率、低功耗、小成本的特点^[2]. 同时,

在 GPU-CPU 异构系统中, 对数据传输的带宽和速率也提出一定的要求. 采用 PCI 总线进行数据传输^[3-4], 可以解决原有的标准总线数据传输率低带来的瓶颈问题.

针对工作在异步模式下, 采用 PCI 核接口芯片外部独立 DMA 控制器的 mpci32, 其使用过程中存在的关键问题是接口时序匹配和跨时钟域信号的处理

理^[5-6],本文设计了基于 mpci32 的 CPU 与 GPU 之间的接口电路,相对于传统设计^[7],该电路实现了采用片外独立 DMA 控制器的 PCI 核接口芯片与 GPU 之间的异步传输和接口时序匹配。

2 mpci32 软核功能概述

mpci32 作为一种连接 PCI 外围设备的软核,实现了 PCI 总线与 PCI 的主/从设备连接并正确传送数据的功能.它可以专用为一个目标设备接口,也可以配置成一个拥有双口 FIFO 的主通道和一个选择性连接的 DMA 控制器接口。

mpci32 的目标通道作为 PCI 核的一个从设备的接口,主要负责处理所有由 PCI 总线主设备发起的操作.从接口主要包含一个目标数据通道和对应的控制逻辑.目标通道数据传输部分的接口分为寄存器类型接口和存储器类型接口,寄存器类型接口主要用于对数据吞吐量要求不高的单数据传输,存储器类型接口用于高数据速率的传输,通常都是线性突发传输的形式。

mpci32 的主通道主要负责处理 mpci32 核的客户端作为主设备时的一些操作.主通道功能的实现由获取数据操作(后端从 PCI 核中读取数据)、存储数据操作(后端向 PCI 核中写数据)、PCI 核对错误的处理三个功能块完成的。

在本文接口电路的设计中,mpci32 分别工作在主从两种模式。

3 接口电路功能定义与方案设计

3.1 CPU 与 GPU 接口电路的功能定义

根据图形处理对数据传输速率的要求,本文采用了 PCI 总线进行了 PCI 后端接口电路的设计.电路主要实现的功能有:

- (1)GPU 系统时钟(250 MHz)与 PCI 时钟(33 MHz/66 MHz)之间的跨时钟域传输;
- (2)外围设备作为从设备时的时序匹配电路,完成 PCI 与 GPU 之间少量数据的传输;
- (3)外围设备作为主设备时的时序匹配电路,完成 PCI 与 GPU 之间大批量数据的传输。

接口电路结构框图如图 1 所示。

要实现以上功能,mpci32 分别充当了主设备和从设备的角色.当 mpci32 作为主设备时,完成 GPU 中 DMA 和命令处理器分别与 PCI 总线间的通信.当 GPU 作为目标设备时,被动的接收由 PCI 发起的操作,完成 PCI 对 GPU 各级流水线中用户可访

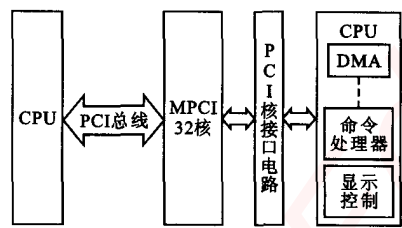


图 1 接口电路结构框图

问寄存器的配置。

3.2 接口电路模块划分

根据 PCI 核后端接口电路要实现的功能,后端接口电路的设计分为 3 个功能模块:BusArbiter 模块、MUX 模块、Configure 模块.各个模块具体实现的功能:

- (1)BusArbiter 模块:完成对总线使用权的仲裁.当 DMA、命令处理器同时请求占用总线时,仲裁器通过对其进行仲裁,决定总线使用权从属的设备。
- (2)MUX 模块:根据 BusArbiter 模块的选择信号,选通相应的数据通道,分别建立 DMA、命令处理器与 PCI 核的连接,完成数据传输。
- (3)Configure 模块:实现 PCI 总线与 GPU 之间少量数据的传输,完成 GPU 显示控制等模块中用户访问寄存器的配置。

PCI 后端逻辑的设计框图如图 2 所示。

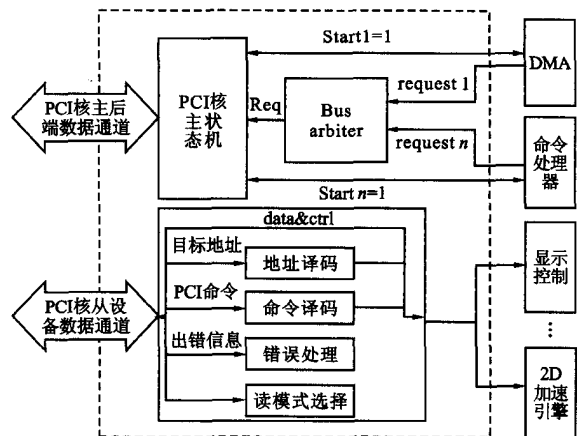


图 2 PCI 核接口电路的设计框图

4 接口电路各功能部件的设计

4.1 BusArbiter 模块

(1)算法概述

为了合理地控制和管理 DMA、命令处理器对总线的占用,依据仲裁算法判决出哪个设备应获得总线控制权.确保任何时刻总线上只有一个主设备发送信息。

总线仲裁优先级算法通常有：固定优先级算法和循环优先级算法^[8-9]；在实际应用中，根据需求，常将两种仲裁算法结合使用。

在 mpci32 与 GPU 之间的时序匹配接口电路设计中，由于 GPU 中的 DMA、命令处理器并没有轻重主次之分，所以采用循环优先级算法。

总线的缺省占用^[10]：本次设计中，由于两个主设备地位平等，没有主次之分，采用最后一次使用总线的设备，为总线缺省拥有者。

(2) 算法设计与实现

在本次设计中用到的是循环优先级算法。由于只有 DMA、命令处理器可以作为主设备，所以用一个状态标志信号 arb_flag 来完成循环优先级仲裁算法的仲裁标志。当 arb_flag=1 时，表示此次选通的是 DMA；如果 arb_flag=0，表示选通的是命令处理器。仲裁算法实现流程图如图 3 所示。

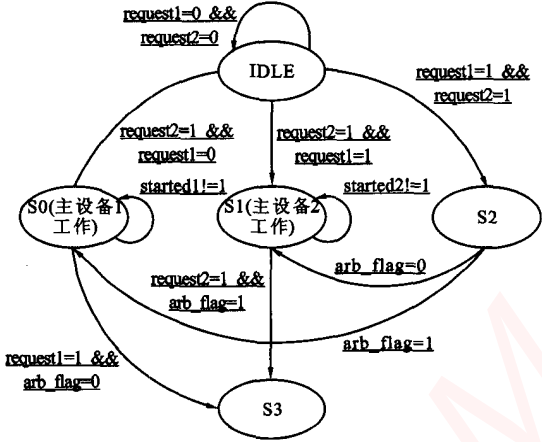


图 3 仲裁算法实现流程图

4.2 MUX 模块

由于在 mpci32 后端时序匹配接口电路的设计中，当 GPU 作为主设备用时，时序匹配接口电路要执行的任务是分别完成 DMA、命令处理器和 mpci32 核之间时序的完全匹配，协调好二者之间的握手信号，同时完成数据在两个模块之间的异步传输。

设计中需要一个命令译码单元，在仲裁判断结束后，后端接口电路根据主设备发出的命令码，判断此次操作是写操作还是读操作，并接收相应的读写控制信号，在写操作的条件下启动后端接口电路中的 Tx_fifo，读操作的条件下启动 Rx_fifo。MUX 模块建立相应数据通路框图如图 4 所示。

后端接口电路要解决的关键问题是握手信号及数据的异步传输问题。为了避免亚稳态效应，后端接口电路对异步信号的处理主要分为：单比特信号的异步处理和多比特信号的异步处理。处理方法结合

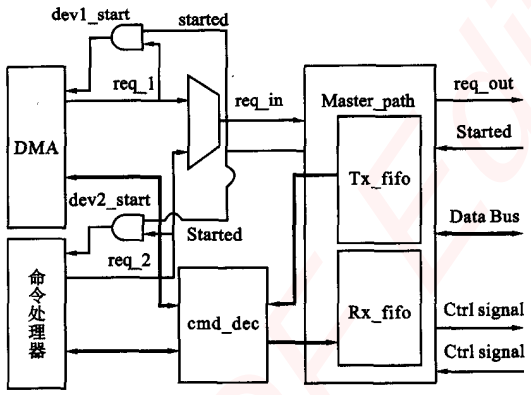


图 4 MUX 模块建立相应数据通路框图

GPU 中具体应用，并参考相应资料，可以完成数据的跨时钟域传输。

4.3 Configure 模块

根据该模块的功能，其具体实现可以通过命令译码单元、地址译码单元及目标通道数据传输模块来完成。

命令译码单元：对 PCI 核送来的 PCI 总线命令码进行译码，决定在此次传输中所用的传输模式。区分读写模式只要看命令码最低位即可，最低位为 0 表示读命令，为 1 表示写命令。在 mpci32 与 GPU 之间的接口时序匹配电路中，应用的都是存储器读写命令。对于存储器读写命令，所有设备都要检查地址位 1 和 0 的状态，以确定在每个数据段结束更新地址计数器时所采用的策略，即线性突发模式和 Cache 行打包模式。

地址译码单元：根据 GPU 中流水线的级数及所用寄存器个数，确定所需的寄存器地址位宽，并进行统一编码。地址译码单元根据 PCI 核传送的地址码进行译码，并选通相应的通道，完成对各个模块寄存器的配置。

对各级寄存器的复位一般分为软复位和硬复位。一般硬复位对流水线各级进行统一复位。如果对流水线各级寄存器的复位是相互独立，就要考虑用软复位来实现相应的复位操作，根据需要进行复位的模块，定义一个寄存器，不同位代表对应模块的复位状态，然后通过软件来实现对不同模块的复位。

5 结果及分析

在 PCI IP 核后端接口电路的设计中，主要实现的是 GPU 分别作为主设备和目标设备的两种工作模式。用 verilog 硬件语言对电路进行了描述，采用 Questsim 进行了功能级仿真，用 VCS 对设计进行了综合，并在 Xilinx Vertex6 760 开发板进行了 FP-

匹配,完成了二者之间的异步数据传输。

针对 GPU 作为主设备和作为目标设备时的读写情况进行了验证。图 5 中可以看出 DMA 在发起读操作之前传输的一些单比特及多比特控制信号的处理,接口电路能够正确的传输 PCI 核与 GPU 之间的控制信号。

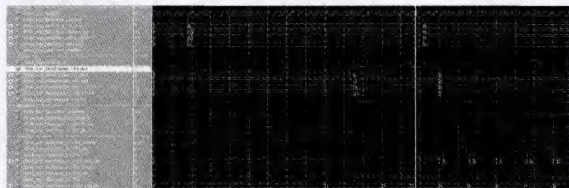


图 5 DMA 控制信号的传输

图 6 所示为 GPU 中 DMA 连续两次向 PCI 发起读请求的操作。时序图证明后端接口时序匹配电路可以准确无误的接收 PCI 经 PCI IP 核发送的数据,并以符合规范要求的时序把数据传送至 DMA。

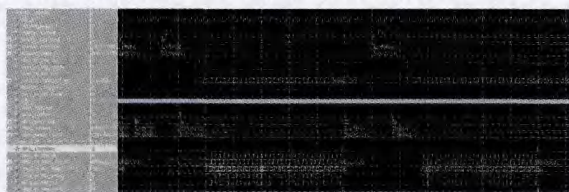


图 6 DMA 连续两次读操作

对与 GPU 作为主设备时的写操作与读操作处理方法基本相同,经验证可知,后端接口电路可以完成 GPU 作为主设备时向 PCI 总线发起的写操作。

GPU 作为目标设备时,后端时序匹配电路主要完成 CPU 对 GPU 中用户层可访问寄存器的配置读写时序的匹配和寄存器数据的异步传输。验证结果如图 7 所示。



图 7 用户层可访问寄存器读写时序

该电路的综合结果如图 8 所示。通过综合结果可以看出,电路的频率可以满足 GPU-CPU 接口电路的要求。

6 结束语

本文基于 mpci32 核,设计并实现了 mpci32 与

Speed Grade: -1

Minimum period: 3.96ns(Maximum Frequency: 252.270MHz)
Minimum input arrival time before clock: 3.501ns
Minimum output required time after clock: 4.185ns
Minimum combinational path delay: 1.227ns

图 8 接口电路综合结果

GPU 之间异步传输模式的接口时序匹配电路,通过仿真验证,mpci32 后端接口时序匹配电路能够实时、准确的完成 GPU 与 CPU 数据传输过程中时序的匹配及跨时钟域传输。

参考文献:

- [1] 徐新海,杨学军,林宇斐,等.一种面向 CPU-GPU 异构系统的容错方法[J].软件学报,2011,22(10):2358-2552.
- [2] 葛万成. CUDA 并行计算的应用研究[J]. 信息技术, 2010,20(4):11-15.
- [3] 张华,胡修林.基于 PCI9656 的高速实时采集存储系统[J]. 计算机应用,2010,11(30):3130-3133.
- [4] Tom S, Don A. PCI system architecture[M]. 4 版. 北京:电子工业出版社,2000.
- [5] 魏堃. 跨时钟域信号同步技术研究[D]. 西安:西安电子科技大学,2009.
- [6] Antonyuk M, Lobur M. Problems of synchronization of the PCI devices for the embedded systems in the real time[C]// IEEE Perspective Technologies and Methods in MEMS Design. Ukraine:IEEE Conference Publications, 2006.
- [7] 李萍,夏雯雯.基于 PCI 软“核”的 PCI 总线接口设计与实现[J]. 计算机应用,2001(11):24-26.
- [8] 王良清,沈绪榜. PCI 总线分组循环仲裁算法的实现[J]. 微电子学与计算机,2002,19(1):1-4.
- [9] 刘海华,陈心浩. PCI 总线加权优先循环仲裁算法[J]. 计算机工程与应用,2003(36):45-47.
- [10] 李贵山,戚德虎. PCI 局部总线开发者指南[M]. 西安:西安电子科技大学出版社,1996.

作者简介:

石茉莉 女,(1985-),硕士研究生.研究方向为专用集成电路设计.

蒋林 男,(1970-),博士,教授,硕士生导师.研究方向为计算机应用技术和电路与系统.

刘有耀 男,(1975-),博士,副教授,硕士生导师.研究方向为集成电路设计与验证.

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 USB3.0 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](#)