基于 SOPC基本信号产生器的设计与实现

倪　亮[1]，吴丽敏[2]，赵鹏飞[1]

(1. 西安电子科技大学 电子工程学院，陕西 西安　710071；2. 西安电子科技大学 技术物理学院，陕西 西安　710171)

摘　要　介绍一种基于 SOPC的基本信号产生器的设计技术，以 Altera公司 EP1C6Q240C8为硬件核心，把软核 CPU嵌入到 FPGA之中构成片上系统 (SOPC)，并结合存储电路、高速 DAC电路、LCD电路、键盘电路、JTAG配置电路以及电源电路等进行了硬件电路的设计，以此实现基本信号产生器。阐述了各主要模块设计方案，并给出软硬件测试图。通过示波器观察，满足了系统设计要求，达到预期目标。

关键词　FPGA；SOPC；Nios II；DDS；基本信号产生器

中图分类号　TN79　　**文献标识码**　A　　**文章编号**　1007 - 7820(2011)01 - 089 - 04

The Design and Realization of the Basic Signal Generator Based on SOPC

Ni Liang[1]，Wu Limin[2]，Zhao Pengfei[1]

(1. School of Electronic Engineering，Xidian University，Xi'an 710071，China；

2. School of Technical Physics，Xidian University，Xi'an 710071，China)

Abstract　This paper introduces a basic signal generator based on the SOPC design technology. In order to achieve the basic signal generator，the design of system hardware circuit takes EP1C6Q240C8of Altera Corporation as the hardware core，embeds the soft-core CPU into FPGA-chip，and unifies the memory circuit，high-speed DAC circuit，LCD circuit，the keyboard circuit，JTAG configuration circuit，the power circuit and so on. The paper describes the main module design and gives some pictures for hardware and software test. Through the oscilloscope observation，system design requirements are met and the desired objectives are achieved.

Keywords　FPGA；SOPC；Nios II；DDS；basic signal generator

SOPC是以 PLD取代 ASIC，更加灵活、高效的 SOC解决方案。SOPC的设计是通过以 IP核为基础、以硬件描述语言为主的设计手段，并借助于以计算机为平台的 EDA工具进行的。它代表一种新型的系统设计技术，也是一种软硬件协同设计技术。可以方便地将硬件系统与常规软件集成在单一可编程芯片中。它可编程的灵活性和 IP设计的重用性保证了产品的差异性，并缩短面市时间，也无需库存和一次性投片费用，降低了投资风险。所以相对于 ASIC具有独特的优势，与 ASIC一起形成共存互补的局面[1]。

1　系统设计方案

本系统采用以 EP1C6Q240C8为核心的设计方案，如图 1所示。

收稿日期：2010-07-24

作者简介：倪亮 (1988 -)，男，硕士研究生。研究方向：电子对抗。吴丽敏 (1987 -)，女，硕士研究生。研究方向：材料科学与工程。赵鹏飞 (1987 -)，男，硕士研究生。研究方向：电子对抗。

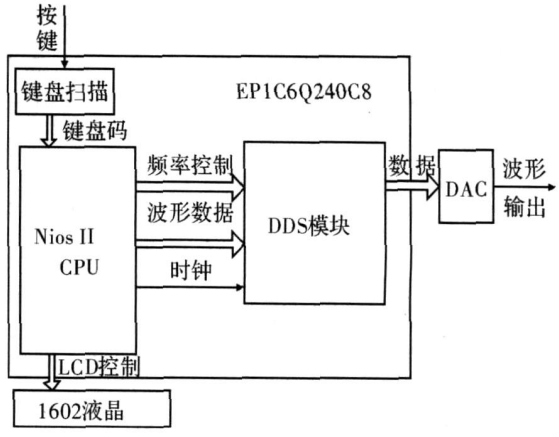

图 1　基本信号产生器系统结构图

方案利用了 FPGA优秀的集成特性，把 Nios II CPU模块、DDS模块、4 ×4键盘扫描模块等集成在 FPGA上实现，外部只接少量的电源模块、DAC模块以及其他输入输出设备。把传统的完全基于硬件的大部分工作转换成在 PC机上通过软件设计编程来实现，减小了系统设计的复杂性[2]。

工作原理如图 1所示。外接 4 ×4键盘根据 1602液晶显示，通过 FPGA的键盘扫描模块向 Nios II CPU

发送键盘扫描码，Nios IICPU根据接收到的扫描码产生相应的信号数据以及控制信号，并通过 PIO传送给FPGA中的 DDS模块，之后 DAC器件将 DDS产生的8位信号数据进行数模转换，从而产生任意频率的方波、三角波、正弦波。

0	1	2	3
4	5	6	7
8	9	Hz	kHz
MHz	方波	三角	正弦

图 2 键盘功能分配

2 系统实现

本系统实现主要分 3个层次：电路板级设计、FPGA硬件设计以及 Nios II软件程序设计。

2.1 电路板级

在电路板级设计中，采用 Altera公司的 EP1C6Q240C8作为设计核心，如图 3所示。由于FPGA配置数据掉电后会丢失，所以需要另外搭配一个配置芯片。EPCS1是 Altera的专用配置芯片，专门用于存贮对 FPGA的配置数据，以保证在 FPGA掉电后还能够保存配置信息，再次上电时 FPGA芯片会自动从 EPCS1中读取数据进行配置[3]。

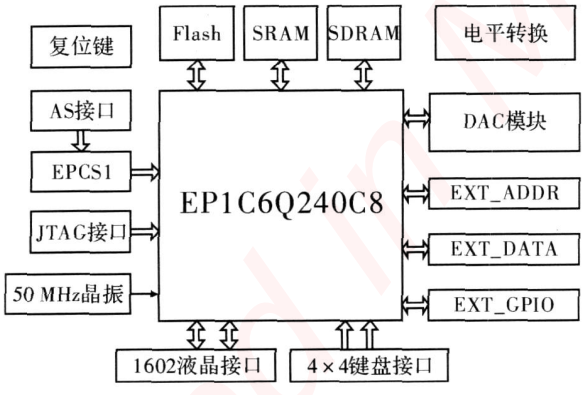

图 3 电路板结构图

为了便于功能更新以及扩展，在 FPGA外加上 Flash、SRAM和 SDRAM作为 FPGA的程序和数据存储器的扩展，地址线通过 EXT_ADDR引出，数据线通过 EXT_DATA引出，增加电路的扩展性。

FPGA中 DDS模块的双口 RAM中输出的数据为8位数字信号，只有通过 DAC转换电路才能将数据转换成相应的模拟信号。综合分辨力、转换速度以及接口方式等要求，本设计采用 ADI公司的 AD9708作为系统 DAC器件。AD9708的数据线和时钟线与FPGA的 I/O脚连接。AD9708的数字地和模拟地在片内是独立的，应通过外部引脚将其连接在一起。同样，模拟电源和数字电源在内部也是独立的，为了减少来自数字电源的噪声，可在模拟电源输入端串联一个磁珠再与数字电源连在一起。

2.2 FPGA硬件设计

FPGA硬件设计是建立在电路板设计基础上的对FPGA芯片功能的设计，将一些可以在电路板上实现的功能在 FPGA内部通过采用硬件描述语言或搭建模块的方式来实现，减少了上层设计的工作量以及系统硬件的风险。通常本层设计是通过通用计算机平台上的可视化编程软件实现的，本设计采用 Altera公司的 Quartus II 8.1系列设计工具[4]。

2.2.1 DDS模块设计

如图 4所示，频率控制字锁存器保存频率设置字 M。双口 RAM的写地址、写数据以及写使能端口完成对 RAM中 1 024 Byte数据的更新，N位累加器输出结果的高 10位作为双口 RAM的读地址。在系统时钟 fclk的作用下累加器根据频率控制字 M输出连续变化或跳跃变化的地址，双口 RAM循环输出相应地址单元中的 8位数据，此 8位数据接到 DAC输入口。假设双口 RAM中存放一个周期的正弦信号数据，那么此时 DAC输出的正弦信号的频率 $f_{out} = f_{clk} \times M / 2^N$，同理，当双口 RAM中存放的是方波或者三角波数据时，DAC也会输出相应频率的信号。

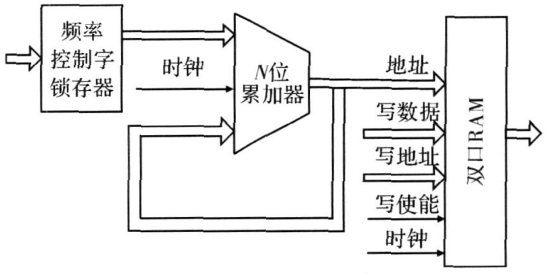

图 4 DDS模块原理图

模块中 32位频率控制字锁存器，是用 Verilog HDL语言实现的，并生成自定义模块以供上层原理图调用。N位累加器和双口 RAM是利用 Quartus II 8.1中的 MegaWizard Plug-In manager定制实现。

2.2.2 定制 Nios IICPU

32位的 Nios II软核是该基本信号发生器的核心模块，主要用于人机界面的控制、键盘值的读取以及控制 DDS模块输出信号的频率和样式。

如图 1所示，CPU与外围设备之间要添加相应的外围接口，通过 Avalon总线与相关部件相连，通过 Avalon的读写时序对各个设备进行操作。在 SOPC Builder中可以提供众多 IP核，通过定制即可完成相应系统的设计[5]。

在软核定制过程中，I/O接口设计充分体现了软核设计的可裁减优势，根据系统设计的要求，任意改变 IO口的个数和类型，使用方便。根据本设计功能的要求，确定 IO口如表 1所示。

表 1 确定 I/O接口

引脚名称	输入/输出 (in/out)	功能
clk	in	系统时钟输入
reset_n	in	系统复位引脚，低电平有效
keys_con[4..0]	in	最高位判断按键的有无，低四位为键值
sys_clk	out	系统时钟输出
fre_con_word[31..0]	out	频率控制字输出
lcd_data[7..0]	out	LCD八位数据输出
lcd_en	out	LCD使能控制信号输出
lcd_rs	out	LCD指令/数据选择信号输出
lcd_rw	out	LCD读/写控制信号输出
write_freq_con_word_en	out	写频率控制字使能信号输出
write_ram_address[9..0]	out	信号数据写地址输出
write_ram_data[7..0]	out	8位信号数据输出
write_ram_en	out	写 RAM使能信号输出

为了使该基本信号产生器系统更加简化，没有扩展 Flash存储器以及 SRAM存储器，而采用 EPCS1和 FPGA内的 RAM来代替。EP1C6Q240C8的 RAM容量为 92 160 bit，在配置时分配空间的大小因程序的大小而定，做到资源的充分合理应用。根据以上分析需要加入的组件有：Nios II CPU Core(CPU核)、片上存储器以及 PIO。配置完成后将会生成如图 5所示定制的 CPU配置表。点击 Generate生成模块，其模块如图 6所示。

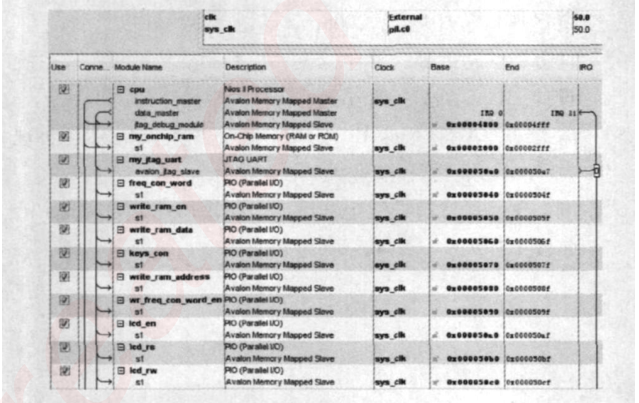

图 5 CPU定制截图

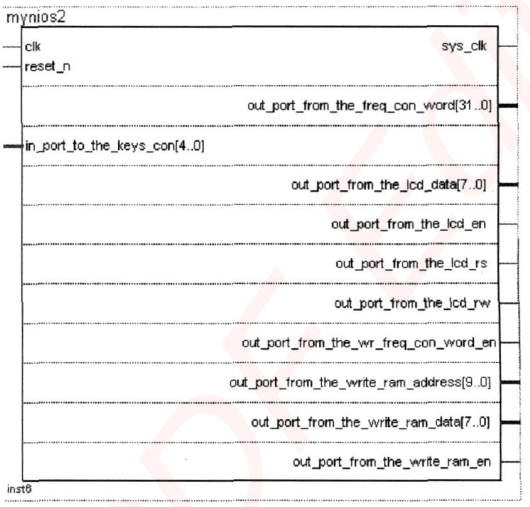

图 6 CPU模块截图

2.3 Nios II软件程序设计

在系统软件设计阶段，采用的开发工具是 Nios II IDE，它是 Nios II系列嵌入式处理器的基本软件开发工具。所有软件开发任务都可以在 Nios II IDE下完成，包括编辑、编译、调试和下载。

本程序实现的主要过程是：系统接收键盘扫描模块发来的 5位扫描码，判断键盘是否按下以及按下的键，根据按键的不同进入不同的子程序以实现 LCD显示、频率控制字的写入、信号数据的生成以及将其写入双口 RAM[6]。

主程序流程如图 7所示。

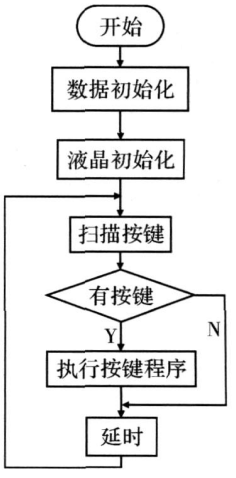

图 7 主程序流程图

3 软硬件测试

(1)在 RAM中加入方波、三角波以及正弦波数据，并设定频率控制字为 0x003fffff。采用 Quartus II 8.1的嵌入式逻辑分析仪。SignalTap II Logic Analyzer观看双口 RAM输出 $q[7..0]$，截图如图 8所示[4]。

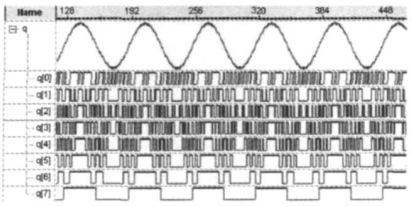

图 8　正弦信号截图

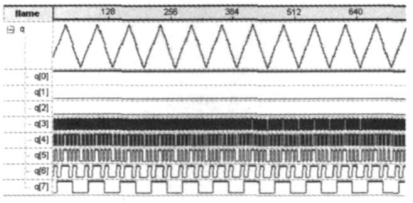

图 9　三角波信号截图

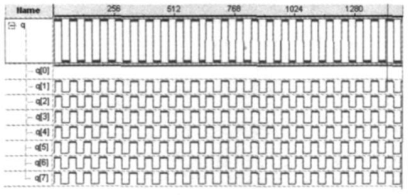

图 10　方波信号截图

(2)通过 AS下载接口将硬件编程文件下载到 EPCS1中，采用 Nios II IDE通过 JTAG接口运行软件程序。通过键盘设置正弦波、方波以及三角波及其频率值输出，并且在 1602液晶上显示相应的提示信息，如图 11所示。

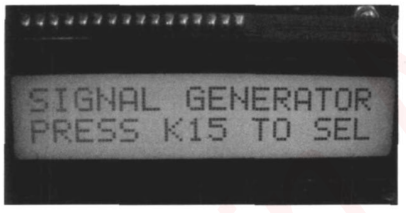

图 11　1602 液晶显示

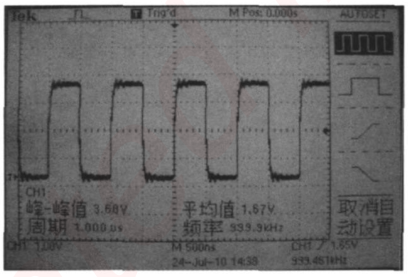

图 12　输出频率为 1 MHz 的方波

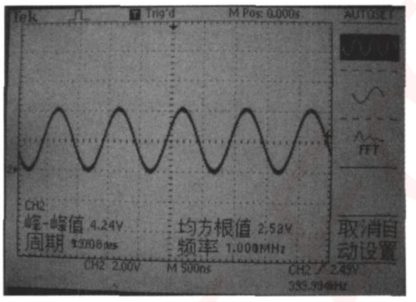

图 13　输出频率为 1 MHz 的正弦波

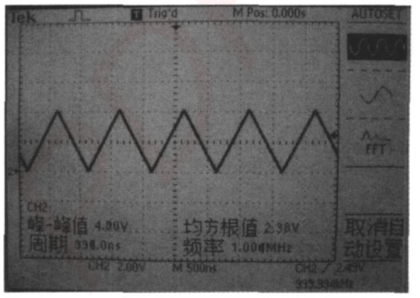

图 14　输出 1 MHz 的三角波

4　结束语

采用人机界面交互方式进行信号选择、频率设置等，Nios II CPU通过判断键盘输入在 LCD上给出相应的显示，提示用户选择相应的按键，输入完毕后，CPU将产生的信号数据和频率控制字传送给双口 RAM和频率控制字锁存器，最后在 DAC输出端输出相应模拟信号。通过示波器观察所产生的正弦波、方波以及三角波，达到了预期的目标，满足了系统设计要求。

参考文献

[1] 潘松，黄继业，曾毓，等. SOPC技术实用教程 [M]. 北京：清华大学出版社，2004.

[2] 吕矿生，周杏鹏. 基于 FPGA & Nios II的任意信号发生器 [J]. 仪器仪表与分析检测，2008，6(6)：24 - 26.

[3] 王书勋. 可重构 DDS信号发生器的设计与实现 [D]. 保定：华北电力大学，2008.

[4] 任爱锋，初秀琴，常存，等. 基于 FPGA的嵌入式系统设计 [M]. 西安：西安电子科技大学出版社，2004.

[5] 朱希志. 基于 Nios II的 DDS信号发生器研究与实现 [D]. 杭州：浙江工业大学，2006.

[6] 刘金华. 任意信号发生器设计与实现 [D]. 哈尔滨：哈尔滨工程大学，2006.

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

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

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

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

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

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

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