

# 基于GPU通用计算的分析与研究

General computing of gpu -A new high-performance computing solutions

田立国<sup>1</sup>, 杜君<sup>2</sup>

TIAN Li-guo<sup>1</sup>, DU Jun<sup>2</sup>

(1.汉中职业技术学院, 汉中 723000; 2.北京装甲兵工程学院 装备指挥与管理学院, 北京 100072)

**摘要:** 基于GPU的通用计算, 目前开创了一个新的研究领域。通过阐述其运算特点、运算平台; 分析它的硬件架构、计算能力、应用领域; 构建实验环境对加速性能进行验证和分析, 综合说明GPU通用计算是一种新的高性能运算。

**关键词:** GPU; 通用计算; 应用领域; 加速性能

**中图分类号:** TP301.4

**文献标识码:** B

**文章编号:** 1009-0134(2011)1(下)-0173-03

**Doi:** 10.3969/j.issn.1009-0134.2011.1(下).59

## 0 引言

最近几年来, 受游戏市场和视景仿真需求的牵引, GPU (Graphic Process Unit 图形处理单元) 得到了迅速的发展, 主流 GPU 的单精度浮点处理能力和外部存储器带宽都远远高于同期的 CPU, 而且 GPU 已经从以往的固定流水线操作模式发展成可编程流水线模式; 另一方面, 受工艺、材料和功耗的物理限制, 处理器的频率不会在短时期内有飞跃式的提高, 新的摩尔定律已经演化为“未来的计算机硬件不会更快, 只会更宽”, 相应的现代 CPU 也采用了多核的架构用来提高 CPU 的并行处理能力。但 GPU 与生俱来就是一种众核并行处理器, 在处理单元的数量上还要远远超过 CPU, 因此 GPU 在未来的高性能计算中将发挥越来越大的作用。

## 1 GPU通用计算概述

### 1.1 GPU通用计算

<sup>[1]</sup> 从系统架构上看, GPU 是针对向量计算进行了优化的高度并行的数据流处理器, 其中包括两种流处理单元: 顶点处理流水线(Vertex Shader)和像素处理流水线(Pixel Shader), 其中顶点流水线是多指令多数据流(MIMD)的处理单元, 而像素处理流水线(Pixel Shader)是单指令多数据流(SIMD)的处理单元。这种以数据流作为处理单元的处理器, 在对数据流的处理上可以获取较高的效率, 因此最近几年, 很多研究人员开创了一个新的研究领域: 基于 GPU 的通用计算 (GPGPU:

General-Purpose Computation on Graphics Process), 其主要研究内容是除了在图形处理方面外, 如何利用 GPU 来进行更为广泛的应用计算。GPGPU 计算通常采用 CPU+GPU 异构模式, 由 CPU 负责执行复杂逻辑处理和事务管理等不适合数据并行的计算, 由 GPU 负责计算密集型的大规模数据并行计算。这种利用 GPU 强大处理能力和高带宽弥补 CPU 性能不足的计算方式在发掘计算机潜在的性能, 在成本和性价比方面有显著的优势。

### 1.2 GPU通用计算的特点

在 GPU 上运行的图形应用程序的算法存在如下几个特征: 算法密集、高度并行、控制简单、分多个阶段执行以及前馈 (Feed Forward) 流水线等, 因此, 符合这些条件或者是可以改写成类似特征的应用程序, 就能够在 GPU 上获取较高的性能。

### 1.3 GPU通用计算平台

GPU 从其诞生之日起就是专门为图形处理设计的处理器, 它有自己的存储单元, 在数据存取方式上也存在一定的特殊性, 而且通常对 GPU 进行开发的是一些游戏厂商, 他们几乎都使非标准编程模式, 如果要想实现通用计算需要花费很大的程序开发能力, 为了便于开发人员使用 GPU 的通用计算能力, 世界上一些知名的软硬件厂商先后推出了一些开发平台和工具, 极大地推动了 GPU 通用计算的发展和进步。目前使用比较广泛的是 NVIDIA 的 CUDA, 主要有以下两个原因:

1) 硬件方面, 支持 CUDA 的 GPU 在构架上有显著的改进, 一是采用了统一的处理架构,

可以更加有效地利用过去分布在顶点渲染器和像素渲染器的计算资源；二是引入了片内共享存储器，支持随机写入和线程间通信。这两项改进使得 CUDA 架构更加适用于 GPU 通用计算。

2) 软件方面，CUDA 不需要借助于图形 API，并采用了比较容易掌握的类 C 语言进行开发。开发人员能够从熟悉的 C 语言比较平稳地从 CPU 编程模式过渡到 GPU 编程模式，而不需要去学习特定的显示芯片指令或者特殊的结构。CUDA 的最大优势在于它极大降低了利用 GPU 进行编程的入门门槛。只要编程人员掌握 C 语言，就可以利用 CUDA 所提供的编程环境和 SDK 基于 GPU 进行软件开发。

## 2 GPU通用计算的应用领域

### 2.1 GPU与CPU硬件架构比较

从硬件架构上看，CPU 和 GPU 是按照不同的设计思路设计的：如图 1 所示。

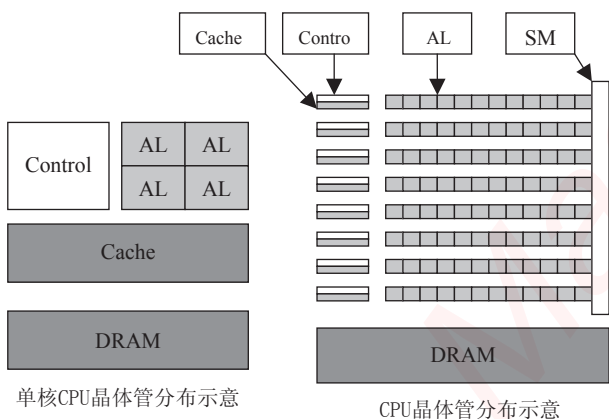


图1 CPU和GPU硬件架构示意图

CPU 的架构是按照兼顾“指令并行执行”和“数据并行运算”的思路而设计，即兼顾程序执行和数据运算的并行性、通用性以及它们的平衡性。因而 CPU 是计算机中设计最复杂的芯片。和 GPU 相比，CPU 核心的重复设计部分不多，这种复杂性来自于实现：如程序分支预测，推测执行，多重嵌套分支执行，并行执行时候的指令相关性和数据相关性，多核协同处理时候的数据一致性等等复杂逻辑。

GPU 其实是由硬件实现的一组图形函数的集合，这些函数主要用于绘制各种图形所需要的运算。这些和像素，光影处理，3D 坐标变换等相关的运算由 GPU 硬件加速来实现。图形运算的特

点是大量同类型数据的密集运算—如图形数据的矩阵运算，GPU 的架构就是面向适合于矩阵类型的数值计算而设计的，大量重复设计的计算单元，这类计算可以分成众多独立的数值计算—大量数值运算的线程，而且数据之间没有像程序执行的那种逻辑关联性。

### 2.2 GPU与CPU计算能力分析

GPU 拥有自己独立的子存储系统—显存，它拥有比系统主存高得多的带宽。GPU 在工作时，工作负载经 PCI-E 总线从 CPU 传入 GPU 显存，按照体系架构的层次自顶向下分发。PCI-E 2.0 规范中，每个通道上下行的数据传输速度达到了 5.0Gbit/s，这样 PCI-E 2.0×16 插槽能够为上下行数据各提供 5.0×16Gbit/s=10GB/s 的带宽，有效带宽为 8GB/s，但是由于 PCI-E 数据封包的影响，实际可用的带宽大约在 5-6GB/s (PCI-E 2.0×16)。

CPU 在浮点运算和并行计算方面的性能远不如 GPU 强大，所以将 GPU 的作为 CPU 的浮点运算协处理器可以使普通 PC 的运算能力提高到一个全新的高度。如果从更深层面考虑，目前 CPU 的发展已经遇到了瓶颈，无论是核心架构的效率还是核心数量都很难获得大幅提升，而 GPU 则是新的突破点，它的潜力很大。

### 2.3 GPU通用计算的应用领域

GPU 通用计算的发展迅速，特别是“CPU+GPU”模式的提出，使得 GPU 通用计算的应用领域逐步深入到科学计算的各个领域，如 FFT（快速傅里叶变换）<sup>[2]</sup>、串匹配算法<sup>[3]</sup>、科学计算可视化<sup>[4]</sup>、实时红外图像生成<sup>[5]</sup>、遗传算法加速<sup>[6]</sup>、数据库操作<sup>[7]</sup>、等。随着软硬件技术的进一步发展，以及越来越多的研究人员和工程技术人员的参与，GPU 通用计算的研究和应用领域将进一步拓宽。

## 3 GPU加速性能验证及分析

GPU 在处理大量的没有逻辑关系的数值计算时，理论上具有明显的加速效果，为了对 GPU 通用计算的加速性能进行原理性的分析和验证，构建了以下的实验环境：

硬件采用 2.2GHz Intel Core2Duo 处理器和 GeForce G210M 显卡（娱乐级显卡），软件环境采用 NVIDIA 的 CUDA SDK 2.0 及 C 语言编译器。

### 3.1 海量浮点数求和运算

选取 0~512\*21056 的浮点型数据，每个整型

数据执行 k 次累加和运算。

$$x[j] = j + \sum_{i=1}^{i \leq K} i (j = 0, 1 \dots 512 * 21056)$$

试验中，通过不断增大 k 的值，来调浮点数的运算量，试验的结果如图 2 所示。其中 GPU 程序执行时间包括数据在显存和主存之间的传递时间。

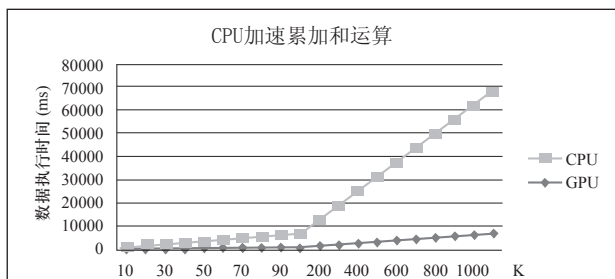


图2 海量浮点数运算

从试验结果可以看出，随着计算量的增加，CPU 和 GPU 执行时间都在增加，但到达一定的计算量后，CPU 执行时间成线性增加，而 GPU 基本保持不变，原因是高密度计算隐藏了 GPU 访存和数据传输带来的延迟。

### 3.2 龙格—库塔法解算弹道方程

龙格—库塔算法是一种常用的微分方程的数值解法，具有较高的精度，相应地其运算量也相对较大。

$$\begin{aligned} \frac{dv_x}{dt} &= -CH(y)F(v)\cos\theta = -C\prod(y)F(v)\cos\theta\tau_{0N}/\tau \\ \frac{dv_y}{dt} &= -CH(y)F(v)\sin\theta - g = -C\prod(y)F(v)\sin\theta\tau_{0N}/\tau - g \\ \frac{dx}{dt} &= v_x \\ \frac{dy}{dt} &= v_y \\ v &= \sqrt{v_x^2 + v_y^2} \\ \theta &= \tan^{-1} \frac{v_y}{v_x} \\ y_{m+1} &= y_m + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ k_1 &= f(t_m, y_m) \\ k_2 &= f(t_m + \frac{h}{2}, y_m + \frac{h}{2}k_1) \\ k_3 &= f(t_m + \frac{h}{2}, y_m + \frac{h}{2}k_2) \\ k_4 &= f(t_m + h, y_m + hk_3) \end{aligned}$$

实验中通过改变解算弹道数，比较分析 CPU 和 GPU 的执行时间，如图 3 所示。

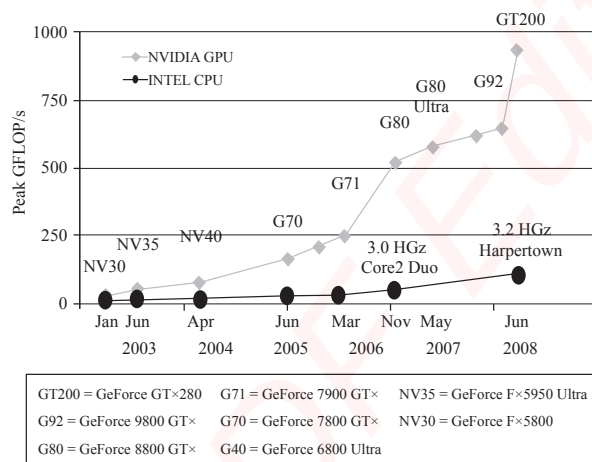


图3 龙格—库塔解算弹道方程

由试验数据可知，在弹道数小于 1024 的时候，GPU 执行速度没有 CPU 快，GPU 的加速性能没有充分发挥。当弹道数大于 1024 时，GPU 加速优势开始显现，随着弹道数增大，GPU 加速优势越来越明显，主要原因是大规模高吞吐量的数据运算也可以隐藏 GPU 访存和数据传输带来的延迟。

## 4 结束语

随着 GPU 计算能力的飞速提高，以及相关软件平台的支持，高性能计算逐步进入桌面领域，对于普通的科研和技术人员而言，高性能计算 (High Performance Computing, HPC) 不再可望而不可及，在这种高性价比的计算资源的支持下，许多以往看起来无法解决的运算问题都可以迎刃而解。

### 参考文献：

- [1] 吴恩华. 图形处理器用于通用计算的技术、现状及其挑战[J]. 软件学报. 2004, 15(10): 1493-1504.
- [2] 冯煌. GPU图像处理的FFT和卷积算法及性能分析[J]. 计算机工程与应用. 2008, 44(2).
- [3] 张庆丹, 戴正华, 冯圣中, 孙凝晖. 基于GPU的串匹配算法研究[J]. 计算机应用. 2006, 7(26): 7.
- [4] 郑杰. 基于GPU的高质量交互式可视化技术研究[D]. 西安电子科技大学. 2007.
- [5] 李勇. 基于GPU的实时红外图像生成方法研究[D]. 西安电子科技大学. 2007.
- [6] 杨正龙, 金林, 李蔚清. 基与GPU的图形电磁计算加速算法[J]. 电子学报. 2007, 6(35): 6.
- [7] 杨正平. 基于GPU计算的直接体视化和遗传算法研究[D]. 中国地质大学文. 2005.

# 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](#)
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](#)

## 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](#)

## 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](#)

## 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 IO 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](#)

## 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](#)

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](#)