

一种基于超大容量 Cache 的 VOD 系统

杨金岗¹, 戎蒙恬²

(上海交通大学电子工程系, 上海 200030)

摘要: 视频服务器是 VOD 系统的核心所在。衡量一个视频服务器性能最重要的指标是它所能支持的最大并发的媒体流的数量, 而硬盘的 I/O 的带宽已经成为进一步提高性能的瓶颈。提出了一种新的基于 SDRAM 的 Cache 方案有效地解决了这个问题。

关键词: VOD; 视频服务器; 磁盘阵列; Cache; SDRAM

A kind of VOD system based on super - large - capacity Cache

YANG Jin-gang¹, RONG Meng-tian²

(Electronic Engineering Department, Shanghai Jiaotong University, Shanghai 200030, China)

Abstract: Video stream server is the core device of a VOD system. The most important performance is the maximum concurrent stream it can support, limited I/O bandwidth of hard disk has been the bottleneck to improve this performance. A new Cache scheme based on SDRAM is proposed in this paper to solve this issue effectively.

Key words: VOD; video stream server; disk array; Cache; SDRAM

0 引言

近年来, 随着计算机网络和多媒体技术的飞速发展, 出现了多种多媒体应用和服务, 如视频会议、数字图书馆、远程教育、新闻点播等。在这些应用中, 视频点播 VOD (video on demand) 技术是一项重要的基础和关键的技术。

VOD 系统的主要功能是向客户提供对存储在 VOD 服务器中视频对象数据的实时访问。视频数据作为连续媒体, 在本质上与离散媒体 (如文本、图像等) 有所不同。首先, 视频对

象 (如影片) 的数据量较大, 需要较大的整体 I/O 带宽 (磁盘带宽与网络带宽)。VOD 服务器必须提供有效的方法, 来高速地存储、检索和传送大量的数据。其次, 视频对象只有在以时间上连续的方式展现时才有意义。因此, VOD 服务器必须确保影片在客户端能够实时地、连续地、平滑地播放。

VOD 系统开发必须解决两个最重要的问题: 处理随机的并发请求问题和大量数据存储问题。对于处理请求问题, 要求视频服务器具有大规模并行处

理能力, 要求网络有足够的带宽; 对于数据存储问题, 要求有大容量的存储器, 磁盘有足够的读吞吐量。为更好的这两个问题, 本文提出了一种基于超大容量 Cache 的 VOD 系统解决方案。

1 VOD 系统特征

1.1 信息流向的不对称性

对于大多数双向通信系统来说, 信息通路两个方向上的信息流量是对称的, 系统要为通信的双方提供同等的通信能力; 而 VOD 采用不对称的双向传输网络将信息提供者与用户连接起来, 用户信息通过窄带的上行信道传到信息中心, 而由信息中心到用户的下行信道则是具有视频音频传输能力的宽带信道。这是一种非对称双工形式的多媒体通信技术, 与多媒体信息检索业务的特点是一致的

1.2 点播信息内容和点播时间的集中性

对于 VOD 系统的广大用户, 点播的信息内容往

收稿日期: 2005 - 05 - 09

作者简介: 杨金岗 (1976 -), 男, 上海交通大学电子工程系硕士研究生, 研究方向为通信与信息系统。

的时间分布也是不均匀的,这正是造成信息流量突发的根本原因。

1.3 信息发送以及重现的实时性与同步要求

VOD系统的信息发送以及重现的实时性与同步要求都较其它信息检索系统高,特别是对视频音频信息的点播必须保证视频媒体与音频媒体内部的自同步以及媒体间的同步,这对系统的延时及抖动特性均提出了较高要求。

2 VOD网络拓扑结构

2.1 分布式结构

分布式结构中,多个视频服务器都配置在每个区域节点的前端,服务器可以很接近调制器。目前大部分VOD系统都是基于分布式结构。

优点:

每个服务器只处理本地业务,所能处理的流的数量有限,对服务器的要求降低。

骨干网只是用于下载新内容,对骨干网带宽要求降低。

缺点:

内容在不同的本地服务器上复制,导致对存储的巨大需求。

内容管理的困难。FTP,组播甚至是磁带都向不同的服务器分发相同的内容。

服务器的维护工作繁重。

存储在头端的内容的安全性不足以保障。

2.2 集中式结构

集中式结构中,服务器都放在主前端,通过数字骨干网将许多传输流传输到不同的地区前端,再由地区前端经调制器进行分配。地区前端仅对视频流作较少的处理,然后进行QAM调制。

优点:

在交换型骨干网络中,由于资源共享,所需的服务器要少于分布式结构。

由于同一内容被大量的用户共享,所需的存储也减少。

由于同一内容不被复制到多个服务器上,内容的管理也更容易。

服务器上的内容会更加安全。

缺点:

需要能够处理大量的并发视频流。

需要更高的网络带宽。

本下降,集中式结构越来越显示出其优越性来。

3 超大容量 SDRAM Cache 机制的提出和分析

3.1 传统 VOD 设备的局限性

VOD设备和传统的有线广播设备的最基本区别在于VOD设备不是一个可扩展的设备。可扩展的设备的特征是其成本结构有利与大多数的用户。广播设备正符合这个特点。由于大量的用户收到同样的内容,所以头端分发设备复杂性和成本都很低,每个用户的服务成本随用户总数的扩大而下降。显而易见VOD设备不是可扩展的,因为每个新用户都点播个性化的节目。于是,头端的输出带宽必须随用户数的增加而增加,而单个用户的成本并不随用户总数的增加而减少,这对于VOD设备是一个巨大的挑战。更为不利的是头端视频服务器的成本比用户数的增长还要快。换句话说,如果系统规模为100个流,每个流的成本少于100元,当系统规模扩大至1000个流时,每个流的成本将大于100元。这样的成本结构的主要成因是当所支持的视频流数目增加时,视频服务器的复杂性的增长将不再是线性增长,而是大于线性增长。当今很多视频服务器通过并行磁盘阵列设计(如RAID)和内容复制(如将相同的内容放在多个服务器上均衡请求)来支持更多的流请求。

市场上已有的视频服务器大多都是基于上述的内容复制技术和RAID技术,包括Seachange, nCube及Concurrent等设备商的产品。当每个用户点播不同的节目时,基于这种架构的设备工作性能较好。即使有几百个用户点播相同的节目时,因为RAID配置可以提供足够的带宽,它仍然可以正常工作。但是当有大量的用户同时点播相同的内容时,这些请求会耗尽磁盘的带宽。这时可以用多个服务器复制同样的内容来支持大量的用户请求,但问题是无法知道哪些是热门内容并事先复制这些内容。实际上热门内容也会随时间的推移而变化,比如电影。

这些因素意味着要根据实时的用户请求来实时的进行内容复制。对于视频服务器来说这样会增加额外的读写带宽负担,导致视频服务器性能的降低。

近年来磁盘的存储容量虽然有了大幅度的提高,但是I/O带宽提高并不多。硬盘的I/O性能瓶颈并不在于磁盘的读取的速度,而在于读写时数据

快越好。SCSI 硬盘的最高转速为 15 000RPM,快于 ATA 的 7 200RPM。磁盘读写数据的智能程度。基于 FC/SCSI 技术的硬盘可实现 R/W 指令的重新排序来减少随机寻道的次数,这使得它们快于基于 ATA 技术的硬盘。但与 SATA 硬盘相比,这个优势并不明显。

对于将硬盘作为主要 I/O 设备的服务器来说,大量对磁盘的随机读写会导致系统崩溃。

3.2 VOD 系统中的 Cache 机制

在计算机应用中,为提高系统的响应速度,大量采用了 Cache 技术。Cache 机制的实现原理都是数据访问首先从较快的存储器中取得,只有当快速存储器中没有所要求的数据时,才从较慢的存储器中访问数据。在视频信息的读取和播放过程中,由于网络的不稳定性,特别是 MPEG 流对时间的不平衡性,使得可能产生如下问题:在某一时刻视频信息的读取和播放速率不匹配,进而导致视频信息播放不稳定。为此采用缓存机制。在该机制的作用下,当信息的读取和播放速率不匹配时,系统仍可以正常工作,内存缓冲区域越多,则缓冲能力越大,系统的稳定性就越强。

目前在 VOD 视频服务器中可见到将磁盘作为 Cache 使用。当视频服务器响应客户的视频请求后,从存储系统读入一部分视频流数据到服务器缓存中,然后服务器缓存中的内容送入网络接口发送到客户端的缓存中,服务器必须保证在客户端缓存中的视频数据播放完之前,将所需的视频数据从存储系统中读出并传送到客户端的缓存中。当有大量的并发请求时,磁盘的寻道访问次数会大幅度上升,由于磁盘的寻道时间较长,寻道时间将占用整个读写时间的很大一部分比例,从而降低读写性能。尤其是当访问较小的数据块时,读写效率则更低。这时就有可能引起视频流播放的抖动和延时。

3.3 基于 SDRAM 的超大容量 Cache 机制

SDRAM 的存取时间很短,一般只有几十纳秒,而磁盘阵列的响应时间平均为 5~10 毫秒;SDRAM 具有很高的随机读写带宽,通常是几百 GBps,而现有最快的磁盘速度只有 400~500Mbps 而已。基于 SDRAM 技术的方案将把随机存取的速度从毫秒级提高到纳秒级,从而极大程度的提高 I/O 性能。

本文提出的基于 SDRAM 的超大容量 Cache 不

点播问题的唯一途径。Cache 容量可达到 1TB,可提供 9 000 个以上的并发流服务。

基于该机制的 VOD 系统有以下几个特点:

(1) 实现了存储和视频流 I/O 的分离

市场上现有的大多数 VOD 设备厂商,他们的 I/O 技术都是以快速 FC/SCSI RAID 阵列加快速的 PC 服务器来实现的。存储和视频流 I/O 的耦合以及有限的磁盘 I/O 性能限制了 VOD 系统的性能提升,其中速度最快的 Midstream 服务器目前也只能提供 440 个流。相比之下,有一个非常大容量的 SDRAM Cache 的 Examax 服务器真正实现了存储和视频流 I/O 的分离。

当有大量并发的对于同一内容的访问时,由于 Cache 中存放着同一内容,所以大量并发访问就表现为对于 SDRAM 的随机存取,这正是 SDRAM 的优势所在。巨大的 Cache 存储量使得大部分用户的请求可直接从 Cache 中获得,从而降低了 I/O 访问的延迟。

(2) 实现了存储数据读取和实时流播放的分离

实时流播放对于视频数据包传输到用户接收端的时序有比较严格的要求。用户接收端,特别是数字机顶盒都是基于标准的视频压缩格式(如 MPEG-2)来设计制造的。这意味着数据包必须按指定的时间间隔进入接收端 Buffer 以避免 Buffer 上溢或下溢。所以视频服务器在传输数据包时既不能太快也不能太慢。更重要的是,传输数据率必须与原始内容的编码比特率相匹配。对于 CBR 数据来说这比较容易,但对于 VBR 数据来说就非常难以控制。

Examax 由于其存储和视频流 I/O 的分离机制及内部 Buffer 的应用,从存储服务器传输数据的速率不再受限于编码比特率,而是可在最大速率以内调度。视频流播放速率则必须按精确的时序传输,两者间的不匹配可通过大容量的 Cache 来消除。

(3) 具有可扩展性

由于 Cache 为所有的用户所共享,每个视频流成本将随用户数量的增加降低,所以基于 Cache 的 VOD 设备将是一个可扩展的设备。

4 Cache 容量、Storage Server 带宽和视频流数量的关系

由图 1 可见,Storage Server 只有当用户点播的内容在 Cache 内没有找到时才会向视频流服务器传送该内容。所以,R 代表当 Cache 没有被命中时所

有利而对 Storage Server 越不利。另一方面,存储和 I/O 的分离对于当所有点播的内容都位于 Cache 内时最为有利,这时所有的流都直接从 Cache 内流出,Storage Server 和流服务器之间没有数据传输。

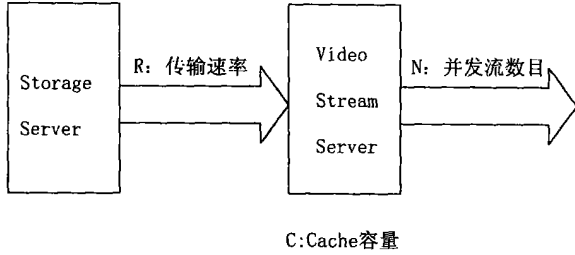


图1 基于 Cache 的 VOD 系统的关键参数

统计上来看,有 Cache hit 和 Cache miss 两种情况的组合。图2说明了节目的访问频率分布情况。

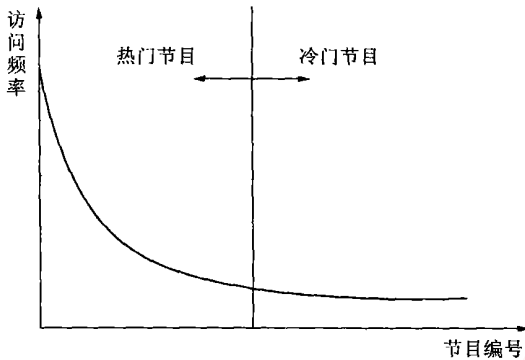


图2 节目的访问频率分布曲线

从图2的曲线上看,热门节目位于左边而冷门节目位于右边。竖直线将该曲线分为两部分,左边代表存储在 Cache 中的内容,也就是发生 Cache 命中的区域,当点播该区域中的内容时无须在 Storage Server 和流服务器之间进行数据传输;右边代表没有存储在 Cache 中的节目内容,也就是发生 Cache 不命中的区域,当点播该区域中的内容时就需要在 Storage Server 和流服务器之间进行数据传输。

当点播请求增加时,Cache 不命中的比例会相应增加,从而增加了流服务器对 Storage Server 的带宽需求,可以从图3的曲线看出这种变化。由该曲线可得出输出带宽应与 Cache 的容量成比例。当输出带宽过高,即支持过多的并发流时,Cache 不命中机率的增加会超出 Storage Server 所能承受的范围。

为了更好的说明这一点,让我们来看图4的曲线。当有更多的节目被存储在 Storage Server 中时,曲线会向上移动。Cache 容量为一定时,由 Cache 容

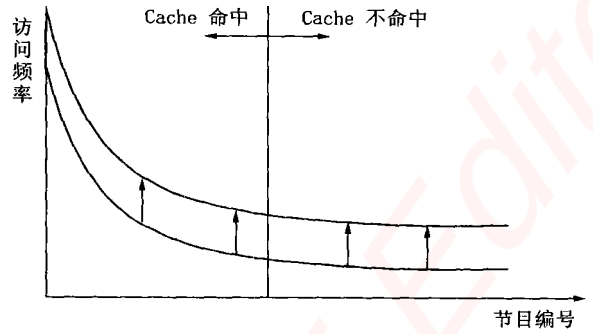


图3 Cache 容量不变时点播请求增加的情形

量所决定的 Cache 命中的节目数保持不变,而因节目请求数增加将使尾部阴影面积增加,而且对直线右边的节目请求次数将比以前增加,这意味着 Cache Miss 的几率会随之增加。阴影面积的大小决定了 Storage Server 的传输速率,点播请求增加时,Storage Server 的传输速率将随之增加。

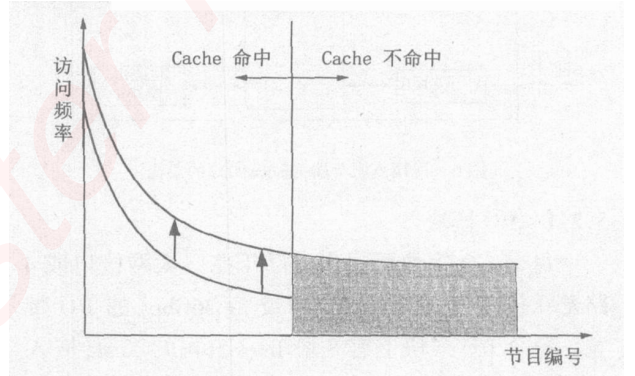


图4 点播请求增加对 Storage Server 传输速率和 Cache 容量的影响

基于以上的分析可得出另一个结论:当支持一定数目的并发流请求时,Storage Server 的最大传输速率与 Cache 容量成反比。表1可以看到这一点。

表1 Cache 容量,Storage Server 带宽和视频流数量的关系

Storage Server 的最大传输速率 R	Cache 容量 C	最大并发流数 N
常数	增大	增大
增大	常数	增大
增大	减少	常数

5 基于超大容量 SDRAM Cache 机制 VOD 系统

5.1 系统架构

Examax9000 是 Exavio 公司推出的新一代高性能视频流服务器平台,该平台可支持基于 SAN 存储架构和基于 NAS 存储架构的 VOD 系统,如图5所示。

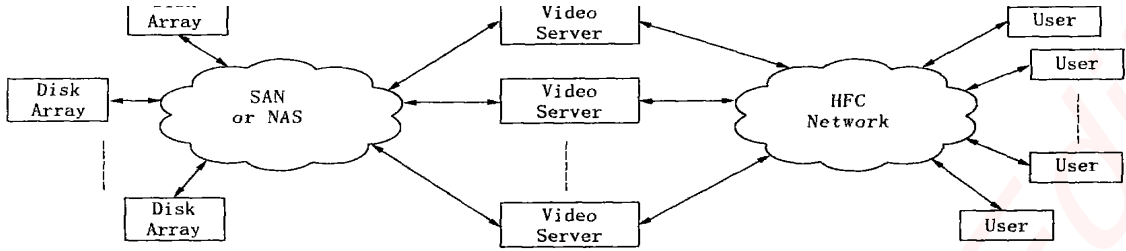


图5 基于 Examax9000 视频流服务器的 VOD 系统架构

5.2 视频流服务器

图 6 为视频流服务器 Examax9000 的架构,由 I/O 模块,背板和 Memory 模块所组成。

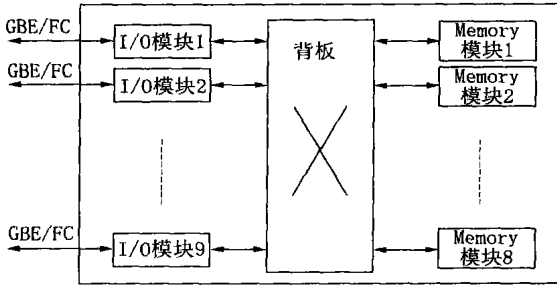


图6 视频流服务器 Examax9000 的架构

5.2.1 I/O 模块

每个 I/O 模块可提供 4 路千兆以太网接口或 4 路光纤接口,9 个 I/O 模块可支持 36Gbps 的 I/O 带宽。每个 I/O 模块上有 8 块 Intel 公司的 Xscale 嵌入式处理器,主频高达 600MHz。该处理器专为网络存储和低功耗应用而设计。多个 CPU 用以完成大量的网络协议栈处理,缓存管理及网管处理。板内芯片之间的互连采用 133MHz,64 位 PCI-X 总线。

5.2.2 背板

背板总线以 72Gbps 的带宽连接多个 CPU 和共享 Memory,通过无阻塞 Switch Fabric 完成高带宽的交换。与千兆以太网交换机中的多点对多点的 Switch Fabric 不同,它是真正的 72Gbps 共享带宽,每个 CPU 都能以 72Gbps 访问 Memory。Switch Fabric 由 Altera 公司的 STRATIX 系列高性能 FPGA 实现。

5.2.3 Memory 模块

Memory 模块直接与背板连接,可以被所有的 CPU 访问到。Memory 模块实现一个超大容量的 SDRAM Cache,每个 Memory 模块可配置 64 条 DIMM,最大容量可达到 $64 \times 2 = 128GB$ 。对 Examax9000 系统来说,36Gbps 的输出带宽需要有 1TB 的 Cache Memory,而且必须有足够带宽的 Storage Server

与之相匹配,带宽大小是根据实验和实际的流量模型所得出的。

作为整个系统的核心所在,Memory 模块有以下的重要特征:

72Gbps 的背板带宽可访问 Memory 中的任意存储空间。

DMA 引擎可将数据搬移的重担从 CPU 中分离出来。

通过电源管理显著降低了功耗。Memory 模块平时工作于低功耗模式,当有读写请求到达时,会有几十个纳秒的时间来“唤醒”Memory,使其进入正常工作状态。由于系统 60% 以上的功耗消耗在 Memory 模块上,这一特征就显得尤为重要。

使用现成的低成本,大容量的 DIMM,并且提供最广泛的兼容性。

通过并行和流水线设计大大提高了 Memory 的带宽利用率。

通过优化的缓存管理算法来提高 Cache 的命中率。

6 结束语

针对现有实现大规模 VOD 系统时所面对的处理随机的并发请求问题和大量数据存储问题,本文提出了一种新的基于 SDRAM Cache 的 VOD 系统解决方案,可以有效的解决这些问题。该目前系统已工作稳定并且显示了其高性能的处理能力。

参考文献:

- [1] 刘志明,彭宇行,陶阳. 并行视频服务器磁盘 Cache 策略研究[J]. 计算机工程应用,2003,(16).
- [2] 林坤辉. 视频点播系统两级存储结构的设计与实现[J]. 厦门大学学报(自然科学版),2002,41(5).
- [3] 郝学坤,胡向晖,马刘非. 中心式 VOD 视频点播系统及其关键技术实现[J]. 电视技术,2002,(10).
- [4] 顾铁成,陈道蓄. 影响多媒体服务器性能的关键因素研究[J]. 小型微型计算机系统,2002,23(12).

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](#)

邀请注册码



关注论坛公众号

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](#)

邀请注册码



关注论坛公众号

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](#)
84. [基于 PCI9656 控制芯片的高速网卡 DMA 设计](#)
85. [基于以太网及串口的数据采集模块设计](#)
86. [MVB1 类设备控制器的 FPGA 设计](#)
87. [MVB 接口彩色液晶显示诊断单元的显示应用软件设计](#)
88. [IPv6 新型套接字的网络编程剖析](#)
89. [基于规则的 IPv4 源程序到 IPv6 源程序的移植方法](#)
90. [MVB 网络接口单元的 SOC 解决方案](#)
91. [基于 IPSec 协议的 IPv6 安全研究](#)
92. [具有 VME 总线的车载安全计算机 MVB 通信板卡](#)
93. [SD 卡的传输协议和读写程序](#)
94. [基于 SCTP 的 TLS 应用](#)
95. [基于 IPv6 的静态路由实验设计](#)
96. [基于 MVB 的地铁列车司机显示系统研究](#)
97. [基于参数优化批处理的 TLS 协议](#)
98. [SSD 数据结构与算法综述](#)
99. [大容量 NAND Flash 文件系统中的地址映射算法研究](#)
100. [基于 MVB 总线的动车组门控系统的设计与仿真研究](#)
101. [多功能车辆总线 MVB 控制](#)
102. [基于 LabVIEW 的 MVB 和 WTB 帧解码方法](#)
103. [基于双 FPGA 的 MVB 通用接口研制](#)
104. [RAID 中 Cache 的设计与实现](#)

邀请注册码



关注论坛公众号

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](#)

邀请注册码



关注论坛公众号

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. [基于 VxWorks 系统的 MiniGUI 图形界面开发](#)
60. [VxWorks BSP 开发中的 PCI 配置方法](#)
61. [VxWorks 在 S3C2410 上的 BSP 设计](#)
62. [VxWorks 操作系统中 PCI 总线驱动程序的设计与实现](#)
63. [VxWorks 概述](#)
64. [基于 AT91RM9200 的 VxWorks END 网络驱动开发](#)
65. [基于 EBD9200 的 VxWorks BSP 设计和实现](#)
66. [基于 VxWorks 的 BSP 技术分析](#)
67. [ARM LPC2210 的 VxWorks BSP 源码](#)
68. [基于 LPC2210 的 VxWorks BSP 移植](#)
69. [基于 VxWorks 平台的 SCTP 协议软件设计实现](#)
70. [VxWorks 快速启动的实现方法\[上电到应用程序 1 秒\]](#)

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 C++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](#)

邀请注册码



关注论坛公众号

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](#)
52. [Linux 基于 W83697 和 W83977 的 UART 串口驱动开发文档](#)
53. [基于 AT91RM9200 的嵌入式 Linux 系统的移植与实现](#)
54. [路由信息协议在 Linux 平台上的实现](#)
55. [Linux 下 IPv6 高级路由器的实现](#)
56. [基于 Android 平台的嵌入式视频监控系统设计](#)

邀请注册码



关注论坛公众号

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](#)
32. [基于 Windows CE 的 GPS 应用](#)
33. [基于 Windows CE 下大像素图像分块显示算法的研究](#)
34. [基于 Windows CE 的数控软件开发与实现](#)
35. [NAND FLASH 在 WINCENET 系统中的应用设计](#)

邀请注册码



关注论坛公众号

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](#)
31. [基于 MPC850 的多功能通信管理器](#)
32. [基于 MPC8640D 处理系统的技术研究](#)
33. [基于双核 MPC8641D 处理器的计算机模块设计](#)
34. [基于 MPC8641D 处理器的对称多处理技术研究](#)
35. [PowerPC 处理器原理](#)

邀请注册码



关注论坛公众号

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](#)
36. [UBoot 在 AT91RM9200 上的移植简析](#)
37. [基于工控级 AT91RM9200 开发板的 UBoot 移植分析](#)

邀请注册码



关注论坛公众号

38. [基于 ARM11 和 Zigbee 的人员定位防丢器](#)
39. [基于 NAND FLASH 的嵌入式系统启动速度的研究](#)
40. [μ COS II 在 ARM7 上的移植](#)
41. [基于 ARM11 的嵌入式视频采集系统设计](#)
42. [基于 ARM11 的视频监控系统设计](#)
43. [μ C-OSII 在 LPC2210 上的移植研究](#)

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 技术研究与发展](#)
30. [GPU 实现的高速 FIR 数字滤波算法](#)

邀请注册码



关注论坛公众号

31. [一种基于 CPU/GPU 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](#)
41. [EFI-Tiano 环境下的 AES 算法应用模型](#)
42. [EFI 及其安全性研究](#)
43. [基于 UEFI Shell 的 PreOS Application 的开发与研究](#)
44. [UEFI Bootkit 模型与分析](#)
45. [UEFI 计算机系统快速调试方法的实现](#)
46. [基于 EFI 系统的多文件系统解决方案](#)
47. [基于 UEFI 的可信 Tiano 设计与研究](#)

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. [用 IPv6 编程接口实现有连接通信的方法](#)
17. [一种战棋游戏的 AI 算法设计与实现浅析](#)
18. [基于 TLS 协议的 ECC 扩展研究](#)

邀请注册码



关注论坛公众号

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. [FPGA 的 VHDL 设计策略](#)
16. [用 FPGA 实现串口通信的设计](#)
17. [GPIB 接口的 FPGA 实现](#)
18. [一种基于 FPGA 的 FFT 阵列处理器](#)
19. [基于 FPGA 的 FFT 信号处理器的硬件实现](#)
20. [CPLD 在 CAN 通讯卡中的应用](#)
21. [用 CPLD 实现同步串口与异步串口的转换](#)
22. [CPLD 在 LED 网络控制器中的应用](#)
23. [基于 CPLD 的双口 RAM 设计与应用](#)
24. [CPLD 在有源电力滤波器中的应用](#)

邀请注册码



关注论坛公众号