RT Embedded http://www.kontronn.com

RAID 中Cache 的设计与实现

缪军海① 朱兰娟 吴智铭②

摘 要 在RAID(廉价冗余磁盘阵列)系统的开发中,Cache 的设计是重要一环,它的效率直接影响到了RAID 系统的性能,本文首先介绍了Cache 的原理和常用的一些技术,并比较了它们的优缺点,然后针对RAID 系统的实际情况,介绍了RAID 中Cache 的设计和实现。

关键词 RAID LUN Cache Write-Through Write-Back

一、引言

RAID(廉价冗余磁盘阵列),通过组合小容量的廉价磁盘来代替大的昂贵磁盘,从而降低大批量数据存储的费用,同时也通过一些冗余信息如镜像或各种校验码数据,使得硬盘发行错误时,可以通过这些信息来恢复硬盘上的数据。现在RAID 系统已经在很多领域取得了应用。如银行及金融系统、ISP(网络服务供应商)、图像处理、多媒体和数据库、服务器、石油化工行业、电信等部门的数据中心等。本文根据本人在实验室里开发RAID 项目的实践经验,来讨论RAID 系统中Cache 的设计和实现。

二、RAID 系统软件总体结构

下面给出RAID 系统软件的总体结构图,用来说明Cache 在RAID 系统中的位置和作用。

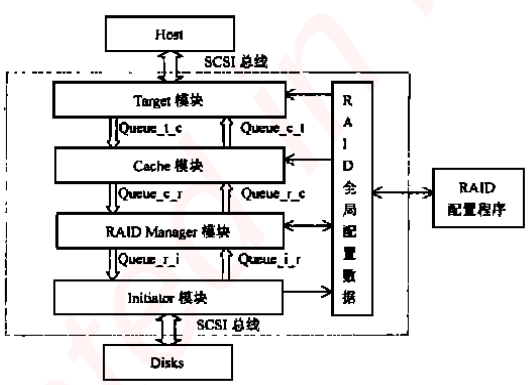

图1 RAID 软件系统总体结构图

RAID 软件系统包括虚线框中的4 个软件模块。这4 个软件模块中,各自的功能如下:

1) Target 模块负责和主机操作系统及Cache 模块的接口,它接收来之主机的命令,然后经过它的处理和转换,发到消息队列Queue-t-c 中,转发给Cache 模块来处理。

2) Cache 模块负责处理数据读写时内存块的管理和调度,如Target 模块发来一个读命令,Cache 负责查找是否命中,如命中则直接返回给Target 模块,如不命中,则根据相应的算法给它分配内存块,再把命令发给Queue-c-r,交给RAID Manager 模块来处理。

3) RAID Manager 模块负责管理RAID 中虚拟盘到物理盘的映射转换和接收及处理来自Cache 的任务,负责整个RAID 系统的错误处理。包含对各个RAID 级别的处理模块:RAID0,RAID1,RAID10,RAID5 等。

4) Initiator 模块负责接收来自RAID Manager 的命令,转换为具体的SCSI 命令,并通过SCSI 总线把它发给某个通道上的某个硬盘。然后由硬盘执行具体的SCSI 命令,从而完成任务。

三、Cache 的原理与技术

Cache,又称为高速缓存,它能够有效地提高系统的I/O 性能,它的原理是将CPU 要写入磁盘的数据先写到由RAM 构成的Cache 中,再由Cache 写到磁盘中,或先将磁盘的数据读到由RAM 构成的Cache 中,然后CPU 再从Cache 中将数据读入。在RAID 中也是如此,因此RAID 系统中的Cache 是系统内存(DRAM)中的一部分(系统和程序使用内存除外的那部分内存),它的速度比硬盘的速度要快得多。RAID 系统可以通过提高数据的命中率来提高读写数据的速度。如读命中可以提高读数据的速度,写命中的情况下,使用Write-Back 技术能够提高写数据速度,有效地提高了磁盘阵列的效率。因此Cache 部分的设计对整个RAID 系统的效率有着重大的影响。

Cache 容量与下一级存储器容量相比,是很小的,它保存的只是下一级存储器内容的一个子集,Cache 与下一级存储器之间内容的交换是以块为单位的,被保存的下级存储块与Cache 中的存储块如何对应起来才能迅速找到所需块? 以及当Cache 已满,无空余位置时,新的存储块到来时必须将一个原有块换出,这种替换该如何进行? 这些涉及到Cache 的组织结构中的映射方式和替换策略。

1 映射方式

Creative Master Free Editor

RT Embedded http://www.kontronn.com

按Cache与下级存储器之间交换数据块的大小把Cache和下级存储器划分为一个个的存储块，Cache的存储块用A_i表示，其中$i=0,1,2\cdots,m-1$，共有$m=2^r$块，下级存储的存储块用B_j表示，其中$j=0,1,2\cdots,n-1$，共有$n=2^s$块。每块有$k=2^t$字节组成。下面给出几种映射方式。

(1) 全相联映射方式(Fully Associative-Mapping)

这种方式是通过将下级存储块的地址(块号)与块的内容一起都存于Cache块中，块地址存放于Cache块的标签(tag)部分中，一个存储块可以拷贝到Cache的任何一个块中，优点是极为灵活，缺点是查找很困难，对于一个指定的存储地址，比较时必须将其块号与Cache所有块的标签相比才能找到，速度太慢。当Cache容量较小，比较次数较少时才适合采用该种方式。映射方式如下图所示：

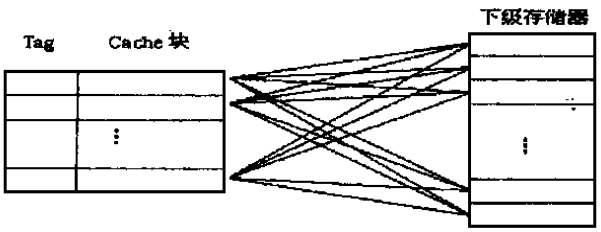

图2　全相联映射方式

(2) 直接相联映射方式(Direct-Mapping)

直接相联映射方式是一对多的映射关系，即一个Cache块可以存储多个对应的下级存储块，但一个下级存储块只能拷贝到Cache的一个特定块里。下级存储块的块号b与Cache的块号a有如下关系：

$$a=b \bmod n_c \quad (n_c \text{是} Cache \text{的总块数})$$

直接相联映射方法的优点是简单，访问速度较快，而且Cache无需地址变换，下级存储地址的低位部分就是Cache的地址，下级存储地址的高位部分与块数据一起存放到Cache块中。该方法主要的缺点就是块的冲突率比较高。当下级存储中的两个或两个以上的块都映射到Cache同一块中，而这些块又是当前常用的，这时就发生了冲突，此时需要把原先已存入的块换出Cache，但很可能过一会又要换入，频繁的换入换出会使Cache的效率降低。即使此时Cache有很多空闲块也用不上。这种方式适合于Cache容量较大的场合。该映射方式如下图所示：

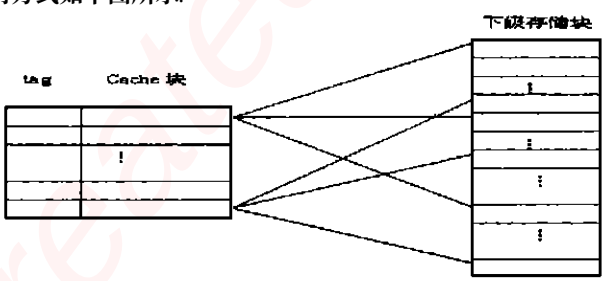

图3　直接相联映射方式

(3) 组相联映射方式(Set-Associative Mapping)

这种算法是兼顾了全相联映射和直接映射两者的优点，又尽量避免两者缺点的一个折衷方案。它将Cache存储块分成a组，每组b块，下级存储中以a个块为一个区来划分区，设k为下级存储块存放到Cache中的组号，j为下级存储块的块号，则有如下关系：

$n_c=a\times b$　　n_c是$Cache$中的总的块数

$k=j \bmod a$

在这种方式下，每个下级存储块存放到那个组中是确定的，至于存在该组中的那一个块中则是可变的。从下级存储的区到Cache的组的映射采用全映射方式。这种算法与直接映射相比，最明显的优点就是块的冲突率大大降低了；与全相联映射相比，它的实现较为容易，查找命中与否的范围小了许多，全相联映射与直接映射可以视为组相联方式的两种特殊情况。a=1时，Cache为分一大组，即为全相联映射方式。b=1时，每组只有一块，即为直接映射方式。下面的图中以a=4，b=2为例说明了这一映射方式。

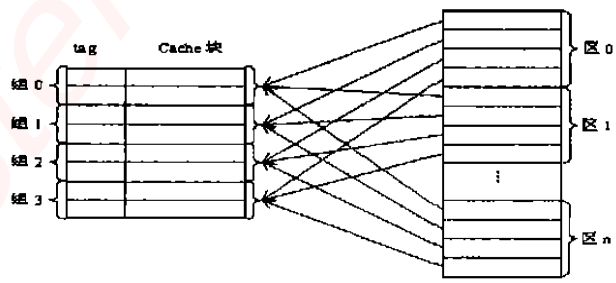

图4　组相联映射方式

2 替换策略

当一个下级存储器中的块要放到Cache中去，而Cache中允许存放此块的位置都被占满了，这就要产生替换，Cache中替换的实现与效率与它的组织结构有关，对于采用直接映射的Cache，下级存储块只有一个特定的位置可以存放，只需要此特定位置上的原存储块替换出去即可。对于全相联或组相联形式的Cache，就要从允许存放新存储块的若干特定块中选取一块换出。常用的算法主要有以下几种。

(1) 最不经常使用(LFU)算法。

LFU (Least Frequently Used)算法是将最近一段时间内被访问次数最少的存储块替换出来。可以通过对每一个存储块设置一个计数器，每个块刚被使用时从0开始计数，以后每访问一次，计数值增加一次。到需要替换时，对这些存储块的计数器的值进行比较，计数值最小的存储块就被替换出去。

(2) 近期最少使用(LRU)算法。

LRU (Least Recently Used)算法是用最近一段时间内访问次数最少的块或页面来替换。这样做是非常合理的，因为目前使用最少的块，将来很可能也很少访问。这样，既充分利用

RT Embedded http://www.kontronn.com

过去的历史信息，又正确反映了程序的局部性。

(3) 随机替换(Random Replacement)

随机替换这种策略实际上就是不采用什么算法，在可被替换的存储块中随便选择一块来替换。这种算法最容易实现，而且速度也最快。但是，它完全没有利用Cache使用情况的历史信息，也没有反映程序的局部性，导致命中率比较低。

四、RAID中Cache的设计与实现

在RAID系统中，可以简单地把内存设计为缓冲区，但这样做的话，RAID系统的性能较低，速度比较慢。而按照Cache的方式来组织的话，可以提高RAID系统的性能，提高读写的速度。我们设计的RAID系统中采用SCSI总线，遵循SCSI-2协议。每个Target最多可以配置8个LUN(逻辑单元)，每个LUN采用何种RAID级别来实现是由用户来可以配置的。下面我按与设计简单的缓冲区相比较的方式，来说明RAID中Cache设地采用的几个主要方案。

(1)Cache中按LUN(逻辑单元)划分区

RAID系统可以配置为多个LUN，各个LUN一般也对应着不同范围或不同用户组的数据，考虑到RAID系统中的内存相对比较小，我们使用的是64M DRAM。为了避免不同LUN的用户在读写过程中使用Cache时，某个LUN的用户抢占过多的Cache存储块而导致另一个LUN的用户无法进行读写操作，我们可以把Cache的整个存储区按实际读写情况动态分配成若干个内存区，每个LUN使用相对应的一个区。缺省情况下按配置中的各个LUN的容易大小划分内存区，每个LUN使用相对应一个区。

(2)Cahce中的内存块与LUN中的存储块之间建立地址映射

在RAID的使用过程中，在一些常见的使用场合下，用户的读写一般会具有一定的局部性，也就是说，在某个具体的时刻，读写的数据会集中在某个局部区域。在这种情况下，数据块被重复读写的概率还是不小的，也就是说可以通过读命中—即直接从Cache中读取数据来加快数据的读取速度，这里涉及到一个命中的查找速度问题，采用地址映射方式来组织Cache的存储块，就是要通过建立和使用映射方式，达到较快的查找速度。比较第3部分叙述的三种映射方式，我们选取的是组映射方式，这种映射方式，查找比较快，而且Cache中存储块的冲突率也比较低。

(3) 替换算法的选择

Cache的容量与RAID的存储容量相比是很小的，在RAID的使用过程中，Cache所有的内存块很快就分配完了，当读写没有命中的时候，就要从可用的存储块中分配一块来存放新的数据，此时就要设计一个合理的算法来从Cache的存储块中选出一块来使用，在我们设计的RAID系统中，根据RAID的实际使用情况，选用第3部分中叙述的LUR(Least Recently Used)，但是，如果每个存储块都只是简单地设置计数器的话，查找时需要把所有可用存储块的计数器要都比较一遍，从中找一个最小的使用，这样做查找效率太低，我们可以对这个算法作了一些改进，对每一个存储块，除了设置计数器，还设置了使用位和被期访问位，在寻找替换块时，在可被替换的存储块(使用位= false)中，先在最近没有被访问的存储块(最近访问位= false)里找计数值最小的块，如果每个可用存储块的最近访问位都是true，则在这些存储块中找一个计数值最小的来用。如果所有的块都不可用，则等到下一轮再查找，直到找到替换块或改任务被取消(如在规定的重试次数内没有找到)。在软件实现中，可以通过把这些状态不同的存储块用链表区分开来，查找时只要对一部分块进行即可，提高了查找替换块的效率。

(4)Cache中存储块大小的确定

在Cache中，对存储块的大小的确定要根据RAID的实际使用情况而定，也可以在使用过程中动态地根据实际应用情况进行改动。存储块的大小如太大，则总的可供使用的块的数目就比较少，同一时刻能处理的读写任务就比较少，在读写数据量比较小的情况下，还会造成存储块的利用率较小。存储块容量太小的话，存储块的数目太多，一方面保存数据块信息的附加数据会较大，另一方面，让SCSI总线每次发送速据量太少的话，总线的使用效率也较低。应此存储块的大小应根据实际应用来定。在我们的设计中，存储块的大小设为8K～128K。

(5)RAID中Cache的一致性处理

在Cache系统中，必须使Cache中的数据与磁盘中的数据保持一致。当在RAID系统中写数据到磁盘中去时，有两种方式，一种是写穿(Write-Through)方式，另一种是写回(Write-Back)方式。在写穿方式中，数据在写到Cache中后，马上就写入磁盘。这种方式控制起来很简单，但是当写命令很多的时候，尤其是当重复写入到某一块中去时，访问磁盘的次数较多，效率较低。在回写方式中，数据写入Cache后，并不马上写入磁盘，只有当它要被置换出去时，才把它写入磁盘。这种方式可以减少写磁盘的次数，加快了写数据的速度，提高RAID系统的性能。缺点是可能会丢失数据，例如当断电时，Cache的数据来不及写入磁盘就丢了。在我们设计的RAID系统中，我们可以在硬件上确保断电时给RAID系统供电，并把没写入磁盘的数据写入磁盘。应此，我们采用的是Write-Back方式，速度较快。

(6)RAID中Cache的动态调整

RAID系统的应用面较广，在不同的应用场合中数据的读写情况也是不同的，Cache的设计应考虑到各种典型的情况，这可以通过调整Cache设计中的参数来实现，例如Cache中内存块的大小，可以通过调整它的大小，来适应具体的应用情况。在实现中，可以暂停读写，把内存中采用Write-Back

(下转第26页)

RT Embedded http://www.kontronn.com

站建立对市局表的快照(Snapshot)和视图(View),通过Snapshot可以实现对远程数据的快速查询,而View则可用于数据的复杂访问。而对第(2)类数据,市局数据库中建立Table,数据存放于数据发生处,如市局向院站转发报文,则有关此文件的内容、处理情况等数据都记录到院站数据库中;在市局数据库中建立链路(Database Link),可以通过直接访问View实现对数据的修改,通过Snapshot可对这些数据快速查询。

第(3)类数据,属于图像(形)的属性数据。可在市局、院站各存放一份ICAS属性数据,市局、院站系统可直接通过ORACLE数据库中的远程Table相连来操作属性数据。随着ORACLE数据库的属性数据的变更,系统的更新程序要时时进行刷新工作,以确保市局、院站二级网络的数据一致性。

上述3类数据使用Oracle数据库系统存储和管理,并提供关系表定义、关系表删除、关系表查询、关系表维护等操作功能。

4. 空间数据的管理

在空间数据库中,每个记录与一个图像或图形相对应,包含空间对象的定义和该对象所对应的数据属性。从而可以从图像入手对数据进行查询和处理。图像的数据属性可以由使用者来定义。

图像(形)数据是指图像本身的信息,如第(4)类数据,这类图像数据管理的对象就是图像。根据像素构成定义了3类,①有CCD摄像机和图像采集卡采集的数字化矢量数据;②由扫描仪录入的点阵数据;③由测试参数分析统计后得到的图表数据。图像(形)数据采用操作系统文件格式存储。常用格式有MG、GIF、TIFF、PCX、BMP和xls等。

空间数据用ICAS管理,ICAS提供接口与ORACLE管理的属性数据相连接。为了使空间数据与属性数据得到有效连接,且实现市局一院站的空间数据同步,可以在院站数据库建立图像(形)修改表,在市局建立对应的View,通过视图读取图像,并修改记录表。即各节点的数据库建立有关的数据对象Database Link,View,Synonym,Snapshot,Table,Index等,使当图像数据受到各种形式的访问时,均会被记录下来,把数据库更新,这样使DDBMS系统即做到了共享性,又确保了一致性[6](实时性)。

六、小结

信息网络是信息化社会发展之趋势,分布式医学数据库是网络数据库发展之必然,图像、图形等空间数据的信息量非常大,是存档与通信之"瓶颈"。根据空间数据本身固有之特性,利用其属性则是解决此"瓶颈"问题的突破口。经过进一步实践和不断完善,相信在未来的信息网络上一定能较好地解决这一问题。

参考文献

[1]Serra R & Zanarini G. Complex System and Cognitive Process. Berlin: Springer- Verlag, 1990

[2]毕亚雷、赵廷、尹宏、许泽卫、田晓敏,试论PACS系统,中国医学影像技术,Vol 13,No. 4, 1997

[3]齐德昱、徐建闽、毛宗源,客户/服务器模型的一种拓广,计算机工程与应用,Vol 35,No. 3, 1999. 3

[4]孟小峰编著,关系数据库管理系统Oracle原理与应用,北京:电子工业出版社,1993年。

[5]胡贯荣、谢美意、庞丽萍、李胜利,分布式实时消息通信的设计与实现,华中理工大学学报,Vol 27,No. 10, 1999. 10

[6]刘云生、李国徽,实时数据库的准一致性可串行化并发控制,计算机学报,Vool 35,No. 3, 1999. 3

(收稿日期:2000年8月30日)

(上接第31页)

技术的内存块刷新到硬盘中去,并重新分配内存块,根据RAID的配置数据重新建立新的映射,整个过程很快完成,并不会影响硬盘上的数据。通过动态调整Cache的设置来提高RAID中读写的效率。

五、小结

在RAID系统中的Cache的设计过程中,要根据RAID系统在实际应用中具体的读写情况来采用相应的技术,而不能简单地照搬计算机中Cache的设计情况。Cache设计的好坏,对RAID的系统有很大的性能影响。

参考文献

[1]Paul Massiglia, The RAID BOOK (Sixth Edition), RAID Advisory Board, February 1997

[2]张昆藏,《计算机体系结构—奔腾PC》科学出版社,1999

[3]郑纬民、汤志忠,《计算机系统机构》清华大学出版社,1998

[4]Friedhelm Schmidt. The SCSI Bus & IDE interface Protocol, Applications & Programming, Addsim Wesley Longman. 2, 1998.

Coated in Master PDF Editor

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

Creator Master PDF Editor

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
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

Creator Master PDF Editor

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

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

Created in Master PDF Editor

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

邀请注册码

关注论坛公众号

Created with Master PDF Editor

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]

邀请注册码

关注论坛公众号

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

Created with Master PDF Editor

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

Creator Master PDF Editor

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

Creat by Master PDF Editor

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
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

邀请注册码

关注论坛公众号

Create by Master PDF Editor

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
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
38. 基于 ARM11 和 Zigbee 的人员定位防丢器

邀请注册码

关注论坛公众号

Created with Master PDF Editor

39. 基于 NAND FLASH 的嵌入式系统启动速度的研究
40. μCOSⅡ在 ARM7 上的移植
41. 基于 ARM11 的嵌入式视频采集系统设计

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
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
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

邀请注册码

关注论坛公众号

Created with Master PDF Editor

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

邀请注册码

关注论坛公众号

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

Create by Master PDF Editor

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

邀请注册码

关注论坛公众号

Created in Master PDF Editor