# RAID 及并行预取技术分析

应红霞

摘　要：介绍目前常用的　RAID级别。并从　I/O响应时间、磁盘调度算法、数据预取策略、　Cache替换算法方面分析了并行预取技术。最后提出了设计并行预取系统的若干建议。

关键词：　RAID；并行预取；　Cache；　I/O响应时间；　I/O操作次数

中图分类号：　TP333　　文献标识码：　C　　文章编号：　1671-9751(　2006)　03-0067-05

## 一、RAID 介绍

### (一) 什么是 RAID

在网络得到了飞速发展的今天，对磁盘的存储容量、存取速度、可靠性及安全性要求不断提高，　CPU、内存与磁盘之间的速度不匹配问题日益突出。实验证明目前硬盘数据传输的“瓶颈”不在 PCI，也不在硬盘总线，而在于硬盘盘体本身；且硬盘的传输速度在现在技术下不可能达到 SCSI 甚至 IDE 的接口速度。在这种大环境下，RAID 磁盘阵列技术应运而生。

RAID (Redundant Array of Independent Disks) 简称磁盘阵列，是一种将多个类型、容量、接口，甚至品牌一致的物理硬盘按不同方式组合起来的硬盘组（逻辑硬盘），能快速、准确、安全地读写磁盘数据，在数据存储容量、数据读取速度和可靠性方面比单个硬盘更优越。

### (二) RAID 技术规范简介

组成磁盘阵列的不同方式成为 RAID 级别 (RAID Levels)。不同的 level 针对不同的系统及应用。在这里重点介绍目前常用的 RAID0、RAID1、RAID10 及 RAID5。

1. RAID0

该级别采用数据分带和块交叉访问的方法。RAID 0 至少使用两个磁盘驱动器，并将数据分成从 512 字节到数兆字节的若干块，这些数据块被交替写到磁盘中。第 1 段被写到磁盘 0 的连续区域中，第 2 段被写到磁盘 1 的连续区域中，……，当系统到达阵列中的最后一个磁盘时，就写到磁盘 0 的下一分段，依次类推。数据分带将 I/O 负载平均分配到所有的驱动器，以避免磁盘访问不均匀。由于多个硬盘可以同时写或读，大大提高了系统的性能，故 RAID 0 是所有 RAID 中速度最快、利用率最高和成本最低的一种模式，也是目前主板支持较多的一种。但是，由于 RAID 0 是不提供数据冗余，因此一旦用户数据损坏，损坏的数据将无法得到恢复。如果其中的任何一块磁盘出现故障，整个系统将会受到破坏，无法继续使用，故可靠性较差。RAID0 磁盘阵列如下图：

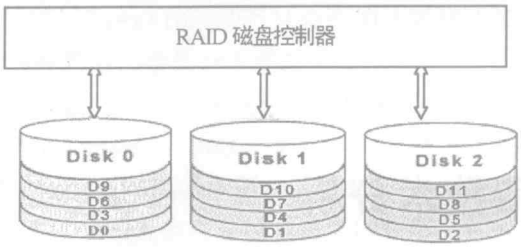



图 1　RAID0 磁盘阵列图

2. RAID1

又称镜像结构，即传统的镜像磁盘。它用两个完全相同的硬盘工作，数据同时写到两个硬盘上，一个作为工作盘，一个作为镜像盘。一旦工作盘发生故障，镜像盘可自动接替工作，使系统工作不中断，这种阵列可靠性最高，但磁盘利用率只有 50%。在性能上，读数据时，多个磁盘可同时进行，速度较快；写入数据时，由于相同的内容要写两次，故速度会慢些。RAID1 一般用于热备份关键数据，适用于银行、证券等要求高可靠性强对成本不敏感的领域。RAID1 磁盘阵列如下图：

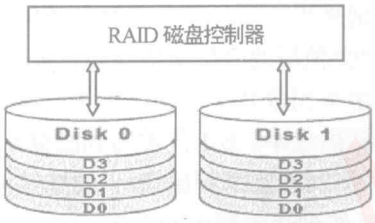



图 2　RAID1 磁盘阵列图

创建镜像时建议使用大小、型号及制造商相同的磁盘。如果镜像分区的大小不同，必须调整热修复重定向表，而且较大分区的多余空间将会丢失。另外，镜像只能防止在硬盘本身出故障时丢失数据，不能防止在硬盘通道其他部件出故障时丢失数据。解决的方法是在磁盘阵列中增加硬盘控制器来增加硬盘通道，这样可以降低在硬盘通道的任何部件出故障时丢失数据的机率。

3. RAID 0+1

RAID 0+1 是 RAID 0 和 RAID 1 的组合形式，也称为 RAID 10。RAID 0+1 是存储性能和数据安全兼顾的方案。它在提供与 RAID 1 一样的数据安全保障时，也提供了与 RAID 0 近似的存储性能。

由于 RAID 0+1 也通过数据的 100%备份提供数据安全保障，因此 RAID 0+1 的磁盘空间利用率与 RAID 1 相同，存储成本高。RAID 0+1 的特点使其特别适用于既有大量数据需要存取，同时又对数据安全性要求严格的领域，如银行、金融、商业超市、仓储库房、各种档案管理等。

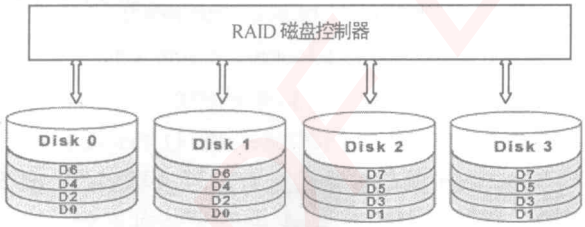
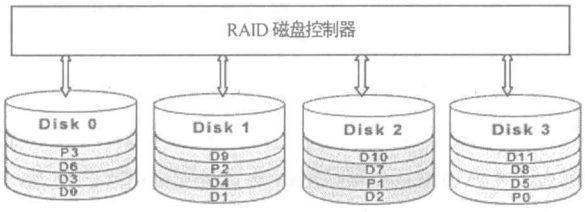



图 3　RAID0 磁盘阵列图

4. RAID5

RAID5 是一种存储性能、数据安全和存储成本兼顾的存储解决方案，它可以理解为 RAID 0 和 RAID 1 的折衷方案。RAID 5 不对存储的数据进行备份，各块独立硬盘进行条带化分割，相同的条带区进行奇偶校验，数据和相对应的奇偶校验信息存储到 RAID5 阵列的各个磁盘上。由于多个数据对应一个奇偶校验信息，故 RAID 5 的磁盘空间利用率要比 RAID 1 高。n 块硬盘组成的 RAID 5 阵列可以有 n － 1 块硬盘的容量，存储成本相对较低。当 RAID 5 的任何一块硬盘上的数据丢失，均可以通过校验数据推算出来。但是如果 1 个硬盘出现故障以后，整个系统的性能将大大降低。故 RAID 5 的安全保障程度要比 RAID1 低。RAID 5 具有和 RAID 0 相近似的数据读取速度，只是多了一个奇偶校验信息；写入数据时，由于要进行奇偶核校验计算，故写入磁盘的速度比对单个磁盘进行写入操作稍慢。RAID5 结构见下图：




RAID5（P0、P1、P2……为奇偶校验）

图 4　RAID5 磁盘阵列图

RAID 级别的选择有 3 个主要因素：可用性（数据冗余）、性能和成本。如果不要求可用性，

选择 RAID0 以获得最佳性能。如果可用性和性能是重要的而成本不是一个主要因素，则根据硬盘数量选择 RAID 1。如果可用性、成本和性能都同样重要，则根据一般的数据传输和硬盘的数量选择 RAID3、RAID5。在实际应用中，具体采用哪种级别的阵列，要根据企事业单位的要求来定。

## 二、RAID 阵列系统中的并行预取技术

### (一) 传统的 RAID 系统分析

在常规的 RAID 系统中，进程在读数据的过程中，向主机返回数据、读状态及接收下一条读命令是完全串行的。加上 CACHE 容量、CACHE 替换算法和小块数据写造成的频繁 I/O 操作等因素的影响，使得一次 I/O 操作的平均时间延长，I/O 操作次数也增加，使系统性能受到很大制约。

下面要谈的并行预取技术可以改善 I/O 响应时间，降低 I/O 操作数次，提高 RAID 的性能。

### (二) 并行预取系统分析

#### 1．并行预取系统的基本思想

并行预取即在磁盘接受 I/O 操作前，根据某些原理或推测，将接下来访问可能性较大的数据块预取到 CACHE。如果访问 CACHE 命中，就可以直接从 CACHE 取出而不必访问磁盘，省略了磁盘驱动器服务时间。如果是写数据，可先写入磁盘 CACHE，等整个应用程序执行结束后，再写入磁盘。另一方面，当进程在磁盘预取数据期间，利用 CPU 暂时空闲的机会，磁盘可以接受下一个 I/O 命令，此时如果要提取的数据刚好在 CACHE 中“命中”的话，便可和上述进程实现并行预取。

并行预取的主要目标是通过对多任务的并行执行和对数据、命令的有效预取来减少 I/O 的平均响应时间和 I/O 操作次数，由此改善系统性能。

要实现并行预取，在磁盘和主机之间需要设置磁盘 CACHE 和主机 CACHE。且容量要达到一定要求，否则会造成 CACHE 频繁替换数据；若替换算法选择不当，也会造成上述情况，严重影响系统性能。

#### 2．并行预取的若干关键技术

根据并行预取系统的设计目标，并行预取的关键技术主要有 I/O 响应时间、数据预取策略、CACHE 容量和 CACHE 替换算法等。

(1) I/O 响应时间

I/O 响应时间即从主机发出 I/O 请求到完成 I/O 操作所需的时间。它由请求访问磁盘等待时间、磁盘服务时间及磁盘适配器接口传输时间 3 部分组成。其中磁盘适配器接口传输时间指主机与磁盘适配器接口之间的数据传送时间，一般由硬件设计时固定。因此主要从请求访问磁盘等待时间和磁盘服务时间两方面来考虑。

1) 请求访问磁盘等待时间

若主机 I/O 请求按泊松分布到达，请求到达率为λ，假设每次主机 I/O 请求的平均磁盘驱动器服务时间为 T（且 T=T 磁盘服务时间 +T 磁盘适配器接口传输时间），T 的方差为δ2，那么根据 M/G/1 排队模型，平均排队等待时间 T 排队 =λ（δ2+T2）/2（1-λT），λT≤1；排队等待时间与 I/O 服务时间成正比，当 I/O 服务时间增大时排队等待时间也相应地增大。

2) 磁盘服务时间

磁盘服务时间由寻道时间、旋转延迟时间和扇区与磁盘适配器接口之间的传输时间组成。其中寻道时间占总时间的 70%到 80%左右，因此寻道时间是关键。影响寻道时间的一个主要因素是移臂调度算法，该算法选择是否恰当直接影响数据预取和并行执行的效率。

在服务器 RAID 阵列系统中，往往有很多同步读请求，此时较好的移臂调度算法是预期调度算法。根据程序执行的局部性原理，相同进程的连续读请求往往会发生在磁盘的相邻的物理块上，若调度器在满足一个读请求后能延迟一小段时间，查看是否有新的读请求发生，若新的读操作能与正在执行的读操作合并，则可以减少寻道时间，增强整个系统的性能。这就是预期调度背后的原理。由于进程在磁盘中预取数据期间，CPU 处于空闲状态，此时可调度下一个读请求，若该读请求所需的数据刚好命中，则相当于省略了寻道时间，从而让两个原本串行执行的读命令达到一定程度的并行预取执行，在很大程度上减少了 I/O 磁盘服务时间。

下面是关于在 LINUX 系统中的 3 种移臂调度算法的一份测试报告，内容是系统在后台读一个大文件的同时，读取内核源代码树目录中的每

个文件，测试结果如下：

| I/O 调度器和内核 | 耗费时间 |
|---|---|
| LINUX 2.4 上的 LINUX 电梯调度 | 30 分 28 秒 |
| LINUX 2.6 上的 LINUX 时限调度 | 3 分 30 秒 |
| LINUX 2.6 上的 LINUX 预期调度 | 15 秒 |

从上述测试结果中可以看出预期调度的性能明显高于其它两种。

(2) 数据预取策略

何时预取数据，预取哪些数据，对降低系统的 I/O 响应时间，减少 I/O 次数有着很大的影响。在这里，主要分析访问磁盘 CACHE 失效时预取和小块写时预取两种情况。

在 RAID 阵列系统中，数据往往是平均分布在阵列的各个磁盘中的，逻辑上相邻的数据块往往存放在相邻的磁盘上，此时，如果某次访问磁盘 CACHE 失效，则根据程序执行的空间局部性原理，一旦某个位置被访问，很有可能它附近的位置也要被访问。所以不但将各块所在磁道的若干数据块一起读出，而且将该数据块所在校验组的其余数据也一起读出。这样可以减少同一磁盘上相邻数据块的寻道时间。研究表明，普通磁盘大约 55 ～ 77%的数据请求是顺序的，因此这样的预取是非常有效的，可以大大减少 I/O 响应时间。

磁盘阵列的小块写问题是制约阵列性能提高的另一关键因素。由于访问磁盘的命令中的数据大小不同，有的数据刚好是一满校验组，这种情况下可并发操作所有盘的读写，仅需一次 I/O 操作即可。如果是不满校验组的读，可直接对涉及的盘进行并发读操作。但如果是不满校验组的写，就必须从同一校验组的其它盘先读出旧的数据和旧的校验，计算出新的校验，再启动磁盘操作，所以总共需至少两次磁盘 I/O 操作。根据程序执行的局部性原理，采用将小块写与预读结合进行的策略。在写小块数据到磁盘的时候，可以对磁盘阵列中处于同一磁道、同一校验组的数据并行预读。这样可以减少由于小块写操作带来的 I/O 操作次数，提高数据传输率。当另外，如果是大量的小块写，可以将小块写合并成满块写，然后再并行读写。

(3) CACHE 容量

CACHE 容量是影响数据预读最直接的因素。首先，增大 CACHE 容量，可以预放更多的数据，只要预取策略合理有效，则可以大大提高访问的命中率。其次，CACHE 腾空的机率也会随之降低，从而减少 I/O 操作次数。再者，许多数据的 I/O 传送可直接在 CACHE 与主机之间重复进行。实验结果表明，通过增大 CACHE 容量甚至可获得大约 300% 的性能增长。当然，CACHE 容量的增大也并不是无限制的，一方面受板卡插槽的限制，另一方面，研究表明，当 CACHE 容量增大到一定程度，会出现性能饱和，那时性价比会急剧下降。

(4) 缓存替换算法

替换分为请求响应替换和预先计划替换。预先计划替换能保证 CACHE 中始终有空闲区。请求响应替换发生在向 CACHE 中预放数据时；或访问 CACHE 失效，要转向访问磁盘，然后将访问过的数据块写入 CACHE 时；此时若 CACHE 满，则应从 CACHE 中淘汰一部分数据块。不管是哪种替换，都应该选择一种合适的替换算法来腾空 CACHE; 在腾空时还要考虑在哪些范围内淘汰数据块，即替换数据的策略。

1) 替换算法：替换算法一般有 FIFO、LRU、LFU 等。在程序和数据顺序访问的系统中，一般采用 FIFO 算法可以达到很好的命中率，如视频系统；在随机访问的系统中，一般采用 LRU 或 LFU 较多，但效果不是很理想，在此我们可以采用一种改进的 LRU 算法，称为 MRU 算法。如下图：

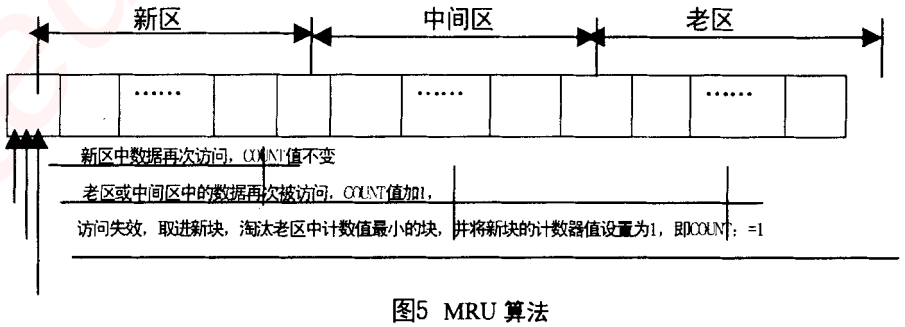



图5 MRU 算法

该算法采用 3 个区，新区用来存放刚被访问过的数据，老区中的数据表示最久没有被访问了，每个数据块用一个计数器 COUNT 来统计被访问的次数。每个区中的数据只要被访问了，就将其移出放到新区头部，其它数据依次后移。若是新区中的数据被命中，则 COUNT 值不变，中间区或老区中的数据被命中，则 COUNT 的值加 1。若访问数据没有命中，则在老区中淘汰 COUNT 值最小的数据块，然后将新数据块放入新区的头部。有了中间区作过渡，大大降低了哪些在新区中没有被快速访问，但其 COUNT 值仍为 1 的数据被快速淘汰的可能性。实验证明，改进后的策略比简单的 LRU、LFU 算法有明显提高，降低了无用数据在磁盘和 CACHE 之间的传输，从而减少磁盘和磁盘缓存之间的 I/O 操作次数。

2）替换策略：为了提高替换的有效性，避免“抖动”现象，一般选择在暂时没有 I/O 请求的进程中选择数据替换。在向 CACHE 中写小块数据时，如果是旧数据块和奇偶校验信息同时存在，则一般淘汰旧数据块。因为旧数据只是在对应数据块回写时用到，而奇偶校验信息在该校验组的任何数据块回写时都能用到，例如对于 N+1 阵列结构，一个奇偶校验块可为同一奇偶校验组中的 N 个数据块回写提供服务，而旧数据只是在该块数据回写时用到.而保存 N 个 CACHE 空间的旧数据块效果和保存一个 CACHE 空间的奇偶校验信息的效果相同。

## 三、总结

经过对并行预取技术的分析，在此提出自己的一点看法，供大家参考。一个高效的并行预取系统可以在如下几方面加以设计或改善：

（一）采用合适的预取算法来生成有效的预取命令。理论上，生成的预取指令数越多，则可合并的命令数越多、预取效果越好，I/O 响应时间越小。移臂调度算法可以采用预期调度算法，可以缩短磁盘服务时间，从而降低系统的平均 I/O 响应时间。

（二）RAID 阵列中的每个磁盘至少配置一个 CACHE，一方面可以预放更多的数据，另一方面可以保证一个 CACHE 满时，至少还有一个可以使用。而且最好采用掉电不丢失信息的 CACHE，对于主机而言，可获得很好的写响应特性。

（三）在设计时，以下三种情况可进行预取：访问 CACHE 失效时、小块数据写时和 CACHE 空闲时。

（四）在顺序访问系统中，CACHE 替换算法采用 FIFO 为佳。但在随机访问系统中，为了减少无用数据在磁盘和 CACHE 之间传输，CACHE 替换算法可以采用改进的 MRU 算法，实验证明，该算法比 LRU 和 LFU 有较好的性能，能够降低 I/O 操作次数。

## 参考文献

[1] 王 宇. 廉价磁盘冗余阵列 RAID 的技术策略 [J]. 哈尔滨商业大学学报（自然科学版），2002,(3).

[2] E B James, J F Michael. Reducing Cache miss rates using prediction Caches. Computer System Laboratory, Stanford University, Tech Rep CSL—TR—96—707, 1996.

[3] 何青林，冯 丹. 面向多应用环境 RAID 系统的智能预取和缓存调度 [J]. 计算机工程与科学，2004,(9).

[4] Stallings 著，陈 渝 译. 操作系统——精髓与设计原理 [M]. 北京：电子工业出版社，2006.

## RAID And Analysis of the Parallel Pre-fetching Technology

YING Hong-xia

**Abstract:** This paper introduces commonly used RAID levels and analyzes the parallel pre-fetching technology on I/O response time, disk dispatch method, data pre-fetching strategy and Cache substitution algorithms. It also puts forward several suggestions on the parallel pre-fetching system design.

**Key words:** RAID, parallel; Cache; I/O response time; I/O operation times

# 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
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

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



## 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



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 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
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

# 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

# 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
35. PowerPC 处理器原理

# 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
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
42. 基于 ARM11 的视频监控系统设计
43. μC-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 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
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

# 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