RT Embedded http://www.kontronn.com

基于 VPX 标准的 RapidIO 交换和 Flash 存储模块设计

李 宾，马晓川，鄢社锋，杨 力
(中国科学院声学研究所，北京 100190)

摘要：针对某阵列信号处理系统中的互连总线和数据存储问题，提出了基于 VPX 标准的 RapidIO 交换和 Flash 存储模块的设计方案，并完成了该模块的软硬件设计。介绍了新一代高速串行总线 RapidIO 和 VPX 标准，根据 VPX 标准设计板卡，利用 Tsi578 交换机设计 RapidIO 网络实现多 DSP 系统的并行处理结构。研究了 RapidIO 的数据传输操作和 TMS320C6455 DSP 的 SRIO 启动方式，给出了软件设计流程。采用 NAND Flash 存储器完成了数据存储功能。最后列出了模块的性能指标，并在实际信号处理系统中测试了 DSP 间 RapidIO 的数据传输速率，高达 600MB/s。通过测试验证了本设计的可行性、先进性和稳定性，可以作为数字系统中互连总线设计和数据存储设计的参考。

关键词：VPX；RapidIO；互连总线
中图分类号：TN911　　**文献标识码**：A　　**文章编号**：1000-3630(2011)-06-0528-05
DOI 编码：10.3969/j.issn1000-3630.2011.06.013

Design of RapidIO switch and flash storage module based on VPX standard

LI Bin, MA Xiao-chuan, YAN She-feng, YANG Li
(*Institute of Acoustics, Chinese Academy of Sciences, Beijing* 100190, *China*)

Abstract: A design of RapidIO switch and Flash storage module is proposed aiming at the interconnect bus and data storage problem met in an array signal processing system. The hardware and software design of the module are realized. Firstly, the serial RapidIO bus and VPX standard are introduced. The module is designed according to VPX standard. Tsi578 switch is used in RapidIO network to realize the multi-DSP parallel processing system. RapidIO data transfer operation and RapidIO boot mode of the TMS320C6455 DSP are both studied. The software design flow is presented then. NAND Flash is used to realize data storage capability. Finally, the module performance is listed and the data transfer speed is tested in the actual signal processing system. The data transfer speed is up to 600MB/s. Experiments show that this design is advanced, realizable and stable, which can be used as a reference design of interconnect bus and data storage.

Key words: VPX; RapidIO; interconnect bus

0 引 言

阵列信号处理技术已经广泛应用于雷达[1]、声纳[2]等各个领域，随着应用的不断深入，阵列信号处理技术也在不断发展，复杂信号处理算法如自适应波束形成、信号检测等对系统的计算能力和实时性要求越来越高。阵列信号处理系统面临着两大挑战：一是计算能力，系统能否满足复杂算法的计算量需求；二是系统内部互连总线，系统内部数据能否进行高速实时交换。随着集成电路制造工艺的提高和更先进的处理器架构的使用，处理器的性能不断提高，新的功能强劲的高性能数字信号处理器(DSP)也相继推出，如 ADI 公司的 TigerSHARC 系列和 TI 公司的 C6000 系列。处理器速度的提高大大改善了阵列信号处理系统的性能，但是系统总线频率的增长速度已经跟不上处理器内核性能的增长速度。处理器速率高于处理器和外设之间的连接或系统中多处理器之间的连接传输速率，系统互连总线已经成为制约系统性能提高的瓶颈。近年来，总线技术向着点对点方式、串行 I/O、差分信号的方向发展，总线频率也不断提高。RapidIO[3]就是面向嵌入式系统开发提出的高可靠、大带宽的新一代高速互连技术，是解决系统间、系统内部板间和芯

RT Embedded http://www.kontronn.com

片间高速实时数据交换的互连方案。所以，利用RapidIO 总线进行互连的多 DSP 阵列信号处理系统，可以同时解决复杂算法对计算能力和互连总线两方面的需求。

阵列信号处理系统在运行过程中都需要实时、可靠地存储大量的关键信息，便于进行历史数据分析，检验系统运行状况，数据存储作为一项不可或缺的重要功能迫切需要一种容量大、体积小、功耗低的存储设备。而近年来，NAND Flash[4]以其存储密度高、功耗低、抗震性好、寿命长等优势广泛应用于各个领域，成为数据存储的最佳选择。

本文针对某阵列信号处理系统中的互连总线和数据存储问题，提出采用 RapidIO 总线实现系统内部板间和芯片间高速实时数据交换，采用 NAND Flash 存储器实现实时可靠地大容量数据存储。介绍了新一代的高速串行总线 RapidIO 和 VPX[5]标准，提出按照 VPX 标准设计系统背板和模块，采用 Tsi578[6]交换机构建 RapidIO 网络实现多 DSP 系统的并行处理结构。研究了 RapidIO 的数据传输操作，阐述了 TMS320C6455 DSP 的 SRIO[7](串行 RapidIO)启动方式。文中还介绍了 NAND Flash 存储器，给出了 NAND Flash 实现数据存储的硬件和软件设计方案。最后列出了该模块的性能指标，并在阵列信号处理系统基础上测试了 DSP 间 SRIO 的数据传输速率。

1 RapidIO 交换的实现

1.1 RapidIO 技术

RapidIO 技术最初是由 Freescale 和 Mercury 共同研发的一项互连技术，目标是为嵌入式互联系统内部提供高可靠、大带宽的高速数据总线。第一个标准 RapidIO 1.0 于 1999 年完成了制定。2003 年 10 月，国际标准组织和国际电工委员会(IEC)一致通过了 RapidIO 互连规范。2005 年 6 月，随着 RapidIO 协议 1.3 版本的发布，RapidIO 技术逐步进入成熟应用阶段。在随后的一段时间内，各大嵌入式芯片厂商相继推出了各自基于 RapidIO1.3 协议版本的芯片产品，如 TI 公司的 DSP 芯片 TMS320C6455，Tundra 公司的交换机芯片 Tsi568、Tsi578。RapidIO 技术采用 3.3V LVDS 接口电路，支持 8/16 位并行和 1x/4x 串行两种接口。其中串行 RapidIO 接口适合长距离传输，采用时钟数据恢复同步技术和 8B/10B 编码机制，支持 1.25 GHz、2.5 GHz 和 3.125 GHz 三种工作频率，能满足多种不同应用的需求。串行差分模拟信号技术满足了系统对管脚数量的限制，以及对背板传输的需求，高达 10Gbps 的带宽、低时延和低软件复杂度的特点使 RapidIO 成为最佳的互连技术。2007 年 6 月随着 RapidIO 协议 2.0 版本的发布，RapidIO 的可选速度在原有的基础又增加了 5Gbps 和 6.255 Gbps 两种，最大的信号传输带宽将可以达到 100Gbps，为嵌入式系统性能的快速增长奠定了坚实的基础。

1.2 VPX 标准

VPX 是基于高速串行总线的新一代总线标准，电气信号定义与原 VME 标准完全不同，但是 VPX 总线标准继承了原 VME 标准中机械结构及导冷抗震方面的优势，是加固嵌入式系统的最佳选择。VPX 标准把所有连接器都换成了支持高速差分信号的 Multi-GigRT2 连接器，以获得更高的传输速度和更多的信号引脚。这种连接器是硅晶片式结构，它的触点及内部布线是利用微型印制电路板设计而成的，具有连接紧密、插入损耗低、误码率低等优点，VPX 的高速串行结构使每对差分信号能够有超过 6.25 Gbps 的数据率。VPX 标准继承了现有的 IEEE1101 3U 和 6U 标准板块尺寸定义，在一块典型的 3U VPX 板上使用一个 8 列 7 行的 RT2 连接器和 2 个 16 列 7 行的 RT2 连接器，分别定义为 P0，P1，P2。

1.3 RapidIO 交换方案设计

本文介绍的基于 VPX 标准的 RapidIO 交换和 Flash 存储模块，是某阵列信号处理系统中的关键部分，该模块的设计对系统的整体架构起决定性作用(图 1)。RapidIO 系统多采用基于交叉交换的拓扑结构，端点设备间不是直接连接而是通过交换机互连，任意两个端点设备间为对等关系，可以方便进行数据交换和并行运算。在该阵列信号处理系统中，采用多 DSP 并行运算结构，运用 RapidIO 交换机设计总线网络，实现系统内部板间和多 DSP 间的高速互连，RapidIO 总线网络的设计提高了系统的灵活性，可以方便地进行共享式或分布式处理。该系统选用 TI 公司的 TMS320C6455 高性能定点 DSP 作为运算单元，C6455 支持 RapidIO1.3 协议的串行接口(SRIO)，支持 1x/4x 连接方式，可选 1.25 Gbps、2.5Gbps 和 3.125Gbps 三种连接速率。

在本设计方案中，RapidIO 交换芯片选用 Tsi578，它是 Tundra 公司的推出的第三代 SRIO 交换芯片，支持高达 80 Gbps 的聚合带宽，具有 8 个 4x 端口，每个 4x 端口可以配置成两个独立的 1x 端口，支持 1.25 Gbps、2.5 Gbps、3.125 Gbps 三种传输

RT Embedded http://www.kontronn.com

速率。SRIO 的路由和交换是通过每个端点设备的 ID 号来实现的，每个端点设备都会分配一个唯一的 ID 号，当一个端点发出一个数据包时，在它的包头中包含有目的终端的 ID 号和发送源端的 ID 号。Tsi578 的每个端口上都有一个交换路由表，根据路由表就可以决定此数据包由哪一个端口送出。同时 TMS320C6455 DSP 上集成了标准的 SRIO 1x/4x 端口，这样就可以实现 DSP 和 Tsi578 之间的无缝连接。

在本设计方案中，RapidIO 交换和 Flash 存储模块是根据 VPX 标准设计的，符合 3U 尺寸，使用 3 个 RT2 连接器。板卡的结构如图 1 所示，Tsi578 交换芯片的一个 4x 端口连接板上的一片 TMS320C6455 DSP，该 DSP 用于对系统的 RapidIO 网络进行维护管理，另外 7 个 4x 端口通过 P1 和 P2 连接器连到背板，用于实现板间 DSP 的高速实时数据交换。背板也是按照 VPX 标准设计的，支持 3U 标准板卡，背板的结构如图 2 所示。

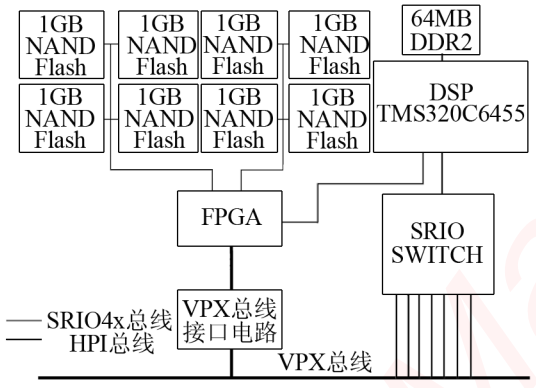

图 1　RapidIO 交换和 Flash 存储模块的结构框图
Fig.1　Block diagram of RapidIO switch and Flash storage module

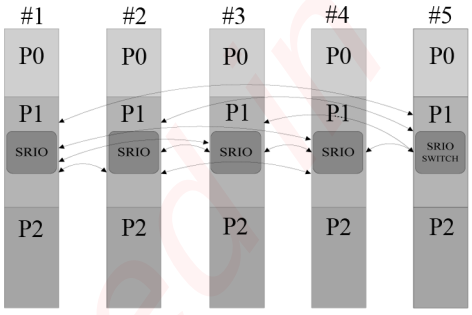

图 2　背板结构框图
Fig.2　Block diagram of back plane

1.4　RapidIO 数据传输研究

RapidIO 采用三层分级的体系结构，分别为：逻辑层、传输层、物理层。逻辑层定义了总体协议和包格式，传输层定义了包交换、路由和寻址机制，物理层定义了电气特性、链路控制、低级错误管理。

RapidIO 操作是基于请求和响应事务的，端点器件通常是通过交换器件互连的，包是系统中的端点器件的基本通信单元，图 3 显示了包是如何在 RapidIO 系统中传输的。一次传输操作开始时，发起者发出一个请求包，该请求包被传送至交换机，交换机发出控制符号确认收到了该请求包，同时将该包转发至目标器件。目标器件收到请求包后完成相应的操作，随后发出一个响应包，该响应包通过交换结构返回到发起者。

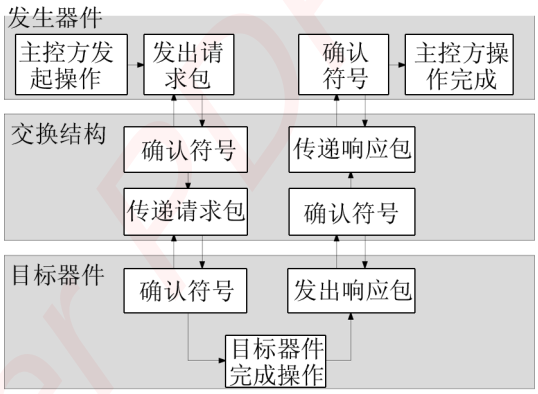

图 3　RapidIO 包的传输过程
Fig.3　RapidIO packet's transfer operation

RapidIO 定义了 6 类请求包格式和 1 类响应包格式，支持读操作(NREAD)、写操作(NWRITE、NWRITE_R)、维护包(Maintenance)、门铃(Doorbell)、消息(Message)等操作。在 TMS320C6455 DSP 中，SRIO 数据传输和 DMA 传输是结合的，当进行 SRIO 传输时，DMA 以自动方式启动。对于发送方来说，DMA 将数据从 L2 内存搬移到 SRIO 端口。对于接收方来说，DMA 将数据从 SRIO 端口搬移到 L2 内存。因此在进行传输时，读写地址是直接包含在包里的，而且此地址就是目标 DSP 中的地址。换句话说，DSP 可以通过 SRIO 对另一 DSP 的 L2 内存直接进行读写操作。

在本方案中，DSP 通过 NREAD、NWRITE 操作实现 DSP 间的数据读写，通过门铃操作对其他 DSP 发起中断请求，通过维护包读写操作实现对 Tsi578 交换机的动态配置。维护包是 Tsi578 交换机识别的唯一一种包格式，通过 DSP 向交换机发送维护包动态配置查找表。查找表包含了每个端口所连器件的 ID，SRIO 传输的包里面包含目标器件的 ID，交换机根据查找表获得该 ID 对应的端口，然后把该包转发到这个端口。图 4 是 SRIO 部分的 DSP 软件设计流程，首先初始化 SRIO 端口，与 Tsi578 端口成功建立链路后，发送维护包动态配置查找表，然后就可以进行读写操作和门铃操作了。

RT Embedded http://www.kontronn.com

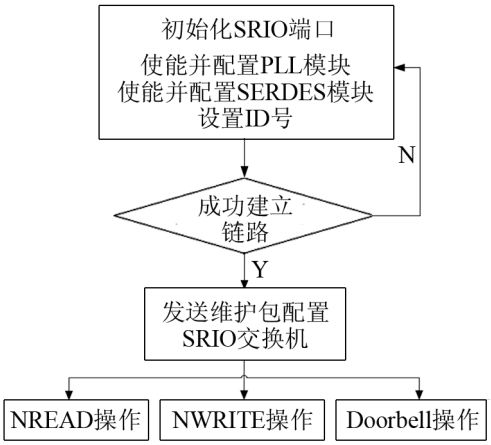

图 4　SRIO 部分的 DSP 软件设计流程
Fig.4　DSP software design flow of SRIO

TMS320C6455 DSP 还支持 SRIO 加载，选择一个主 DSP，其他作为从 DSP，将从 DSP 的启动方式设为 SRIO 启动，这时系统上电时就会对从 DSP 的 SRIO 进行初始化，主 DSP 配置好交换机后，把程序下载到从 DSP 的 L2 内存中，然后主 DSP 发起一个门铃操作向从 DSP 请求中断，从 DSP 就从地址 0x800000 处开始执行程序，至此加载过程完成。

2　Flash 存储的实现

2.1　NAND Flash 介绍

NAND Flash 是一种非易失性存储设备，它没有机械部件，无需寻道时间，所以具有很快的数据访问速度，同时具有存储密度高、体积小、功耗低、抗震性好等优点，非常适合数据存储和文件存储，使得 NAND Flash 广泛应用于嵌入式系统、数码电子产品和军工设备中。在本方案提到的阵列信号处理系统中，需要实时记录采集到的数据和一些关键信息，便于进行历史数据分析，检验系统的运行状况。本设计方案选用的 NAND Flash 芯片是 SAMSUNG 公司的 K9G8G08，其存储容量为 1G 字节，结构上分为 4096 块(block)，每个块由 128 页(page)构成，每页的大小是(2K+64)字节。

NAND Flash 的操作主要有读、编程和擦除三种操作，页是读操作和编程操作的最小单位，块是擦除的最小单位。NAND Flash 的数据线和地址线复用为 8 条 I/O 线，提供单独的控制信号线，命令、地址和数据均通过 8 条 I/O 线传输。NAND Flash 内部有命令、地址和数据三种寄存器，通过命令锁存信号 CLE 和地址锁存信号 ALE 的不同组合选通不同的寄存器。

2.2　NAND Flash 存储设计

在 RapidIO 交换和 Flash 存储模块的设计中，我们选用了 8 片 K9G8G08 NAND Flash，总存储容量达到 8G 字节，硬件设计如图 1 所示。我们选用 Xilinx 公司的 Xc5vlx30t 高性能 FPGA 实现对 Flash 的控制，8 片 Flash 分为两组连接到 FPGA 上，每组 Flash 共用 I/O 线和除片选(CE#)外的控制线，两组 Flash 可以并行工作，大大扩展了数据带宽。DSP 通过 EMIF 总线实现与 FPGA 的数据交换。

软件设计的结构如图 5 所示，在 FPGA 中设计了 3 个 FIFO，即地址 FIFO、数据 FIFO1 和数据 FIFO2。在对 Flash 进行编程操作时，DSP 通过 EMIF 总线向地址 FIFO 写入地址，向数据 FIFO1 写入数据，然后发出一个开始命令，FPGA 则会产生相应的时序，将数据 FIFO1 中的数据写入 Flash，完成编程操作。在对 Flash 进行读操作时，DSP 通过 EMIF 总线向地址 FIFO2 写入地址，然后发起一个开始命令，FPGA 则会产生相应的时序实现对 Flash 的读操作，将 Flash 中的数据写入数据 FIFO2，DSP 再通过 EMIF 总线将数据从 FIFO2 读出。

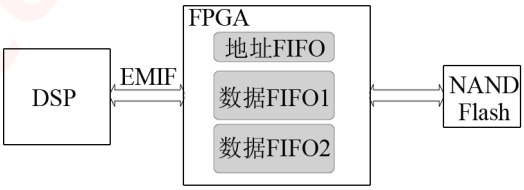

图 5　Flash 存储设计的软件结构框图
Fig.5　Block diagram of Flash storage software design

3　性能测试和分析

本文针对某阵列信号处理系统中的互连总线和数据存储问题，设计了一种基于 VPX 标准的 RapidIO 交换和 Flash 存储模块，实物如图 6 所示，表 1 列出了模块的性能指标。

实际的数据传输速率与理论速率之间往往有

表 1　模块的性能指标
Table 1　Module performance

	性能指标
处理器	1 个 TMS320C6455 DSP 1 个 Virtex-5 FPGA
存储器	DSP 带 2MB 内部 RAM 和 64MB DDR2 8GB NAND Flash
通信能力	任意两个 RapidIO 端点间 10Gbps 全双工通信
计算能力	每秒 20 亿次 32bit 定点乘加运算 或每秒 20 亿次 32bit 定点乘加运算
功耗	小于 10W

RT Embedded http://www.kontronn.com

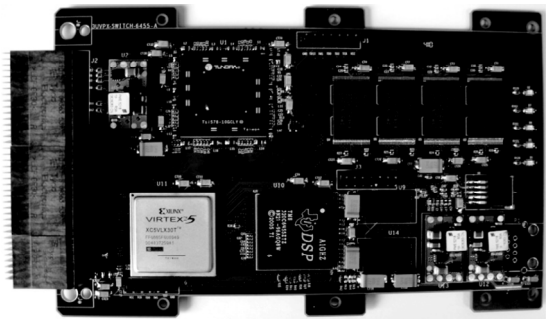

图 6　RapidIO 交换和 Flash 存储模块的实物图
Fig.6　RapidIO switch and Flash storage module

所差距，对本模块在阵列信号处理系统环境中进行了测试，测得 SRIO 端口在 4x 模式下，DSP 间的数据传输速率如表 2 所示。可以看出，对于单包发送 256 byte 数据的传输速率是比较低的，随着包大小的增加，传输速率也在增加，SRIO 的包最大可以传送 4096 byte 的数据，在单包发送 4096 byte 的情况下，SRIO 的传输速率是最高的。

表 2　实测 DSP 间的数据传输速率
Table 2　Actual data transfer speed between DSPs

速率模式 /Gbps	包大小 /Byte	NREAD 操作速率/(MB/s)	NWRITE 操作速率/(MB/s)
1.25	256	179	176
	512	274	271
	1024	346	348
	2048	414	415
	4096	467	462
2.5	256	181	176
	512	313	306
	1024	404	403
	2048	502	501
	4096	573	574
3.125	256	331	326
	512	442	435
	1024	560	563
	2048	608	612
	4096	646	650

4　结 论

本文提出的基于 VPX 标准的 RapidIO 交换和 Flash 存储模块设计，采用 RapidIO 总线作为系统总线，其高可靠、大带宽的特点保证了系统内部板间和芯片间的高速实时数据交换；采用 NAND Flash 存储器实现数据存储，具有容量大、体积小、功耗低的优点；采用 VPX 标准设计背板和各个模块，系统集成密度高、可靠性好。该模块已经成功应用于某阵列信号处理系统中，运行稳定可靠，为系统提供了高性能的数据交换和存储功能。本设计具有很高的通用性，可以作为多处理器系统中互连总线设计和数据存储设计的参考方案。

RapidIO 代表了总线技术的发展方向，随着 RapidIO 技术的进一步完善和发展，采用 RapidIO 总线的体系结构将为多处理器系统的互连带来更好的解决方案，以及更为强大的性能。

参　考　文　献

[1] Skolnik M I. Introduction to Radar Systems[M]. New York: McGraw-Hill, 1980.
[2] Nielsen R O. Sonar Signal Processing[M]. London Artech House, 1991.
[3] Association R T. RapidIO™ Interconnect Specification Annex 1: Software/System Bring Up Specification[EB/OL]. 2008.
[4] Toshiba America Electronic Components Inc. NAND Flash Applications Design Guide[EB/OL]. Revision 1.0, 2003.
[5] ANSI/VITA4.6.0-2007. American National Standard for VPX Baseline Standard[S]. 2007.
[6] TUNDRA Inc. Tsi578 Serial RapidIO Switch User Manual [EB/OL]. 2007.
[7] TI Inc. TMS320C645x DSP Serial RapidIO (SRIO) User's Guide [EB/OL]. SPRU976C, 2009.

Created 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

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

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

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

Created in Master PDF Editor

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

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

Create with 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

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

Created with Master PDF Editor

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

Programming:

1. 计算机软件基础数据结构 - 算法

Created in Master PDF Editor