PCI Express 流控机制的研究与实现

夏寅星，罗浩丁，章建雄

（中国电子科技集团公司第三十二研究所，上海 200233）

摘　要： PCI Express 总线协议通过流控机制降低数据重传概率、调度虚拟通道传输事务，提高数据链路的传输效率。文中对 PCI Express 协议的流控机制进行分析，并提出一种实现流控机制的设计方案，包括发送流控模块、接收流控模块等。该方案在一款 PCIe 接口的以太网卡控制器中实现，通过对流控系统的仿真验证，结果表明该设计方案实现了 PCI Express 流控功能。

关键词： PCI Express 总线；流控；信用量；事务层；数据链路层

中图分类号： TN915　　**文献标识码：** A

Research and implimentation of PCI express flow control mechanism

XIA Yin-xing，LUO Hao-ding，ZHANG Jian-xiong

（**The 32nd Research Institute of China Electronics Technology Group Corporation，Shanghai** 200233，**China**）

Abstract：PCI Express bus protocol can improve the efficiency of data transmission through flow control mechanism to reduce the probability of data retransmission and scheduling virtual channel transmission services. In this paper，the flow control mechanism of PCI Express protocol is analyzed，which proposed adesign to achieve it，including the transmission flow control module and the receives flow control module. The designis implemented in a PCIe interface Ethernet card controller. Through the simulation of the flow control system ，the results show that the design implements PCI Express flow control.

Key words：PCI Express bus；flow control；credit；transaction layer；data link layer

0　引言

PCIExpress（Peripheral Component Interconnect Express，简称 PCIe）总线协议由于具有高带宽、高可靠性、低电压、多通道以及与上一代 PCI 总线技术良好的兼容性等特点[1-3]，被广泛运用于 PC、服务器、嵌入式设备中。PCIe 协议以流量类别（TC）、虚通道（VC）、TC/VC 映射及优先级仲裁机制为事务层数据包（TLP）传输提供有区别的服务质量（QoS）[4]。为了使数据传输不受阻塞，PCIe 收发端均会提供多个虚通道。当链路建立时，配置软件会建立虚拟信道保证 TLP 包可以通过不同的虚通道传输。如果某一虚通道的流控缓冲区已满，则可以报告发送端暂停发送，同时发送端就可以使用另一个虚通道并发送与该虚通道相关的事务。流控机制能够提高数据传输效率，故 PCIe 链路必须实现流控。

通过对 PCIe 流控机制的研究，结合一款 PCIe 接口的以太网网卡控制器，本文提出一种实现 PCIe 流控功能的设计方案。本文的设计方案，在保证功能及效率的情况下通过设置合理的模块参数及模块复用、信号复用等方法，减少模块接口的同时也减少所需的连线，由此降低芯片的面积和功耗。本文用 VerilogHDL 语言对设计进行描述并通过仿真验证流控模块功能的正确性。

1　PCIe 流控机制分析

1.1　PCIe 系统结构

PCIe 协议定义了一种三层传输体系结构[5-7]，从上到下依次为事务层、数据链路层和物理层，每一层又分为发送和接收两部分。发送时，在事务层接

收稿日期： 2015-09-01

作者简介： 夏寅星（1990-），男，硕士研究生，研究方向为 FPGA 设计。

收应用层传入的数据打包成 TLP，并在数据链路层为其添加序列号和冗余校验码(CRC)，在物理层给数据包加上起始符和结束符后编码为串行差分信号发往另一端的 PCIe 设备。在接收部分，处理过程与之相反，物理层接收到串行数据信号通过解码并逐层拆解，最终发送到应用层。数据链路层生成的数据链路包(DLLP)用于实现流控协议、ACK/NAK 协议、电源管理信息传输等[8]。

PCIe 流控协议以“流控信用”(Flow Control Credit，FCC)[9]为单位来报告缓冲区的空间大小。完成包的头 FCC 单位代表 4DW(16 字节)，请求包的头 FCC 单位代表 5DW(20 字节)。数据 FCC 一个单位代表 4DW(16 字节)。不同类型的 TLP 分别用不同的流控信用单独控制，具体分为报告(P)、非报告(NP)和完成(CPL)三种类型。不同类型的事务又包含头缓冲区和数据缓冲区，每个缓冲器里含有 6 种不同的缓冲区[10-11]。流控信用在数据链路层组包，但是其数据是由事务层提供并且在事务层起作用[12]。在链路初始化时，各个接收器通过流控 DLLP 向链路另一端的端口报告其缓冲区的大小(以流控信用为单位)。

1.2 PCIe 流控原理

PCIe 设备在发送数据之前，先检查接收方是否有足够的接收缓存空间。因此，接收方需要通过流控信用将接收方的缓存容量信息传递给发送方。图 1 展示了数据传输过程中的流控信息传递过程。

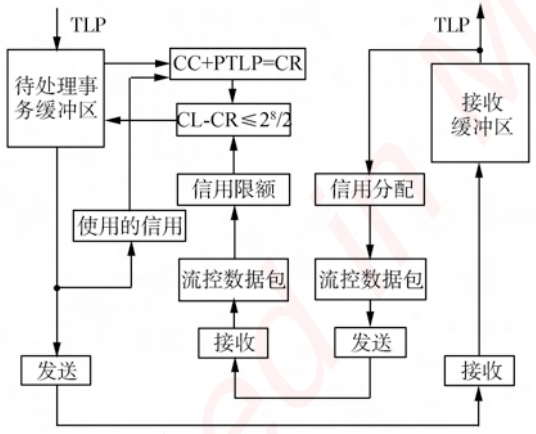

图 1 数据传输和流控信息传输

在发送端设置一个待处理事务缓冲区，其作用是缓存将要发送的 TLP。同时从接收流控 DLLP 数据包，提取出接收方的信用限额 CL(Credit Limit)。根据待处理事务缓冲区中 TLP 的信息可以知道该 TLP 包所需的信用量(PTLP)，加上已经使用的信用量 CC(Credit Consumed)可以得出本次发送所需的信用量 CR(Credit Required)。根据公式(1)判断接收端是否有足够的缓冲区空间来接收待发送的 TLP(PTLP)，其中 fieldsize 为计数器中的位宽。

$$CL-(CC+PTLP)\bmod 2^{[fieldsize]} \leqslant 2^{[fieldsize]}/2 \quad (1)$$

此检查可以保证 CC 加上待发送的下一个数据包所需的信用不会超过 CL。在接收端同样设置一个缓冲区用来保存 TLP，同时根据接收数据包的信息更新信用的分配量 CA(Credit Allocated)。当 TLP 进入接收缓冲区时，CA 并不会发生改变，只有当 TLP 发往应用层后，缓冲区空间增大，CA 才会增加。当流控更新时 CA 会更新到发送端的 CL。根据实际情况和最大缓冲区容量，本文将计数器宽度(即上文中的 fieldsize)设置为头流控使用模 256 计数器(8 位宽)，数据流控使用模 4096 计数器(12 位宽)。

PCIe 中定义了无限流控信用值，一个通告无限流控信用的设备在初始化之后不需要发送流控更新数据包，永远也不会阻塞发送器发送事务。在初始化流控时，设备通过在 FC_Init1 DLLP 的信用字段中发送一个 0 来通告无限信用。一般把 CPL 事务的流控模式设为无限信用模式，因为完成包不能及时到达可能导致设备误以为事务没有完成而造成错误。

2 流控模块设计方案

2.1 流控实现结构

通过对流控原理的分析，根据流控信息的传递过程及其中所需的判断处理仲裁，设计发送流控模块、接收流控模块以及缓冲区来实现对应的功能。本设计使用了 8 个 VC，每个通道的设计完全相同，其中发送流控的数据经过 VC 仲裁器进行优先级判断后传入链路层。接收到的数据则直接进入相应 VC。流控实现的结构如图 2 所示。

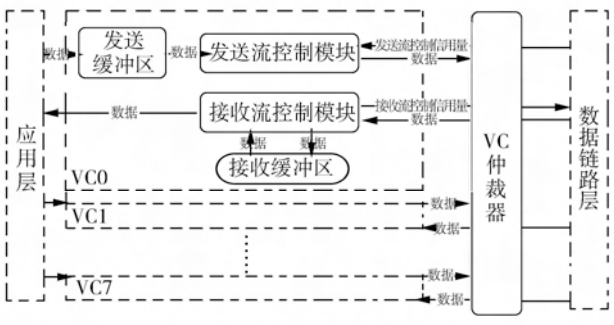

图 2 流控实现结构

每个 VC 中的流控模块包含了一个发送流控模块和一个接收流控模块。发送流控模块接收发送流控信用量并判断另一端是否有足够的空间接收，如果有则从发送缓冲区中提出数据并发送，同时更新内部的信用量。接收流控模块则接收来自另一端的数据并放入接收缓冲区，当应用层从接收缓冲区中

读取数据时,接收缓冲区的变化将更新模块内部的信用量,并把接收流控信用量数据更新到数据链路层通过 DLLP 传输到另一端设备。在本文中,考虑到实际情况把每个 VC 缓冲区设为 8kB。但在设计中该缓冲区大小为可配置的,该数值在今后的应用中可以根据需要进行调整,使之能够适用于其他 PCIe 总线系统。

2.2 发送流控模块设计

发送流控模块分为 3 个层次,最外层的模块主要通过状态机进行转发 TLP,里面包含一个数据处理的子模块,其中又例化了 3 个相同的流控计数子模块分别对 P、NP 以及 CPL 的信用量进行分配管理。发送流控模块结构如图 3 所示。

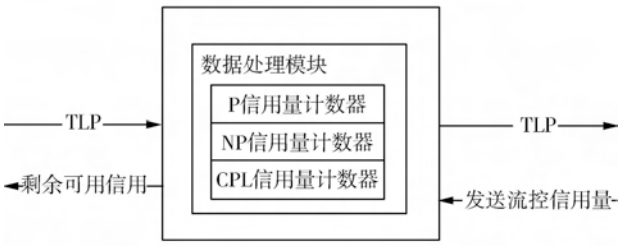

图 3 发送流控模块

当 VC 选通且数据链路层准备接受数据后模块开始判断流控信用量并从缓冲区中请求 TLP 发送。模块的流控更新采用了信号复用的方式,把 3 种类型更新使能独立出来,复用一个 20 位的信号记录信用量,高 8 位为头信用量,低 12 位为数据信用量。这种方式能减少模块的接口数量及模块间的连线。不同类型的信用可用量和允许 TLP 发送的发送信号会反馈给缓冲区。在内部的数据处理主要作用是对传输的 TLP 进行分析计算出其所需的信用量以及对不同类型的包分类并传输给相应的流控计数模块处理。3 个流控计数模块功能完全相同,它会分别处理头信用量和数据信用量,其原理如 2.2 节。计数模块采用了无符号的计数方式能够避免最大值翻转回 0 时产生错误。

发送模块中的 TLP 发送状态机如图 4 所示。当处于 IDLE 时,只有用户端请求发送且流控信用量足够发送下一 TLP 才转换到 ACCEPT 状态。当处于 ACCEPT 状态时,只有数据链路层准备好接收 TLP 且应用层没有发送请求时才回到 IDLE 状态。

2.3 接收流控模块设计

接收流控模块主要结构分为 2 个层次,外层对 TLP 进行接收和分析同时转存到缓冲区,并把相应类型的流控信用更新从子模块传输到数据链路层。由于实际应用中把 CPL 流控设为无限流量模式故

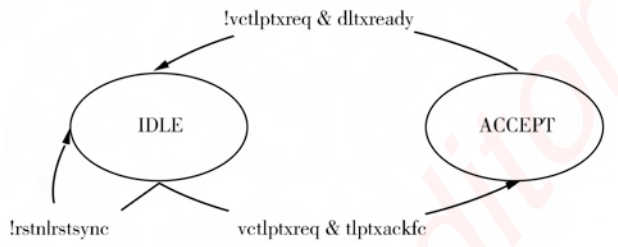

图 4 发送流控模块状态机

接收流控计数模块只需要例化 2 次,分别用于 P、NP 类型的 TLP 传输控制,其结构如图 5 所示。

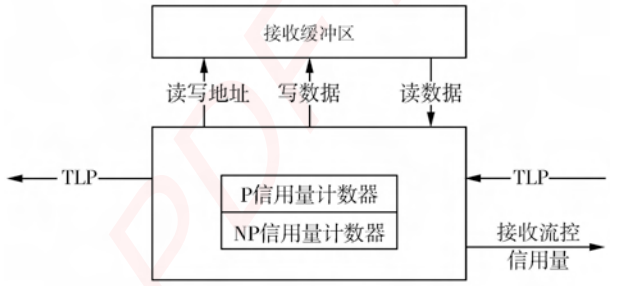

图 5 接收流控模块

接收模块接收的 TLP 会先存储到 VC 缓冲区中,当应用层发送接收请求时再发向应用层。该设计把缓冲区设计为 TLP 的中转区,可以减少缓冲区与接收流控模块之间的通信次数提高响应速度。当从缓冲区发向应用层后,流控信用量才会更新。故接收流控模块中有 2 个状态机,一个用于从数据链路层接收 TLP 写入到缓冲区,另一个用于从缓冲区读取 TLP 到应用层。如图 6 左图,输入 TLP 状态机主要控制 TLP 从数据链路层进入缓冲区中各个状态的变化。当 TLP 包开始传输时,进入 HD0 状态,然后当数据有效时进入缓冲区,如果是结束就进入更新状态。该状态机的作用是为 TLP 在缓冲区中分配空间。

如图 6 右图,TLP 输出状态机主要控制 TLP 从缓冲区输出到应用层的各个状态的变化过程。当缓冲区中有 TLP 时,进入 REQ 状态。收到来自应用层的回应同意接收缓冲区 TLP 时进入 SEND 状态并向应用层发送 TLP。当一个 TLP 发送完时,进入更新状态,此时更新接收计数子模块的信用量,也就是说此时才会更新信用分配量。

3 功能仿真及结果分析

流控模块的验证采用了 Questasim 仿真软件,通过搭建一端为 PCIe 接口的以太网网卡控制器一端为 Microblaze 控制的根节点的系统,在系统中收发以太网帧,使整个 PCIe 流控模块处于工作状态。本节通过对 TLP 以及流控信号的分析,判断流控模块

的功能的正确性。下文是对收发流控模块分别的仿真分析。

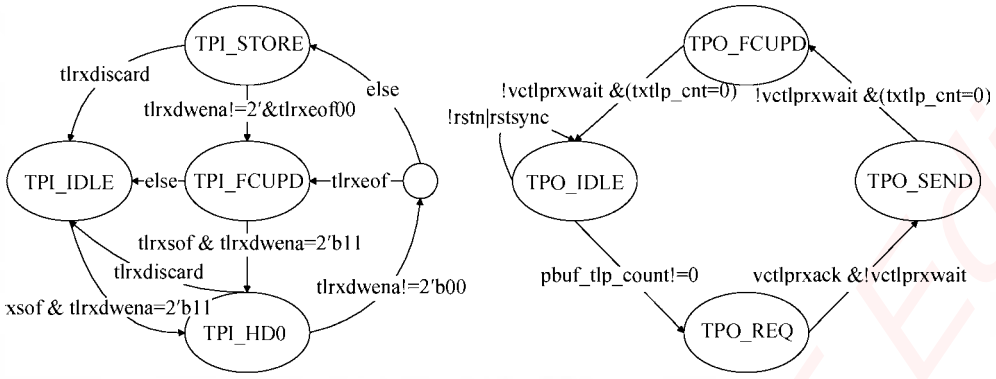

图6　接收流控模块状态机

3.1　发送流控仿真分析

图7所示，分别初始化P、NP、CPL三种不同类型的流控信用。从图中可以看出流控信用量采用了信号复用的方式，通过3种类型更新使能信号控制不同的信用量更新。根据流控信号格式可得出：P类型的头信用量为32，数据信用量为77；NP类型的头信用量为12，数据信用量为12；CPL则通过“0”信号被设为无限流控模式。

图7　流控信用初始化

由图8可以看出发送一个CPL的TLP包时由于处于无限流量控制的模式，所以头信用量和数据信用量不再作为判断TLP包是否发送的标准，该模式下的包不受流控信用量的影响。由此可见，无限流控发送功能完全实现。由于P和NP除了需要判断TLP类型不同外其余功能完全相同，故只用P类型的TLP发送作为分析。从图中可以看出，头使用1个信用量，头信用量限制值为23，故相减后满足下一个TLP包的发送，把头信用可用信号设为1。同时，发送的TLP包带有32DW的数据故需要占用8个数据信用量，数据信用量减去8后仍然大于系统所设置的最大TLP包所需信用量8，把数据信用可用信号设为1。可以知道P和NP的发送流控功能完全实现且信用量计数具有良好的实时性。

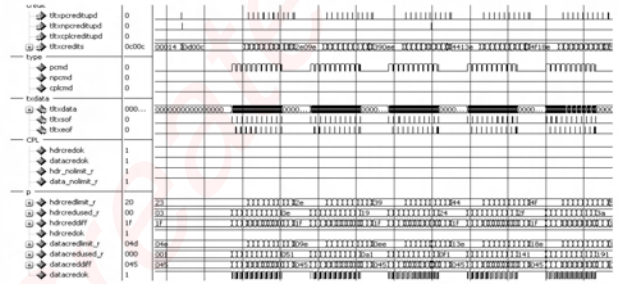

图8　发送流控信号

3.2　接收流控仿真分析

从图9中可以看出，内部的2个状态机工作完全正常符合设计要求。初始信用量符合缓冲区的大小设置，且数据写入缓冲区及从缓冲区读出数据传输到应用层的功能从波形上看出能够实现。在设计时把CPL设置成无限流控信用模式，故CPL的流控信号一直为20位“0”。P和NP类型的接受流控信用会随着从缓冲区读出TLP并转发而进行更新。每个TLP被读出后头信用量加1，数据信用量也会根据TLP数据的长度增加。综上可见接收流控模块的功能符合设计要求，且流控信用量在TLP被读出的下一个时钟周期更新，由此可知具有良好的实时性。

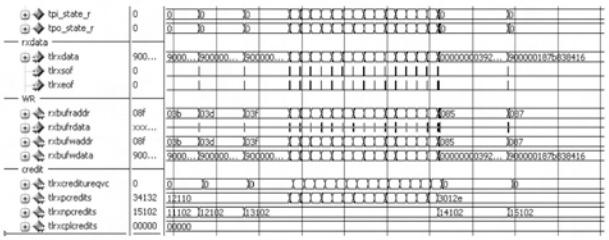

图9　接收流控信号

4　结束语

本文通过研究分析PCIe协议流控机制，设计出一种在PCIe接口的千兆网卡控制器中实现流控协议的系统结构，在设计时采用了模块复用、信号复用等方式进行优化，并且把缓冲区设置为可配置的以便移植到其他PCIe系统中。在实现上采用VeriogHDL语言对其模块进行了描述，最后通过仿真验证表明所设计的流控各个模块能够有效地工作并且具有良好的实时性。文中的PCIe流控模块可根据实际需要改变部分参数用于各种PCIe系统的设计中，对今后设计其他PCIe总线系统有一定参考性。

（下转第137页）

100%。为了方便直观地显示整个实验流程,给出了MATLAB 编写的 GUI 界面,如图 5 所示。

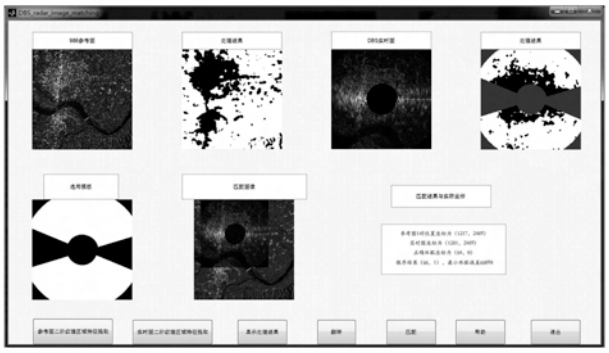

图 5　雷达景象分割与匹配界面

表 2　基于二阶纹理特征分割的景像匹配结果

序列号	实时图坐标	参考图坐标	实际坐标	计算坐标	坐标误差
1	(1281,2489)	(1016,2375)	(265,114)	(271,116)	(6,2)
2	(1289, 2905)	(1076,2774)	(213,131)	(216, 131)	(3,0)
3	(945,2577)	(920,2294)	(25,283)	(26, 286)	(1,3)
4	(1433,2177)	(1238,2150)	(195,27)	(201,26)	(6, -1)
5	(1417,2473)	(1382,2432)	(35,41)	(41,41)	(6,0)
6	(1113,2129)	(1055,2108)	(58,21)	(61,16)	(3, -5)
7	(1561,2897)	(1391,2702)	(170,195)	(176,196)	(6,1)
8	(1225,2249)	(1079,2159)	(146,90)	(151,91)	(5,1)
9	(1065,2857)	(851,2618)	(214,239)	(216,241)	(2,2)
10	(1385,2913)	(1280,2654)	(105,259)	(113,253)	(8, -6)
11	(1060,2488)	(1030,2258)	(30,230)	(31,231)	(1,1)
12	(1260,2488)	(960,2188)	(300,300)	(306,301)	(6,1)
13	(1260,2518)	(1160,2388)	(100,130)	(106,136)	(6,6)
14	(1260,2458)	(1210,2438)	(50,20)	(56,21)	(6,1)

3　结束语

本文采用生物视觉中的“滤波 - 整流 - 滤波”的二阶流程提取图像的纹理特征,并用于 SAR 雷达景像的城区分割和匹配。实验初步表明,该方法在 SAR 图像具有较复杂的几何变形、分辨率变化和光照变化等情况下,仍能达到较好的匹配效果,具有一定的鲁棒性。本文提出的基于 DOG 的二阶纹理特征的方法,对于雷达景像的分割和识别具有较高的准确性,但仍不完善,由于本文所采用的雷达景像图并非真实雷达图,而是由雷达仿真软件生成的仿真图,虽然具有一定的可比性,但实际的雷达图还是有不小的差别。在实际的雷达景像中所包含的干扰,对算法有更高的要求。虽然目前还没有具体的算法,但随着相关理论的发展和完善,对于雷达景像的分割和识别方法必将趋于成熟。

参 考 文 献:

[1] 李禹,计科锋,粟毅. 合成孔径雷达图像分割技术综述[J]. 宇航学报,2008,29(2):17 - 22.

[2] 高馨,曹宗杰. 基于稀疏约束的 SAR 目标特征提取方法研究[J]. 雷达科学与技术,2012,10(6):618 - 623.

[3] 杜文超,孟小芬,刘钦,等. 基于 SAR 图像的快速景象匹配方法[J]. 雷达科学与技术,2014,12(1):39 - 43.

[4] 杨朝辉,陈鹰,邵永社,等. 基于 SIFT 特征的合成孔径雷达景象匹配方法[J]. 计算机应用,2008,28(9):2404 - 2406.

[5] 刘婷,阮锋,张江华. 末制导 DBS 图像匹配方法研究[J]. 航空计算技术,2011,41(2):105 - 107.

[6] Baker C. A processing stream in mammalian visual cortex neurons for non-Fourier responses[J]. Science,1993,261(5117):98 - 101.

[7] Zheng Qing-qing, Sang Nong, Wang Yue-huan, et al. Texture segmentation based on combination of second-order features and spatial information[C]//Proc of the 6th International Symposium on Multispectral Image Processing and Pattern Recognition, Yichang, China, 2009, V.7495: 749519 - 1 - 749519 - 7.

[8] Duda R O, Hart P E, Stork D G. 模式分类[M]. 北京:机械工业出版社,2003.

[9] ZTTOVA B, FLUSSER J. Image registration methods: a survey[J]. Image and vision computing,2003,21(11):977 - 1000.

责任编辑:么丽苹

(上接第 133 页)

参 考 文 献:

[1] 王齐. PCI Express 体系结构导读[M]. 北京:机械工业出版社,2010.

[2] PCI,PCI-X 和 PCI Express 的原理及体系结构[M]. 北京:清华大学出版社, 2007.

[3] 梁小虎, 王乐, 张亚棣. 高速串行总线 RapidIO 与 PCI Express 协议分析比较[J]. 航空计算技术,2010,40(3):127 - 129.

[4] Budruk R, Anderson D, Shanley T. PCI Express 系统体系结构标准教材[M]. 田玉敏,王松, 张波,译. 北京: 电子工业出版社, 2005.

[5] 孟会, 刘雪峰. PCI Express 总线技术分析[J]. 计算机工程, 2006, 32(23): 253 - 255.

[6] 许军, 李玉山, 贺占庄,等. PCI-Express 总线技术研究[J]. 计算机工程与科学, 2006, 28(5): 141 - 143.

[7] Sig P C I. PCI Express Base Specifications Revision 1.1 [J]. PCI SIG, 2005.

[8] Mayhew D, Krishnan V. PCI Express and Advanced Switching: evolutionary path to building next generation interconnects[C]// High Performance Interconnects, 2003. Proceedings. 11th Symposium on. IEEE, 2003: 21 - 29.

[9] Kelley R A, Neal D M. Transaction credit control for serial I/O systems: U.S. Patent 6,760,793[P]. 2004 - 7 - 6.

[10] 关凯锋. PCIe 事务层及数据链路层的实现与验证[D]. 西安: 西安电子科技大学, 2013.

[11] 缪露鹏. PCI Express 端点 IP 核设计[D]. 成都:电子科技大学, 2011.

[12] 张大为, 梁宇琪, 刘迪. PCI Express 协议实现与验证[J]. 现代电子技术, 2012, 35(4): 123 - 127.

责任编辑:么丽苹

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

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

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

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

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

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

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