

基于 CPLD 的双口 RAM 设计与应用

王 琴, 全书海

(武汉理工大学 自动化学院, 湖北 武汉 430070)

摘 要: 介绍了在磁悬浮的主轴控制器中实现双 CPU 之间数据通信的双口 RAM 设计。采用复杂可编程逻辑器件(CPLD), 用基于原理图和 VHDL 语言两者相结合的方法实现了多字节双口 RAM 的设计, 并在设计过程中采用数字逻辑方法解决了 2 个 CPU 对双口 RAM 同时进行写操作时产生冲突的问题, 在磁悬浮主轴控制器中获得了成功应用。

关键词: 复杂可编程逻辑器件; VHDL; 数字信号处理器; 双口 RAM

中图法分类号: TP311.1

文献标识码: A

1 概 述

磁悬浮轴承技术是一门多学科交叉结合的高新技术, 其中的主轴控制器是其研究的核心部分。它主要通过 10 个通电线圈控制主轴的 5 个自由度, 因此磁悬浮的主轴控制器要完成采集数据、预处理、输出控制和算法实现等功能。由于这些工作单靠一个 CPU 难以胜任, 故在此采用双 CPU 结构, 让 CPU 1 完成采集数据、预处理和输出控制的任务, CPU 2 完成控制算法的功能。在此具体采用数字信号处理器(DSP)来实现这些功能。由于 2 个 DSP^[1]之间要求快速通信, 故采用双口 RAM 实现两者之间的通信。采用双口 RAM 有 2 种方法: 用现成的双口 RAM 器件实现 2 个 DSP 之间的通信; 因为在主轴控制器中已经使用了 CPLD^[2]这种器件, 所以为了不增加硬件电路, 可以使用 CPLD 实现双口 RAM 功能。双口 RAM 和 2 个 DSP 之间数据通信接口结构图如图 1 所示, r_1, w_1 分别代表双口 RAM 端口 1 对 RAM 的读写信号; $/cs_1$ 代表端口 1 的片选信号; $d_1 [7..0]$ 代表双口 RAM 端口 1 的 8 位双向数据线; $a_1 [3..0]$ 表示双口 RAM 端口 1 的地址信号线, $/int_1$ 表示双口 RAM 端口 1 的中断信号线, ready 信号线用于当 2 个 CPU 同时对双口 RAM 进行写操作时, 通知 CPU 2 的数据没有被送到双

口 RAM 中。端口 2 的信号线与此类同。

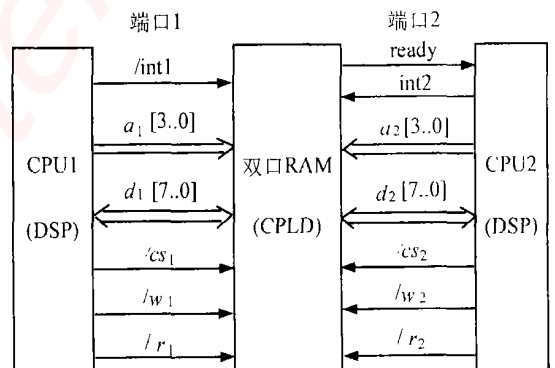


图 1 双口 RAM 和 2 个 DSP 之间数据通信接口结构图

2 双口 RAM 的内部结构

从图 1 可知, 采用 CPLD 设计双口 RAM, 还必须进行内部电路的设计, 双口 RAM 的内部结构图如图 2 所示, 图 2 中存储器阵列模块就是 16 字节的 RAM 单元, 两端口的 4 根地址线 $a_1 [0] \dots a_1 [3]$ 和 $a_2 [0] \dots a_2 [3]$, 通过各自的地址译码器进行译码, 选中 16 字节中的某一个单元, $/cs_1, /cs_2$ 分别是 CPU 1 和 CPU 2 对双口 RAM 的片选信号线, 仲裁和中断逻辑模块要实现 2 个功能: 当一端口对双口 RAM 进行写操作时, 先检测另一端口是否执行写操作, 如果是, 则中断写操作。当两端口同时对双口 RAM 进行写操作时, 保证高

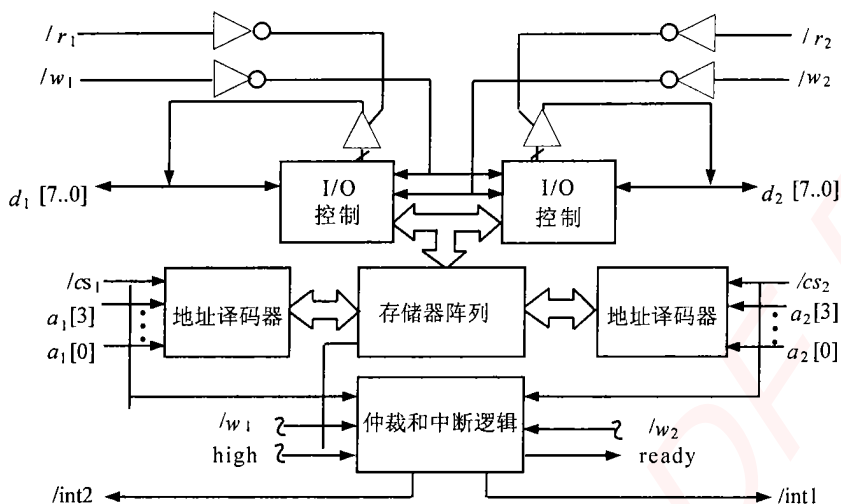


图2 双口RAM 内部结构图

优先级端口的数据被安全送入双口RAM中,并且低优先级端口产生 ready 信号,通知CPU 2 它的数据没有被送到双口RAM中;2个I/O控制模块根据读写信号线控制数据的写入和读出。

3 双口RAM 的设计

3.1 双口RAM 原理图

基于CPLD器件的设计电路,其设计方法常采用两种:一种是基于原理图的交互式图形法;一种是基于硬件描述语言VHDL^[3-5]的方法。在这里采用两者相结合的方式,图3中16字节的双口RAM单元(BYTE16模块)和两端口片选逻辑电路部分(ENA模块)以及中断逻辑电路部分(INT1模块、INT2模块)分别用VHDL语言实

现,然后用原理图的方法调用这些模块完成一个双口RAM的设计,16字节双口RAM原理图如图3所示。对图3中的信号分配引脚后,进行编译,把产生的sof文件下载到CPLD芯片中,CPLD就可以实现双口RAM的功能。

BYTE16模块实现了图2中的存储器阵列模块和I/O控制模块的功能,2个ENA模块分别实现图2中两端口的地址译码器模块的功能,INT1、INT2模块分别实现图2中端口1和端口2的仲裁与中断逻辑模块的功能, r_1, w_1 代表端口1对双口RAM的读写信号, $a_1[3..0], d_1[7..0], int_1, cs_1$ 代表端口1的地址、数据、中断和片选信号。另外一边类同,只是多了一个两端口同时进行写操作时,低优先级端口产生的 ready 信号。

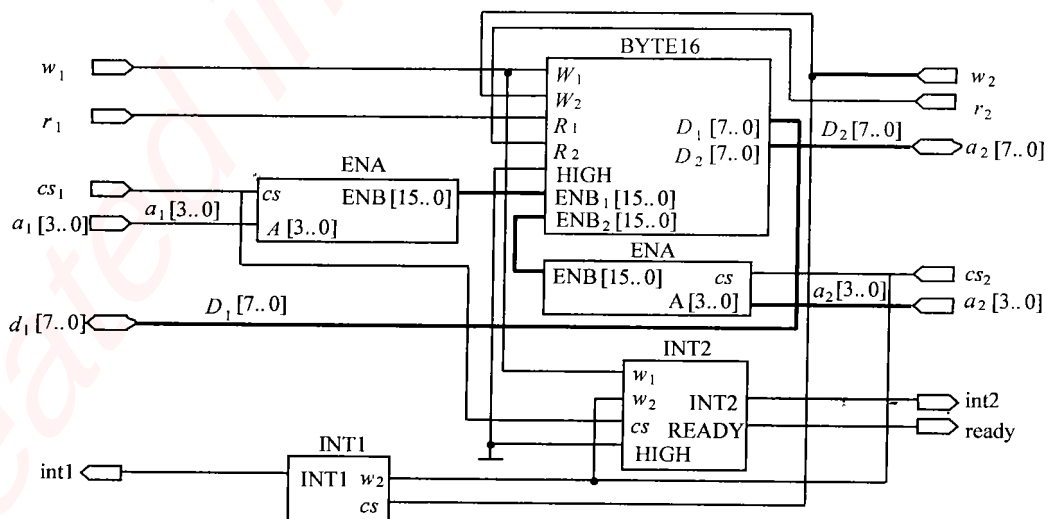


图3 16字节双口RAM 原理图

3.2 一个字节双口RAM 模块设计

设计一个 16 字节的双口RAM, 必须先设计 1 字节的双口RAM 模块。采用 VHDL 语言设计时, 调用了 8 个 D 触发器作为 8 位数据的存储单元。每个位都具有相同的使能端和脉冲输入端, 只是数据输入端和数据输出端不同, 因此在调用 8 个 D 触发器进行端口映射时, 把使能端和脉冲输入端映射到同一个端口, 而 8 个 D 触发器的数据端映射到 8 个不同的数据输入输出端口。具体来说, 先在实体描述中定义 $w_1, w_2, r_1, r_2, ena_1, ena_2, high$ (因为 D 触发器有 CLRN 和 PRN 端, 即清零端和置位端, 但没用上, 所以接高电平 high 屏蔽) 等输入端和 d_1, d_2 这 2 个 8 位数据输入输出向量信号。在接下来的结构体描述中先声明调用了 MAX+ PLUS II 自带的 D 触发器模块, 由于每个输入端并不是与 D 触发器直接相连, 而是经过逻辑电路, 因此在结构体说明中定义 w_1_out, w_2_out, cs, en 这 4 个中间变量和 din, IO 这 2 个 8 位中间变量向量, 如下所示。

```
SIGNAL w1_out: STD LOGIC;
SIGNAL w2_out: STD LOGIC;
SIGNAL cs: STD LOGIC;
SIGNAL en: STD LOGIC;
SIGNAL din: STD LOGIC VECTOR (7
DOWN TO 0);
SIGNAL IO: STD LOGIC VECTOR (7
DOWN TO 0);
```

声明了这些中间变量后在结构体逻辑描述中说明 $w_1_out, w_2_out, cs, en, din$ 与 $w_1, w_2, ena_1, ena_2, d_1, d_2$ 等的逻辑关系如下。

```
w1_out <= w1 or (not w2_out); (1)
w2_out <= w2 or (not w1_out) or ((not
w1) and (not w2)); (2)
cs <= w1_out and w2_out; (3)
en <= (not ena1) or (not ena2); (4)
din(0) <= (d1(0) or (not w2_out)) and
(d2(0) or (not w1_out)); (5)
...
din(7) <= (d1(7) or (not w2_out)) and
(d2(7) or (not w1_out));
```

这相当于完成了 D 触发器以外的部分逻辑电路功能。其中式(1)和式(2)解决当两端口同时进行写操作时的总线冲突问题, 定义端口 1 的优先级高于端口 2, 由式(1)和式(2)可知: 当 w_1 和 w_2 同时为低电平时, w_1_out 为低电平, w_2_out

为高电平, 由式(5)可知 $din(0) = d_1(0)$, 端口 1 的数据被送到双口RAM 中。接下来就要进行 8 个 D 触发器的端口映射, D 触发器的 D 端映射到中间变量 $din(i)$, 脉冲输入端 clk 映射到 cs , 使能端 ena 映射到 en , 数据输出端 Q 映射到中间变量 $IO(i)$ 。下面在一个进程中说明读操作的逻辑过程, 当 ena_1 和 r_1 都为低电平时, 中间变量 IO 送到输入输出端口 d_1 , 当 ena_2 和 r_2 都为低电平时, 中间变量 IO 送到输入输出端口 d_2 , 这样就完成了一个字节双口RAM 的 VHDL 设计。编译通过后, 生成符号文件, 将编译产生的所有相关文件和符号文件都复制到 MAX+ PLUS II 的 $max2lib$ 文件的 $prim$ 文件中, 在实现 16 个字节的 RAM 时就可以通过调用它完成。

3.3 16 字节双口RAM 的设计与实现

在介绍 16 字节双口RAM 模块的设计前, 先介绍 16 对 1 多任务器的实现, 如图 4 所示, 其中虚线表示映射, 空心圆圈表示中间变量, 实心圆圈表示 16 字节双口RAM 的端口引脚。由于 16 字节双口RAM 模块只有一个 8 位的数据端, 而 16 字节中的每个字节都有 8 位的数据端, 多任务器的作用是根据控制信号同时只允许一个字节的数据端和 16 字节的数据端相连。具体实现如下: 在实体中定义了 1 个 16 位的输入向量 s , 16 个 8 位的输入向量 m_0, m_1, \dots, m_{15} 和 1 个 8 位的输入输出向量 $IO[7..0]$ 。在结构体描述中说明当 16 位输入向量 s 某一位为低电平时, $m_0[7..0] \dots m_{15}[7..0]$ 中的某一个与 $IO[7..0]$ 通信。从图 4 可以看出设计 16 字节双口RAM 调用了 16 个 1 字节双口RAM 和 2 个 16 对 1 多任务器, 在实体中定义了 $w_1, w_2, r_1, r_2, high$ 这 5 个输入端, enb_1, enb_2 这 2 个 16 位的向量, 以及 d_1 和 d_2 这 2 个 8 位的输入输出向量(如图 4 实心圆圈所示), 由于进行端口映射时, 每个字节的数据端不能直接和多任务器的数据端口相连, 因此对两端口定义了 16 个 8 位的中间变量向量。 $DIN_0 \dots DIN_{15}$ 和 $DI_0 \dots DI_{15}$, 1 字节双口RAM 和 16 字节双口RAM 的端口映射如图 4 所示, 一个字节双口RAM 的 $w_1, w_2, r_1, r_2, high$ 端口分别映射到 16 字节双口RAM 的 $w_1, w_2, r_1, r_2, high$ 端口, ena_1 映射到向量 enb_1 中的某一位, ena_2 映射到向量 enb_2 中的某一位, 16 位的输入向量 enb_1, enb_2 同时只有一位是低电平, 它对应某个字节被选中, 数据端 d_1 通过 16 对 1 多任务器(A) 映射到 16 个中间变量 $DIN_0 \dots DIN_{15}$ 中的一个, 数据端 d_2 映射到 16 个中间变量 $DI_0 \dots DI_{15}$ 中的

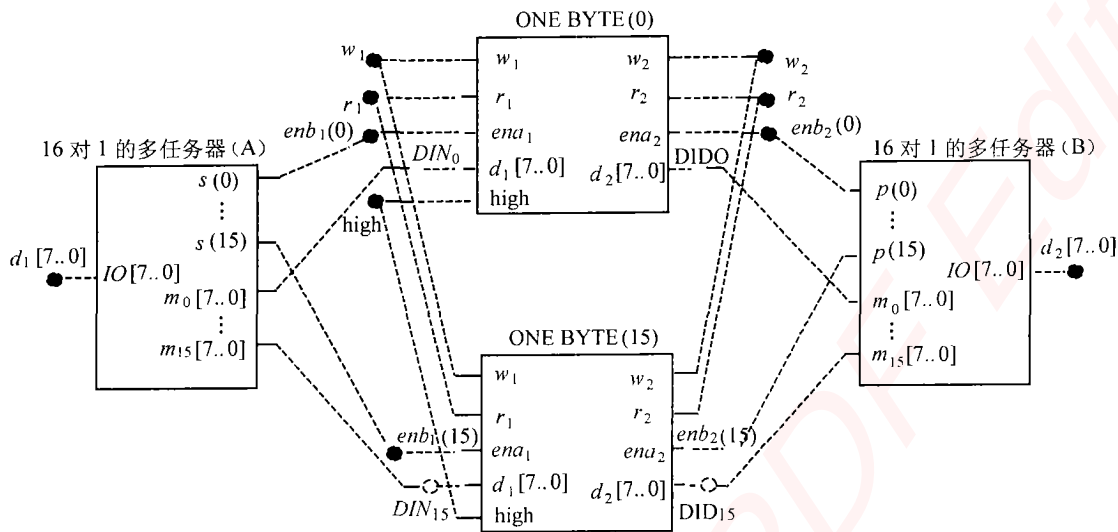


图4 16字节双口RAM 内部映射图

一个, 中间变量向量与多任务器 $m_0[7..0] \dots m_{15}[7..0]$ 端口分别映射实现双口RAM 与CPU 之间的数据通信。

3.4 地址译码模块的设计

16 字节双口RAM 地址译码模块(见图5)的实现是在实体中定义了输入信号片选 cs , 4 位的地址向量 $a[3..0]$; 输出信号 16 位的使能信号向

量 $ENB(0) \dots ENB(15)$ 。在结构体描述中对 4 位地址线进行译码, 当片选信号有效, 4 位地址线译码使 16 位的向量 ENB 中只有一位为低电平时, 表示选中 16 字节的某一个字节。在外部直接将 16 字节双口RAM 模块与这个模块(图3的 ENA 模块)相连即可。

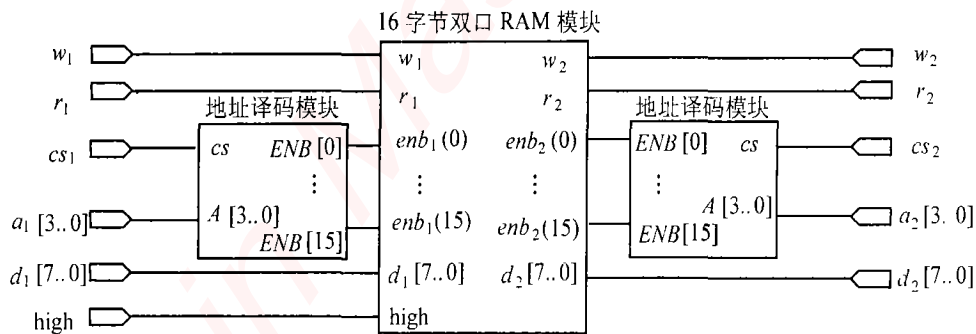


图5 16字节双口RAM 地址译码模块

3.5 中断模块的设计

为了避免总线冲突, 设计了一个中断模块, 也就是图2的仲裁和中断逻辑模块。因为对两端口设置了不同的优先级, 因此两端口的中断模块不相同, 高优先权模块的设计, 即图3的 $NT1$ 模块, 将端口2的片选信号 cs_2 的非信号和写信号线 w_2 的非信号进行与运算后的 $int1$ 信号作为端口1的中断信号, 当这个中断信号为高电平时, 说明对方端口正在进行写操作, 暂缓写操作。低优先权端口由于比高优先权端口多了个 $ready$ 信号, 所以设计复杂一些, 为此给出它的原理图如图6所示。在

实体说明中定义 2 个端口的写信号 w_1 和 w_2 以及对方端口的片选信号线 cs_1 和 $high$ 这 4 个输入信号, $int2$ 和 $ready$ 这 2 个输出信号, 在结构体中声明调用 D 触发器(保持 $ready$ 信号), 并定义 2 个中间变量 a 和 b , 在结构体中除了像 $NT1$ 模块那样将端口 1 片选信号 cs_1 的非信号与对方的写信号 w_1 的非信号进行与运算后的结果作为 $int2$ 外, 还要将 w_1 的非信号与 w_2 的非信号进行与运算的结果即中间变量 a 映射到 D 触发器的 D 端, w_2 的非信号即中间变量 b 映射到 D 触发器的脉冲输入端, $ready$ 信号映射到 D 触发器的 Q 端, 由于 D 触

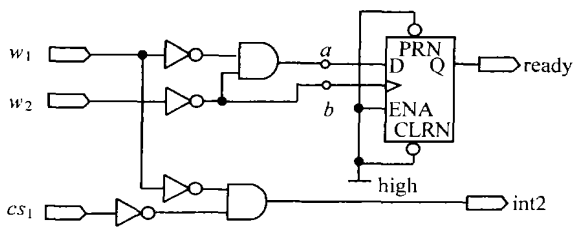


图6 低优先权端口中断模块示意图

发器的其他几个脚都不需要用, 故将 high 信号映射到 D 触发器的 PRN, CLRN 和 ENA 端。

4 双口 RAM 在磁悬浮主轴控制器中的应用

磁悬浮轴承是一种近年来才发展起来的高新技术。传统的轴承在轴转动的过程中和轴承的表面有相对摩擦, 从而产生大量的热, 使轴承很快因为过热而报废。磁悬浮轴承可以很好地解决这个问题, 它通过对磁场中的线圈通以一定的电流从而产生磁力使轴悬浮, 通过调节各个线圈的电流可使轴保持在中心位置。为了实现这个目标, 需要不断地测试轴的位置并将它反馈回去, 通过一定的算法来改变通过线圈电流的大小从而使轴回到中心位置。采用双 DSP 来完成这个过程, DSP1 完成对轴位置的检测, 并对检测到的数据预处理和输出控制任务, DSP2 接收这些数据并根据一定的算法来控制通过各个线圈的电流, 用双口 RAM 来实现两者的通信。用上述完成的程序下载到 CPLD 中, 使 CPLD 实现双口 RAM 的功能, 在设计中采用的 CPLD 是 FLEX 系列的 EPF10K10LC84-4 这种型号。这种系列内含门数达两万以上, 在本设计中仅用了一小部分, 它还可

实现其他功能(只要引脚分配不冲突)。CPLD 的用途十分广泛, 可用于任何数字逻辑系统。CPLD 将半定制的门阵列电路的优点和可编程逻辑器件的用户可编程特性两者结合在一起, 使设计的电子产品达到小型化, 集成化, 而且器件具有用户可编程特性, 大大缩短了设计周期, 减少了设计费用, 降低了设计风险。故 CPLD 广泛地用于通信, 工业控制, 数据处理和军事等各个领域。

5 结束语

笔者介绍了采用原理图和 VHDL 语言相结合的方式设计 16 字节的双口 RAM, 通过设置两端口的优先权成功地解决了当两端口同时对双口 RAM 进行写操作时的总线冲突问题, 并且低优先权端口产生 ready 信号通知 DSP2 它的数据没有被送到双口 RAM 中, 实现了 DSP 与 DSP 之间高速安全的数据通信, 成功地应用于磁悬浮轴承主轴控制器设计中。

参考文献:

- [1] 张雄伟, 曹铁勇 DSP 芯片的原理与开发应用(第 2 版)[M] 北京: 电子工业出版社, 2000
- [2] 廖裕评 CPLD 数字电路设计[M] 北京: 清华大学出版社, 2001
- [3] 曾繁泰 VHDL 程序设计[M] 北京: 清华大学出版社, 2001
- [4] 赵俊超 集成电路设计 VHDL 教程[M] 北京: 希望电子出版社, 2002
- [5] Armstrong J R. VHDL 设计(表示与综合)[M] 北京: 机械工业出版社, 2002

Design and Application of Dual- Access RAM Based on CPLD

Wang Qin, Quan Shuhai

Abstract: The design of dual- access RAM which can realize data communication between two CPU s in main shaft controllers of magnetic suspension is introduced. By virtue of complex programmed logical data (CPLD), dual- access RAM is designed based on the combination of graphics and VHDL. In the designing process, the conflict of the two CPU s in writing operation at the same time is solved using the data logical method. This design is successfully applied in main shaft controllers of magnetic suspension.

Key words: CPLD; VHDL; DSP; dual- access RAM

Wang Qin: Postgraduate; School of Automotive Engineering, WUT, Wuhan 430070, China

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

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](#)

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

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](#)
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](#)

邀请注册码



关注论坛公众号

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 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

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](#)
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

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](#)

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](#)
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

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](#)

邀请注册码



关注论坛公众号