

用 CPLD 实现同步串口与异步串口的转换*

简思平¹ 周学才² 张基宏²

(1.深圳大学信息工程学院, 广东 深圳 518060 2.深圳信息职业技术学院, 广东 深圳 518029)

摘要: 常用串口通信标准种类繁多, 系统设计者必须解决不同串行接口标准之间的转换问题。业界常用串口分为同步串口和异步串口。该文以常用 DSP 芯片的 McBSP 同步串口与 UART 异步串口之间的转换为例, 探讨以 CPLD 来实现串行接口标准之间相互转换的低成本方案, 并详细论述了该方案的软、硬件实现。该方案硬件连接简单, 成本低廉, 软件编程方便, 可用于 DSP 与 PC 机间的串行通信等工程场合。

关键词: 异步串行收发器; 多通道缓冲串行接口; 复杂可编程逻辑器件; 数字信号处理器; 标准串
中图分类号: TP212 **文献标识码:** A **文章编号:** 1672-6332 (2004) 01-0005-06

DSP 技术在各种具体应用中展现出的优异性能, 使其在电子工业领域得到了越来越广泛的应用。在 DSP 应用系统设计中, 必不可少的是各种通信接口的设计。与并行接口相比, 串行接口的最大特点是减少了器件引脚数目, 大大增加了信号的传输距离, 降低了接口设计复杂性和电路成本。目前业界串行通信接口标准种类繁多, 互不兼容。这就导致在某些应用场合, 需要解决不同标准串行接口器件 (或系统) 之间的数据通信问题。

例如, 通用 PC 机的 RS-232 接口为通用异步接口 UART (Universal Asynchronous Receiver and Transmitter), 而 MOTOROLA 公司的串行外围设备接口 SPI、队列 SPI (QSPI)、PHILIPS 公司的内部 IC 总线 (I2C), National 公司的微总线 (MICROWIRE) 均为同步串行协议。美国 TI 公司 TMS320C54xx 系列的 DSP 则仅提供 McBSP 多通道缓冲同步串口^[1-3]。这样, 在实际的应用中, 往往需要实现同步串口与异步串口之间的数据通信。本文以 McBSP 多通道缓冲同步串口与 UART 之间的转换为例, 研究并实现了一种简单、可靠的同步串口转异步串口的方案。

1 DSP 应用系统中异步串口扩展方案

实现同步串口转异步串口的方案很多, 归纳起来, 主要有以下四类。

(1) 在 DSP 的并行总线上扩展 UART 芯片 (如 TI 公司的 TL16C552), 用硬件实现异步数据传输。优点是软件实现简单, 缺点是在总线上还需扩展其它设备, 这样做使目标系统复杂化, 增大系统体积。

(2) 利用 DSP 的 McBSP 和 DMA, 在不扩展其它硬件的情况下, 用软件实现异步数据传输格式。这种方法的优点在于硬件简单, 但软件复杂, 加大了 CPU 的负担, 所以不适合通信数据量大的场合。

(3) 利用 DSP 的 McBSP 同步串行接口, 在扩展适当硬件的情况下, 将同步数据变换为 UART 异步数据格式进行传输。这样可充分利用 DSP 的片上资源, 使硬件系统尽量简单化。如应用美国 MAXIM 公司的 MAX3111 串行异步收发器。该方案的缺点是成本太高。

(4) 利用 DSP 的 McBSP 同步串行接口, 但不

外扩 MAX3111 之类的硬件，而使用 CPLD 可编程逻辑器件，直接将同步串口转换为异步串口。这样，不但充分利用了 DSP 的片上资源，使软硬件都简单化，而且成本低，占用板上空间少。

本文采用第四种方案，应用美国 XILINX 公司的 XC95144XL 可编程逻辑器件，将 DSP 的 McBSP 口与 UART 器件相连的收发线直接接入 CPLD，硬件上再加一片 MAX232 之类的通用电平转换芯片。此时由于异步数据的发送和接收由 CPLD 以硬件形式转换成 McBSP 的标准通信格式，软件编程需要考虑的只是通常 McBSP 的同步通信问题，硬件则用 VHDL 编程后合成下载到 CPLD。这样，用最简单廉价的硬件连接和软件编程就能实现同步到异步的串行数据格式转换。

2 UART 接口时序及 DSP 的多通道缓冲串口标准

2.1 UART 接口协议

通用异步收发协议是异步串行通信的一个标准，因为它是异步的，所以无需通过链路发送时钟信号。取而代之的是收发双方独立的预置频率串行时钟。协议中有一起始位和停止位帮助收信方与接收数据同步。如图 1 所示，数据队列中的一个下降沿的发送表示数据传送开始，紧接着按先低位 (LSB) 的顺序串行地发送数据位，每一帧数据以停止位结束。收发双方预置频率的串行时钟通常是发送波特率的 16 倍。UART 器件控制在一比特间隔的中间采样，这样就能实现同步接收。

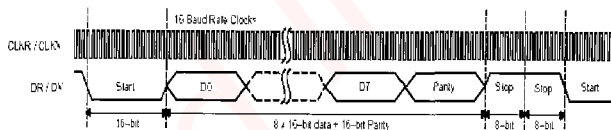


图 1 UART 的时序图

2.2 McBSP 的功能与特点

TMS320C54XX 系列 DSP 芯片具有 2~3 个高速、全双工、多通道缓冲串行接口 (McBSP)，其数据流控制可使其与大多数同步串行外围设备连接。McBSP 是在标准串行接口的基础上对功能进行扩展的，除具有标准串口的功能特点外，其灵活性体现在如下几个方面：

(1) 双缓冲区发送，三缓冲区接收，允许连续数据流传输；

(2) 可与 SPI、IOM-2、AC97 等兼容设备直接连接；

(3) 可编程帧同步、数据时钟极性，支持外部移位时钟或内部频率可编程移位时钟；

(4) 拥有相互独立的数据发送和接收帧同步脉冲和时钟信号；

(5) 多通道发送和接收，最多可达 128 个通道，速度可为 100Mbit/s。

因 McBSP 的标准工作模式是一同步串口，自然不能与 UART 器件直接对接。它有三根信号线，分别为数据、时钟信号线、帧同步线。其标准传输格式如图 2 所示：



图 2 McBSP 标准工作模式的时序图

3 CPLD 中异步串口 UART 协议与 McBSP 标准工作模式的格式转换

为确定起见，本文假定 UART 的数据格式为 1 位起始位，8 位数据，一位停止位。

3.1 DSP 接收时 UART 异步串口转换成 McBSP 同步串口

仔细观察 UART 的时序图和 McBSP 标准工作模式的时序图就会发现，通用异步串口 UART 接收信号线 Rx 上有一比特间隔的低电平起始标志位，紧接着是 8 比特数据，最后是一比特间隔的高电平停止位。与此相对应，同步串口 McBSP 在标准工作模式下，接收帧同步信号线上有 1 比特间隔宽的帧同步脉冲 FSR，紧接着在接收数据线 DR 上给出 8 比特的数据，最后是让接收数据线 DR 处于高阻态。

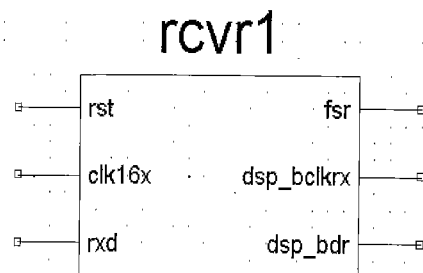


图 3 接收外特性原理图符

由此对比可看出，在 DSP 接收时，可由 CPLD 检测异步串口 UART 起始位的下降沿，一检测到，便在 DSP 同步串口的接收帧同步信号线 FSR 上输出高电平，同时使能两个 4 比特宽度的计数器。一个计数器 clkdiv (4: 0) 对 16 倍于异步串口比特率的时钟信号进行计数。该计数器的最高有效比特作逻辑非运算后就可作为 DSP 同步串口需要的接收时钟信号 CLKR。另一个计数器 no_bits_rcvd (4: 0) 对接收时钟信号 CLKR 进行计数。在接收时钟信号线 CLKR 的第一个上升沿处，复位 DSP 同步串口的接收帧同步信号线 FSR，便有了一完整的接收帧同步脉冲。与此同时，开始将异步串口 Rx 直接连到 DSP 同步串口的数据线 DR 上。直到第二个计数器计到 10，即接收时钟信号 CLKR 的第十个上升沿处，复位起始位的边沿检测器，关闭两个计数器，并复位到全零态。等待下一个字符的到来再重复此过程。

如此，通过简单的计数器、寄存器和其它组合逻辑，在外部晶振时钟信号 (3.6864MHz) 的协助下，便完成了接收异步数据格式与 McBSP 同步数据格式的转换。设计用 VHDL 语言实现。其外特性原理图符见图 3。图 4 为使用 ModelSim5.6 生成的时序仿真图。

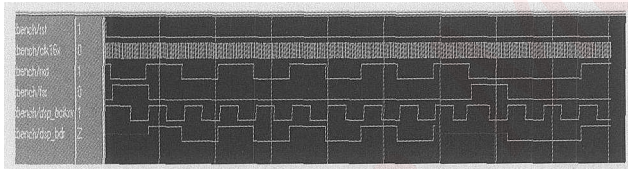


图 4 接收异步串口转换成同步串口时的时序图

其 VHDL 源程序如下：

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity rcvr1 is
port (
rst, clk16x, rxd: in std_logic;
fsr: out std_logic;
dsp_belkrx: out std_logic;
dsp_bdr: out std_logic
);
```

```
end rcvr1;
architecture v1 of rcvr1 is
signal rxd1: std_logic;
signal rxd2: std_logic;
signal clk1x_enable: std_logic;
signal clk1x: std_logic;
signal clk1xx: std_logic;
signal clkdiv: unsigned (3 downto 0);
signal no_bits_rcvd: unsigned (3 downto 0);
signal bitsbit: unsigned (3 downto 0);
begin
process (rst, clk16x, no_bits_rcvd)
begin
if rst = '1' or clk1x_enable = '0' then
bitsbit<="0000";
elsif clk16x'event and clk16x = '1' then
if std_logic_vector (no_bits_rcvd) = "1001" then
bitsbit<= bitsbit + "0001";
end if;
end if;
end process;
process (rst, clk16x)
begin
if rst = '1' then
rxd1 <= '1';
rxd2 <= '1';
elsif clk16x'event and clk16x = '1' then
rxd2 <= rxd1;
rxd1 <= rxd;
end if;
end process;
process (rst,clk16x,rxd1,rxd2,bitsbit)
begin
if rst = '1'or ( std_logic_vector ( bitsbit) = "1110") then
clk1x_enable <= '0';
elsif clk16x'event and clk16x = '1' then
if rxd1 = '0' and rxd2 = '1' then
clk1x_enable <= '1';
end if;
```

```

end if;
end process;
fsr <= '1' when (std_logic_vector (no_bits_rcvd) =
"0000" and clk1x_enable = '1') else
'0';
process (clk16x,clk1x_enable)
begin
if rst = '1' or clk1x_enable = '0' then
clkdiv <= "0000";
elsif clk16x'event and clk16x = '1' then
if clk1x_enable = '1' then
clkdiv <= clkdiv + "0001";
end if;
end if;
end process;
clk1x <= not (clkdiv (3));
clk1xx <= not (clkdiv (3));
dsp_bclkrx<=clk1x;
dsp_bdr<= rxd when (std_logic_vector
(no_bits_rcvd) >= "0001" and std_logic_vector
(no_bits_rcvd) <= "1000") else
'Z';
process (rst,clk1x,clk1x_enable,no_bits_rcvd)
begin
if rst = '1' or clk1x_enable = '0' then
no_bits_rcvd <= "0000";
elsif clk1x'event and clk1x = '1' then
if clk1x_enable = '1' then
no_bits_rcvd <= no_bits_rcvd + "0001";
end if;
end if;
end process;
end;

```

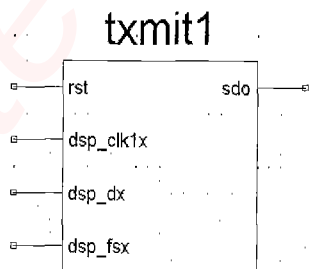


图5 发送外特性原理图符

3.2 DSP 发送时 McBSP 同步串口转换成 UART 异步串口

发送时，发送时钟信号 CLKX 由 CPLD 提供给 DSP，发送帧同步信号 FSX 由 DSP 生成并送往 CPLD，CPLD 检测该发送帧同步信号 FSX，由此生成低电平起始位，同时开始使能相关计数器，把同步串口的发送数据送往异步串口发送数据线 Tx。在计数到最后的一位时，由 CPLD 生成一个停止位并将停止位维持到下一个发送帧同步脉冲信号的到来。设计用 VHDL 语言实现。其外特性原理图符见图 5。图 6 为使用 ModelSim5.6 生成的时序仿真图。

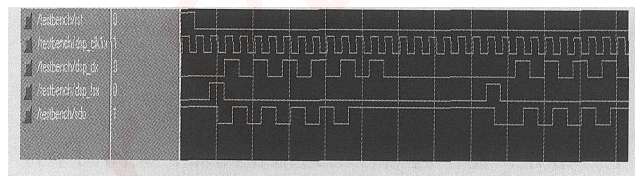


图6 发送异步串口转换成同步串口时的时序图

其 VHDL 源程序如下：

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity txmit1 is
port (
rst: in std_logic;
dsp_clk1x: in std_logic;
dsp_dx: in std_logic;
dsp_fsx: in std_logic;
tx_clk1_dis: out std_logic;
sdo: out std_logic
);
end txmit1;

architecture v1 of txmit1 is
signal sdi: std_logic;
signal clk1x_enable: std_logic;
signal no_bits_send: unsigned (3 downto 0);
begin

```

```
process (rst,dsp_clk1x,dsp_fsx,no_bits_send)
begin
if rst='1' or std_logic_vector ( no_bits_send) ="
1001"then
clk1x_enable<='0';
elseif dsp_clk1x'event and dsp_clk1x = '0' then
if dsp_fsx = '1' then
clk1x_enable<='1';
end if;
end if;
end process;
```

```
process (rst,dsp_clk1x,clk1x_enable,no_bits_send)
begin
if rst='1' then
no_bits_send <= "0000";
elseif dsp_clk1x'event and dsp_clk1x = '1' then
if clk1x_enable = '1' then
no_bits_send <= no_bits_send + "0001";
else
no_bits_send <= "0000";
end if
end if;
end process;
```

```
process (dsp_clk1x,no_bits_send)
begin
if dsp_clk1x'event and dsp_clk1x = '0' then
if std_logic_vector (no_bits_send) ="1000" then
tx_clk1_dis <= '0';
else
tx_clk1_dis <= '1';
end if;
end if;
end process;
```

```
sdi<='0' when dsp_fsx='1' else
dsp_dx when clk1x_enable='1' and dsp_fsx='0' else
'1' when clk1x_enable = '0' else
'1' ;
```

```
process (rst,dsp_clk1x)
begin
if rst='1' then
sdo <= '1';
elseif dsp_clk1x'event and dsp_clk1x = '0' then
sdo <= sdi;
end if;
end process;
end v1;
```

4 DSP 方的通信编程

DSP 方的初始化编程与普通的 McBSP 编程一样，只是有几点要注意。一是时钟，发送与接收的时钟信号都由 CPLD 提供。二是发送帧同步信号由 DSP 在数据由 DXR 复制到 XSR 产生，而接收帧同步信号 CPLD 提供。三是单相，单帧，每帧 1 字，每字 8 比特，最低有效比特先发。

5 结束语

针对现有的多数数字信号处理器 (DSP) 芯片上不提供异步串行收发器 (UART) 接口，而只有同步串行接口的情况，本文通过简单的 CPLD 设计将同步接口转换为异步串口，充分利用了 DSP 的在片硬件资源，很好地解决了 DSP 的异步串口扩展问题。此方法在工程实践中已经得到应用。实践证明，在各种波特率下 (最高可为 230.4kbps)，其查询和中断方式数据传输正确、可靠，各元件工作正常，并且在此硬件连接的基础上，利用 DSP 的 DMA 功能进行串行数据接收及发送收到了良好效果，进一步提升了应用系统的性能。

【参考文献】

- [1] TI Datasheet, TMS320C54 DSP CPU And Peripherals Reference, Texas Instruments, Vol11, April 2001
- [2] 戴明祯, 周建江. TMS320C54X DSP 结构、原理及应用 [M]. 北京: 北京航空航天大学出版社, 2001.
- [3] 薛志宏, 刘建业. TMS320C54XX 系列 DSP 与 PC 机间串行通信的实现 [J]. 电子技术应用, 2003, (5).

Interfacing from Synchronous to Asynchronous Serial Port with Complex Programmable Logic Device

Jian Siping¹, Zhou Xuecai², Zhang Jihong²

(1. Shenzhen University, Shenzhen 518060, P. R. China)

(2. Shenzhen Institute of Information Technology, Shenzhen 518029, P. R. China)

Abstract There are several kinds of serial port interface standards in industry. The system designers usually encounter how to interface between the different serial port standards. The conventional serial ports are classified as Synchronous Serial Port and Asynchronous Serial Port. With low cost in mind, this paper presents a scheme for interfacing from Synchronous to Asynchronous Serial Port with Complex Programmable Logic Device (CPLD) on Digital Signal Processor (DSP) chip. This scheme features of low cost, simple connection in hardware and easy programming. It can be widely used in those cases which request for interfacing between DSP and PC.

Keywords UART; McBSP; complex programmable logic device; digital signal processor; serial port interfacing

(责任编辑: 贺小凤)

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. [基于磁盘阵列引擎的 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](#)

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](#)
52. [基于 Tilcon 航海标绘台界面设计](#)

邀请注册码



关注论坛公众号

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](#)
18. [Android 简单 mp3 播放器源码](#)

邀请注册码



关注论坛公众号

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](#)

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 技术研究与实现](#)
30. [GPU 实现的高速 FIR 数字滤波算法](#)
31. [一种基于 CPU/GPU 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](#)

邀请注册码



关注论坛公众号

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](#)

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](#)

邀请注册码



关注论坛公众号