

The Realization of Programmable FIR Filter on FPGA*

XIE Haixia SUN Zhixiong*

School of Electronic and Information Engineering Qiongzhou University Sanya Hainan 572022 China)

Abstract The paper introduced an architecture of linear phase FIR filter and SD coding algorithm of FIR filter coefficients gave a digital filter index and then tap coefficients were gained by MATLAB finally the 16 order low-pass filter was achieved by using Verilog HDL language and simulating on Quartus II. The results of waveform simulation and the theoretical value met each other by making a comparison between them. And the programming data files are downloaded to FPGA chip at the last. For the performance of different FIR filter the tap coefficients were changed , therefore the design of the tap coefficients were online reconfigured that the different FIR filters were achieved.

Key words FIR filter FPGA Quartus II Verilog HDL

EEACC 1200 1270F

doi 10.3969/j.issn.1005-9490.2012.02.026

可编程 FIR 滤波器的 FPGA 实现*

谢海霞 孙志雄*

琼州学院电子信息工程学院 海南 三亚 572022)

摘 要 介绍了 FIR 滤波器的基本的线性相位结构及 FIR 滤波器的抽头系数 SD 算法编码。给定滤波器的数字指标 ,用 MATLAB 设计抽头系数 ,最后用 Verilog HDL 语言实现了一个 16 阶的 FIR 低通滤波器并在 Quartus II 上仿真 ,并对仿真结果与理论值进行比较 ,波形仿真结果和理论值相吻合 ,最后将编程数据文件下载到 FPGA 芯片上。对于不同性能的 FIR 滤波器 ,抽头系数是变化的 ,因此只要对本设计的抽头系数重新在线配置 ,就可以实现不同的 FIR 滤波器。

关键词 FIR 滤波器 FPGA Quartus II Verilog HDL

中图分类号 TN911.72

文献标识码 A

文章编号 1005-9490 2012 02-0232-04

随着信息技术、计算机和半导体集成电路的飞速发展 ,数字滤波器的理论与实现技术也获得极大的发展 ,并已经应用及渗透到许多重要学科和技术领域中。根据单位冲激响应函数的时域特性数字滤波器可分为 IIR 滤波器和 FIR 滤波器 ,其中 FIR 滤波器是数字信号处理中常用部件 ,它的最大优点在于设计任何幅频特性时 ,可以具有严格的线性相位 ,这一点对数字信号的实时处理非常关键 ,因而受到人们的青睐^[1-5]。

FIR 滤波器的设计实现大致有两种方法 ,一种是软件实现 ,使用高级语言如 C/C++、MATLAB 等 ,在通用的计算机上实现 ,这种方法多用于教学或算法仿真 ,不能实现实时性。另一种是硬件实现 ,采用可编程逻辑器件 (PLD/FPGA) FPGA 具有灵活的可编程逻辑 ,突破了并行处理与流水级数的限制 ,它所具

有的查找表结构非常适用于实现实时快速可靠的 FIR 滤波器 ,在加上 Verilog HDL 语言灵活的描述方法以及与硬件无关的特点 ,使得使用 Verilog HDL 语言基于 FPGA 芯片实现 FIR 滤波器成为研究方向^[3]。本文对基于 FPGA 的 FIR 数字滤波器实现进行了研究 ,并设计一个 16 阶的 FIR 低通滤波器^[6-10]。

1 FIR 滤波器的结构

FIR 有限脉冲响应滤波器是数字滤波器的一种 ,它的特点是单位脉冲响应是一个有限长序列 ,其数学表达式为 :

$$y(n) = \sum_{k=0}^{N-1} h[k] \cdot x[n-k] \quad (1)$$

式中 N 表示 FIR 滤波器的抽头数 , $y(n)$ 表示第 n 时刻的输出样本 , $h(k)$ 表示 FIR 滤波器的第 k 级抽头系

数 $x[n-k]$ 表示延时 k 个抽头的输入信号。根据式 (1) 可得 FIR 滤波器的直接型结构,如图 1 所示。

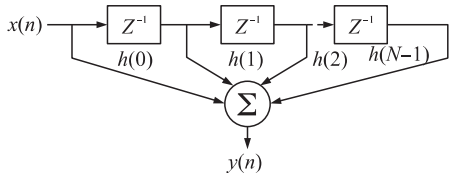


图 1 直接型 FIR 滤波器的结构示意图

用加法器和乘法器不难实现这种结构的 FIR 滤波器,传给每个乘法器的操作数就是一个 FIR 系数。所需乘法次数是 N ,加法次数是 $N-1$ 。但这种直接实现的 FIR 滤波器不论速度上还是资源耗用上都不理想。在使用 FIR 滤波器的实际系统中,常常都会利用了 FIR 滤波器的线性相位的新特点,因此根据线性相位 FIR 滤波器的系数具有对称性这一特点,即 $h[n] = h[N-1-n]$ 或 $h[n] = -h[N-1-n]$ 。因此,线性相位 FIR 滤波器的直接型结构可改进为如图 2 所示。

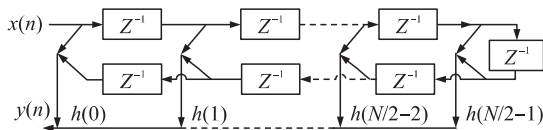


图 2 线性相位 FIR 滤波器的直接型结构改进图 (N 为偶数)

在改进的结构中, N 次乘法减少为 $N/2$ 次,而加法次数增加了 $N/2$ 次,总的运算量得以减少。以乘法次数表示,其总运算量为 $N/2$ 次,这种直接型结构简单明了,系统调整方便^[11-12]。

2 FIR 滤波器的抽头系数的 MATLAB 设计

2.1 FIR 滤波器的设计指标

本文设计的 FIR 数字滤波器是低通滤波器,其设计指标为:采样频率是 5 MHz,截止频率是 1.5 MHz,输入、输出数据宽度是 8 位,阶数是 15 阶,通带衰减 $A_p=4$ dB。

2.2 参数提取

使用 MATLAB6.5 软件中 Filter Design Toolbox 工具箱中的 FDATool,选择低通滤波器,Kaiser 窗设计方法, $F_s=5$ MHz, $F_c=1.5$ MHz,阶数为 15 阶,线性相位。图 3 是软件仿真结果图,如图 3 所示,实线条是滤波器的幅频响应,频率为 1.5 MHz 处幅频响应衰减 $A_p=4.8$ dB,虚线条是相频响应,相频特性是线性,仿真结果满足设计指标要求。最后导出所设计的滤波器抽头系数,对其四舍五入处理后,16 个抽头系数如下:

$$h[0]=h[15]=0.0399 \quad h[1]=h[14]=-0.0145 ;$$

$$\begin{aligned} h[2]=h[13]=-0.0453 \quad h[3]=h[12]=-0.0560 ; \\ h[4]=h[11]=0.0270 \quad h[5]=h[10]=-0.1265 ; \\ h[6]=h[9]=0.0654 \quad h[7]=h[8]=0.5149 . \end{aligned}$$

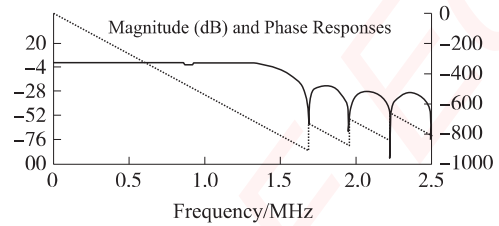


图 3 FIR 数字低通滤波器频域特性图

3 抽头系数的最佳 SD 编码

FIR 滤波器的抽头系数多为小数,且有符号,因此抽头系数的编码是必须考虑的一个问题。常用的编码方式有二进制补码、反码、有符号数值表示法等。

本例中采用以下的编码方式,将十进制数用 2^n 数相加、减的形式表示出来,这种编码方法也被称 SD (Signed Digit Numbers) 编码,该编码与传统的二进制编码不同,它使用三个值来表示数字,即 0、1、-1,其中 -1 经常写为 $\bar{1}$ 。例如:

$$27_{10} = 32_{10} - 4_{10} - 1_2 = 100\bar{1}0\bar{1}_{SD} \quad (\text{下标表示进制})$$

通常可以通过非零元素的数量来估计乘法的效率,比如乘法操作 $A^* x[n]$,其具体实现过程如下。

若 $A_{10} = a_{k-1}a_{k-2} \cdots a_0_2$ 则

$$\begin{aligned} A_{10} * x[n] &= a_{k-1}a_{k-2} \cdots a_0_2 * x[n] \\ &= a_{k-1}2^{k-1}x[n] + a_{k-2}2^{k-2}x[n] + \cdots + a_0x[n] \quad (2) \end{aligned}$$

可以明显地看到,乘法的成本与 A 中非零元素 a_k 的数量有直接的关系。而普通二进制编码需要 4 个加法器,用 SD 编码只需要 3 个加法器。

SD 编码通常不是唯一的,比如:

$$\begin{aligned} 15_{10} &= 16_{10} - 1_{10} = 1000\bar{1}_{SD} \\ 15_{10} &= 16_{10} - 2_{10} + 1_{10} = 100\bar{1}1_{SD} \\ 15_{10} &= 16_{10} - 4_{10} + 2_{10} + 1_{10} = 10\bar{1}11_{SD} \end{aligned}$$

在上面的 SD 编码中,由于第 1 种方式具有最少数量的 1 和 $\bar{1}$,因此它的乘法成本最低,因此应该尽量减少编码中 1 和 $\bar{1}$ 的数量,以将乘法器实现的成本降低到最低,通常将这种包含最少 1 和 $\bar{1}$ 数量的 SD 编码称为最佳 SD 编码。

在滤波器系数的处理上,采用最佳 SD 编码方式,以减少对器件资源的在耗用。滤波器系数的 SD 编码如下,每个抽头系数均先左移 7 位,乘以 128。

$$\begin{aligned} 128 \cdot h[0] &= 128 \times 0.0399 = 5.1072 \\ &= 4 + 2 - 0.5 - 0.25 - 0.125 - 0.03125 = 1(10.\bar{1}\bar{1}10\bar{1}_{SD}) \\ 128 \cdot h[1] &= 128 \times 0.0145 = 1.856 = 2 - 0.125 = 10.\bar{0}0\bar{1}_{SD} \end{aligned}$$

$$\begin{aligned}
 128 \cdot h[2] &= 128 \times 0.0453 = 5.7984 \\
 &= 4 + 1 + 0.5 + 0.25 + 0.0625 = 1(01.1101)_{sb} \\
 128 \cdot h[3] &= 128 \times 0.0560 = 7.168 = 8 - 1 + 0.125 + 0.0625 \\
 &= 1(001.0011)_{sb} \\
 128 \cdot h[4] &= 128 \times 0.027 = 3.456 = 4 - 0.5 - 0.065 \\
 &= 1(00.1001)_{sb} \\
 128 \cdot h[5] &= 128 \times 0.1265 = 16.192 = 16 + 0.125 + 0.0625 \\
 &= 1(111.0011)_{sb} \\
 128 \cdot h[6] &= 128 \times 0.0654 = 8.3712 = 8 + 0.5 + 0.125 \\
 &= 1(000.101)_{sb} \\
 128 \cdot h[7] &= 128 \times 0.5149 = 65.9072 \\
 &= 64 + 2 - 0.0625 - 0.03125 = 1(00010.00011)_{sb}
 \end{aligned}$$

4 FIR 滤波器的 FPGA 实现

FIR 滤波器采用对称结构,每个抽头的输出分别乘以相应加权的二进制值,再将结果相加。同时利用滤波器系数的偶对称特性,把输入信号 $x(n)$ 进行以下等效:

$$\begin{aligned}
 t_0 &= x(0) + x(15) \quad t_1 = x(1) + x(14) \quad t_2 = x(2) + x(13) \\
 t_3 &= x(3) + x(12) \quad t_4 = x(4) + x(11) \quad t_5 = x(5) + x(10) \\
 t_6 &= x(6) + x(9) \quad t_7 = x(7) + x(8)
 \end{aligned}$$

这样,16 阶 FIR 滤波器的输出就可以用下面的算式得到:

$$\begin{aligned}
 \text{sum} &= t_0 \ll 2 + t_0 \ll 1 + t_0[7] + t_0[7:1] + \\
 &t_0[7] + t_0[7] + t_0[7:2] + t_0[7] + t_0[7] + t_0 \\
 &[7:3] + t_0[7] + t_0[7] + t_0[7] + t_0[7] + t_0[7] + \\
 &5 + t_0 \ll 1 + t_0[7] + t_1[7] + t_1[7] + t_1[7:3] + t_1 \\
 &\ll 2 + t_1 - t_1[7] + t_1[7:1] + t_1[7] + t_1[7] + t_1[7:4] + t_1 \\
 &3 + t_1 - t_1[7] + t_1[7] + t_1[7] + t_1[7:3] + t_1[7] + t_1 \\
 &[7] + t_1[7] + t_1[7] + t_1[7:4] + t_1 \ll 2 + t_1[7] + t_1 \\
 &[7:1] + t_1[7] + t_1[7] + t_1[7] + t_1[7:4] + t_1 \\
 &\ll 3 + t_1 \ll 2 + t_1 \ll 1 + t_1 - t_1[7] + t_1[7] + t_1 \\
 &[7:3] + t_1[7] + t_1[7] + t_1[7] + t_1[7:4] + t_1 \\
 &\ll 3 + t_1[7] + t_1[7:1] + t_1[7] + t_1[7] + t_1[7] + t_1 \\
 &[7:3] + t_1 \ll 6 + t_1 \ll 2 + t_1[7] + t_1[7] + t_1[7] + t_1 \\
 &[7] + t_1[7:4] + t_1[7] + t_1[7] + t_1[7] + t_1[7] + t_1[7] + \\
 &t_1[7:5]
 \end{aligned}$$

在得到结果后,再将结果右移 7 位,即得到正确的结果。根据以上设计思路,用 Verilog HDL 语言设计。仿真结果如图 4 所示。

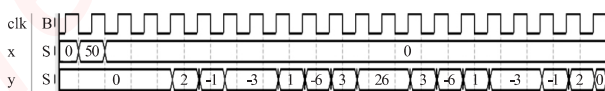


图 4 16 阶 FIR 滤波器的仿真结果

对 Quartus II 仿真结果和 MATLAB 计算的理论值进行比较分析,它们之间的误差如表 1 所示。

表 1 理论值与仿真结果

	经 MATLAB 计算的 理论值	经 Quartus II 仿 真结果值	误差 /%
输出 结果 y(n)	y(0) ⇒ y(15) ⇒ 1.995	y(0) ⇒ y(15) ⇒ 2	0.3
	y(1) ⇒ y(14) ⇒ -0.725	y(1) ⇒ y(14) ⇒ -1	38
	y(2) ⇒ y(13) ⇒ -2.265	y(2) ⇒ y(13) ⇒ -3	32
	y(3) ⇒ y(12) ⇒ -2.8	y(3) ⇒ y(12) ⇒ -3	7
	y(4) ⇒ y(11) ⇒ 1.35	y(4) ⇒ y(11) ⇒ 1	26
	y(5) ⇒ y(10) ⇒ -6.325	y(5) ⇒ y(10) ⇒ -6	5
	y(6) ⇒ y(9) ⇒ 3.27	y(6) ⇒ y(9) ⇒ 3	8.3
	y(7) ⇒ y(8) ⇒ 25.745	y(7) ⇒ y(8) ⇒ 26	1

由上面仿真波形可以读出结果。经比较,仿真结果与输出信号理论吻合,除了一个 38% 的误差较大,其它的误差都在可接受范围内,且波形基本没有毛刺,设计符合要求。部分比较大的误差出入是由于在设计中系数量化时引入的量化误差,以及舍位时产生的舍位误差,在数据比较小的时候。这种误差较为明显。仿真结果正确后把生成的编程数据文件下载到 FPGA 芯片中即可。

5 结束语

在实际使用的时候,大多数情况下,人们希望能够根据应用的不同场所改变滤波器的参数和性能,这就需要对滤波器的系数进行在线配置。本文设计的可编程滤波器能满足这一要求。可编程滤波器在数据处理上分为两个过程:第 1 步完成滤波器系数的配置,将系数配置到抽头延迟线上;第 2 步执行乘积和的计算,对乘积进行一位有符号扩展,并加到前面的部分乘积上。本文所介绍的设计、验证的方法都是可行的,具有很高的应用价值。

参考文献:

- [1] Shousheng He, Mats Torkelson. FPGA Implementation of FIR Filters Using Pipelined Bit-Serial Canonical Signed Digit Multipliers [J]. IEEE Custom Integrated Circuits Conference, 1994, 1(4 May): 81-84.
- [2] 刘在爽, 卢莹莹. 高阶 FIR 滤波器面向 FPGA 的多种实现方法 [J]. 中国有线电视, 2008, 2(1): 64-168.
- [3] 李莹, 路卫军, 于敦山. 一种在 FPGA 上实现 FIR 数字滤波器的资源优化算法 [J]. 北京大学学报(自然科学版), 2009, 45(2): 222-226.
- [4] 杨丽杰. 基于 FPGA 的 FIR 滤波器设计方法的研究 [J]. 东华大学学报, 2006, 6(1): 93-97.
- [5] 刘悦. FIR 数字滤波器的设计与实现 [J]. 信息技术, 2009, 2(1): 8-9.
- [6] 江国强. EDA 技术与应用 [M]. 北京: 电子工业出版社, 2006. 1-15. [M]. 清华大学出版社, 2010.

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](#)

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](#)

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](#)
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](#)

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](#)

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](#)

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](#)

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 COM Express 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](#)
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](#)

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](#)

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.