

流水线技术在用 FPGA 实现高速 DSP 运算中的应用

周政海, 邓先灿

(杭州电子工业学院微电子 CAE 研究所, 浙江 杭州 310037)

摘要: 现代数字信号处理需要处理大量的数据, 迫切需要高速 DSP 运算。随着超大规模可编程器件 FPGA/CLD 和流水线技术的迅速发展, 使得高速 DSP 运算的快速编程实现成为了可能。文章讲述了流水线技术的原理和结构, 提出了在 FPGA 芯片中应用流水线技术, 完成高速数字信号处理运算的思路、方法与具体实现。通过软件综合比较, 测试数据表明应用流水线技术大大提高了 DSP 的运算速度。

关键词: 流水线技术; 现场可编程逻辑阵列; 数字信号处理

中图分类号: TN911.72

文献标识码: A

文章编号: 1001-9146(2002)04-0005-04

0 引言

随着大规模可编程逻辑器件越来越多的应用到数字信号处理 (DSP) 领域, 传统的设计方法显示出一定的局限性, 因为在数字信号处理领域, 芯片需要处理的数据量很大, 为了得到所要求的功能并且具有很好的实时性, 这就要求采用新的设计方法来提高芯片的最高工作速度。在系统设计中采用流水线技术, 可以充分利用硬件内部的并行性, 提高单位时间里的数据处理能力, 提高系统的工作速度。

1 流水线技术的原理

流水线的各个部分就相当于装配车间的传送带, 并行的进行处理, 使系统的吞吐率得到极大的提高。图 1 是一个三级流水线的示意图, 在第 N 个指令译码时, 其前面的一个指令即 $N-1$ 个指令在取指, 后一个指令即 $N+1$ 个指令正在执行。为了将组合逻辑转换成相同逻辑功能的流水线设计, 必须在输入输出之间插入寄存器组, 转换过程如图 2 所示。

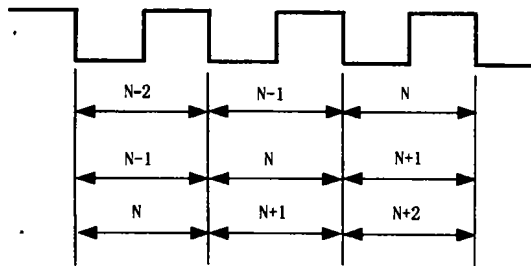


图 1 一个三级流水线操作示意图

对图 2 所示的系统进行分析可知, 图中组合逻辑的延迟包括两级, 第一级是 T_1 、 T_3 中的较大值, 第二级就是 T_2 的延迟。因此, 为了得到最后输出结果, 从输入到输出总的等待时延为: $T_{D1} = \max(T_1, T_3) + T_2$ 。插入寄存器组转换成流水线结构后, 第一级寄存器组所具有的时延为 T_1 、 T_3 时延的较大值加上寄存器组的触发时间 T_{CO} , 同样, 第二级寄存器组的时延为 T_2 加上 T_{CO} 。为了获得第一个计算结果, 必须经过两级逻辑和两级寄存器组, 因此, 该流水线设计获得第一个计算结果需要等待的时延为 \max

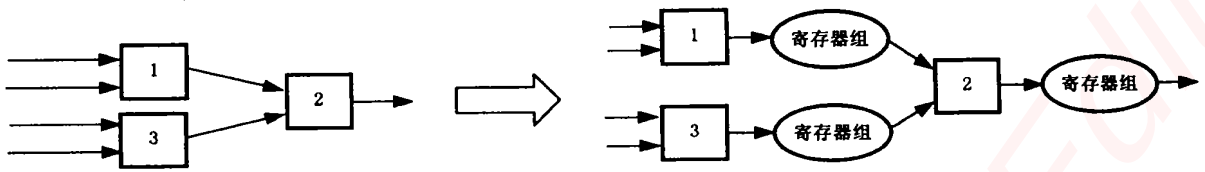


图2 组合逻辑转换成流水线设计

$(T_1, T_3) + T_{CO} + (T_2 + T_{CO})$, 这也被称为流水线设计的首次延时 (First Latency), 但是, 由于加上了中间的寄存器组, 每级计算后的中间结果都能暂时保存, 下一个时钟到来时, 可以直接参与下一级逻辑的运算, 无需等待从系统的输入端传来的数据, 因此, 第二个结果及以后的运算结果的获得只需要一个时钟周期, 等待的时延为: $T_{D2} = \max[(\max(T_1, T_3) + T_{CO}), (T_2 + T_{CO})]$ 。在 FPGA/CPLD 中, 器件的延时 T_1, T_2, T_3 要比寄存器的触发时间长得多, 比较 T_{D1} 和 T_{D2} , 可知 $T_{D1} > T_{D2}$, 因此, 尽管流水线结构的首次延时比较长, 但是执行一次重复性操作所需的时间也就是流水线的吞吐延时要小于普通组合逻辑。流水线的级数越多, T_{D1} 和 T_{D2} 相差就越多, 减少的延时就越显著 (当然, 由于存在“瓶颈”现象, 流水线的级数不能很多, 时延也不可能无限减少, 一般流水线的级数都在 10 级以内), 大大提高了吞吐量 (throughput), 其代价是消耗了一些寄存器资源, 但是对于有着丰富寄存器资源的 FPGA/CPLD 来说, 流水线技术绝对是一个非常好的提高芯片工作速度的选择。

2 用流水线技术提高 DSP 运算速度的设计

在 DSP 处理中, FIR 滤波器是常用的结构。就以 FIR 滤波器为例来说明流水线技术在提高 DSP 运算速度的应用。

2.1 FIR 滤波器的结构

FIR 滤波器的差分方程可表示为:

$$Y(n) = \sum_{i=0}^{N-1} a_i \cdot x(n-i) \quad (1)$$

式中: N 为 FIR 滤波器的阶数, a_i 为滤波器的系数, $x(n)$ 、 $y(n)$ 分别为输入输出, 其直接式的运算结构如图 3。为了便于实现, 减少资源的占用率, 减少输入输出时延, 一般具体实现时可采用滤波器的倒

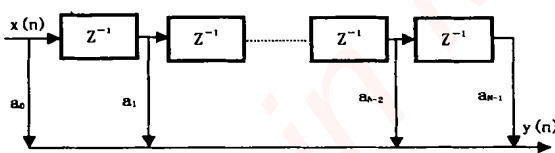


图3 FIR 滤波器的直接式结构

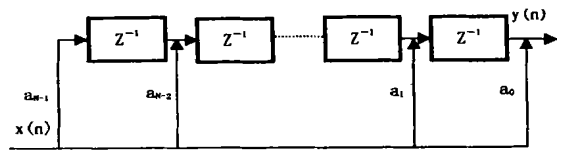


图4 FIR 滤波器的倒置型结构

置型结构, 如图 4 所示。由 FIR 滤波器的差分方程式 1 可以看出, 数据的乘法和加法运算是实现 FIR 滤波器的关键, 它们构成了 DSP 系统的基本结构。因此, 为了用流水线技术实现 DSP 的高速运算, 就是要用流水线技术来改善加法器和乘法器的速度, 实现快速加法和乘法。

2.2 流水线式加法器设计

对于一个 $8 + 8$ 位的加法运算: $S = A + B + C$; $A = [a_7, a_6, a_5, a_4, a_3, a_2, a_1, a_0]$, $B = [b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0]$; $S = [s_7, s_6, s_5, s_4, s_3, s_2, s_1, s_0]$ 。组合逻辑的加法器是由四个 1 位加法器串联而成, 造成的时延主要是逐位向前进位产生的, 虽然可以使用超前进位加法器, 但当加法器的位宽变长时, 超前进位加法器的时延也要增加。采用流水线结构, 可以把位宽较大的加法转换成若干各位宽较小的加法, 而时延与位宽小的加法相同。其计算步骤是先计算低位的数据, 添加寄存器组来暂存中间结果, 再计算高位的数据。流水线式加法器设计框图见图 5。

图 5 框图中画的是将 8 位加法器转换成两个 4 位加法器来实现, 在实现的过程中加入了流水线的

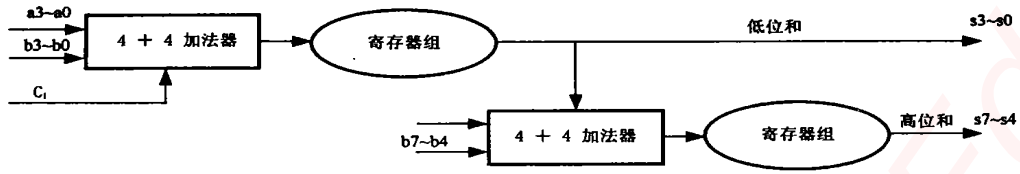


图 5 8 + 8 位流水线加法器框图

技术。图中的 4 + 4 加法器可以利用同样的方式进行分解,化简为更少位的加法。在用 VHDL 等硬件描述语言来实现时,一定要把输入数据与中间得到的中间结果也加入到流水线中,才能使各个部分周而复始的重复的进行流水线式的操作。由于操作数据源源不断的加入到运算器输入端,因此从总的效果来看加法器平均耗费的时间等于寄存器组的时钟周期。为了更进一步满足速度的要求,还可以采用进位保留技术(carry save addition)。

2.3 流水线式乘法器的设计

对于乘法器,归根结底是要转换成乘积项之和,可以采用像加法器一样化简为低阶的的乘法器,也可以直接在加法树乘法器的基础上实现流水线结构。加法树结构的 8 × 8 乘法器是将乘法分解为 8 个 1 × 8 乘法器,为了节省资源,对加法树结构作了一些变动,如图 6 所示。图 6 中,最左侧的是一组多路器,完成 b_{7-0} 与 a_i 的乘法,因为 1 × 8 位乘法器的结果要么为 0,要么为 b_{7-0} ,因此等效的结果就是二选一多路器。为了使相加的时候得到化简,去除部分的和项,多路器的输出组合方式如下: $[a_7, a_3]$ 、 $[a_6, a_2]$ 、 $[a_5, a_1]$ 、 $[a_4, a_0]$ 。以 $[a_7, a_3]$ 的组合为例,假定 a_7 所在的多路器输出为 $x(7-0)$, a_3 所在的多路器输出为 $y(7-0)$,因为 a_3 和 a_7 所在的位权相差四位,这样相加的应该是 $\{x(7-0), 0000\}$ 和 $\{0000, y(7-0)\}$,特定的位置总为 0,进位的计算能够得到化简,从而使时延减少。在图 6 中的每一级后面加上寄存器组,就可以构成流水线结构,更进一步提高运算速度。

2.4 流水线式 FIR 滤波器设计

除了在加法器和乘法器上实行流水线外,还可将流水线技术用到 FIR 滤波器结构上,也就是在实现 FIR 滤波器功能的时候,在不同的阶段插入寄存器组,形成流水线的工作方式,如图 7 所示。

图中的 FIR 滤波器结构除了使用流水线形式外,还运用了改进的 Booth 编码算法、进位保留技术、符号位扩展等手段,达到减少乘积项、加快运算速度的目的。

2.5 高速 DSP 的软件综合测试结果

根据以上所述的电路结构,可以硬件描述语言编写出相应模块的代码。为了使软件综合器生成流水线式的电路,还必须对原来的代码进行改写,施加必要的设计约束。现在大多数综合器可接受得到一种流水线描述结构就是采用多个进程并行工作的方式,在 VHDL 中设计风格如下:

```

proc1: process (input-port) begin          - - process 1 description
    a <= operation (input-port);
end process;
R1: process begin
    wait until clk 'event and clk = '1';
    a-reg <= a;
end process;
proc2: process (input-port) begin        - - process 2 description
    b <= operation (input-port);
end process;
R2: process begin
    wait until clk 'event and clk = '1';
    a-reg <= b;
end process;

```

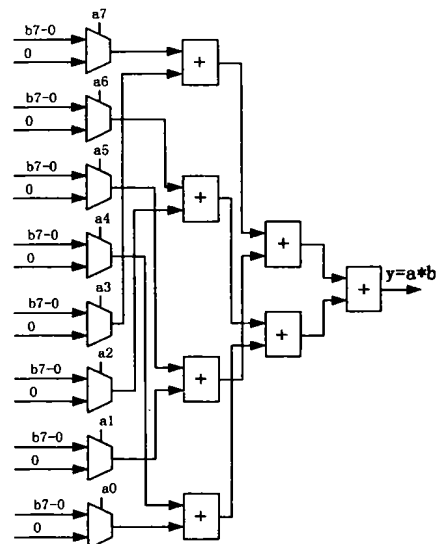


图 6 改进的加法树乘法器的结构图

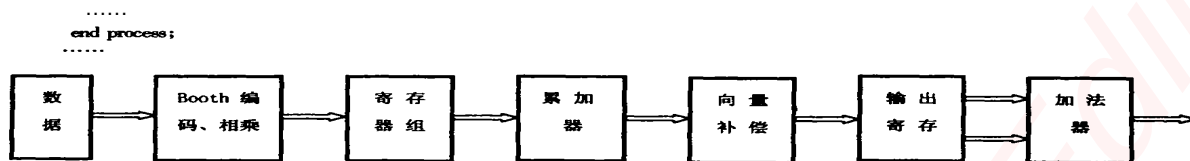


图 7 一种流水线式 FIR 滤波器的结构

经过软件综合,用软件仿真测试延时,确定最高工作速度,如表 1 所示。

表 1 软件综合测试结果

参数	普通加法器	8+8 流水加法器	普通 8×8 乘法器	12 位并行乘法器	8×8 流水乘法器	FIR 滤波器
时延	15.4 ns	9.9 ns	38.1 ns	53 ns	15.3 ns	20 ns
最高工作频率	65 MHz	100 MHz	26.2 MHz	18.8 MHz	65.3 MHz	50 MHz

从表中可以看出,如果直接使用 FPGA/CPLD 中提供的加法器和乘法器,则因为时延过大导致最高工作频率偏低,实现高阶的 FIR 运算。使用了流水线结构的乘法器和加法器后,FIR 滤波器关键路径上的时延仅为 20ns,最终能以 50MHz 的时钟进行运算。

3 结 论

随着 DSP 技术在 ASIC 和超大规模可编程器件中的应用,为了满足 DSP 处理的数据量大、实时性强的要求,必须采用流水线结构等提高系统工作速度的新的设计方法。本文讲述了流水线结构的原理,提出了以 FIR 滤波器为代表的 DSP 运算中采用流水线技术的思路和方法,提高 FIR 滤波器的吞吐率,使得在 FPGA/CPLD 中实现高阶运算成为可能。经过软件逻辑综合,在应用流水线结构的基础上,再采用各种优化的算法,完全可以提高系统的工作速度,实现在 FPGA/CPLD 中实现 DSP 高速运算。

参考文献:

- [1] 夏宇闻. 从算法设计到硬线逻辑的实现[M]. 北京:高等教育出版社,2001.109 - 114.
- [2] 金昕,黄捷. 一种用 FPGA 实现的 FIR 滤波器结构[J]. 微电子学,1999,29(1):58 - 61.
- [3] Kventus A Y, Hung H T, Wilsson Jr A N. An architecture for high - performance/small - area multipliers for use in digital filtering application[J]. IEEEJ Solid - State Circuits, 1994,29(2):117 - 121.
- [4] Goslin G R. A guide to using field programmable gate array (FPGA) for application specific digital signal processing performance [EB/OL]. <http://www.xilinx.com/appnotes/dspguide.pdf>, 1995.
- [5] 徐加全,侯朝焕. 18×18 并行流水乘法器芯片设计[J]. 电子学报,1995,23(2):82 - 84.
- [6] 戴明桢. 数字信号处理的硬件实现[M]. 北京:航空工业出版社,1998.32 - 36.

Application of Assembly - Line Technique in Realization of DSP High - speed Operation with FPGA

ZHOU Zheng-hai, DENG Xian-can

(Microelectronics CAE Center, Hangzhou Institute of Electronic Engineering, Hangzhou Zhejiang 310037, China)

Abstract: Modern digital signal process needs processing a great deal of data, and high - speed calculation is required urgently. With the rapid development of VLSI programmable device such as FPGA/CPLD and assembly - line technique, the fast programmable design method of high - speed DSP becomes possible. The principles and structures of assembly - line technique are described. The paper gives the idea and method to use pipeline technique into designing digital signal process (DSP) with FPGA/CPLD and proves that it is absolutely possible that high - speed DSP operation can be realized with assembly - line technique.

Key words: assembly - line technique; FPGA; DSP

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 3.0 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](#)
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](#)

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](#)

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 I/O 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 技术研究与展望](#)
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](#)
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](#)
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.

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.