

高速浮点运算单元的 FPGA 实现

张小妍, 邵杰

(南京航空航天大学信息科学与技术学院, 江苏省南京市 210016)

摘要: 运用流水线技术对单精度浮点乘法和加法运算单元进行了优化设计。浮点加法器采用了改进的双路径结构, 重点对移位单元和前导 1 检测单元的结构进行了优化。浮点乘法器在对被乘数进行 Booth 编码后, 采用改进的 4-2 压缩器构成 Wallace 树, 在简化逻辑的同时, 提高了系统的吞吐率。经过仿真验证, 在 Virtex-4 系列 FPGA (现场可编程门阵列) 上, 浮点加法器的最高运行速率达到 405 MHz, 浮点乘法器的最高运行速率达到 429 MHz。

关键词: 浮点运算; FPGA; 流水线技术

中图分类号: TN911.72

0 引言

浮点加法是数字信号处理中的一种非常频繁且非常重要的操作, 在现代数字信号处理应用中, 浮点加法运算几乎占到全部浮点操作的一半以上。浮点乘法器是高性能 DSP (数字信号处理器) 的重要部件, 是实时处理的核心, 其速度直接影响 DSP 的速度, 因此国内外学者对提高浮点乘加单元的性能进行了大量的研究^[1-5]。浮点运算单元的设计主要是在速度和所占用资源之间进行权衡。文献 [2-3] 对浮点单元的速度和所占用资源进行了深入分析, 并以矩阵相乘为例进行应用实现, 使得处理器性能得到了显著提高。之后, 文献 [4] 对浮点单元的性能进行了进一步的优化。而以应用于实时信号处理为目的的浮点运算单元设计考虑速度和所占用资源时, 速度的优先级别要高于所占用资源。

本文以实时信号处理为应用背景, 首先介绍了单精度浮点格式, 然后从速度和占用资源这两个角度出发, 详细介绍了浮点乘法单元和浮点加法单元的流水线结构。经过仿真验证, 在 Virtex-4 系列 FPGA (现场可编程门阵列) 上, 浮点加法器的最高运行速率达到 405 MHz, 而浮点乘法器的最高运行速率达到 429 MHz。

1 浮点格式简介

常用的浮点格式为 IEEE 754 标准。该标准定义了单精度浮点数、双精度浮点数和扩展双精度浮点数 3 种格式, 单精度为 32 位, 双精度为 64 位, 扩展双精度为 80 位以上, 位数越多则精度越高, 表示范围也越大。在通常的数字信号处理应用中, 单精度浮点数即

可满足处理精度要求。单精度浮点数如图 1 所示。

$$S \quad E[30:23] \quad M[22:0]$$

图 1 IEEE 754 单精度格式

S 为符号位, S 为 1 时表示负数, S 为 0 时表示正数; E 为指数, 取值范围为 $[1, 254]$, 0 和 255 表示特殊值; M 有 23 位, 再加上小数点左边一位隐含的 1 总共 24 位构成尾数部分。由它表示的浮点数的值 V 可以表示为:

$$V = (-1)^s \times 2^{E-127} (1.M) \quad (1)$$

2 浮点加法运算单元

本文采用改进的流水线型浮点加法器结构^[5], 如图 2 所示。可以用对阶移位、加/减操作和结果选择 3 个主要功能模块来描述。设 2 个浮点操作数分别为 A, B , 其中 E_a, E_b 为指数操作数, M_a, M_b 为尾数操作数, S_a, S_b 为符号位, E, M, S 分别是结果的指数、尾数和符号位。

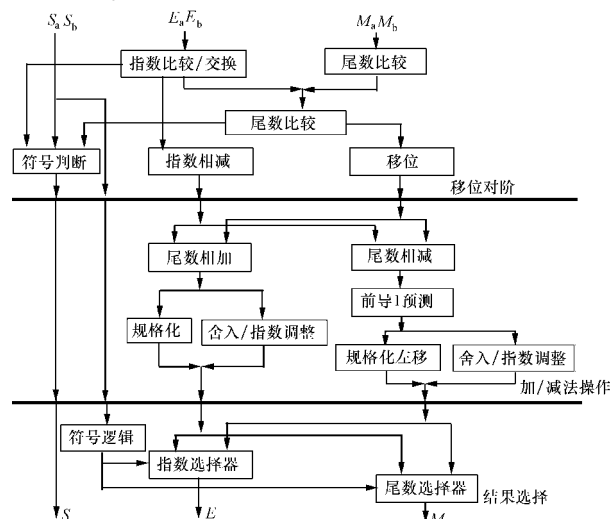


图 2 浮点加法运算单元结构

2.1 对阶移位模块

如图 2 所示,对阶移位模块主要完成 2 个操作数的比较和尾数移位。比较 2 个操作数 a 、 b 的大小,首先要比较指数 E_a 、 E_b 的大小,然后比较尾数 M_a 、 M_b 的大小。为了减少系统延时,采用并行的双路径结构,将指数和尾数的比较分开,然后将比较结果寄存,同时也为了简化电路,将输出寄存的较小尾数解规格化,也就是将尾数的第 24 位赋 1。经过指数比较和尾数比较之后,无论进行的是加法操作还是减法操作,都保证了最终结果为正数,简化了负数求补的模块,既降低了系统的时延,也大大简化了电路的规模。

对较小的尾数进行移位操作无疑是限制此模块主频提高的瓶颈。首先,移位位数取决于指数差 $shift_num[7:0]$,必须先求出指数差,然后判断指数差是否大于有效位的位长,再进行移位。对于 32 位浮点加法运算单元,这样串行执行的时延太长。因此,在求指数差的同时,将移位的中间结果进行输出寄存。同时用 3 个 8 选 1 数据选择器和 1 个 4 选 1 数据选择器,插入 2 级流水线结构实现该移位器,如图 3 所示。经仿真测试,单独移位单元的工作频率可以达到 500 MHz。

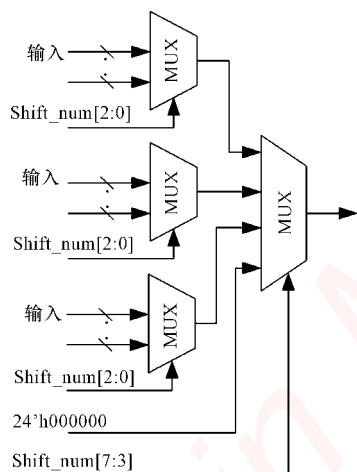


图 3 移位操作的结构

2.2 加/减法操作模块

加/减法功能模块主要完成 2 个操作数的定点加/减操作,同时对结果进行规格化移位、舍入和指数调整。由于加法器和减法器都产生了较大的时延,硬件复杂度也相应较高。在设计实现时,将加/减法操作分离,使用并行的 2 条路径,有效地降低了系统时延,获得了较高的工作频率。

规格化移位操作是对定点加/减法结果进行移位、规格化。由于设计实现时加/减法操作是分离的,其规格化也不同。

a) 加法结果规格化移位:加法操作的规格化移位相对较简单,因为对于加法结果来说,可能产生 1 位进

位,或者没有进位,只要对加法结果的最高位进行判断,即可决定规格化移 1 位还是不移位。

b) 减法结果规格化移位:对减法操作结果进行前导 1 的判断,共有 24 种可能,之后移位器对尾数进行左移,这里又出现了一个限制系统工作频率提高的瓶颈,而采用基于检测树并行修正的 LOP (前导数 1 预测) 结构^[6-7],实现起来相当的复杂,同时也大大增加了整个系统的硬件复杂度。在本设计中采用了一种易于流水线实现的前导 1 检测器,将减法结果 (Sub_result_temp) 各个位的位值作为级联选择器的选择信号,并可与规格化移位操作紧密结合。如图 4 所示,用多个 2 选 1 数据选择器和 1 个 3 选 1 数据选择器,加入简单的标志位逻辑,并且插入一级流水线。改进的前导 1 检测器的实现,有效提高了系统主频。

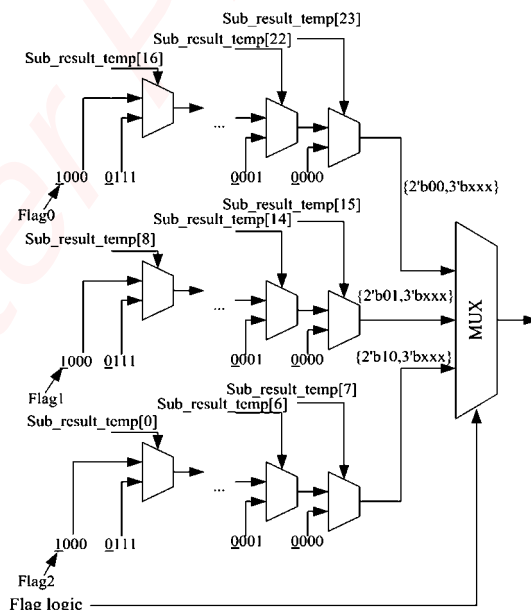


图 4 前导 1 检测器的实现

舍入 指数调整操作主要是对规格化移位之后的结果进行舍入和指数调整。由于设计实现时,加/减法操作是分离的,规格化移位操作分离的,对应的舍入/指数调整单元也分离操作。

2.3 结果选择模块

这个模块主要对加法/减法结果进行选择。同样利用并行的双路径结构,对指数、尾数分别进行选择,本质上就是 2 个 2 选 1 数据选择器,在降低系统延时的同时提高了工作频率。

3 浮点乘法运算单元

本文采用了一种改进的流水线型浮点乘法运算单元结构,如图 5 所示。

浮点乘法操作包括符号位产生、指数位相加和尾

数位相乘 3 个部分。其中,符号位和指数位运算相对简单。对于 32 位的浮点乘法来说,其指数为 8 位,因而可以用一个 9 位的 CSA (进位选择加法器)来实现,经过指数调整后(减去偏置)得到 8 位的指数。符号位可以用一个异或操作来完成。尾数的乘法和指数的加法可以同时进行。尾数相乘可以看成是 2 个 24 位的定点数相乘,从得到的 48 位乘积中取高 25 位规格化后得到 23 位的尾数,此部分具有较高的硬件复杂度,限制了系统主频的提高。所以设计一个 32 位的浮点乘法器关键就是设计一个 24 位乘 24 位的乘法器。

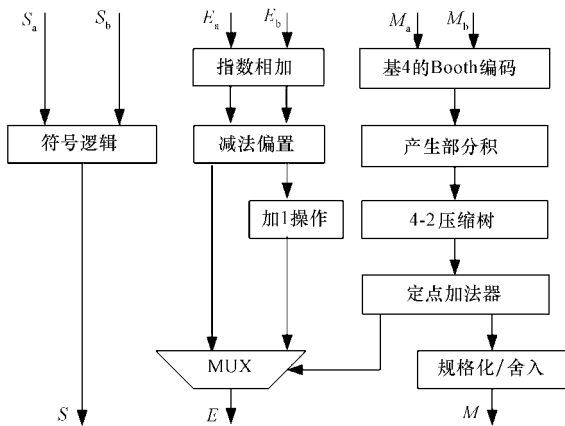


图 5 浮点乘法运算单元结构

3.1 基 4 的 Booth 编码

一个乘法器的延迟主要决定于相加的部分积个数,为了减少部分积,采用改进的基 4 的 Booth 编码方案。在基 4 的 Booth 算法中,乘数的各位分成相互交叠的 3 位,其中前 2 位是当前位,第 3 位是下一组的低位位。这样,每个组的低阶位都要被检查 2 次,每个 3 位组产生一个部分积,从被乘数的倍数 { +2A, +A, 0, -A, -2A } 中选择部分积。Booth 算法结构见图 6。

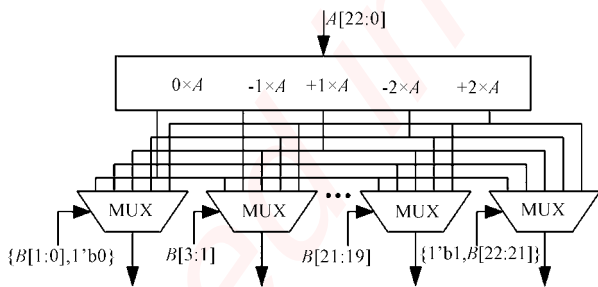


图 6 Booth 编码器

3.2 改进的 4-2 压缩器

在部分积求和网络模块中,为了获得较高的速度,本文采用了改进的 4-2 压缩器结构。4-2 压缩器通过将具有相同权值的 4 个部分积减少到 2 个来提高并行度。传统 4-2 压缩器是由 2 个串行连接的全加器所组成(如图 7 所示)。本文采用一种改进的 4-2 压缩

单元^[8],如图 8 所示。

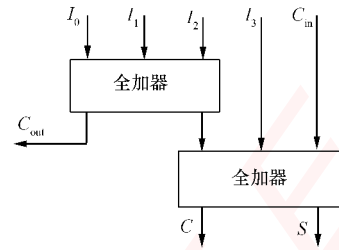


图 7 传统的 4-2 压缩单元逻辑

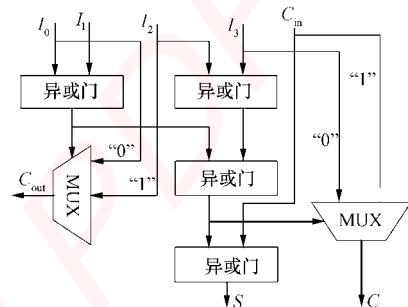


图 8 改进的 4-2 压缩单元逻辑

对基于图 7 的由 2 个全加器设计的传统电路进行了研究,得到了下列表达式:

$$S = I_0 \oplus I_1 \oplus I_2 \oplus I_3 \oplus C_{in} \quad (2)$$

$$C = (I_0 \oplus I_1 \oplus I_2) \cdot I_3 + (I_0 \oplus I_1 \oplus I_2) \cdot C_{in} + I_3 \cdot C_{in} \quad (3)$$

$$C_{out} = I_0 \cdot I_1 + I_0 \cdot I_2 + I_1 \cdot I_2 \quad (4)$$

由于它在关键路径上需要 4 个 XOR 门的延时,而且输入信号 I_0, I_1, I_2, I_3 不是同时参与运算,累加过程容易产生问题。为此,对此电路进行改进,采用优化的逻辑电路。对式 (3)、式 (4) 进行逻辑等价变换可以得到式 (5) 和式 (6):

$$C = (I_0 \oplus I_1 \oplus I_2 \oplus I_3) \cdot C_{in} + (I_0 \oplus I_1 \oplus I_2 \oplus I_3) \cdot I_3 \quad (5)$$

$$C_{out} = (I_0 \cdot I_1) \cdot I_2 + (I_0 \cdot I_1) \cdot I_3 \quad (6)$$

对于式 (5) 和 (6),电路可由 4 个异或门和 2 个 2 选 1 的数据选择器构成,此电路结构不但缩短了关键路径的延时,同时也使 4 个输入同时参与运算,并行程度得到有效提高。

3.3 Wallace 树

加法器阵列常用的两种结构,一个是 IA 阵列,即重复阵列,另一个是 Wallace 树结构。Wallace 树结构由于它的并行操作性,理论上它的速度是最快的。在 32 位浮点乘法运算时,对尾数处理所产生的部分积,通过 Booth 编码产生 13 个部分积,基于前述改进的 4-2 压缩器,对 13 个部分积构造了 Wallace 树的压缩阵列,如图 9 所示,最终将其压缩为 2 个伪和的部分积。

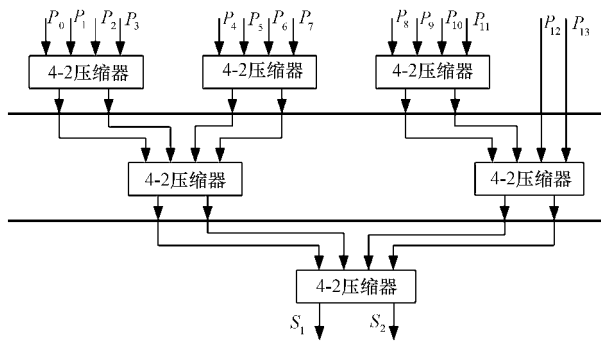


图 9 Wallace树的压缩结构

4 实验结果和分析

根据上述的改进算法结构,采用 Verilog HDL 硬件描述语言在逻辑级进行描述,采用 Xilinx公司的综合工具 ISE 9.2 在 XC4VLX200 FPGA 上仿真实验,完成了流水线结构的单精度浮点加法器和浮点乘法器的设计。所设计的浮点运算单元与 Xilinx公司 IP核的性能比较见表 1 和表 2。表 1 中的结果表明:浮点加法器的吞吐速率达到 405 MHz,速度资源比为 99.5%,比 Xilinx的 IP核提高了 10.8%。表 2 中结果表明:浮点乘法器的速度资源比为 58.3%,比 Xilinx的 IP核略有下降,但是整个流水线的最高工作频率达到了 429 MHz,在浮点运算处理系统中,非常有利于整个系统工作频率的提高,更适于实时信号处理。

表 1 浮点加法器性能比较

浮点加法器	流水线深度	资源片	速度 / MHz	速度资源 (%)
本文的结构	10	407	405	99.5
Xilinx IP core	13	415	368	88.7

表 2 浮点乘法器性能比较

浮点加法器	流水线深度	资源片	速度 / MHz	速度资源 (%)
本文的结构	6	736	429	58.3
Xilinx IP core	8	391	274	70.1

5 结束语

本文运用了流水线技术单精度浮点加法和乘法运算单元进行了优化,在降低时间延迟的同时,提高了系统的吞吐率。在设计过程中,多路选择器在移位器、前导 1 检测器和 Booth 编码器等模块的设计中得到了有效应用。浮点乘法器采用了改进的 4-2 压缩器构成 Wallace 树,简化了逻辑,提高了速度。仿真结果表明:浮点加法器在速度和所占用资源上均得到了优化,速度资源比相对于 Xilinx 的 IP 核提高了 9.32%。而浮

点乘法器在速度方面得到了显著提高,能有效提高整个运算系统的主频,更适用于实时信号处理。

参考文献

- [1] PANATO A, SILVA S, WAGNER F, et al Design of very deep pipelined multipliers for FPGAs [C] // Proceedings of Design, Automation and Test in Europe Conference and Exhibition: Vol 3, Feb 16-20, 2004, Paris, France. Los Alamitos, CA, USA: IEEE Computer Society, 2004: 52-57.
- [2] GOVINDU G, ZHUO L, CHOIS, et al Analysis of high-performance floating-point arithmetic on FPGAs [C] // Proceedings of the 18th International Parallel and Distributed Processing Symposium, Apr 26-30, 2004, Santa Fe, NM, USA. Los Alamitos, CA, USA: IEEE Computer Society, 2004: 149-156.
- [3] DNIZPC, GOVINDU G Design of a field-programmable dual-precision floating-point arithmetic unit [C] // Proceedings of 16th International Conference on Field Programmable Logic and Applications Aug 28-30, 2006, Madrid, Spain Piscataway, NJ, USA: IEEE, 2006: 1-4.
- [4] HEMMERT K S, UNDERWOOD K D. Open source high performance floating-point modules [C] // Proceedings of the 14th Annual IEEE Symposium on Field-Programmable Custom Computing Machines, Apr 24-26, 2006, Napa, CA, USA. Los Alamitos, CA, USA: IEEE Computer Society, 2006: 349-350.
- [5] SHAO Jie, YE Ning, ZHANG Xiaoyan An IEEE compliant floating-point adder with the deeply pipelining paradigm on FPGA [C] // Proceedings of 2008 International Conference on Computer Science and Software Engineering, Dec 12-14, 2008, Wuhan, China. Los Alamitos, CA, USA: Computer Society, 2008: 50-53.
- [6] BRUGUERA J D, LANG T. Leading one prediction with concurrent position correction [J]. IEEE Transactions on Computer, 1999, 48(10): 1063-1097.
- [7] MALIK A, KO Seok-Bum. Effective implementation of floating-point adder using pipelined LOP in FPGAs [C] // Proceedings of Canadian Conference on Electrical and Computer Engineering, Mar 1-4, 2005, Saskatoon, Canada. Los Alamitos, CA, USA: IEEE Computer Society, 2005: 706-709.
- [8] GOPNEEDI P D, THAPLIYAL H, SRINIVAS M B, et al Novel and efficient 4-2 and 5-2 compressors with minimum number of transistors designed for low-power operations [C] // Proceedings of the 2006 International Conference on Embedded Systems and Applications, Jun 26-29, 2006, Las Vegas, VA, USA. 2006: 160-168.

张小妍 (1985-), 女, 硕士研究生, 主要研究方向为信号处理。

表 1 测试用例执行结果

测试用例名称	执行结果	失败原因	执行时间 (2009-7-16)
[M1] SIP用户 PAP鉴权	成功	无	19 18 44 406
[M2] SIP用户 CHAP鉴权	成功	无	19 18 59 406
[M2] SIP用户发送反向数据报文	成功	无	19 19 14 406
[M2] SIP用户接收前向数据报文	失败	脚本超时	19 20 15 406
[M2] SIP用户建立辅连接和流	成功	无	19 20 44 406
[M2] SIP用户协商 MPPC压缩	成功	无	19 20 59 406
[M2] SIP用户数据携带 GRE扩展	成功	无	19 21 14 406
[M2] SIP用户跨 PCF切换	成功	无	19 21 42 406

成功率: 87.5%

本次测试总共运行 8 个测试用例,用时 2 min 24 s, 错误用例数 1 个,其他用例均正确,并给出错误用例失败原因。可以看出,与传统手工测试相比,测试效率有了很大提高。

4 结束语

本文介绍了一种基于 TCL 的 ATE 设计方案。预先为每个测试用例编好自动化测试脚本后,该测试工具可以一次批量执行测试用例,并将执行结果显示或打印,并以 Excel 表的形式对测试结果进行统计。该

测试工具可以昼夜运行,长时间脱离人力操作,极大地提高了测试效率,节省了测试人力。利用 TCL 语言开发测试系统,使系统具有协议描述能力强、灵活性高、移植性好的特点。本测试工具具有很强的可移植性,针对不同的测试实体的不同信令,只需对 Control tcl、Event tcl 和 Message tcl 这 3 个脚本文件的内容以及编解码模块的 DLL 部分做相应的修改即可。

参考文献

- [1] 刘丽娜,高西全,卢朝阳. 一种 WCDMA 中 PS 域的自动化测试方案 [J]. 中国测试技术, 2005, 31 (3): 59-61.
- [2] 李小将. 基于 CIS 的消息驱动的 GUI 自动测试方法 [J]. 计算机工程, 2004, 30 (5): 12-14.
- [3] LAIR. A survey of communication protocol testing [J]. The Journal of System and Software, 2002, 62 (1): 21-46.
- [4] DUSTN E, RASHKA J, PAUL J. Automated software testing: introduction, management, and performance [M]. Reading, MA, USA: Addison-Wesley, 2002: 18-53.
- [5] FEWSTER M, GRAHAM D. 软件测试自动化技术与实例详解 [M]. 舒智勇,包晓露,译. 北京:电子工业出版社, 2000: 135-213.

赵丽 (1984-), 女, 硕士研究生, 主要研究方向为无线通信与移动计算网络。

A Study and Implementation of Automated Testing Tool Based on TCL Script Language

ZHAO Li

(College of Communication, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

Abstract: As software becomes larger and its complication increases, software testing becomes more and more difficult. Automated testing can efficiently decrease the difficulty of software testing and improve the testing efficiency. Based on idea of automatic testing, an automated testing tool based on TCL script language is proposed in this paper. Specifically, the process of design and implementation are discussed in detail. The tool is transplantable. It can be used to test multi-entities by properly modifying some parts.

Keywords: TCL; script; automated testing

Design of High-speed Floating-point Arithmetic on FPGAs

ZHANG Xiaoyan, SHAO Jie

(College of Information Science and Technology, Nanjing University of Aeronautics & Astronautics, Nanjing 210016, China)

Abstract: For optimizing floating-point units, a design based on pipeline techniques is described in this paper. For design of the floating-point adder, an improved dual-path architecture is used to reduce the latency of critical path. The reduction is due to a modified shifter unit and a Leading One Predictor (LOP) unit. By using a structure of 4-2 column compression tree with the Radix-4 Booth encoding and a modified 4-2 compression algorithm, the logical function of floating point multiplier is simplified and the throughput of the system is increased. The simulation results show that floating-point adder achieves 405 MHz and the floating-point multiplier achieves 429 MHz on Xilinx Virtex-4 series FPGA.

Keywords: floating point arithmetic; FPGA; pipeline

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](#)

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 技术研究与发展](#)
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](#)

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](#)

RT Embedded <http://www.kontronn.com>

2.

WeChat ID: kontronn