

# 基于 FPGA 和 DSP 的 DBF 实现

徐欣, 银庆宏, 李楠, 孙兆林

(国防科技大学电子科学与工程学院, 长沙 410073)

**摘要:** FPGA 和 DSP 通过带通采样、I/Q 支路的数字化和数字滤波进而实现基带信号数字波束合成(DBF)。为了更加合理地利用各种现代电子开发工具来充分发挥 DSP 突出的运算能力, 以及 FPGA 逻辑处理能力强、速度快的优点, 可将二者结合使用。在 FPGA 中, 首先利用高速高精度 AD 对中频调制信号进行带通采样, 然后利用 DDS 对采集数据进行 I/Q 分解并用数字滤波器进行滤波, 然后对滤波后数据进行加权求和从而实现系统功能。在 DSP 中, 主要进行波达方向估计, 并用直接矩阵求逆算法计算权值。实验结果表明, 该方法硬件简单, 软件配置灵活, 并且性能良好, 能很好的满足应用需要。

**关键词:** 数字波束合成; 现场可编程逻辑阵列; 数字信号处理; 模数转换

DOI 编码: 10.3969/j.issn.1002-2279.2010.05.008

中图分类号: TP274 文献标识码: A 文章编号: 1002-2279(2010)05-0024-04

## The Realization of Digital Beamforming Based on FPGA and DSP

XU Xin, YIN Qing-hong, LI Nan, SUN Zhao-lin

(College of Electronic Science and Engineering, National University of Defense and Technology, Changsha 410073, China)

**Abstract:** FPGA and DSP make the realization of signal beam forming (DBF) for baseband signal through band-pass sampling, I/Q digital branch decomposition and digital filtering. In order to make more rational use of modern electronic tools, make full use of DSP's computing power, give prominence to FPGA's strong logic processing capacity and fast speed, a combination of both will be used. In the FPGA, firstly band-pass sample the intermediate frequency modulated signal using high-speed high-precision frequency Ads; next, adopt DDS to conduct I/Q decomposition on the collected data and then filter it through the digital filter; finally weight and sum the filtered data and system function is achieved. While, the main work of DSP is to estimate the DOA and calculate the needed weights by the algorithm of matrix inversion. Experimental results show that, the hardware design is straightforward, the software does have a good flexibility in configuration, and the system adapt well in application needs.

**Key words:** Digital Beamforming; FPGA; Digital signal processing (DSP); AD

## 1 引言

数字波束合成是采用阵列天线同时接收多路信号。由于各个阵元接收信号具有不同的波程差, 通过对每路信号进行复数加权, 校正这种差别, 达到方向图主瓣对准期望, 零陷对准干扰的效果, 从而提高系统的输出信噪比, 在通信、雷达、声纳中广泛采用<sup>[1-3]</sup>。DBF 系统一般包括天线阵、信道变频和基带信号处理 3 个部分。而信道变频部分经常是采用模拟混频实现变频并在数字化后进行基带部分的信号处理, 因此一旦硬件设计成型就很难实现灵活改

变。本方案采用数字信道变频使外围电路更加简单, 实现方式也会变得更加灵活。在此方案中使用经典的 FPGA 加 DSP 共同实现基带部分的信号处理<sup>[4-5]</sup>, 充分发挥各自的优点从而更加容易实现 DBF, 满足实现处理的要求。

## 2 总体方案设计

为了描述简便, 假设只存在一个期望信号的情况下由图 1 描述其总体结构。图中  $f(t)$  为阵列天线接收到的中频调制信号, 其中调制信号最高频率为  $f_H$ , 最低频率为  $f_L$ , 中心频率  $f_0 = (f_H + f_L)/2$ , 带宽

$B = f_H - f_L, \tau$  为天线接收信号的时间延迟。根据带通采样定理, 如果其采样频率  $f_s$  满足:

$$f_s = \frac{2(f_H + f_L)}{2(n + 1)} \quad (1)$$

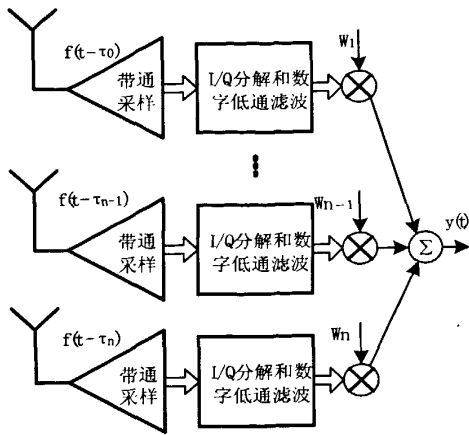


图1 系统框图

式中,  $n$  取能满足  $f_s \geq 2(f_H - f_L)$  的最大整数 ( $0, 1, 2, \dots$ ), 则  $f_s$  用进行等间隔采样所得到的信号采样值能准确的确定原始信号。由于采集下来的信号具有幅度和相位信息, 因此必须对其进行 L/Q 分解得到基带信号的复数表示形式。在实现其过程时还必须对输出结果进行低通滤波, 具体实现如图 2 所示。其中 DDS 输出的频率大小将由采样频率  $f_s$  和调制信号的中心频率所决定, 信号输出  $x(t)$  为复数信号, 其将于复数加权信号  $w$  进行复相乘, 最后对所有结果进行叠加输出实数信号  $y(t)$ 。

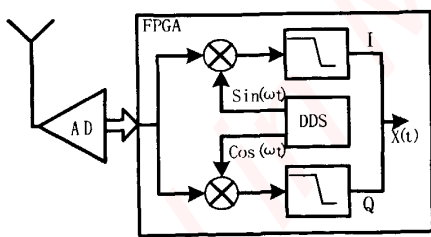


图2 L/Q分解与低通数字滤波

由天线接收到的信号主要包括三种信号, 即期望信号  $f$ , 干扰信号  $s_j$  和噪声信号  $n$ , 采集后信号的数学表达式为:

$$x = fv(\theta_T) + \sum_{j=1}^J s_j v(\theta_j) + n \quad (2)$$

其中  $x$  为  $N$  个 AD 采集通道采集数字化后的基带复信号, 即经过 L/Q 分解和数字低通滤波后的信号, 为  $N \times 1$  维矢量,  $\theta_T$  为期望信号的波达方向 (Direction Of Arrival, DOA),  $v(\theta_T)$  为期望信号对应的方向矢量, 它由天线阵列的几何结构和信号频率所决定, 对于最简单的线阵天线来说, 期望信号对应的方

向矢量可以表示为<sup>[6]</sup>:

$$v(\theta_T) = [1, e^{-jw_m \Delta \tau_m}, L, e^{-jw_m(N-2)\Delta \tau_m}, e^{-jw_m(N-1)\Delta \tau_m}]^T \quad (3)$$

其中  $w_m$  为天线接收信号的角频率,  $\Delta \tau_m$  为相邻两天线之间的时间差, 与相邻天线之间的距离  $d$  和波达方向  $\theta_T$  有关, 其具体表示方式为:

$$\Delta \tau_m = \frac{d \sin(\theta_T)}{v} \quad (4)$$

式中  $v$  为光速, 这样期望信号的方向矢量可以表示为:

$$v(\theta_T) = [1, e^{-jw_m \frac{d \sin(\theta_T)}{v}}, L, e^{-jw_m(N-2) \frac{d \sin(\theta_T)}{v}}, e^{-jw_m(N-1) \frac{d \sin(\theta_T)}{v}}]^T \quad (5)$$

对于不同的天线阵, 由于  $\Delta \tau_m$  将会不同, 从而决定了信号的方向矢量。公式(2)中  $\theta_j$  为干扰信号的方向, 而  $v(\theta_j)$  则为信号对应的方向矢量。由于输出信号为接收信号的复数加权, 则

$$y = w^H x \quad (6)$$

根据最小均方误差准则, 最优权矢量解就是维纳解<sup>[7]</sup>, 其表达式为

$$w_{opt}^H = \frac{v^H(\theta_T) R_x^{-1}}{v^H(\theta_T) R_x^{-1} v(\theta_T)} \quad (7)$$

由上式可以看出,  $R_x^{-1}$  为接收数据协方差的逆, 可以通过多次快拍估计得出。因此  $w_{opt}$  由信号的方向矢量  $v(\theta_T)$  决定, 从式(4)中可知  $v(\theta_T)$  是由输入信号频率以及波达方向  $\theta_T$  所决定的。由于信号输入频率  $w_m$  是已知条件, 所以只有波达方向  $\theta_T$  需要进行估计。一旦实现了对波达方向  $\theta_T$  的估计, 就可以实现波束的合成。

根据以上分析, 要实现波束数字合成主要要实现 3 个部分, 一是基带信号的数字化, 并将信号进行数字化处理得到采集信号的协方差矩阵  $R_x$  以及逆矩阵  $R_x^{-1}$ ; 二是实现对波达方向  $\theta_T$  的估计; 三是对基带信号进行加权叠加, 实现波束合成输出。基带信号数字化后逆矩阵的实现采用采样矩阵求逆 (SMI) 算法, 它是一种数字开环计算方法, 通过直接求解阵列协方差矩阵  $R$  来估计权矢量, 从而避免了闭环自适应中收敛速率对输入相关矩阵特征值的依赖性, 其复杂度主要由天线阵列数  $N$  和采样精度所决定。为了有效实现整个系统的设计, 必须根据 FPGA 和 DSP 各自的优势对硬件进行有效设计, 在两者之间合理的分配任务。FPGA 具有强大的逻辑处理功能, 其实时性非常强; 而 DSP 具有较强的运算功能, 其数字信号处理能力非常强。因此对于实时性要求较高的基带信号数字化和加权输出在 FPGA 中实现, 而矩阵的求解、矩阵的求逆以及 DOA

的估计等运算比较复杂的具体实现将由 DSP 实现。FPGA 将数字化的基带信号传输给 DSP 进行复杂的数字处理,而 DSP 又实时将处理出来的加权信号反馈回 FPGA。

## 3 基带信号的数字化及处理

### 3.1 基带信号的数字化

基带信号的数字化主要包括三个部分,中频调制信号的带通采样、I/Q 支路的数字分解和数字低通滤波。带通采样不但实现了模拟信号的数字化,同时还在频域实现了频谱的搬移,可以将窄带信号搬移到低频段。带通采样频率的选择必须根据被采样信号的中心频率和其带宽以及被数字化后信号数字化处理的能力等因素共同确定。首先采样频率必须满足带通采样理论,其次采样频率必须控制在后端信号处理能力范围内。由于 FPGA 输出时钟的抖动范围比较大,前端 AD 采样时钟均可以通过 FPGA 对时钟芯片的控制而产生所需要的采样时钟。在采样数据输入 FPGA 后数据可以分为两路,分别与 DDS 产生的正弦和余弦信号相乘得到 I 和 Q 两支路信号。由于对输入信号 I/Q 支路的实时性要求比较高,但 FPGA 对乘法资源消耗比较大,综合考虑到 FPGA 相对 RAM 资源比较丰富且对于逻辑运算功能非常强大,故采用读取列表代替数乘,若当输入信号宽度分别为  $m$  和  $n$ ,则可以采用  $m+n$  输入  $m+n$  输出的 RAM 实现  $m \times n$  的乘法器,因此若当天线阵列为  $N$  时,就需要  $2N$  个这样的 RAM。实现 I/Q 支路的 DDS 频率主要取决于采样信号的中心频率  $f_0$  和采样频率  $f_s$ ,其选择的频率可以把所采样的信号搬移到零中频,从而经过低通滤波后得到数字化后的基带信号。数字滤波器是得到数字化基带信号前关键的一步,其性能的好坏直接影响到信号质量,因此对数字滤波器的阶数选择非常关键,必须满足用户对精度的要求。

### 3.2 基带信号的数字处理

由数字低通滤波器输出的基带信号将分成两部分,一部分进行加权求和输出,另一部分将用于估计加权系数  $w_{opt}$ 。用于加权输出的部分可以根据由 DSP 返回的权值进行相乘然后叠加直接输出,同理其相乘部分还是采用读取 RAM 来实现,其输出结果即为 DBF 输出结果;另一部分将被传入 DSP 进行复杂的数字信号处理,计算出协方差矩阵并估计出波达方向和加权系数。由于计算协方差矩阵和对权值的估计需要  $N$  个快拍,所以在传输数据之前必须先对数据进行缓冲,因此对  $N$  个天线阵的复数进行

缓冲时就需要  $2N$  个 FIFO 实现对数据的缓冲。每当传送完  $N$  个数据后并发出一个中断,DSP 根据中断信号对每次传送来的  $N$  组数据对权值进行估计。由于协方差矩阵的求解是 DOA 估计和权值求解的前提,因此必须首先利用式(8)完成协方差的计算

$$R_x = \frac{1}{N} \sum_{k=0}^{N-1} x(k)x^H(k) \quad (8)$$

由于 AD 采样数据的输出格式是无符号整型数,而 DSP 对数据处理的时候需要有符号整型数或浮点数,因此在把数据从 FPGA 传入 DSP 之前必须先将无符号整型数转换成为有符号整型数或浮点数。由于数据转换属于逻辑转换,因此在 FPGA 中将比较容易实现,且实时性比较好。将数据转换后再写入 FIFO,然后再通过 FPGA 和 DSP 之间的扩展口传送到 DSP 中。值得注意的是当将无符号整型数转换为有符号整型数时必须考虑到将 DSP 中的浮点数转换为整型数的同时还要考虑溢出的问题,因此在 DSP 的设计时必须要有防溢出的冗余设计。

### 3.3 DOA 及权值的估计

DSP 的主要工作是实现权值的求解。由公式(7)可知,要实现权值的求解必须先对波达方向进行估计,在此之前还必须求出协方差矩阵的逆。协方差矩阵的逆可以直接通过 DMI 算法计算得到。对于一个自适应 DBF 系统,必须实时更新加权系数从而保证目标始终还保持在天线阵主瓣波束半功率带宽内。当 DBF 处于目标搜索阶段时,不需要对来波方向进行判定,可以对各个方向进行扫描。当对目标进行跟踪时就需要实时更新波达方向从而改变加权系数来改变天线阵波束的主瓣方向。当目标移动速度比较快时,就必须提高加权系数的更新速度,从而对信号处理速度有了更高的要求。此时可以采用多块 DSP 并行工作来提高信号处理速度。若目标是缓慢移动或者是间歇移动则可以采用单片或简单的 DSP 实现。

高分辨率的 DOA 估计算法在实践中已得到广泛应用<sup>[8-10]</sup>,如 MUSIC、ESPRIT 和 MNM 等。但是因为在实际中天线阵列孔径有限,所以现有算法所能分辨的信号数及能达到的谱分辨率受到了极大限制,从而影响到波束合成的质量。由于系统处理的信号具有非圆对称特性<sup>[11]</sup>(例如 BPSK 信号和 MASK 信号),文献<sup>[12]</sup>对阵列孔径进行了虚拟扩展,并提出了非圆算法及其求根形式。这些算法能够对多于实际阵元数的信号进行高分辨率的 DOA 估计,有助于在智能射频通信中实现对期望信号的高质量接收。尽管如此,现有非圆算法仍需要进行计算量

较大的特征分解,不利于快速稳定地估计信号 DOA。因此采用快速虚拟 DOA 算法,它可以避免特征分解,计算量远小于已有算法。

通过 DOA 的估计,把估计值带入式(7),可以得到最优权矢量,但是该权矢量是复数,其实部或者虚部为有符号浮点数或有符号整型数。为了在 FPGA 中顺利进行加权求和,必须把加权系数的格式转变成无符号定点数,即用二进制原码表示。为了保证自适应要求,在 FPGA 中必须及时的对加权矢量进行更新,当有多个目标时可以同时实现多目标的波束合成,如图 3 所示。

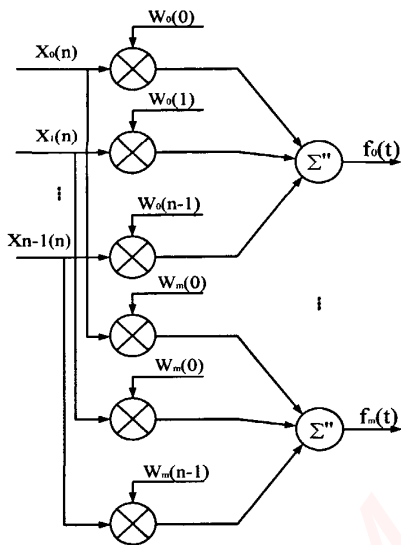


图 3 多目标 DBF 的实现

当进行多波束合成时,就必须同时进行多个加权系数的求解,这样就对信号处理部分提出了更高的要求。

## 4 结束语

基于 FPGA 和 DSP 的 DBF 实现充分发挥了 FPGA 速度快、逻辑处理能力强和 DSP 数据处理能

力强的优势,共同实现了功能复杂的数字波束合成,在 AD 采样前端尽可能的接近天线,也符合软件无线电的设计要求。这就使该系统设计更具有广泛性和实用性。

## 参考文献:

- [1] Godara L C. Application of antenna arrays to mobile communications, part I: Performance improvement, feasibility, and system considerations [J]. Proc. of the IEEE, 1997, 85(7):1029 - 1030.
- [2] Godara L C. Application of antenna arrays to mobile communications, part II: Beam-forming and direction-of-arrival considerations [J]. Proc. of the IEEE, 1997, 85(8):1195 - 1245.
- [3] Krim H, Viberg M. Two decades of array signal processing research [J]. IEEE Signal Processing Magazine, 1996, 13(4):67 - 94.
- [4] 程伟,左继章,许悦雷. 数字波束形成器的 FPGA 实现 [J]. 现代雷达, 2003, 25(5):34 - 36.
- [5] 刘志英,万卫华. 基于 FPGA + DsP 的数字波束形成的实现 [J]. 甘肃科技, 2007, 23(10):27 - 29.
- [6] 杨小牛,楼才义. 软件无线电原理与应用 [M]. 北京:电子工业出版社, 2001.
- [7] 张贤达,保铮. 通信信号处理 [M]. 北京:国防工业出版社, 2002.
- [8] SCHMIDT R. Multiple Emitter Location and Signal Parameter Estimation [J]. IEEE Trans. On AP, 1986, 34(3):276 - 280.
- [9] ROY R, KAILATH T. ESPRIT - Estimation of Signal Parameters Via Rotational Invariance Techniques [J]. IEEE Trans. on ASSP, 1989, 37(7):984 - 995.
- [10] KUMARESAN R, TUFTU D W. Estimating the Angles of Arrival of Multiple Plane Waves [J]. IEEE Trans. on AES, 1983, 19:134 - 139.
- [11] PICINBONO B. On Circularity [J]. IEEE Trans. on SP. 1994, 42:3473 - 3482.
- [12] 黄蕾,张曜. 一种 DOA 估计新算法及其求根形式 [J]. 系统工程与电子技术, 2007, 29(12):2026 - 2028, 2076.

(上接第 23 页)

## 5 结束语

(1) 数字化的“逻辑功效”能够刻画逻辑门或逻辑路径的复杂度,这种方法能够对比不同电路的拓扑结构,发现哪些结构比其他结构更好。

(2) 当每一级的功效延迟都接近相等并且这些延迟近似等于 4 的时候,路径的速度是最快的。

(3) 为获得“更低的门延迟”而采用较少的级数并不能提高电路的速度。将门做的更大也不能提高电路的速度,那只会增大面积和功耗。

(4) 需要计算的有效数字不超过 1~2 位,因此许多估算工作都可以在头脑中完成。所选择的晶体管尺寸不一定要精确的符合理论值,而且如果设计比较合理,那么微调晶体管的尺寸所带来的好处并不大。

## 参考文献:

- [1] R Jacob Baker. CMOS Circuit Design, Layout, and Simulation [M]. 北京:机械工业出版社, 2003.
- [2] 刘树林,张华曹,柴常春. 半导体器件物理 [M]. 北京:电子工业出版社, 2005.
- [3] 朱正涌. 半导体集成电路 [M]. 北京:清华大学出版社, 2001.
- [4] 张建成. MOS 集成电路分析与设计基础 [M]. 北京:电子工业出版社, 1987.

# 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. [USB3.0 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 USB3.0 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. [USB3.0 物理层中弹性缓冲的设计与实现](#)
55. [USB3.0 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 CC++ 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