2004年全国第五届嵌入式系统学术交流会

论 文 集

中国计算机学会微机专业委员会　主编

北京航空航天大学出版社

需要全本请访问 www.ertongbook.com

内 容 简 介

本书为“2004年全国第五届嵌入式系统学术交流会”的论文集,共收集论文112余篇,书中论文内容反映了近年来嵌入式系统领域的一些新兴、前沿和热点技术,内容新颖,范围宽广。

本书共6章,为综合性论述,软件技术及其应用,网络、通信与总线技术,微处理器与芯片技术,FPGA/CPLD与SoC设计,典型应用设计。

本书适合于嵌入式系统业界专家、科技工作者、产品开发人员以及高等院校教师和研究生等参考阅读。

图书在版编目(CIP)数据

2004年全国第五届嵌入式系统学术交流会论文集/中国计算机学会微机专业委员会主编. —北京:北京航空航天大学出版社,2004.9

ISNB 7-81077-550-2

Ⅰ.2… Ⅱ.中… Ⅲ.微型计算机—系统设计—学术会议—文集 Ⅳ.TP360.21-53

中国版本图书馆CIP数据核字(2004)第094163号

2004年全国第五届嵌入式系统学术交流会论文集

中国计算机学会微机专业委员会 主编

责任编辑 王鑫光等

*

北京航空航天大学出版社出版发行

北京市海淀区学院路37号(100083) 发行部电话:(010)82317024 传真:(010)82328026

http://www.buaapress.com.cn E-mail:bhpress@263.net

北京市松源印刷有限公司印装 各地书店经销

开本:787×1 092 1/16 印张:33.25 字数:1150千字

2004年9月第1版 2004年9月第1次印刷

ISBN 7-81077-550-2 定价:80.00元

2004 年全国第五届嵌入式系统学术交流会
中国第十四届微机专业委员会年会

2004 年 9 月 27 日～28 日

北　京

会议主题:

嵌入式系统与 SoC 开发应用

主办单位:

中国计算机学会微机专业委员会

支持指导单位:

国家八六三超大规模集成电路设计专项专家组

国家八六三软件重大专项专家组

承办单位:

北京大学信息科学技术学院

《电子产品世界》杂志社

协办单位:

北京航空航天大学出版社

《电子设计应用》杂志社

大会指定网站:

www .embeddedworld .com .cn

前　　言

什么是嵌入式系统？随着 IT 技术的发展,嵌入式系统已深入到我们的生活之中。从玩具、家电、汽车一直到航天飞机,都有嵌入式系统隐身其中。其应用范围是如此之广,种类是如此繁多,要给嵌入式系统下一个明确的定义实在是太困难了。但嵌入式系统都具有一些基本的特征:就是应具有高可靠、低成本、体积小、实时性强等特点。

随着实际应用的提高,嵌入式系统越来越复杂。如何快速推出产品,并提高产品的可靠性、降低系统成本、缩小产品的重量和体积、降低系统的功耗是我们所要重点解决的问题。SoC 技术是解决这个问题的重要途径。SoC 是一种技术,更是一种方法学。SoC 技术可以指导我们如何合理地进行系统的软硬件划分,以降低系统成本和系统设计的复杂度;如何进行设计重用以缩短产品的开发周期。

SoC 技术是将来嵌入式系统开发的方向。以"嵌入式系统和 SoC 开发及应用"作为本届年会的主题就是希望通过会议使大家了解 SoC 技术,重视 SoC 技术,应用 SoC 技术。

本论文集收录的内容包括作者投稿论文 112 余篇。投稿作者工作在各个领域,所投稿件也是作者实际工作的经验总结和心得体会,希望本论文集会给读者一定的帮助。

感谢给本次会议投稿的作者们对微机专业委员会的支持。承办单位北京大学信息科学技术学院和《电子产品世界》杂志社在会议的组织、稿件评审等方面做了大量的工作,在此表示感谢。协办单位北京航空航天大学出版社加班加点完成了论文集中稿件的编辑和出版工作,在此表示衷心的感谢。感谢百忙之中仍抽出时间参加会议的特邀专家,感谢与会的代表们对微机专业委员会的支持。

中国计算机学会
微机专业委员会
2004 年 9 月

目　录

第一篇　综合性论述

嵌入式开发平台的搭建与软件移植 …… 于导华　张丽芬(2)
SoC 近十年如何最快挖到 SoC 第一桶金 …… 梁合庆(5)
嵌入式系统中的逻辑文件系统 …… 向　玢　赵宏图(9)
网络计算机与嵌入式 Linux 应用支撑环境研究综述
…… 曾　立　吴华瑞　赵春江　吴　平(12)
千兆线速防火墙系统结构与性能分析 …… 张春红　朱新宁(16)
嵌入式系统的电磁兼容设计 …… 邢建泉(21)
基于多处理器通信系统研制 …… 马新平　田小峰　马元奎　奚后玮(24)
测控系统中复杂通讯模式研究 …… 顾伟峰　翟性泉　赵小兵(27)
三种 SoC 片上总线的分析与比较 …… 张丽媛　章　军　陈新华(30)
嵌入式系统启动过程中中断向量的建立方法 …… 王晓春(36)
嵌入式处理器中提高数据 Cache 性能的设计考虑 …… 崔　亮　陈章龙(41)
闪烁存储器与高速数字处理器的接口设计及编程技巧 …… 张蓬鹤　王　宇　周晓惠(46)
嵌入式实时系统中 DSP 的应用与技术分析 …… 刘亚东(50)
基于 B/S 结构的 Web SCADA 系统设计 …… 符意德　任力伟　陈宜磊(55)
低功耗无线数字传输模块设计与应用 …… 申长军　吴庆宏(60)
Protel 软件环境下 PCB 设计 …… 朱然刚　叶春逢(65)
基于 SOFM 网纹理分类算法在遥感图像军事目标识别中的应用
…… 杨　斌　赵红漫　赵宗涛　张　乐(69)
VC 调用 MATLAB 图形图像处理方法研究 … 李旭辉　李俊山　李建军　胡善岳(73)
嵌入式系统学科建设和人才培养的探讨 …… 刘利民(78)
漫谈当前的嵌入式系统教学 …… 张福炎　俞建新(82)
32 位嵌入式系统的 ARM 浪潮 …… 费浙平(85)

第二篇　软件技术及其应用

实时操作系统 RTEMS 结构研究 …… 刘忠仕　戴金海　桂先洲(91)
一种嵌入式系统的通用通信软件设计 …… 刘　诚　岳春生(94)
基于构件的网络化嵌入式系统的中间件技术…… 李巍巍　刘荣兴　杨晓霞(97)
一种基于文件的通用嵌入式设备驱动程序开发模型 …… 吕为工　杨书华(101)
一个通用的轻量级的嵌入式 GUI 的设计与实现 …… 吕为工　杨书华(105)
J2ME 应用程序图形化开发方法的研究 …… 吴　斌　张雪兰(110)
嵌入式实时操作系统 I/O 子系统方案设计 …… 孙海峰　胡宝成　唐小丰(114)
基于 μClinux 嵌入式系统开发平台的建立 …… 肖　杰　李仁发　徐　成(117)
基于嵌入式 LINUX 的通用 PCMCIA 存储卡驱动程序编写…… 韩吟吟　章坚武(122)
VxWorks 下的串口调试…… 谢李李(125)
基于 VxWorks 的矿井多功能接入网关 …… 孙彦景　王　智(130)
VxWorks 下声纳基阵稳定控制系统的软件实现 …… 谢李李(134)
嵌入式实时操作系统 μC/OS－Ⅱ的分析及改进 …… 原变青　刘荣兴　张　忠(138)

需要全本请访问 www.ertongbook.com

基于 μC/ OS 的 μC/ GUI 系统的实现与应用
…… 杨光友 丁 毅 周国柱 张道德(142)
基于 μC/ OS－II 系统的便携式心血管血流参数检测仪的设计
…… 杜 旭 吴水才 侯立亚 常 宇(147)
ARM 处理器启动代码的分析与设计 …… 甘 泉 杨 健 陈永泰(151)
基于 MCS－51 单片机的 C5402 HPI Bootloader 实现
…… 王 玥 王树勋(155)
用 Driver Studio 开发 WDM 设备驱动程序…… 高金锐 熊庭刚(159)
嵌入式 Java 虚拟机 KVM 在 CRTOSII 上的移植实现 …… 金新宝 熊光泽(164)

第三篇 网络、通信与总线技术

一种低成本的嵌入式 Internet 系统接口 …… 彭宣戈 廖金明(170)
TCP/ IP 在凌阳单片机上实现的特点和方法 …… 邓永德 王志刚 郭 颖(175)
基于 S3C44B0x 的以太网接口设计 …… 闫效莺 田 泽 车晓萍 杨 峰(178)
嵌入式 Internet 在远程监测系统中的应用 … 刘玉红 井海明 车轩玉 高静巧(183)
基于 LAN91C111 的嵌入式以太网卡设计 …… 王雪瑞 戴紫彬(187)
CAN 智能模块在电动车上的应用 …… 姚国春 张维刚 朱小林 杨卫列 (192)
基于 CAN 总线和 DSP 的仿人机器人运动控制系统研究 ……
…… 冯金光 周华平 马宏绪(196)
基于 CANBUS 的智能饮水控制系统设计 …… 杨仁弟 周常森 杜培宇(201)
基于 PC104 的 CAN 总线数据状态监控程序的开发……
…… 姚国春 张维刚 杨卫列 朱小林(204)
基于 USB2 .0 的实时视频图像采集系统设计 …… 黄传铭 周山宏 林政剑(207)
基于 ET44M210 芯片的 USB 接口应用设计 …… 熊小明 朱仁龙 黄 勇(211)
USB 方式下 MC68HC908JB8 与 PC 机之间的通信编程 …… 刘雪兰 王宜怀(218)
用 VC 实现 PC 机与 SPCE061A 单片机的串行通信…… 何 静 王志刚 王蒙军(223)
基于 EPP 模式的两台微机并行通信 …… 胡景春 王志华 陈培明(227)
一种无线调度专网的判选控制器设计 …… 邓洪波 姜福广(230)
用无线数传模块构建嵌入式系统的无线数据通信…… 张文军 姜 焱 陈 洁(235)
基于 DSP 的网络视频监控系统 …… 孔梁萍 李宇成 张志芳 黄江海(242)
嵌入式 H .324 可视电话终端系统设计与实现 …… 曾园园 胡瑞敏(246)
基于 IPSec 的嵌入式安全网关的设计与实现 …… 匡晋湘 徐 成(254)
基于 Linux 操作系统 SNMP 卫星通信网管代理的设计与实现 …… 杨巧丽(259)
嵌入式系统与 GPS 的接口技术与导航信息提取软件设计 …… 张文军 姜 焱(263)

第四篇 微处理器与芯片技术

嵌入式处理器中嵌入式 ICE 的设计 …… 孙桂花 涂时亮(271)
ADS 环境下 ARM 系统两种初始化方式分析 …… 高超然 徐 成 李仁发(276)
基于 ARM CPU 嵌入式主机板系统结构的研究 …… 杨书华 吕为工 王志明(282)
ARM 映像文件机理分析及其应用 …… 刘智臣 徐 成 李仁发(287)
基于 S3C44B0X 的嵌入式系统设计 …… 马天军 李晓辉(293)
基于 S3C2410 的 WLAN 智能住宅控制终端的设计 …… 柯建伟 周嘉农(299)
TMS320VC5402DSP 与外部模拟设备的 HPI 通信研究
…… 李玉峰 张宗麟 沈春林(303)
基于 TMS320C5402 DSP 的 PID 温度控制算法的实现
…… 肖洪兵 蒋天伟 关晓丹 樊志成(306)

MSP430 单片机和 TMP100 温度传感器的接口设计 …… 黄艳秋(309)
SDRAM 控制器的设计及其在控制系统中的应用
…… 朱海君　敬　岚　乔卫民　陆　军(313)
ET13X210/ 221 射频收发芯片原理及应用 …… 徐　晋　赵俊逸　黄　勇(317)
基于 MB90092 的动态视频字符叠加器设计 …… 钱怀风(324)
基于 Neuron 芯片的电力参数测试节点的设计 …… 曲金鹏　刘晓光　凌志浩(329)
基于 SPMC701fm0a 的自主移动机器人导航算法的研究
…… 周爱玉　李　欣　刘振山(334)
基于 TI S6700 收发芯片的防数据碰撞读写器实现 …… 鲁公羽　陈　雄(339)
基于 HMS30C7202 嵌入式应用接口设计 …… 车晓萍　田　泽　闫效莺　陈群英(344)
基于凌阳 SPCE061A 的嵌入式数据采集系统的设计 …… 王蒙军　王志刚　杨　珉(350)
基于 PCI9054 的嵌入式 PCI 模块接口设计 …… 刘红甫(354)

第五篇　FPGA/ CPLD 与 SoC 设计

硬进程的 HDL 描述 …… 朱　勇(361)
基于 FPGA 的可重构计算系统 …… 罗毅辉　李仁发　彭蔓蔓(367)
基于 FPGA 技术伪码同步的设计与实现 …… 陆海峰　马　宏(372)
32 位软件处理器 MicroBlaze™ 在软件无线电系统中的应用
…… 王　磊　周贤伟　杨　军(377)
Microblaze 微处理器 IP Core 的结构及应用 …… 龙霞飞　李仁发(383)
Nios 软核处理器的研究与应用 …… 刘连东　郁　滨　戴紫彬(389)
一种嵌入式微处理器设计 …… 陈　雷　高德远　樊晓桠　彭和平　张　妍(393)
USB 到 RS－232 转换芯片设计 …… 王伟煌　杨准洲　于敦山(397)
SoC 中基于 RTL 的低功耗 BIST 测试技术 …… 汪　滢　王　宏　李金凤(401)
基于状态机和流水线技术的 3DES 加密算法 FPGA 设计 …… 邵金祥　陈利学(405)
基于 CPLD 的 FSK 信号产生器设计 …… 杨仁弟　周常森(409)
基于 FPGA 的 APB 总线与 I^2C 总线接口控制器的设计 …… 韩少锋　王国峰　许　茁(413)
SDRAM 控制器的 FPGA 设计与实现 …… 李　卫　王　杉　魏急波(418)
基于 FPGA 的实时图像采集系统 …… 张　崇　邓长军　于晓琳　金　剑(422)
基于 EDA 的机械手控制系统的设计与实现 …… 戴立江　尤一鸣(425)
基于 FPGA 的蓝牙系统中 AFH 算法的研究与实现
…… 周小平　何　丰　黄　涛　李春泉(431)
基于 DSP 和 FPGA 的数据采集与信号处理系统设计 …… 刘育浩　杨文辉　黄新生(436)

第六篇　典型应用设计

蒸汽发生器自动控制系统的多模态控制性能分析
…… 杨晓红　张邦成　王占礼　王海波　吴　狄(441)
嵌入式高速 SmartMedia 卡控制器模块设计 …… 林　凡(445)
基于双目视觉和 Hough 变换实现的测距研究
…… 洪初阳　潘晓露　詹跃东　李一民(452)
一种基于 DSP 的视频会议终端硬件平台设计与实现 …… 刘树波(457)
立体仓库模型控制系统 …… 李志明(462)
一种嵌入式 WEB 服务器的设计及在 SAW 压力传感器中的应用
…… 李金梁　景　博　何鹏举　王　旭　胡炜涛(466)
C8051F02X 单片机在波浪补偿起重机控制系统上的应用
…… 赵小兵　翟性泉　顾伟峰(473)

具有通信功能智能氧量测量仪的设计 …… 王 平 侯 耀(476)
双 MCU 实现对出租车的税务管理 …… 刘大为(480)
基于 MCU 的智能音乐门铃设计 …… 陈存辉 张新宇 胡 骁(486)
基于单片机的提花机控制系统设计 …… 项功宏 陈 奇(490)
基于单片机的移动存储卡接口设计 …… 姚放吾 李 晖(495)
用 AT89C2051 设计超声波流量计 …… 郭 颖 王志刚 邓永德(501)
基于高速 SoC 单片机的磁探测系统 …… 王连福 王 华(504)
基于 CS5460A 的低成本配电控制系统 …… 陈智高 魏学业(508)
一种单片机实验教学仪器设计 …… 李 耀(513)

第一篇

综合性论述

嵌入式开发平台的搭建与软件移植

于导华　　张丽芬
北京理工大学计算机科学工程系,北京,100081

摘　要　嵌入式应用是近年来流行最快的微电子产业,嵌入式开发包含着内容丰富的技术底蕴。针对目前嵌入式开发广泛的市场需求,对比存在的各种嵌入式开发方式,提出了一种基于网络和串口的交互式软件开发平台,阐述了开发平台目标板的软件结构、开发平台的硬件连接及软件设置方法。在研究目标板文件系统的基础上,通过 Minigui 移植讲解了该平台下进行软件移植的过程、测试方法及系统配置方法,并对进行全面系统的嵌入式开发提出了几点建议。

关键词　ARM, Minigui, Ramdisk,宿主机,目标板,引导程序,内核,交叉编译

1　引　言

在后 PC 时代,嵌入式的发展逐渐取代 PC 成为时代主流,无论从生活领域、工业控制领域、军事领域、航空航天领域都有很好的发展前景。在硬件技术、软件技术以及网络信息技术日益成熟的今天,嵌入式突现出了强大的优势。它以其功能的可裁减和扩充性使得成本大大减少,成为商家关注的焦点,以其产品的小巧和灵便性赢得了最终用户的青睐。

2　了解嵌入式开发

由于嵌入式软/ 硬件对各模块的可用性要求比较苛刻,在软/ 硬件设计的时候应尽量减少资源使用,提高集成度。软件开发工具不可能安装到嵌入式硬件上,嵌入式软件的开发只有借助于 PC 完成。这就产生了开发和测试的问题,软件开发必然要伴随着测试。嵌入式软件由于需要特定的硬件运行平台,测试起来更加复杂。

嵌入式软件开发平台有通过仿真器模拟硬件平台进行开发,也有通过 JTAG、U 盘等外部设备把软件写入硬件进行开发等多种。本论文介绍一种通过网口和串口进行的软件开发平台。

3　建立嵌入式开发平台

3.1　开发平台硬件连接

目标板采用华恒 HHARM2410-FEC-USB-LCD-R1 开发套件。该套件有网口和串口支持,串口实现终端控制,网口实现局域网访问和数据传输,其硬件连接如图 1 所示。

宿主机的串口与目标板的串口相连,宿主机和目标板的网口接到同一个 HUB 上,同时保证宿主机和目标板的 IP 地址在同一网段。

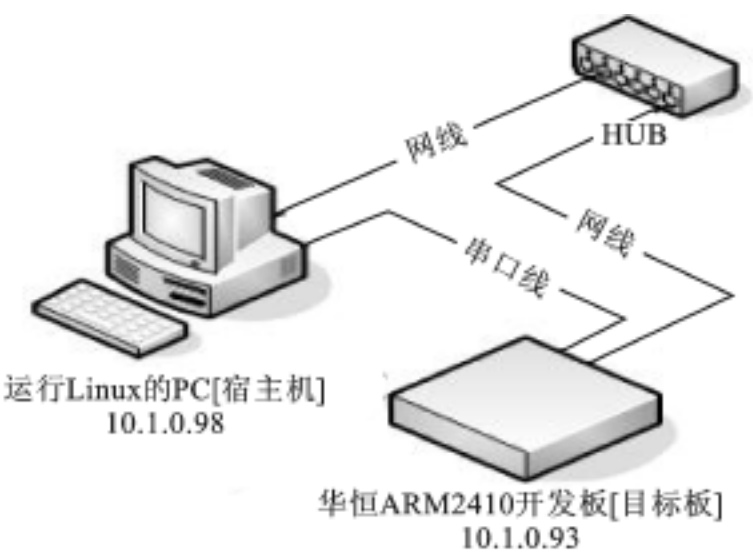

图 1　开发平台硬件连接图

3.2　开发平台软件设置

宿主机完全安装 RedHat 9.0,启动 TFTP 服务,将宿主机文件系统的根路径配置为 NFS 可读/写,禁用 FireWall。

建立宿主机与目标板的软件连接:

在宿主机上启动 minicom 作为目标板的仿真终端,启动目标板文件系统;

运行目标板中的 mount 命令,挂载宿主机文件系统(假设宿主机 IP 为 10.1.0.98):

mount - o nolock 10.1.0.98:/ / mnt

通过这种方法,在仿真终端的/ mnt 中就能看到宿主机的文件系统。宿主机上用交叉编译器编译过的软件能够直接在仿真终端中测试运行。测试运行成功,再把软件打包到目标板的文件系统供目标板烧写。

4　目标板的软件结构

包括三部分: bootloader(ppcboot)、zImage 和 ramdisk .image .gz。ppcboot 是编译好的引导程序,引导系统执行内核模块;zImage 是编译好的 kernal,配置硬件模块的驱动,使硬件能够在软件作用下正常工作;ramdisk .image .gz 是 ramdisk 文件系统映像,相当于文件系统,包含系统文件、命令和加入的应用。

目标板启动时,首先进入 ppcboot 引导程序,然后把 kernal 文件 zImage 复制到内存的相应位置解压缩并执行,初始化硬件驱动,接着复制 ramdisk .image .gz 到内存,解压缩,运行文件系统。

5　Minigui 文件系统移植

Minigui 是一个自由软件项目,为基于 Linux 的实时嵌入式系统提供了一个轻量级的图形用户界面支持系统。它提供了一组轻量级的窗口和图形设备接口。利用这些接口,用户可以方便地建立自己的嵌入式图形应用程序。如下是 Minigui 在本目标板的移植过程。

5 .1　下载源代码

下载 Minigui 源代码,当前版本是 libminigui-1 .3 .3 .tar .gz、mde -1 .3 .0 .tar .gz、minigui-res-1 .3 .3 .tar .gz、mg-samples-1 .3 .0 .tar .gz。

ibminigui-1 .3 .3 .tar .gz 是 Minigui 的图形库函数包。

mde-1 .3 .0 .tar .gz 是 Minigui 的综合演示程序。主要的演示程序有控件演示程序、对话框演示程序等。

minigui-res-1 .3 .3 .tar .gz 是针对 Minigui 1 .3 .x 版本发布的基本资源包,其中包含了运行 Minigui 1 .3 .x 需要的基本字体(ISO8859-1)、鼠标光标、图标和位图等文件。

mg-samples-1 .3 .0 .tar .gz 是 Minigui 的配套示例程序包。

在宿主机的根目录下建立工作目录 minigui- free,将下载的源代码包复制到该目录,分别解压缩源文件包到各自目录。在该目录中建立 nfsroot 目录用以存放生成的文件。

5 .2　在 libminigui-1 .3 .3 目录中配置 lib

终端中运行以下命令:

./ configure --host = arm - unknown-linux --enable-jpgsupport = no--enable-pngsupport = no --enable-gifsupport = no --disable-lite--prefix =/ minigui-free/ nfsroot --enable-smdk2410ial = yes

修改 configure,在文件开头处增加:

CC =/ opt/ host/ armv4l/ bin/ armv4l-unknown-linux-gcc

CPP =/ opt/ host/ armv4l/ bin/ armv4l-unknown-linux-cpp

LD =/ opt/ host/ armv4l/ bin/ armv4l-unknown-linux-ld

AR =/ opt/ host/ armv4l/ bin/ armv4l-unknown-linux-ar

RANLIB =/ opt/ host/ armv4l/ bin/ armv4l-unknown-linux-ranlib

STRIP =/ opt/ host/ armv4l/ bin/ armv4l-unknown-linux-strip

指明交叉编译器,必须保证在用户宿主机的/ opt 目录中存在上述编译器。

修改 lib 中的源文件 2410 .c,它存在于 src/ ial 目录中。告诉 Minigui 确定 lcd 的设备名和如何从内核中读取触摸屏的坐标值。

生成的文件在/ minigui-free/ nfsroot 目录中,包括 lib 等目录。因为采用的是动态连接,所以删除 * .a 和 * .la 等静态链接库文件,保证获得较小的文件系统。

执行/ opt/ host/ armv4l/ bin/ armv4l-unknown-linux-strip * 使库文件缩小。

参照 Minigui 的配置手册,按照需要用同样的方法配置 mde-1 .3 .0、minigui-res-1 .3 .3、mg-samples-1 .3 .1,安装到/ minigui-free/ nfsroot 目录下。

5 .3　制作 Ramdisk

创建一个简单的基于 Ext2fs 的 8M Ramdisk:

```
mke2fs -vm0 / dev/ ram 8192
mount -t ext2 / dev/ ram / mnt
```

需要全本请访问 www.ertongbook.com

```
cd / mnt
cp / bin,/ sbin,/ etc,/ dev  . . .files in mnt
                          // 将编译好的 Minigui 复制至/ mnt 并进行相应的系统配置
                          // 将 Minigui 演示程序 treeview 复制至/ bin
cd  ./
umount / mnt
dd if =/ dev/ ram bs = 1k count = 8192  of = ext2ramdisk
```

mke2fs 是用于在任何设备上创建 ext2 文件系统的实用程序。它创建超级块、索引节点以及索引节点表等等。

在上面的用法中,/ dev/ ram 是上面构建有 8192 个块的 ext2 文件系统的设备。然后,将这个设备(/ dev/ ram)挂装在名为 / mnt 的临时目录上并且复制所有必需的文件。一旦复制完这些文件,就卸装这个文件系统并且设备(/ dev/ ram)的内容被转储到一个文件(ext2ramdisk)中,它就是所需的 Ramdisk(Ext2 文件系统)。上面的顺序创建了一个 8 M Ramdisk,并用必需的实用程序文件来填充它。

Ramdisk 制作好以后,将其压缩为 .gz 文件烧写到目标板的 FLASH 中,重新启动目标板,直接执行 treeview,就能看到 treeview 的演示效果了。

6 结束语

以上介绍了利用串行口和网络建立嵌入式软件开发平台的方法,并讲解了利用这种方法进行 minigui 的移植过程。虽然嵌入式系统相对于计算机系统来说比较简单,但由于嵌入式系统力求小巧,模块独立性相对较弱,“麻雀虽小,五脏俱全”,要进行深入的嵌入式开发必须要对操作系统底层知识有良好的掌握。只有这样才能找到嵌入式软件与硬件的最佳结合点,实现完美的嵌入式开发。

参 考 文 献

1 刘文峰, 李程远 . 嵌入式 Linux 操作系统的研究 . 浙江大学学报(工学版),2004
2 北京飞漫软件技术有限公司 . MiniGUI 用户手册, 2003
3 华恒科技 . 华恒科技 HHARM2410 技术手册 . 2004
4 于明俭,陈向阳,方汉 . Linux 程序设计权威指南 . 北京:机械工业出版社,2001
5 W .Richard Stevens . UNIX 环境高级编程 . 北京:机械工业出版社,2000

Embedded Platform Building and Software Transplant

Yu Daohua Zhang Lifen
Dept . of Computer Science& Engineering, Beijing Institute of Technology, Beijing 100081, P . R . China

Abstract Embedded apply is the quickest prevailing micro-electronics domain in recent years . Emb edded development includes abundant technology . Aimed at broadly requirement on embedded development today, here brings forward an interactive software developing platform based on network and serial, and discusses the target software architecture, the hardware connection and the way of soft setting on developing platform . Based on research of target file system, here explains the process of software transplant and software testing, method of system configuration with the platform . In the end, a little advice is been put forward on entirely embedded development .

Key Words ARM, Minigui, Ramdisk, Host, Target, Bootloader, Kernel, Cross Compile

SoC 近十年如何最快挖到 SoC 第一桶金

梁合庆
国家电网公司南京电力自动化研究院,南京,5566636

摘　要　本文简介近十年 SoC、pSoC 的一般情况及其新的投资机遇。

关键词　SoC, pSoC, IP 核

1　SoC 基本情况

1.1　什么是 SoC

20 世纪 90 年代中期,受用 ASIC 技术实现芯片组的启发,萌生用各种功能块在一颗硅片上组合完整计算机应用系统的想法。这种芯片,初始起名叫 System-on-a- Chip(SoC),直译的中文名是系统级芯片。

如何界定 SoC,认识并未统一。但可以归纳如下:

oC 应由可设计重用的 IP 核组成。IP 核是具有复杂系统功能的能独立出售的 VLSI 块。

IP 核应采用深亚微米以上工艺技术。

SoC 中可以有一个或多个 MPU、DSP、MCU 或其复合的 IP 核。

由上可见,无论从功能、工艺和应用技术上,SoC 都是站在高的起点上利用重用技术克服同等水平复杂功能 VLSI 的重复设计,最大限度地在重用基础上设计新的 VLSI。

1.2　硬核、软核、固核

IP 核实现的是可设计重用的核,因此依设计阶段的不同而有:硬核、软核、固核之分,现概述如下:

① 硬核(Hard Core)　专为版图硬核。它以版图形式提交。硬核已经过性能、功耗、几何尺寸的优化,并映射为指定的工艺和经过实际流片的验证。只要符合整体 IC 工艺满足本核物理限制的就可使用。它的使用最为简单,但没有重用的灵活性。

② 软核(Soft Core)　专为 RTL 阶段重用的核叫软核。它以综合后的 RTL 描述提交,或以通用库元素的网表形式提交。此类重用块只通过性能和时序的验证,其他实现内容及相关测试验证事项均属使用者的责任。因后续的布局布线、功耗、几何尺寸,优化、工艺映射等等尚未进行,故用户有最大的灵活性。

③ 固核(Firm Core)　设计过程介于软核及硬核之间使用的核叫固核。它以综合后的代码或是通用库网表的形式提交。因只在布局规划的层面上对性能和几何尺寸等进行过结构性的或拓扑意义的优化和初步做过多种工艺的映射,因而用户的灵活性介于硬、软核之间。

选购何种 IP 核为佳?若着眼于:可被重用的频度、可向其他工艺移植的可能性、重用的灵活性等方面来说,则以软核最好,固核其次,硬核最差。若着眼于:保证确定的性能和规范、短的上市时间、自己付出劳动少并不在乎高价,则正好是倒过来,硬核最好,固核其次,软核最差。

1.3　SoC 的一般构成

从大处来分,SoC 含有:

逻辑核　包括 CPU、时钟电路、定时器、中断控制器、串并行接口、其他外围设备、I/O 端口以及用于各种 IP 核之间的粘合逻辑等等。

存储器核　包括各种易失、非易失以及 Cashe 等存储器。

模拟核　包括 ADC、DAC、PLL 以及一些高速电路中所用的模拟电路(如 1.6 Gbps@800 MHz 数据率的 RDRAM 的低摆幅信号传输部分、1394/ 1394.B 串行总线因传输彩色视频信号传输率达到 1 Gbps 的物理层以及 PCI 总线为克服 I/O 端口瓶颈而采用的 Hyper_LVDS 高速缓冲等)。

1.4　SoC 开发的两大问题

开发设计重用 IP 核的 SoC 存在设计和测试两方面的难题。

传统的设计方法和测试、开发工具和资源依然是 SoC 起步的参考。但是,又必须与时俱进,充分考虑到设计重用引发的诸多方法学、黑箱测试验证以及深亚微米技术引起的天线效应、信号流失等的特殊问题。硬/ 软

件协同设计新理论、协同设计新工具、硬软固核的开发导则和提交规范、核库的建设、可测性设计、可验证性设计以及方便前述各项实施的新型 EDA 高级设计验证综合平台的完善等等。1997 年伊始,VSIA(虚拟插座接口联盟)、Si2(硅上集成创始者)、RAIPD 网站(可重用专用 IP 开发者)等组织以及散见于其他杂志的前驱者们都发表过许多原创性的概念和规范文本,奠定了基于重用核 SoC 的基础。目前 SoC 论文与日俱增,成书及产品的已经不少,但是真正达到各个公司之间共享知识产权决非一蹴而就的事。

SoC 的根本难点在于产品上市时间的苛刻和必须是一步到位地直接送去流片产生,其次是要求技术的特点前沿。只因逼上梁山,非用整块复杂的知识产权去叠罗汉不能满足要求。眼下最好的例子就是手机,越是高档的,其内部越是只有一片技术含量越高的 SoC。

2 IC 业第二次重组

早期,IC 业一直是少数 IDM(集成器件制造厂)垄断的行业。IDM 从接受订单到器件的设计,到 IC 的加工等等都是自己封闭地完成,如颇具实力的 Intel、Motorola、IBM 等。随着 PC 机的普及和技术的发展,IDM 独立地自己扩大新建生产线,不仅耗资巨大而且时间上也来不及,外包生产自然顺理成章。这就是 20 世纪 80 年代末期,第一次出现 IC 业重组的动因。这次重组的特点是生产环节的外包。不料竟因此造就了一批新兴的专事生产晶园加工器件的独立产业,叫做 Foundy(代工厂)。IDM 将加工外包之后,剩下来的设计部门及器件销售部门等等泛称为 Fabless IDM,无对应的合适中文名。

20 世纪 90 年代末,进入 IP 核为基的 SoC 时代,设计部门因技术复杂,任务繁重,又进一步重组,留下系统设计部分而将 IP 核的设计部分实行外包。可以预断必将再造就一批新兴的 IP 设计服务产业。进一步外包后的 IDM 所剩下的部分便叫做 Chipless IDM,也无对应的合适中文名。

专事重要 IP 核设计的独立产业,其实 20 世纪 80 年代末已经出现,90 年代初便展露头角,它是专利授权公司。擅长 MPU 设计的 ARM 公司已执本行业的牛首。

有的专家预计,随着嵌入式产品竞争的进一步加剧,产品更新以季度为周期,促使 IDM 进一步地把剩下来的系统设计部门也狠心外包出去,自己则新成立一个产品创意部门,依赖原有实力,专门构想实用新型和另类应用产品,蜕化为概念设计。这时的 IDM,剩下的只是靠无形资产吸纳订单等部门了,名之为 Designless IDM,本无对应的中文名,如要硬套,便是皮包公司(非贬义)了。这次的外包,再造就的将是一批独立的 SoC 集成服务公司。

3 pSoC 捷足先登

晶体管的发明开创了 IC 产业。使用不同功能的 IC,可构成硬逻辑的专用型产品。如早期的计算器、早期的投币自动售货机和早期的键盘等。因为追求产品的灵活性,才发明了当今运行软件程序的微处理器的计算器。如果把软件固化,整体产品转回纯硬件逻辑,便是所谓 ASIC(专用集成电路)的专用型产品。有正必有反,与 ASIC 相对抗的是另一大类的硬件可编程逻辑器件 PLD。现在所使用的是 PLD 有两个子类:CPLD(复杂可编程逻辑器件)和 FPGA(现场可编程逻辑阵列)。20 世纪 90 年代末期提倡的 SoC 算是第三代的专用型产品。21 世纪初出现的 pSoC 则是第三代的通用型产品。现在归纳一下上述不同时期 IC 芯片的特征:

硬逻辑产品	第一代专用	软件不可编程	硬件不可配置
MPU(计算机产品)	第一代通用	软件可编程	硬件不可配置
ASIC(硬逻辑产品)	第二代专用	软件不可编程	硬件不可配置
PLD(硬逻辑产品)	第二代通用	软件不可编程	硬件可配置
SoC(计算机产品)	第三代专用	软件可编程	硬件不可配置
pSoC(计算机产品)	第三代通用	软件可编程	硬件可配置

pSoC 是硬件可配置的 SoC。因 SoC 本是软件可编程的,所以 pSoC 的用户可获得软、硬两个方面的编程灵活性。

pSoC 一般是在 FPGA 硅片上,嵌入 MCU IP 核的 SoC。

2000 年,Altera 公司生产了首例嵌入 32 位 ARM IP 核的 Excolibur pSoC。接着又生产出 32/ 16 位 Nios MPU 软核的 Stratix pSoC。Xilinx 生产的是 32 位 PowerPC MPU 核的 Virtex II pSoC。其他 PLD 制造商也都纷纷的在自己的 FPGA 上嵌入了不同位数的 MPU/ MCU。

一般各 pSoC 生产厂都提供一块装有 pSoC 的开发板,为用户备好硬件/ 软件同时开发的环境,帮助目标产品的开发及提前上市。这些 PLD 制造商都提供现成的硬件编程工具和丰富的逻辑元件库供用户使用。

使用 pSoC 开发 SoC 产品比直接 SoC 产品开发有如下优点:

① 设计重用 IP 核开发 SoC 方法尚处建设阶段,而 pSoC 方法已经现实可用。

② pSoC 著名厂商多俱 20 多年坚实功底,其硬件编程的各种工具和兼顾第三方的功能库都已精良完善。基于插表的 FPGA 已经达到,在单片上构建多核 DSP 乃至多个 DSP 处理器的水平,满足当前通信、音视频和图像应用的需要。

③ pSoC 绕开了 SoC 方法虚拟插口的困扰、IP 的黑箱认证的困难,同时可避免了 SoC 整个的 IC 设计阶段和整个流片生产过程。所谓开发仅是对现成 pSoC 进行硬件系统的配置和软件的编程。

④ 有力于大幅度地缩短上市时间。

⑤ 经济实惠。

不足之处:

① pSoC 芯片上的资源不免会比 SoC 有所浪费。

② 片上处理器受限于厂家的设定,缺乏选择的灵活性。

4　老草开新花

前面介绍的是 32 位 MPU 的 pSoC,服务于当前热门的高端嵌入式消费类应用。现在看 8 位 MCU 的 pSoC,适于低端嵌入式控制应用。

8 位 MCU pSoC 的现状归纳如下:

① MCS－51 核的居多,有 ST 公司的 μPSD32xx、Cypress 的 CY8C264xx 和 CY8C266xx、Triscend 的 E5。Sygnal 的 C8051Fxxx 不属于 pSoC。

② 非 MCS－51 核的 pSoC 只发现一家。它是 Atmel 公司的 8 位 RISC AVR 核 MCU pSoC, AT94K40。

③ 它们各自都提供配置硬件的软件工具,有的还是免费的。

下面选 ST 公司的 μPSD32xx 来说明一下 pSoC 的结构。选它的理由是:

① MCS－51 核是 MCU 的事实标准,应用最广,其原理和开发已为广大工程界所熟知。仅需学习它的硬件编程部分和配置硬件。它的硬件编程工具是免费。

② 硬件编程的 PLD 部分是过去大家熟知的 WSI 公司的技术,即当初专门与 MCU/ MPU 配套的可编程外转芯片。因 WSI 公司现被 ST 收购,专门在其上嵌入 51 核构成 pSoC。片上沉淀者 WSI 20 多年实践经验。

图 1 是 μPSD32xx 的简要框图。下面是图 1 的说明:



图 1　pSoC 方块图(ST μPSD32xx)

① 图 1 分两大部分,硬件固定部分和硬件可配置部分。

② 硬件固定部分有两块:MCU 核和 MCU 的扩展。硬件可配置部分有四块:存储器块、DPLD、CPLD、PPLD。

③ MCU 核是 8032 核。

④ MCU 扩展块有:2 个 I^2C 核、第二个 UART、5 通道 PWM、USB、复位/ 监控/ 看门狗等,补充 8052 核的不足。众所周知,8051 已有 20 多年的历史,是最老的 MCU,目前有数十个厂家在更新和继续生产,更新集中在六个方面,详情请见文献 4(2003 年底出版新书)。

⑤ 存储器块包括:1-FLASH 256 KB、2-FLASH 32KB、带备用电池 SRAM 8 KB。每块存储器的任意部分都可通过 DPLD 为它们配置片选信号。配置每一存储块到数据或程序空间、外部或内部空间。运行中间可以动态地改变所在的存储空间。在页寄存器的帮助下 DPLD 可以突破原来 64K 寻址空间的限制,而一直扩大到 16 MB,满足现今对大存储的需要。

⑥ DPLD 是地址解码 PLD,专门用于配置片选信号,已如上述。

⑦ CPLD 是复杂 PLD。因为具有宏单元,专为配置额外的 I/ O 端口之用。

⑧ PPLD 是外设 CPLD。它也具有宏单元,用于配置其他需要的逻辑部件,这部分可以与前项的 CPLD 合并使用。

⑨ 硬件的配置通过 JTAG 接口实现,无需另外的烧写器。FLASH 的烧写也同样。

⑩ 使用独立的 1-FLASH、2-FLASH 实现动态的 IAP,更新软件不必停止当前程序运行。

1 合理地配置存储器的分页,能实现比一般方法简单得多的跨页超大程序的运行。

2 免费提供不用 HDL 语言的 μPSDSoft Express 软件。用它完成硬件配置,简单易学。

5 如何最快地挖到 SoC 的第一桶金

5.1 SoC 高层次挖金方案

由 1、2 部分知,进入 SoC 领域挖金需要有计算机和微电子行业较广的知识和深入的专业技术。容易挖到金的地方如下:

① Foundy(代工厂)是专门负责晶园到 IC 的加工厂,不负责 IC 的电子电路设计。Foundy 适合单纯资金优势的投资者经营。它的投资额度较大,IDM 自己多不愿重金投入初始吃不饱的生产线,正好是专门承揽加工业务的 Foundy 发展的良机。投资瞄准的方向应是前途看好的、最先进的生产线。

② IP 设计的服务产业和 SoC 集成服务公司都是纯技术性投入的新兴产业。它立足于自己开发,投资不需很大。对于技术水平勉强并无真实的感悟者,恐怕不能冒然进入这样的行业。尤其是目前关于 IP 核和 SoC 尚无公认的标准,风险很大,尤其目前,还有捷足先登的基于 FPGA 的 32 位 pSoC 的竞争。

5.2 32 位 pSoC 层次挖金方案

开辟基于 FPGA pSoC 的设计服务产业,是当前比较稳健而又现实的业务。现在,立地就可以开发产品;将来,在穷尽 pSoC 片上 FPGA 资源时也许就会裂变,走出另一条通向 SoC 的捷径也说不定。这是一条适合高中级技术人才发展的丘陵之地。

5.3 8 位 pSoC 层次挖金方案

嵌入 8 位 MCU pSoC 产品的开发是一般技术人员都可走出的一条最容易的挖金坦途。8 位 MCU 极容易上手,于是你手上就握有一把软件编程的剑;pSoC 又教你掌握另一把硬件可编程的剑。如果所选 pSoC 嵌入的是核,你只需磨快硬件可编程的剑,学习难度一下子就减少了一半。前面已经讲过。MCS-51 与 PLD 的结合,能轻松地突破 64K 存储界限,能轻松取得灵活的动态存储管理,能获得动态 IAP 等独到的长处等等。故,从 8 位 pSoC 切入硬件可编程的门槛,对谁都是最简单的进取之路。掌握之后,再换用 32 位 pSoC,展现眼前的又将是新的一马平川。只要你使用 MCS-51 核 pSoC 去更新老的 MCS-51 产品,任何人都有条件轻松挖到第一桶金。祝君成功!

参 考 文 献

1 Rochit Rajsuman . System-on-a-Chip: Design and Test . Artech House

2 Altera . 20 Years of Altera Innovation . June, 2003

3 Xilinx . Data sheets . 2003

4 梁合庆 . 增强核闪存 80C51 教程 . 北京:电子工业出版社, 2003

SoC Review and New Investing Fields

Liang Heqing

Abstract Reviewing the progress on SoC and pSoC, discussing the new investing fields of SoC as well .

Key Words SoC, pSoC, IP Core

嵌入式系统中的逻辑文件系统

向　玢　　赵宏图
北京理工大学电子工程系,北京,100081

摘　要　本文给出了对逻辑文件系统的理解,并介绍了一个嵌入式系统中逻辑文件系统的构建方法。文章从一个普通 Linux 系统的启动过程入手引出了逻辑文件系统。分析了一个重要的配置文件 inittab。并给出了一个简单的逻辑文件系统的制作流程,重点是/ bin、/ sbin、/ dev、/ etc 等目录的构建。通过介绍 busybox、tinylogin 工具,介绍了/ bin、/ sbin 目录的构建。根据实际情况介绍了/ etc 目录中的主要配置文件,注意到了与 pc 机的区别,并给出了一个简单的启动脚本。联系开发板的具体情况给出了/ dev 目录的构成。最后对文件系统进行了打包和烧写工作。

关键词　逻辑文件系统,Inittab,Busybox,嵌入式系统

1　引　言

我们通常所说的文件系统分为两个含义:一个含义是磁盘和磁盘机制的文件系统;另一个含义是用户看得见并能操作的逻辑文件系统。在本文中,笔者所介绍的是一个典型嵌入式系统中的逻辑文件系统的构建。开发所处的硬件环境是基于 XScale 构架 PXA255 处理器的开发平台,软件环境是 redhat 9 .0,编译环境是 arm-linux-gcc,开发板所使用的 bootloader 为 redboot,操作系统为 linux-2 .4 .19,文件系统为 jffs2。

2　从启动说起

说到逻辑文件系统的构建就不得不从系统的启动说起。redboot 启动以后会由它来引导内核,当内核启动了自己之后(已被装入内存、已经开始运行、已经初始化了所有的设备驱动程序和数据结构等等),通过启动用户级程序 init 来完成引导进程的内核部分。因此,init 总是第一个进程(它的进程号总是 1)。

内核在几个位置上来查寻 init,但是 init 的最适当的位置(在 Linux 系统上)是/ sbin/ init。如果内核没有找到 init,它就会试着运行/ bin/ sh;如果还是失败了,那么系统的启动就宣告失败了。

当 init 启动后,init 读取/ etc/ inittab 配置文件。etc/ inittab 中的行由 4 个冒号限定的域组成:

id:runlevels:action:process

id 是确定文件中的一行。对于 getty 行来说,指定了它在其上运行的终端(设备文件名/ dev/ tty 后面的字符)。对于别的行来说,是没有意义的(除了有长度的限制),但它必须是惟一的。runlevels 为该 process 的运行级别。action 描述了用户想让系统采取什么行动,即 respawn、askfirst、wait 等。process 为要运行的可执行程序的完整路径以及相应传递给它的任意参数,中间用空格符分隔。

以下是一个的/ etc/ inittab 文件的例子(只列出了比较重要的几行):

id:3:initdefault:

系统默认的运行级别为 3。

si::sysinit:/ etc/ rc .d/ rc .sysinit

系统启动时会自动运行 rc .sysinit 脚本,它主要完成基本的系统初始化命令,如激活交换分区、检查并挂上文件系统、装载部分模块等。

id:*:wait:/ etc/ rc .d/ rc *

当进入不同运行级别时,将运行对应的/ etc/ rc .d/ rc * .d 脚本。init 将等待其返回,其中 0 为终止系统,1 为用户模式(用于特别管理),2～5 为正常操作,6 为系统重启。这些脚本都是到/ etc/ init .d 中启动脚本的符号链接。由这些脚本来启动或终止系统的一系列服务。

1:2345:respawn:/ sbin/ getty tty1

在 2～5 各个级别上以 tty1 为参数执行/ sbin/ getty 程序,打开 tty1 终端用于用户登录。如果进程退出则再次运行 getty 程序。在这一行中,id 要与所使用的 tty 的次设备号对应。

需要全本请访问 www.ertongbook.com