

<<达芬奇技术开发基础、原理与实>>

图书基本信息

书名：<<达芬奇技术开发基础、原理与实例>>

13位ISBN编号：9787121169311

10位ISBN编号：7121169312

出版时间：2012-6

出版时间：电子工业出版社

作者：高玉龙，白旭，吴玮 编著

页数：302

字数：499000

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<达芬奇技术开发基础、原理与实>>

内容概要

DaVinci技术是TI公司专门为视频和音频处理而设计的基于DSP的系统解决方案。它不仅仅是DSP+ARM双核架构的SoC，而且包括一整套的软件开发包，涉及嵌入式系统、ARM、DSP及数字图像处理等多个学科的知识，有着较为复杂的开发流程。但由于其集成度高、性能卓越而受到越来越多人的关注。目前达芬奇技术相关资料偏少，给开发过程带来了很大困难。本书从达芬奇技术最基本的要求——Linux操作系统入手，全面介绍基于达芬奇技术的图像编解码系统开发的整个过程及所需的相关知识，主要包括软硬件环境的搭建、bootloader、文件系统的设计、应用程序设计、驱动程序设计、FPGA接口设计等内容。逐步引导读者进行达芬奇技术的开发，最后设计和实现一个图像无线传输系统。本书能使读者掌握达芬奇技术的开发、调试方法，使毫无任何DaVinci知识的开发者快速进行实际系统的设计开发。

书籍目录

第1章 Linux操作系统基础知识

1.1 引言

1.2 Linux文件和文件系统

1.2.1 Linux文件概念

1.2.2 Linux文件系统结构

1.2.3 简单文件操作

1.2.4 简单目录操作

1.3 基于用户权限的文件保护

1.3.1 用户分类

1.3.2 文件操作/访问权限的分类

1.3.3 改变文件或目录的访问权限的命令

1.4 文本编辑工具Vi

1.4.1 Vi的三种工作模式

1.4.2 Vi常用操作命令

第2章 基于Linux操作系统的C语言程序开发工具

2.1 引言

2.2 GCC编译器

2.2.1 GCC的基本用法

2.2.2 交叉编译

2.3 工程管理工具make

2.3.1 Makefile文件介绍

2.3.2 make命令的使用

2.4 GDB调试

2.4.1 GDB的使用

2.4.2 GDB中运行程序

2.4.3 暂停/恢复程序运行

2.5 Linux系统中的C语言库函数

2.5.1 标准I/O函数

2.5.2 格式化输入、输出函数

2.5.3 字符串函数

第3章 视频、音频编解码标准

3.1 引言

3.2 视频压缩编码的基本原理

3.2.1 视频压缩原理

3.2.2 视频压缩算法的研究进展

3.2.3 现有的图像压缩标准概述

3.3 JPEG 2000压缩标准

3.3.1 JPEG 2000核心编码系统结构

3.3.2 小波变换在JPEG 2000标准中的应用

3.3.3 优化截断块编码

3.3.4 JPEG 2000标准算法压缩性能分析

3.4 H.264压缩标准

3.4.1 H.264标准的结构框架

3.4.2 H.264标准的档次

3.4.3 H.264标准的编解码器结构

<<达芬奇技术开发基础、原理与实>>

- 3.4.4 H.264编解码新技术
- 3.4.5 H.264的主要技术特点
- 3.4.6 H.264的网络与容错技术
- 3.4.7 H.264的应用前景
- 3.5 音频编码算法和标准
 - 3.5.1 音频信号冗余度
 - 3.5.2 音频压缩的关键技术
 - 3.5.3 音频编码标准概述
- 第4章 达芬奇技术软硬件原理及开发流程
 - 4.1 引言
 - 4.2 达芬奇技术概述
 - 4.2.1 达芬奇处理器
 - 4.2.2 达芬奇开发工具
 - 4.2.3 达芬奇软件
 - 4.2.4 达芬奇第三方系统专业技术支持
 - 4.3 达芬奇处理器TMS320DM365芯片
 - 4.3.1 TMS320DM365概述
 - 4.3.2 TMS320DM365配置
 - 4.3.3 ARM子系统
 - 4.3.4 DSP协处理器
 - 4.3.5 视频处理前端
 - 4.3.6 视频处理后端
 - 4.3.7 异步外部存储器接口
 - 4.3.8 IIC总线
 - 4.4 达芬奇处理器软件
 - 4.4.1 达芬奇软件算法遵循规则——xDAIS和xDM
 - 4.4.2 达芬奇软件架构和算法开发步骤
 - 4.4.3 编解码引擎和服务
 - 4.4.4 CMEM和DSPLINK
 - 4.5 基于达芬奇平台的应用程序的启动过程和开发流程
 - 4.5.1 达芬奇芯片中各种程序的启动过程
 - 4.5.2 达芬奇芯片开发流程
- 第5章 构建达芬奇技术开发的Linux系统环境
 - 5.1 引言
 - 5.2 Linux操作系统的安装及配置
 - 5.2.1 Fedora 10的安装
 - 5.2.2 根用户登录方法
 - 5.2.3 分辨率的修改
 - 5.2.4 系统时间的修改
 - 5.3 服务器配置
 - 5.3.1 TFTP服务器的安装和配置
 - 5.3.2 DHCP服务器的安装和配置
 - 5.3.3 NFS服务器的安装
 - 5.3.4 其他服务的安装
 - 5.4 minicom的安装及使用
 - 5.4.1 minicom的设置和使用
 - 5.4.2 故障现象解决方法

<<达芬奇技术开发基础、原理与实>>

5.5 达芬奇开发软件的安装

5.5.1 目标Linux软件的安装

5.5.2 DVSDK的安装

5.5.3 安装A/V demo文件

5.5.4 网络文件系统的设置

5.5.5 交叉编译环境的测试

第6章 BOOTLOAD及其在达芬奇芯片中的应用

6.1 引言

6.2 BOOTLOAD概述

6.2.1 BOOTLOAD的作用和启动模式

6.2.2 BOOTLOAD的分类

6.2.3 BOOTLOAD的启动过程

6.3 U-BOOT介绍

6.4 U-BOOT的常用命令和变量

6.4.1 U-BOOT的常用命令

6.4.2 U-BOOT的常用变量

6.5 U-BOOT在达芬奇芯片中的应用

6.5.1 更新或升级U-BOOT

6.5.2 采用仿真器和CCStudio烧写UBL和U-BOOT

6.6 U-BOOT的编译与烧写

6.6.1 交叉编译器的安装与配置

6.6.2 U-BOOT的编译

6.6.3 U-BOOT的烧写

第7章 达芬奇技术的嵌入式Linux内核

7.1 引言

7.2 嵌入式系统内核概述

7.2.1 UNIX简介

7.2.2 Linux简介

7.2.3 操作系统和内核简介

7.2.4 Linux内核版本

7.2.5 Linux发行版

7.2.6 嵌入式Linux的发行版

7.3 嵌入式Linux内核组成

7.3.1 信号

7.3.2 系统调用

7.3.3 进程和进程调度

7.3.4 内存管理

7.3.5 虚拟文件系统

7.3.6 进程通信

7.3.7 设备驱动

7.4 内核编译及移植

7.4.1 内核源码结构

7.4.2 Makefile在内核编译时的作用

7.4.3 Kconfig的作用

7.4.4 Linux内核配置选项

7.4.5 Linux内核编译的过程

7.4.6 Linux内核的编译

<<达芬奇技术开发基础、原理与实>>

第8章 达芬奇芯片的文件系统

8.1 引言

8.2 Linux文件系统的概念

8.3 Linux的文件系统

8.3.1 Ext2文件系统

8.3.2 Ext3文件系统

8.3.3 Ext4文件系统

8.3.4 JFFS2文件系统

8.3.5 YAFFS文件系统

8.4 移植Busybox

8.5 构建根文件系统

8.5.1 建立根文件系统目录

8.5.2 构建etc目录下的配置文件

8.6 制作根文件系统映象文件

8.6.1 制作YAFFS映象文件

8.6.2 制作JFFS2映象文件

第9章 嵌入式Linux下的驱动程序设计

9.1 引言

9.2 设备驱动程序的基本知识

9.2.1 嵌入式Linux设备

9.2.2 用户空间和内核空间

9.2.3 I/O端口操作

9.3 字符设备驱动程序

9.3.1 数据结构

9.3.2 dchar字符设备的设计

9.3.3 设备注册、打开和释放

9.3.4 设备的读和写

9.4 块设备驱动

9.4.1 注册

9.4.2 块设备操作

9.4.3 请求处理

第10章 基于H.264的图像传输系统设计与实现

10.1 引言

10.2 图像传输系统设计

10.3 H.264图像压缩板硬件方案设计

10.3.1 压缩端硬件方案设计

10.3.2 电源管理

10.3.3 时钟管理

10.3.4 模数转换模块

10.3.5 存储器模块

10.3.6 调试接口部分

10.4 发送部分基带板设计

10.4.1 发射机信道编码芯片

10.4.2 扩频和调制

10.5 发射机射频前端

10.6 接收机射频前端

10.7 接收机基带处理

<<达芬奇技术开发基础、原理与实>>

- 10.7.1 解扩解调
- 10.7.2 信道解码
- 10.8 图像解压缩方案设计
- 第11章 H.264压缩解压缩软件方案设计与实现
- 11.1 引言
- 11.2 TMS320DM365的H.264编解码Buffer机制和参数设置
- 11.2.1 H.264编解码Buffer机制
- 11.2.2 H.264编解码参数含义和设置
- 11.3 H.264压缩程序设计与实现
- 11.3.1 主线程
- 11.3.2 控制线程
- 11.3.3 视频线程
- 11.4 EMIF驱动程序设计与实现
- 11.5 FPGA压缩端程序设计与实现
- 11.5.1 锁相环altpll0子模块
- 11.5.2 压缩feq_division子模块
- 11.5.3 image_process子模块
- 11.6 FPGA解压缩端程序设计与实现
- 11.6.1 锁相环altpll0子模块
- 11.6.2 解压缩feq_division子模块
- 11.6.3 write_to_EMIF子模块
- 11.7 解压缩读驱动程序设计与实现
- 11.8 解压缩软件方案设计
- 11.8.1 主线程
- 11.8.2 数据接收线程
- 11.8.3 视频解码线程
- 11.8.4 视频显示线程
- 11.8.5 解压缩线程的交互
- 11.8.6 控制线程
- 11.9 自启动流程设置
- 附录A FPGA程序
- A-1 压缩feq_division子模块
- A-2 image_process子模块
- A-3 解压缩feq_division子模块
- A-4 write_to_EMIF子模块
- 附录B EMIF驱动程序
- 参考文献

章节摘录

版权页：插图： JFFS (Jouralling Flash File System) 日志闪存文件系统是特别为嵌入式系统开发设计的文件系统，该文件系统基于Linux2.0的内核。

JFFS文件系统基于MTD驱动层，主要用于NOR型闪存。

JFFS文件系统是基于哈希表的日志型文件系统，它提供了崩溃/掉电安全保护和“写平衡”支持，并且该文件系统可读写，支持数据压缩。

JFFS2是RedHat公司基于JFFS开发的闪存文件系统，主要是为RedHat公司的eCos开发的。

JFFS2的特点是采用非顺序日志结构以及支持数据压缩、硬链接和多种节点类型。

JFFS2提高了对闪存的利用率，降低了闪存的消耗。

JFFS2的碎片收集对象和删除对象基于一个扇区而不是整个文件系统，从而缩短了碎片整理和删除操作的时间。

当遇到坏扇区时，则对其进行标记而使用可用的扇区，增强了设备的写生命周期。

JFFS文件系统的缺点也非常明显：当文件系统将满或已满时，JFFS的运行速度大大降低。

同时，JFFS不适合用于类似NAND Flash这样的大容量闪存。

8.3.5 YAFFS文件系统 YAFFS (Yet Another Flash File System) 文件系统是专门针对NAND闪存设计的嵌入式文件系统，目前有YAFFS和YAFFS2两个版本，两个版本的主要区别之一在于YAFFS只针对页大小为512字节的NAND Flash，而YAFFS2能够更好地支持大容量的NANDFlash芯片。

YAFFS文件系统与JFFS文件系统相似，但YAFFS文件系统特别针对NAND Flash芯片进行了优化，使芯片的寿命更长，运行速度更快，占用内存更少，但其稳定性较差。

<<达芬奇技术开发基础、原理与实>>

编辑推荐

《达芬奇技术开发基础、原理与实例》集理论、实现、案例于一体，适合已具备C语言、VHDL语言基本知识的通信工程、电子工程、电气工程、计算机工程及计算机科学与技术等专业方向的高年级本科生和研究生作为相关课程的参考书和教材，也可供相关的教师和工程技术人员参考使用。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>