

《达芬奇技术》

图书基本信息

书名：《达芬奇技术》

13位ISBN编号：9787121071942

10位ISBN编号：7121071940

出版时间：2008-9

出版社：电子工业出版社

作者：彭启琮 编

页数：308

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu000.com

《达芬奇技术》

前言

数字视频正在带来一场革命。增加视频功能，可以使各种数不胜数的电子产品及嵌入式应用大大增加附加值；用户可以和视频交互，开启各种基于视频服务的大门，包括视频点播、互动和导向等：先进的监视系统可以采集视频信号并实时处理，自动跟踪目标，还可以改善服务的可靠性；基于视频的医疗服务，使医生和护士得到的信息比文本的指令要有效得多。然而，随着基于视频应用的迅速增加，开发者面临着更大的压力，因为数字视频的实现是极其复杂的事情。开发者往往需要花很长的时间来熟悉各种多媒体的标准，而后者又在不停地改变。

《达芬奇技术》

内容概要

《DSP应用丛书·达芬奇技术:数字图像/视频信号处理新平台》讲述达芬奇技术是一种数字图像、视频、语音、音频信号处理的新平台，一经推出，就受到热烈欢迎，以其为基础的应用开发层出不穷。该技术是一种内涵丰富的综合体，包含达芬奇处理器、软件、开发环境、算法库和其他技术支持等。正因为涉及的技术面广，因此有比较高的技术门槛。

书籍目录

第1章 概述	1.1 DaVinci技术所面临的主要挑战和创新	1.1.1 数字视频所面临的挑战	1.1.2 API的强大功能	1.1.3 支持实时操作系统Linux及windows CE	1.1.4 实际的视频	1.1.5 编程的灵活性																																																																																																			
1.2 DaVinci技术的内涵	1.2.1 Davinci处理器	1.2.2 DaVinci软件	1.2.3 DaVinci的开发工具	第2章 DaVinci处理器	2.1 TMS320C64x+DSP	2.1.1 TMs320C64x/TMs320C64x+DsP的特性	2.1.2 TMs320c64x / TMS320C64x+DSP的结构	2.1.3 TMS320C64x+指令集	2.1.4 TMS320C64x/TMS320C64x+DSP的寻址方式																																																																																																
2.2 TMS320DM644x DMSoC DSP子系统	2.2.1 概述	2.2.2 TMS320C64x+大模块	2.2.3 存储器映射	2.2.4 ARM内部存储器	2.2.5 片内外设	2.2.6 器件外设	2.2.7 ARM—DSP集成	2.2.8 DSP子系统的时钟	2.2.9 电源管理	2.2.10 引导和复位																																																																																															
2.3 TMS320DM644x DMSoC中的ARM子系统	2.3.1 ARM子系统概述	2.3.2 ARM核	2.3.3 系统存储器	2.3.4 器件时钟	2.3.5 PLL控制器	2.3.6 电源与休眠控制器	2.3.7 电源管理	2.3.8 中断控制器	2.3.9 系统控制模块	2.3.10 复位	2.3.11 引导方式	2.3.12 ARM—DSP集成																																																																																													
2.4 TMS320DM644x的外设	2.4.1 概述	2.4.2 异步外部存储器接口 (EMIF)	2.4.3 音频串口 (ASP)	2.4.4 ATA控制器	2.4.5 DDR2存储器控制器	2.4.6 增强的直接存储器访问 (EDMA) 控制器	2.4.7 以太网媒体访问控制器 (EMAc) / 管理数据输入 / 输出 (MDIO) 模块	2.4.8 通用输入 / 输出 (GPIO)	2.4.9 内部集成电路 (I2C) 模块	2.4.10 内部直接存储器访问 (IDMA) 控制器	2.4.11 DsP子系统中断控制器 (INTc)	2.4.12 多媒体卡 (MMC) / 安全数字 (sD) 卡控制器	2.4.13 节电控制器 (PDC)	2.4.14 脉冲宽度调制器 (PwM)	2.4.15 串行外设接口 (SPI)	2.4.16 64位定时器	2.4.17 通用异步收发器 (uART)	2.4.18 通用串行总线 (uSB)	2.4.19 VLYNQ接口	2.4.20 视频处理后端 (vPBE)	2.4.21 视频处理前端 (vPFE)																																																																																				
2.5 其他的DaVinci处理器	2.5.1 TMS320DM6441处理器	2.5.2 TMS320DM643x处理器	2.5.3 TMS320DM355处理器	2.5.4 TMS320DM64x处理器	2.5.5 TMS320DM6467处理器	第3章 DaVinci的软件	3.1 xDAIS和xDM	3.1.1 xDM和xDAIS之间的关系	3.1.2 xDAIS界面	3.1.3 xDAIs和xDM的优越性	3.1.4 如何建立和验证符合express DsP的算法	3.1.5 建立和验证符合xDAIS和xDM的算法	3.1.6 修改自己的算法, 使其符合xDM和xDAIs	3.1.7 测试Codec Engine里的用户算法	3.1.8 多媒体框架产品 (MFP)	3.2 Codec Engine	3.2.1 为什么要使用Codec Engine	3.2.2 Codec Engine在应用程序结构中的位置	3.2.3 用户的角色	3.2.4 Codec Engine的安装和设置	3.2.5 Codec Engine的目录结构	3.2.6 使用范例应用程序	3.2.7 使用Codec Engine API	3.2.8 VA类型: 视频、图像、语音和音频	3.2.9 关于dSP存储器	3.2.10 DSP的实时处理问题	3.2.11 软件跟踪	3.3 Codlec Engine框架和xDAIS算法包	3.3.1 启动	3.3.2 建立包	3.3.3 建立一个发布包	3.3.4 开发一个xDM Codec	3.3.5 支持非xDM的算法	3.3.6 建立Codec Engine扩展	3.3.7 设计一个新的应用程序界面	3.3.8 开发存根和骨架	3.3.9 打包和配置核算法	3.3.10 非xDM存根和骨架的范例: SCALE	3.4 DSP / BIOS LINK	3.4.1 DSP / BIOS LINK的软件结构	3.4.2 DSP / BIOS LINK的主要成分	3.4.3 DSP / BIOS LINK源代码的布局	3.4.4 定制和配置所建立的开发环境	3.4.5 建立源代码	3.5 Linux	3.5.1 概述	3.5.2 Linux的主要开发工具	3.5.3 构建基于Linux的嵌入式系统	3.5.4 MontaVista Linux的特点	3.5.5 DaVinci的Linux开发流程	3.6 基于Davinci的数字媒体软件	3.6.1 H.264 Baseline Profile (BP) 编码器和解码器	3.6.2 wMV9 解码器	3.6.3 WMA9 (WindOWS Media 9 Series Audio) 解码器	3.6.4 MPEG-4 / H.263 COdec	3.6.5 MPEG-2解码器	3.6.6 解内插库 (Deinterlacing Library)	3.6.7 JPEG编码器 / 解码器	3.6.8 G.711编码器, 解码器	3.6.9 MPEG—4 AAC—HE解码器	3.6.10 MP3解码器	3.7 TMS320C64x+IMGLIB图像 / 视频处理库	3.7.1 概述	3.7.2 特性与优点	3.7.3 安装IMGLIB	3.7.4 使用IMGLIB	3.7.5 IMGLIB函数介绍	第4章 开发工具与软件	4.1 适用于DaVinci的CCS	4.1.1 v3.2的更新	4.1.2 CCS IDE v3.34.2 XDC (eXDress DSP Components)	4.2.1 XDC术语	4.2.2 使用基于xDc的软件包	4.2.3 写C代码	4.2.4 处理配置	4.2.5 编译和连接	4.3 数字视频评估模块 (DVEVM) 及其使用	4.3.1 DVEVM概述	4.3.2 DVEVM的硬件设置	4.3.3 运行演示软件	4.3.4 DvEvM的软件设置	4.3.5 启动建立开发环境	4.3.6 为用户的目标板重建DvEVM软件	4.3.7 建立一个新的Linux核	4.3.8 引导新的Linux核	4.4 数字视频软件开发包 (DVSDK)	4.4.1 概述	4.4.2 DVSDK的主要特点	4.4.3 可视化数据分析器	4.5 DaVinci系统级指标测试	4.5.1 概述	4.5.2 DVEVM演示范例软件说明	4.5.3 所需要的设备和软件	4.5.4 测量演示范例程序的处理器负载	4.5.5 关于DM644x soC分析器的结果	4.5.6 演示范例的存储器使用	4.5.7 演示范例的功耗测量	第5章 DaVincj技术应用案例	5.1 建立一个小的Linux核	5.1.1 所需的环境	5.1.2 性能选择及建立核的步骤	5.1.3 建立一个RAM DISK文件系统	5.1.4 支持应用程序	5.1.5 将信息复制到NOR Flash	5.1.6 引

《达芬奇技术》

导5.2 DVEVM / DVsDK用于编码的演示范例5.2.1 概述5.2.2 应用程序设计5.2.3 配合应用程序5.3
DVEVM / DVSDK用于解码的演示范例5.3.1 概述5.3.2 应用程序设计5.3.3 配合应用程序5.4
DVEVM / DVSDK用于编码和解码的演示范例5.4.1 概述5.4.2 应用程序设计5.4.3 用其他的Codec
来替换该编码和解码算法5.5 运行在DM6446上的活动JPEG演示范例5.5.1 演示范例简介5.5.2 演示
范例包的内容5.5.3 从DM642移植到DM64465.5.4 运行5.5.5 重新编译演示范例参考文献

《达芬奇技术》

章节摘录

互动数字视频正在大踏步地进入人们的工作和生活，各种电子产品和嵌入式应用，由于增加视频功能而提高了价值。例如，用户可以像对待图像和声音文件那样，与视频文件互动；各种基于视频的服务，丰富多彩，如视频点播、导向等；救援系统可以通过视频来提高及时性与可靠性；对于医护人员，基于视频的医疗设备比文字指令要有效得多，等等。然而，数字视频的实现，是十分复杂的事情，开发人员要花很长的时间来熟悉多媒体的标准，而这些标准又随着技术的发展在不停地改变。已有的数字视频实现，往往和特定的硬件平台和操作系统紧紧地联系在一起，使得开发人员只能使用手工编程。这是复杂、耗时、高成本的过程。为了应对这种强劲的需求，2005年，TI推出了DaVinci（达芬奇）技术，其应用目标就是数字视频。它将固定功能器件的高效率和可编程器件的灵活性结合在一起，支持各种数字视频的终端设备，诸如IP机顶盒、视频会议系统、便携式媒体播放器及数码相机等。

《达芬奇技术》

编辑推荐

关于介绍“达芬奇技术——数字图像”的教学用书，是为了满足市场对掌握达芬奇技术的技术人员的迫切需求，也为了满足广大学生和技术人员学习和掌握达芬奇技术的迫切愿望而编写的。全书概括地介绍了达芬奇技术所包含的各个方面，以便读者有一个总体的了解，为进一步学习和研究打好基础。

《达芬奇技术》

精彩书评

1、其实你懂的，这类的书就是翻译芯片手册。细读后发现其实还是有些好玩的东西的，总之可以看看其实你懂的，这类的书就是翻译芯片手册。细读后发现其实还是有些好玩的东西的，总之可以看看其实你懂的，这类的书就是翻译芯片手册。细读后发现其实还是有些好玩的东西的，总之可以看看其实你懂的，这类的书就是翻译芯片手册。细读后发现其实还是有些好玩的东西的，总之可以看看其实你懂的，这类的书就是翻译芯片手册。细读后发现其实还是有些好玩的东西的，总之可以看看

《达芬奇技术》

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:www.tushu000.com