

DSP应用丛书

达芬奇技术

——数字图像/视频信号处理新平台

彭启琮 主编

DSP



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

DSP 应用丛书

达芬奇技术

——数字图像/视频信号处理新平台

彭启琮 主编

电子工业出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

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。

本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。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目（CIP）数据

达芬奇技术：数字图像、视频信号处理新平台 / 彭启琮主编. —北京：电子工业出版社，2008.9
（DSP 应用丛书）

ISBN 978-7-121-07194-2

I. 达… II. 彭… III. ①数字图像处理 ②视频信号—信号处理 IV. TN911.73 TN941.1

中国版本图书馆 CIP 数据核字（2008）第 115984 号

责任编辑：万子芬（wzf@phei.com.cn）

印 刷：北京天宇星印刷厂

装 订：三河市皇庄路通装订厂

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本：787×1 092 1/16 印张：20 字数：512 千字

印 次：2008 年 9 月第 1 次印刷

定 价：48.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

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。已有的数字视频的实现，往往与特定的硬件平台和操作系统绑定在一起，开发者只能通过编写和修改代码来进行开发和改变。总的来说，数字视频的实现，是复杂、费时和昂贵的过程。

但随着达芬奇（DaVinci）技术的出现和发展，这一切都在发生改变。

DaVinci 技术是一种内涵丰富的综合体。

DaVinci 处理器将高性能可编程的核与存储器及外设集成在一起，它包括一个可编程的 DSP 处理器，以及面向视频的硬件加速器，为实时的压缩-解压缩算法及其他通信信号处理提供所需的计算功能；它还将一个 RISC 处理器和 DSP 组合在一起，增加对控制界面和用户界面的支持，使之更加易于编程；所集成的视频外设，降低了系统的成本，简化了设计。这种多处理器的硬件结构，构成了开放软件结构的基础，便于灵活、快速地开发各类包含数字视频的产品。

更为重要的是，DaVinci 技术不仅包含了处理器以及传统的开发工具和应用支持，还包含了嵌入式的操作系统，开发多媒体应用所需的基础软件，以及针对视频、图像、音频、语音处理的标准化库程序，使开发者能更快、更容易地实现数字视频。其战略目标是，使开发者不用编写 DSP 代码，就可以将数字视频迅速地带入应用。

开发者所面临的挑战是，数字视频可以是各种不同格式的编码，可以使用各种机制来存储，可以用不同的方式来访问，其内容可以用多种方式操作，等等。

理想的使用数字视频的应用，应该可以支持这些不同的情况，有关的实现细节应该对用户透明。

当用户不希望考虑特定的视频流源时，就必然要求数字视频设备理解不同的视频源和目标之间的区别，因为其机制和特点都各不相同，例如，通过网络传输的视频，需要考虑抖动对视频质量的影响；通过无线传输的视频，就要考虑丢包，根据现有的网络条件，动态地调整码率；通过天线接收的广播视频，必须具有纠错机制，因为被破坏的帧无法重传；从 DVD 读视频，要考虑盘片表面的划伤所导致的画面冻结和死锁。

所有这一切，都可能使情况变得非常复杂，为数字视频的实现带来很大的困难。用户当然希望应对这些挑战，也不希望因为选择的技术不同，带来产品和性能的巨大差异。

DaVinci 技术能够简化数字视频实现的一个关键因素，就是建立在已有的标准上，不需要开发者重新熟悉新的全套 API。DaVinci 技术充分利用嵌入式操作系统(如 Linux 和 Windows CE)的驱动程序。例如，通过高层 Linux 的支持、标准的 API 及产品化的底层驱动器，可以大大降低设计数字视频系统的复杂性，提供完整的实现方法，从而简单而有效地将数字视频集成到需要它的系统里。

由此可见，DaVinci 技术在简化数字视频开发的同时，也提高了学习和熟练使用它的技术门槛。原来熟悉嵌入式操作系统和编写驱动软件的开发者，需要重新学习有关的多媒体标准、算法及其实现；以往熟悉 DSP 技术的开发者，需要更多地了解和掌握嵌入式操作系统、API 及各种驱动程序；所有的开发者，都需要学习和掌握一系列的开发环境、算法标准、库函数，等等。

既然 DaVinci 技术是内涵丰富的综合体，就没有可能在一本小册子里对其进行完整和详细的介绍。本书的目的，仅仅是对以上问题进行概括性的描述，起到一种引导入门的作用，使完全不熟悉 DaVinci 技术的读者，对其内涵有一个初步的了解和认识，为以后继续深入学习和开发打下一定的基础。

本书由彭启琮主编，林静然博士参加了部分内容的编写工作，研究生刘月、雷蕾、靳翼、胡定勇参加了部分资料的翻译工作。

由于 DaVinci 技术相当新，内容又很丰富，我们的了解还非常有限。因此，这本书仅仅希望起到抛砖引玉的作用，肤浅和不当之处，敬请读者和专家指正。

本书的编写和出版，得到了德州仪器公司（中国）大学计划和电子工业出版社的大力支持和帮助，作者对他们表示深切的谢意。

作 者

2008 年 5 月于电子科技大学

目 录

第 1 章 概述	(1)
1.1 DaVinci 技术所面临的主要挑战和创新	(2)
1.1.1 数字视频所面临的挑战	(2)
1.1.2 API 的强大功能	(2)
1.1.3 支持实时操作系统 Linux 及 Windows CE	(3)
1.1.4 实际的视频	(3)
1.1.5 编程的灵活性	(4)
1.2 DaVinci 技术的内涵	(5)
1.2.1 DaVinci 处理器	(5)
1.2.2 DaVinci 软件	(6)
1.2.3 DaVinci 的开发工具	(7)
第 2 章 DaVinci 处理器	(8)
2.1 TMS320C64x+ DSP	(8)
2.1.1 TMS320C64x/TMS320C64x+DSP 的特性	(8)
2.1.2 TMS320C64x/TMS320C64x+DSP 的结构	(10)
2.1.3 TMS320C64x+指令集	(20)
2.1.4 TMS320C64x/TMS320C64x+DSP 的寻址方式	(32)
2.2 TMS320DM644x DMSoC DSP 子系统	(36)
2.2.1 概述	(36)
2.2.2 TMS320C64x+大模块	(37)
2.2.3 存储器映射	(42)
2.2.4 ARM 内部存储器	(42)
2.2.5 片内外设	(42)
2.2.6 器件外设	(43)
2.2.7 ARM-DSP 集成	(43)
2.2.8 DSP 子系统的时钟	(43)
2.2.9 电源管理	(43)
2.2.10 引导和复位	(44)
2.3 TMS320DM644x DMSoC 中的 ARM 子系统	(44)
2.3.1 ARM 子系统概述	(44)
2.3.2 ARM 核	(46)
2.3.3 系统存储器	(50)
2.3.4 器件时钟	(53)
2.3.5 PLL 控制器	(57)
2.3.6 电源与休眠控制器	(61)

2.3.7	电源管理	(68)
2.3.8	中断控制器	(70)
2.3.9	系统控制模块	(74)
2.3.10	复位	(78)
2.3.11	引导方式	(82)
2.3.12	ARM-DSP 集成	(85)
2.4	TMS320DM644x 的外设	(90)
2.4.1	概述	(90)
2.4.2	异步外部存储器接口 (EMIF)	(91)
2.4.3	音频串口 (ASP)	(91)
2.4.4	ATA 控制器	(92)
2.4.5	DDR2 存储器控制器	(92)
2.4.6	增强的直接存储器访问 (EDMA) 控制器	(92)
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	(93)
2.4.8	通用输入/输出 (GPIO)	(93)
2.4.9	内部集成电路 (I2C) 模块	(93)
2.4.10	内部直接存储器访问 (IDMA) 控制器	(94)
2.4.11	DSP 子系统中断控制器 (INTC)	(94)
2.4.12	多媒体卡 (MMC) /安全数字 (SD) 卡控制器	(95)
2.4.13	节电控制器 (PDC)	(95)
2.4.14	脉冲宽度调制器 (PWM)	(95)
2.4.15	串行外设接口 (SPI)	(95)
2.4.16	64 位定时器	(96)
2.4.17	通用异步收发器 (UART)	(96)
2.4.18	通用串行总线 (USB)	(97)
2.4.19	VLYNQ 接口	(97)
2.4.20	视频处理后端 (VPBE)	(98)
2.4.21	视频处理前端 (VPFE)	(99)
2.5	其他的 DaVinci 处理器	(100)
2.5.1	TMS320DM6441 处理器	(100)
2.5.2	TMS320DM643x 处理器	(105)
2.5.3	TMS320DM355 处理器	(110)
2.5.4	TMS320DM64x 处理器	(114)
2.5.5	TMS320DM6467 处理器	(116)
第 3 章	DaVinci 的软件	(122)
3.1	xDAIS 和 xDM	(122)
3.1.1	xDM 和 xDAIS 之间的关系	(123)
3.1.2	xDAIS 界面	(123)
3.1.3	xDAIS 和 xDM 的优越性	(124)
3.1.4	如何建立和验证符合 eXpress DSP 的算法	(124)

3.1.5	建立和验证符合 xDAIS 和 xDM 的算法	(125)
3.1.6	修改自己的算法, 使其符合 xDM 和 xDAIS	(125)
3.1.7	测试 Codec Engine 里的用户算法	(125)
3.1.8	多媒体框架产品 (MFP)	(125)
3.2	Codec Engine	(126)
3.2.1	为什么要使用 Codec Engine	(127)
3.2.2	Codec Engine 在应用程序结构中的位置	(127)
3.2.3	用户的角色	(128)
3.2.4	Codec Engine 的安装和设置	(130)
3.2.5	Codec Engine 的目录结构	(130)
3.2.6	使用范例应用程序	(131)
3.2.7	使用 Codec Engine API	(132)
3.2.8	VISA 类型: 视频、图像、语音和音频	(134)
3.2.9	关于 DSP 存储器	(138)
3.2.10	DSP 的实时处理问题	(140)
3.2.11	软件跟踪	(140)
3.3	Codec Engine 框架和 xDAIS 算法包	(146)
3.3.1	启动	(146)
3.3.2	建立包	(148)
3.3.3	建立一个发布包	(148)
3.3.4	开发一个 xDM Codec	(149)
3.3.5	支持非 xDM 的算法	(152)
3.3.6	建立 Codec Engine 扩展	(154)
3.3.7	设计一个新的应用程序界面	(154)
3.3.8	开发存根和骨架	(155)
3.3.9	打包和配置核算法	(159)
3.3.10	非 xDM 存根和骨架的范例: SCALE	(160)
3.4	DSP/BIOS LINK	(161)
3.4.1	DSP/BIOS LINK 的软件结构	(162)
3.4.2	DSP/BIOS LINK 的主要成分	(162)
3.4.3	DSP/BIOS LINK 源代码的布局	(164)
3.4.4	定制和配置所建立的开发环境	(167)
3.4.5	建立源代码	(174)
3.5	Linux	(176)
3.5.1	概述	(176)
3.5.2	Linux 的主要开发工具	(177)
3.5.3	构建基于 Linux 的嵌入式系统	(180)
3.5.4	MontaVista Linux 的特点	(182)
3.5.5	DaVinci 的 Linux 开发流程	(184)

3.6	基于 DaVinci 的数字媒体软件	(186)
3.6.1	H.264 Baseline Profile (BP) 编码器和解码器	(186)
3.6.2	WMV9 解码器	(186)
3.6.3	WMA9 (Windows Media 9 Series Audio) 解码器	(187)
3.6.4	MPEG-4/H.263 Codec	(187)
3.6.5	MPEG-2 解码器	(188)
3.6.6	解内插库 (Deinterlacing Library)	(189)
3.6.7	JPEG 编码器/解码器	(189)
3.6.8	G.711 编码器/解码器	(190)
3.6.9	MPEG-4 AAC-HE 解码器	(190)
3.6.10	MP3 解码器	(191)
3.7	TMS320C64x+ IMGLIB 图像/视频处理库	(191)
3.7.1	概述	(191)
3.7.2	特性与优点	(192)
3.7.3	安装 IMGLIB	(192)
3.7.4	使用 IMGLIB	(193)
3.7.5	IMGLIB 函数介绍	(194)
第 4 章	开发工具与软件	(197)
4.1	适用于 DaVinci 的 CCS	(198)
4.1.1	v3.2 的更新	(198)
4.1.2	CCS IDE v3.3	(201)
4.2	XDC (eXpress DSP Components)	(207)
4.2.1	XDC 术语	(207)
4.2.2	使用基于 XDC 的软件包	(208)
4.2.3	写 C 代码	(211)
4.2.4	处理配置	(215)
4.2.5	编译和连接	(218)
4.3	数字视频评估模块 (DVEVM) 及其使用	(220)
4.3.1	DVEVM 概述	(221)
4.3.2	DVEVM 的硬件设置	(222)
4.3.3	运行演示软件	(225)
4.3.4	DVEVM 的软件设置	(229)
4.3.5	启动建立开发环境	(234)
4.3.6	为用户的目标板重建 DVEVM 软件	(235)
4.3.7	建立一个新的 Linux 核	(235)
4.3.8	引导新的 Linux 核	(236)
4.4	数字视频软件开发包 (DVSDK)	(236)
4.4.1	概述	(237)
4.4.2	DVSDK 的主要特点	(237)

4.4.3	可视化数据分析器	(239)
4.5	DaVinci 系统级指标测试	(242)
4.5.1	概述	(243)
4.5.2	DVEVM 演示范例软件说明	(243)
4.5.3	所需要的设备和软件	(245)
4.5.4	测量演示范例程序的处理器负载	(246)
4.5.5	关于 DM644x SoC 分析器的结果	(249)
4.5.6	演示范例的存储器使用	(254)
4.5.7	演示范例的功耗测量	(255)
第 5 章	DaVinci 技术应用案例	(257)
5.1	建立一个小的 Linux 核	(257)
5.1.1	所需的环境	(257)
5.1.2	性能选择及建立核的步骤	(258)
5.1.3	建立一个 RAM DISK 文件系统	(261)
5.1.4	支持应用程序	(262)
5.1.5	将信息复制到 NOR Flash	(264)
5.1.6	引导	(266)
5.2	DVEVM/DVSDK 用于编码的演示范例	(268)
5.2.1	概述	(268)
5.2.2	应用程序设计	(268)
5.2.3	配合应用程序	(274)
5.3	DVEVM/DVSDK 用于解码的演示范例	(276)
5.3.1	概述	(276)
5.3.2	应用程序设计	(277)
5.3.3	配合应用程序	(286)
5.4	DVEVM/DVSDK 用于编码和解码的演示范例	(288)
5.4.1	概述	(288)
5.4.2	应用程序设计	(289)
5.4.3	用其他的 Codec 来替换该编码和解码算法	(293)
5.5	运行在 DM6446 上的活动 JPEG 演示范例	(294)
5.5.1	演示范例简介	(295)
5.5.2	演示范例包的内容	(297)
5.5.3	从 DM642 移植到 DM6446	(299)
5.5.4	运行	(301)
5.5.5	重新编译演示范例	(304)
参考文献		(306)

第1章 概述

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。

TI 通过 DaVinci 技术，提供了新的产品开发工具和支持。其第一批处理器 TMS320DM6446（简称 DM6446）和 DM6443 是基于 ARM+DSP 的 SoC，集成了为加速数字视频开发所专门设计的协处理器引擎。它超越了传统的开发工具和应用支持，扩展到包含了开发数字视频所需要的基本软件，通过应用编程接口（API），开发者可以直接调用已经编好的、符合接口标准的 DSP 库程序。

以下是 DaVinci 技术应用的一些例子，随着技术和市场的发展，更多的应用领域正在不断地开发出来。

- 汽车视觉及视频信息；
- 数码相机；
- 数字媒体适配器；
- 数码相框；
- 数码摄像机；
- IP 网络摄像机；
- IP 机顶盒；
- 机器视觉；
- 媒体网关；
- 医学图像处理；
- 多点会议单元；
- 便携式媒体播放器；
- 机器人；
- 电视广播代码转换；
- 视频会议；

- 可视电话；
- 视频监视 DVR/DVS。

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

- 开发者必须首先确定编码格式，例如，MPEG-2、MPEG-4、WMV、DivX、H.264、H.263，等等。
- 数字视频可以用多种方式存储，例如，Flash 等非挥发性的存储器、本地硬盘、远程服务器、DVD 或 VCD 等固定介质、摄像机或媒体播放器设备，等等。
- 数字视频可以用多种方式来访问，例如，固定文件、广播媒体、实时网络流、非实时流、有线网络和无线网络等。
- 数字视频的内容可以用多种方式来操作，例如，改变分辨率、变换编码格式、存储在不同的设备以便重放等。

由于上述原因，开发者面临很多需要解决的问题。例如，通过网络传输视频，就要克服抖动对视频质量的影响；视频的无线传输，可能丢包，因而必须根据网络的条件，动态地调整数据速率；无线接收的广播视频，必须具有纠错机制，因为不可能重传；从 DVD 调用视频时，要考虑因为盘片的损坏导致的图像冻结和死循环，等等。

这些看似细节的问题，所涉及的问题则可能很复杂，解决起来，也并非轻而易举。用户所期待的，是这些都不要成为问题，也不要考虑产品所采用技术的不同。开发者也不希望陷入耗时费力的技术细节之中。

1.1.2 API 的强大功能

应用编程接口（Application Programming Interfaces, API）是 DaVinci 技术所集成的主要功能之一，使开发者可以集中精力于应用的开发，不用将时间花费在一些开发的细节上。例如，从应用的角度来看，开发者所关心的，只有视频流的访问，通过一个 API，只需要简单地调用函数 `GetVideo()` 即可完成。调用下一帧视频的所有复杂细节，都已经包含在这个函数里了。API 屏蔽了应用层面的复杂性，可以访问各种不同来源、不同方式所实现的视频流，而且不用对应用程序代码进行大的改变。

这正是提出 DaVinci 技术及其集成成分的关键价值之一。通过提供必要的软件和 API 结构，DaVinci 的 API 使开发者通过简单地调用函数，实现数字视频，不再需要关注编码/解码（CODEC）实现的技术细节。

与在特定的平台上，针对特定的应用，花费数月的开发时间，来设计算法并对其优化相比，开发者现在可以使用做好了的驱动器和 CODEC，将其注意点和开发资源集中于应用层的软件，这才是其附加价值之所在。

API 还可以保证 OEM 对其代码的已有投资。

当 OEM 成功地将产品投放市场，并获得一定的市场份额后，往往希望扩展产品线，提供新的应用。理想的情况是，将他们已有的应用程序代码用到新的应用中。

API 可以通过提高代码的可移植性,来达到跨产品的平滑过渡。将实际的硬件实现细节从具体的应用中分离出来,就可以很容易地将已有的应用程序代码与新的平台和操作系统接口,不用重写所有的应用程序代码。不仅如此,开发者还比较容易跟上多媒体格式的改变。DaVinci 的驱动器和 CODEC 是基于最新的标准,使开发者得以保证,其设计可以支持最新的流行多媒体格式。

1.1.3 支持实时操作系统 Linux 及 Windows CE

目前, DaVinci 支持基于 MontaVista 专业版的 Linux 2.6.10 和 Windows CE。

从内联的角度看,可以使用标准的 API 调用来进行通信,例如,文件的开、关、读、写等。用户可以简单而直接地访问各种外设和视频源,也可以通过文件系统接口访问 NAND-、CF-、MMC-、ATA 等存储器里的数据。可以通过 API 配置基于硬件的视频引擎,以便与特定应用的需要匹配。例如,设置参数来调整图像的大小、每像素的位数、刷新速率、调色板,以及管理帧缓冲器和处理器内置的屏幕显示能力。DaVinci 的声音驱动器可以记录与重放,并提供多种采样率(8 kHz、16 kHz、22 kHz、44.1 kHz、48 kHz 以及 96 kHz)。

所有这些复杂细节的实现,都是由低层的驱动器来处理,开发者只需要通过公共的 Linux 和 Windows CE 的 API 来调用这些驱动器。

基于 DaVinci 技术的 TMS320DM644x 处理器的计算资源包括:优化的 DSP 公用程序、基于硬件的加速引擎、强化的 DMA 外设(EDMA)等。

1.1.4 实际的视频

以下是几个例子,说明如何用 DaVinci 技术,直接而简单地实现数字视频。

考虑一个家用的媒体网关,用户可以从各种信号源,连接到以太网的 PC、USB 2.0 的数码相机、通过 ATA 连接的硬盘上的文件等来重放视频。

因为 DaVinci 技术基于标准的 Linux 驱动器,视频流一旦打开,服务于视频数据的应用程序代码就像文件传输一样简单。

例 1.1-1 用 HDD 显示 H.264 视频的伪码。

```
InitPeripherals()  
InitEthernet();  
InitUSB();  
InitATA();  
InitDisplay();  
InitH264Decoder();  
OpenCodec()  
OpenH264Channel();  
StartDecode(ATA.Address, H264.Channel, Display.Address);
```

这就是从任意源访问视频数据的全部代码。重要的是,同样的代码可以用于各种情况,使应用程序代码灵活而通用。在决定视频的目标时,这种灵活性同样重要。考虑一个机顶盒,连接到家庭影院的显示器或 PC 显示器,或下载到个人的媒体播放器。每个显示器都具有不同的分辨率,都可能不支持所收到的视频原来的格式。

同样，当视频流一旦打开和配置，其应用程序代码是直截了当的。

例 1.1-2 改变显示大小的伪码。

```
InitPeripherals()  
// Display driver is passed arguments to move away from default values  
InitDisplay(QCIF | CIF | D1 | 720p | 1080i);
```

因为 DaVinci 技术使用了业界所接受的 API，从而可以简单地实现先进的视频特性。例如，时移特性，用户要暂停实时的视频源，就要求将实时的视频存储到盘里，以免丢失部分节目。当用户重放时，设备要存储新进来的视频（视频的即时编码/解码）。

例 1.1-3 暂停和重放视频的伪码。

```
[Action] Pause Button is pressed  
OpenCodec()  
OpenH264Channel;  
//Start compressing and storing Incoming data from Ethernet to HDD  
StartEncode(Ethernet.Address, H264.EncodeChannel, ATA.Address);  
...  
[Action] Resume Button is pressed  
StartDecode(ATA.Address, H264.DecodeChannel,  
Display.Address);
```

1.1.5 编程的灵活性

DaVinci 的灵活性，来自其可编程的结构。例如，TMS320DM644x 是双核结构，ARM+DSP。开发者使用开源的 ARM 工具来编写应用程序代码，并不需要对 DSP 编程，因为可以通过 API 访问已经编好的可执行 DSP 代码。

可编程的优点是显而易见的。ASIC 可以提供完全的实现，但只能用在特定的产品里。而且，如果该产品的功能和性能发生改变，就得费时费钱重新设计 ASIC。FPGA 是可以编程的，但难以提供完整的实现，因为开发者必须在完全不同的开发环境里，付出极大的努力来设计，这使开发的效率成为很大的问题。

DaVinci 技术兼顾了两者的优点，因为其结构是可编程的 DSP 和 ARM 核，开发者可以通过代码和驱动器的更新，使系统优化，并能跟上潮流。TI 及其合作伙伴提供必要的 API 代码和驱动器，使数字视频得以完全实现。

依靠 DaVinci 处理器的可编程性，围绕单种应用所开发的 IP 可以重复使用。而 ASIC 对于每种 CODEC 的系统支持，要求不同的硬件。

使用 TMS320DM644x 的硬件资源，可以动态地重新配置，支持不同的编码/解码器，从而最大限度地保护对硬件系统的投资。

1.2 DaVinci 技术的内涵

DaVinci 是一项综合性的技术，包括处理器、软件、开发工具和第三方合作伙伴的技术。

1.2.1 DaVinci 处理器

DaVinci 处理器是建立在 TI 最新的 TMS320C64x+DSP 核基础上的 SoC，包含 DSP 核、ARM 核、加速器和外设，从而保证各种数字视频终端设备对价格、性能及将来可能发展的苛刻要求。首批处理器包括：

- **TMS320DM6446 DaVinci 处理器**

用于视频的解码和编码，其编码能力是解码能力的 2~3 倍，以适应各种格式的数字视频的采集和处理。

- **TMS320DM6443 DaVinci 处理器**

专门用于便携式多媒体设备的视频解码，片上包括模拟和数字的视频输出及屏幕显示引擎。

图 1.2-1 是 DaVinci 处理器的框图。

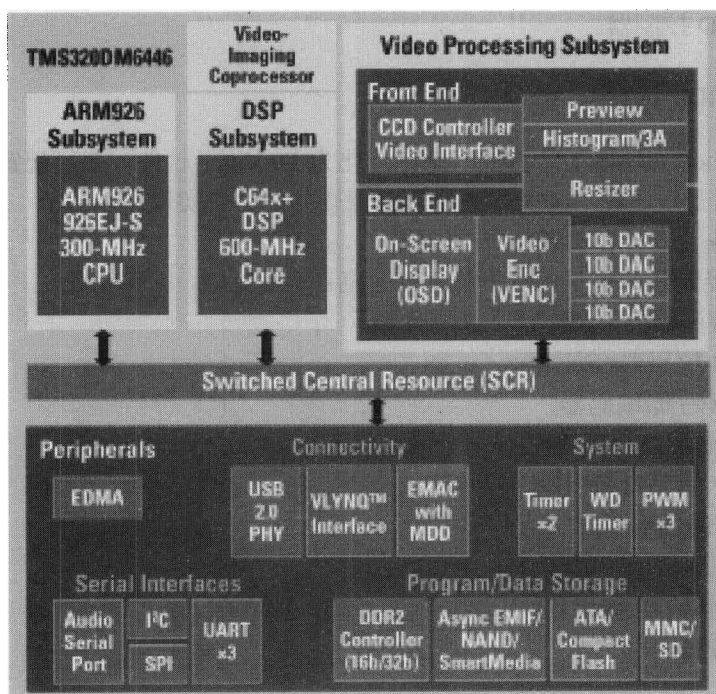


图 1.2-1 DaVinci 处理器的框图

DaVinci 处理器的外设和加速器包括：

- **视频加速器**

- 视频 I/O 处理子系统；
- 视频和图像加速器。

- **音频**
 - 用于声音编码/解码器界面和通信的串口（ASP）。
- **片外存储器接口**
 - 双倍数据率（DDR2）；
 - 片上 NAND Flash 控制器；
 - 用于 NAND/NOR Flash 的异步 EMIF。
- **视频显示**
 - 集成了对常用的解决方案和界面的支持。
- **连接**
 - USB 2.0 高速主机和客户端功能；
 - 全速 10Mb/s /100Mb/s Ethernet MAC；
 - I²C 总线接口；
 - 专门的 FPGA 接口（VLYNQ）。
- **数据存储接口**
 - ATA（硬驱动）；
 - 紧凑的 Flash 控制器；
 - MMC/SD 控制器。

1.2.2 DaVinci 软件

运行在 DaVinci 处理器上的软件，都归一化为 DaVinci 软件，在可配置的框架里，通过规范的 API，运行于广泛使用的操作系统，以便软件的尽快实现。

- 视频、图像、音频和话音编码/解码器
- 目前，TI 已经制作和提供的标准化软件包括：
- H.264 解码器/编码器；
 - MPEG-4 解码器/编码器；
 - H.263 解码器/编码器；
 - WMV9 解码器/编码器；
 - MPEG-2 解码器/编码器；
 - JPEG 解码器/编码器；
 - AAC+解码器/编码器；
 - WMA9 解码器/编码器；
 - MP3 解码器；
 - G.711；
 - G.728；
 - G.723.1；
 - G.729ab。

- 操作系统

DaVinci 技术支持 Linux、Windows CE 等操作系统。

- 应用编程接口 (API)

这些 API 有利于 DSP 和 ARM 的软件开发人员。当 DSP 平台保持向 DSP 专家开放时，ARM 开发者可以对 API 编程，并保证可以和下一代数字视频处理所要求的编码/解码器组合在一起，进行测试。

- 框架

DaVinci 的软件框架，是为了保证所有基于 DSP 的 SoC 硬件和软件可以无缝地组合起来，将系统开发者从多处理器 SoC 内的工作解脱出来，大大地减少研究与开发、系统结构、实现与测试等所花费的时间。

1.2.3 DaVinci 的开发工具

DaVinci 技术提供完整的开发工具，包括低成本的初始套件、完整的开发工具包，以及面向应用的参考设计等。

ARM/DSP 集成开发环境 (IDE)、操作系统，以及 DSP 工具，这些使开发者在熟悉的环境里工作。