

基于达芬奇技术的 H.264 视频编码器的实现 · 实用设计 ·

成 嘉, 张文雄, 李善劲

(华南理工大学 电子与信息学院, 广州 510640)

【摘要】介绍了 H.264 视频编码标准和达芬奇技术内部的编解码引擎 (Codec Engine) 框架, 描述了如何利用 Codec Engine 来实现 H.264 编码器的方法。

【关键词】H.264 标准; 达芬奇技术; 视频编码器; 编解码引擎

【中图分类号】TN919.81

【文献标识码】A

Realization of H.264 Video Encoder Based on DaVinci Technology

CHENG Jia, ZHANG Wen-xiong, LI Shan-jin

(School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510640, China)

【Abstract】This paper gives an introduction of H.264 video coding standard and the framework of Codec Engine in TI DaVinci technology. Then it presents the method of how to achieve an H.264 video encoder with Codec Engine.

【Key words】H.264; DaVinci technology; video encoder; code engine

1 H.264 压缩编码标准

相对于 MPEG-4, H.264 标准着重于解决高效率视频压缩和高可靠性传输。H.264 标准可分为 3 个档次, 分别是: 1) 基本档次: 利用 I 片和 P 片(注意没有 B 片), 支持帧内和帧间编码, 支持利用基于上下文自适应变长编码(CAVLC)进行的熵编码。2) 主要档次: 支持隔行视频, 采用 B 片的帧间编码和采用加权预测的帧内编码; 支持利用基于上下文的自适应的算术编码(CABAC)。3) 扩展档次: 支持码流之间的有效切换(SP 和 SI 片)、改进误码性能(数据分割), 但不支持 CABAC。H.264 标准具有多个优点: 1) 视频编译码结构采用分层设计, 视频编码层(VCL)负责高效率视频编码压缩, 网络适配层(NAL)负责网络的适配; 2) 多参考帧模式预测, 以改善运动目标估值性能, 提高译码器的误码恢复能力; 3) 高精度运动估计, 在 H.264 中采用 1/4 像素的运动估计; 4) 统一的可变长度编码(UVLC)码表, 同时, 进行视频编码, 使量化后的 DCT 变换系数使用 CAVLC; 5) 编码 INTRA 图像可用帧内预测有效提高预测质量; 6) 宏块(MB)更精细与多样化分解, 大大提高了运动目标估值精度^[1]。

2 达芬奇技术 Codec Engine 框架研究

2005 年底 TI 公司推出了达芬奇技术, 专门用于开发多媒体应用系统, 使开发人员只要对 Codec Engine 提供的应用程序编程接口(API)加以应用就可控制和使用 H.264 编码器, 根据自身需要产生各种档次的视频图像。达芬奇技术的基础是集成了可编程 DSP 内核以及 ARM 处理器内核的 SoC。达芬奇技术提供了一整套包括数字

视频评估模块(DVEVM)在内的软硬件系统、开发工具等部件, 可高效地开发数字多媒体应用系统^[2]。

2.1 Codec Engine 定义

达芬奇技术体系中引入了 Codec Engine, 并创建了一整套的应用开发平台。Codec Engine 为通用处理器(GPP)上的开发者提供更为简单的开发环境。Codec Engine 是一系列用于表示和运行数字多媒体标准化 DSP 算法接口(XDAIS)及算法的 API。XDAIS 定义了一整套的多媒体算法编程接口, 可单独在 GPP 或 DSP 上运行, 也可在 DSP 上运行, 而 GPP 通过 Codec Engine 对其实行控制。对于所有支持的运算器结构、运行方式及操作系统, Codec Engine 都有相同的 API。Codec Engine 定义了 4 类编解码器算法接口标准, 分别是视频、图像、语音、音频, 简称 VISA。

2.2 Codec Engine 框架分析

Codec Engine 在解决达芬奇双处理器架构问题时首先引入了远程过程控制(RPC)的概念。RPC 是指在另一个处理器上远程执行一个命令或过程。RPC 有客户端和服务端, 客户端通过某种通信协议并最终通过物理链路向服务端发送一个命令, 服务端执行命令并通过相应的通信方式将结果返回给客户端。达芬奇 DM6446 芯片将 ARM 作为客户端、DSP 作为服务端, 两者之间的物理链路就是两个处理器之间共享的 DDR2 存储器, 而将物理链路上的通信协议称为 DSPLink。图 1 给出了这种框架的结构示意图。图中, 共享 DDR2 存储器是 ARM 与 DSP 之间共享的内存, 通过它进行物理数据的交换。DSPLink 是达芬奇架构双处理器间通信的基础软件, 为开发人员

提供了一个通用的 API, 用于描述 ARM 与 DSP 之间通信的物理链路的特征。引擎功能层用于管理算法对象的实例, 例如创建一个对象的具体过程等。VISA 层是面向引擎功能层的一个接口, 用于定义创建、删除和使用一个特定符合 XDM(XDAIS 的扩展, 专用于多媒体应用)标准的算法的进程。由于 Codec Engine 的主要任务是从 ARM 发起的 VISA 命令控制远程的 XDM 算法起到一个管道的作用, 因此 VISA 层实际上就是代表了 XDM 接口层。

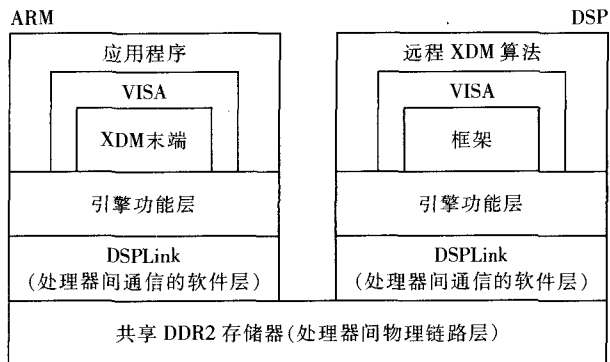


图 1 Codec Engine 框架示意图

3 利用 Codec Engine 框架设计 H.264 编码器

3.1 Codec Engine 框架中的应用开发角色

采用 Codec Engine 技术开发多媒体应用系统时, 开放的标准接口使得开发人员从不同角度分工解决不同的问题, 分为 4 种不同的角色^[3]:

1) 算法开发。所谓算法开发, 就是开发 XDAIS 算法并封装算法, 以确保算法可被 Codec Engine 使用 and 配置。算法开发工程师可利用前人在 TI DSP 平台上开发的运算算法开发 Codec Engine。如果算法与 XDM 标准兼容, 则可不经修改而直接被 Codec Engine 的 VISA API 远程执行; 如不兼容, 算法开发工程师需为其提供 Codec Engine 的框架和终端接口。

2) 服务集成。在双核环境中, 如果 Codec Engine 是在远程的处理器(如 DM6446 中的 DSP)上执行, 而操作系统是由另外一个处理器(如 DM6446 中的 ARM)通过 VISA API 控制 Codec Engine, 必须先要在远程的 DSP 生成一个 Codec Server。Codec Server 是后缀为 .x64P 的文件, 是一个集成了编解码器、框架元件和系统代码的二进制库。在达芬奇 DM6446 平台中, Codec Server 包含了一个为相应客户端(ARM)生成编解码器、提供系统运行信息(包括指令和内存使用情况)的 DSP/BIOS 任务线程; 客户端(ARM)的应用程序通过使用 VISA API 与在远程

DSP 上的编解码器联系。

3) 引擎集成。引擎集成工程师将通过 RTSC 配置工具做大量的关于引擎配置的工作, 包括引擎的名字、每个引擎中包含什么编解码器、编解码器是本地执行还是远程执行等, 如果这些编解码器在远程执行的, 则还需配置相应的 Codec Server。一般将所有的配置写到一个 .cfg 文件里面, 这个配置文件包含的内容有: Codec Engine 运行的系统环境, 包括 xDC 通用模块和客户端操作系统类型等; 指明 Codec Engine 需要用到的编解码器模块; 配置 Codec Engine, 指明 Engine 包含的内容。

4) 引擎应用。引擎应用工程师利用 Codec Engine 的应用编程接口(Engine APIs)创建/删除配置好的引擎实例, 进而创建/删除控制编解码器。由于 Codec Engine 不参与任何的 I/O 操作, 因此应用工程师需要复杂管理 I/O, 包括文件操作(文件的打开、读写与存储等)和外设操作(外设的打开、关闭与控制等)。应用工程师负责编写应用代码并将合适的内容加到最终的可执行文件中去。应用工程师可使用以下资源: 从算法开发工程师得到的大量的编解码器软件包; 从服务集成工程师得到一个可以在 DSP 上运行的 Codec Server 二进制文件, 一般为 .x64P 文件; 从引擎集成工程师得到一个引擎配置文件, 一般为 .cfg 文件。应用工程师编写应用代码, 并结合这些资源再链接、编译生成最终的可执行代码。

3.2 Codec Engine APIs 的分析

开发者利用 Codec Engine 提供的 API 控制和使用 H.264 编码器^[4]。Codec Engine 具有两类 APIs, 一类为核心引擎应用编程接口(Core Engine APIs), 用于实现 Codec Engine 的打开、关闭以及对 CPU 和内存使用情况的统计等, 表 1 给出了 Core Engine 模块的相关信息。Codec Engine 通过一系列的软件包来发布, 其软件包的名字刚好对应了软件包的目录, 例如, ti.sdo.ce 软件包对应一个以 /ti/sdo/ce 作为结尾的目录。

表 1 Codec Engine 核心引擎应用模块

描述	模块缩写	软件包名称	头文件
初始化函数	CERuntime_	ti.sdo.ce	CERuntim.h
编解码器引擎运行控制	Engine_	ti.sdo.ce	Engine.h
操作系统内存抽象层	Memory_	ti.sdo.ce.osal	Memory.h

另一类 APIs 为具体的多媒体算法应用编程接口(VISA APIs), 用于实现对视频、图像、语音、音频等多媒体算法的具体操作, 本文只使用视频编码, 表 2 给出了相应的视频模块信息。

表 2 Codec Engine 视频应用模块

描述	模块缩写	软件包名称	头文件
视频编码接口	VIDENCx_	ti.sdo.ce.video	videnc.h
视频解码接口	VIDDECx_	ti.sdo.ce.video	videc.h

其中定义了4个最主要的功能函数接口,分别是:VIDENC_create(),用于创建编码器实例;VIDENC_process(),用于处理编码器实例以完成编码工作;VIDENC_control(),用于传递动态参数以控制编码器实例;VIDENC_delete(),用于删除编码器实例,释放创建编码器实例时占用的系统资源。另外还定义了必要的数据结构,其中最主要的数据结构是VIDENC_Params和VIDENC_DynamicParams。VIDENC_Params是视频编码器创建时的原始参数,此结构包括了编码形式、输出码率控制方式、输出数据的格式、支持的最大图像高度、最大图像宽度、最大帧率、最大比特率等。VIDENC_DynamicParams是编码器运行时的动态参数,此结构包括了输入图像的高度、宽度、帧率、输出的图像帧率、编码器输出的码率、相邻两个I帧之间的其他帧(P或B)数等。上述的两种数据结构在所有视频编码器中是通用的,对于各种具体的编码器,Codec Engine又提供了另外的数据结构用于详细设置编码器参数,且这些数据结构是通用数据结构的扩展。H.264编码器专门提供了针对H.264编码器参数设置的数据结构:IH264VENC_Params是H.264编码器创建时的原始参数,不仅包括了通用数据结构VIDENC_Params,还包括编码器的档次设置、编码器的级设置、码率控制表使用的算法和搜索范围设置;IH264VENC_DynamicParams是H.264编码器运行时的动态参数,不仅包括了通用数据结构VIDENC_DynamicParams,还包括I帧的初始量化参数设置,P帧的初始量化参数设置,环路滤波器的开闭,1/4像素内插法的开闭。

3.3 H.264 编码器的设计流程研究

本文H.264编码器的设计流程如下:

1) 开发基于C64系列DSP的符合XDM标准的编解码器算法,或封装不兼容XDM的算法,使其与XDM标准兼容。本文采用的DSP算法是TI发布H.264 Baseline Profile编码器的测试评估样板。

2) 编写相关配置文件,编译生成的Codec Server为一个.x64P文件。再通过RTSC配置工具做关于引擎配置的工作,生成.cfg配置文件。

3) 编写基于ARM-Linux与Codec Engine的应用程序,用于调用Codec Engine的APIs进行视频的编码操作。图2是H.264编码器软件设计流程,编码器的输入数据来自视频处理前端的视频采样数据。

4 小结

笔者采用TI发布H.264 Baseline Profile编码器的测试评估样板,并通过TI的第三方合作伙伴闻亭公司的研究成果得到一个已编译的、可直接在DSP端运行.x64P的文件。接着完成引擎集成的任务,利用Codec

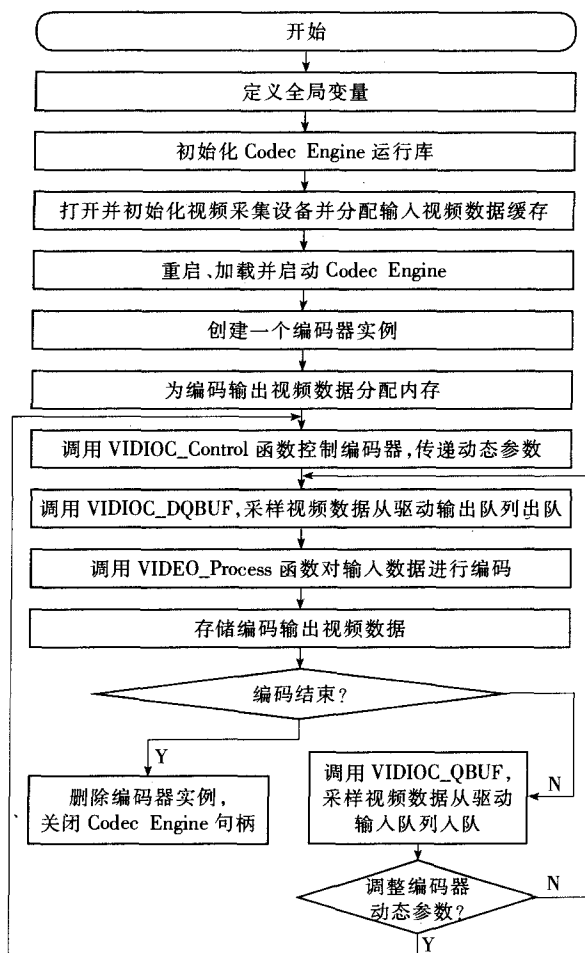


图2 H.264编码器软件设计流程图

Engine APIs编程实现一个具体的视频编码器实例。这种开发模式可大大降低开发成本,加快产品上市时间,提高市场竞争力。

参考文献

- [1] 毕厚杰. 新一代视频压缩编码标准——H.264/AVC[M]. 北京:人民邮电出版社, 2005.
- [2] IAIN E G R. H.264 and MPEG-4 Video compression Video coding for Next Generation Multimedia[M]. [S.l.]:Wiley Press, 2003.
- [3] Texas Instruments Incorporated. TMS320DM6446 Digital Media System-on-Chip (Rev. E) [S]. 2007.
- [4] Texas Instruments Incorporated. Codec Engine Application Developer User's Guide [S]. 2006.

作者简介:

成嘉(1982-),硕士生,研究方向为网络视频编码与传输;

张文雄(1982-),硕士生,研究方向为视频编码算法;

李善勤(1948-),副教授,硕导,研究方向为程控交换技术、电视技术、嵌入式系统设计与应用、通信新技术等。

责任编辑:任健男

收稿日期:2007-09-25