

i.MX51 平台视频硬件解码的研究与应用 *

周时伟^{1,2}, 谢维波^{1,2}, 郭利全^{1,2}

(1. 华侨大学 计算机科学与技术学院, 福建 厦门 361021)

2. 厦门软件园 华侨大学嵌入式技术开放实验室, 福建 厦门 361008)

摘要: 提出了一种视频在 i.MX51 平台的硬件解码方案, 详细描述了使用 VPU 进行视频硬件解码的逻辑处理过程及解码流程。使用本方案将 MPEG4 格式的视频文件在 Android 平台进行解码并获得成功。对于使用 i.MX51 进行多媒体开发具有重要的指导意义。

关键词: i.MX51 开发板; 硬件解码; VPU; MPEG4

中图分类号: TP311

文献标识码: A

文章编号: 1674-7720(2013)10-0013-04

Research and application of video hardware decoding on i.MX51 platform

Zhou Shiwei^{1,2}, Xie Weibo^{1,2}, Guo Liqian^{1,2}

(1. College of Computer Science & Technology, Huaqiao University, Xiamen 361021, China;

2. Huaqiao University the Open Laboratory of Embedded Technology, Xiamen Software Park, Xiamen 361008, China)

Abstract: This paper presents a video hardware decoding scheme on the i.MX51 platform, Giving out the logic of VPU and decoding process. The MPEG-4 format video file is decoded successfully in the Android platform by using this program. This study has important guiding significance for the i.MX51 to conduct multimedia development.

Key words: i.MX51; hardware decoding; VPU; MPEG4

随着多媒体技术的迅速发展, 数字视频的应用越来越广泛。人们对高清视频的狂热追逐给视频解码带来了巨大的压力^[1]。传统的软件解码方法已经很难满足视觉要求。高清视频的分辨率远远高于一般格式视频, 其码率也很高, 再加上视频编码标准的压缩率高, 使得解码运算量很大。因此, 常规地直接用 CPU 解码会极大地消耗 CPU 的运算能力, 于是转向硬件解码。硬件解码是通过显卡的视频加速功能对高清视频进行解码, 能够将 CPU 从繁重的视频解码运算中释放出来, 使计算机具备流畅播放高清视频的能力。本文通过研究主流 Android 开发板 i.MX51 上的 VPU, 给出了使用 VPU 实现视频硬件解码的主要流程和方法。

1 i.MX51 EVK 平台硬件视频解码方案

1.1 i.MX51 处理器的特性

i.MX51 是飞思卡尔公司自主研发的 ARM 体系结构的中央处理器单元。i.MX51 处理器采用了先进、高效率的 ARM Cortex A8 内核, 处理器运行在 800 MHz 的高速下^[2]。相比于 ARM11 产品, 其性能被大大提高。它支持

最高 200 MHz 的 DDR2 和移动 DDR DRAM 时钟速率, 拥有 32 KB 的指令缓存和数据缓存以及 256 KB 的二级缓存^[3]; 处理器内部集成了 DDR/DDR2 内存控制器; 同时集成了矢量运算、浮点运算以及 ARM NEON SIMD 媒体和信号处理器, 为多媒体信息娱乐终端提供了强大的处理核心支持。

1.2 i.MX51 VPU

i.MX51 VPU(视频处理单元)是一个支持多标准编解码操作的高性能的视频编解码引擎。VPU 编解码器支持 MPEG-1/2、MJPEG、H.264、MPEG-4 的编解码。VPU 在 i.MX51 上支持的解码高达 HD 水平, 编码为 SD 水平。VPU 通过 32 bit 的 APB 总线和 64 bit 的 AXI 总线与系统联络, APB 总线用于系统控制, AXI 总线用于数据的传输。同进, 利用片上存储器实现高性能。

VPU 中的视频硬件模块被最佳化设计以用来在不同的视频标准中共享使用, 并提供强大的性能和超低的功耗, 如图 1 所示。

VPU 提供一套主控接口寄存器, 通过这些寄存器主处理器可以简单、高效地控制 VPU。VPU 拥有一个 16 bit 的 DSP 内核——位处理器, 通过该位处理器可以控制内
《微型机与应用》2013 年 第 32 卷 第 10 期

* 基金项目: 福建省自然科学基金项目(2010J01334)

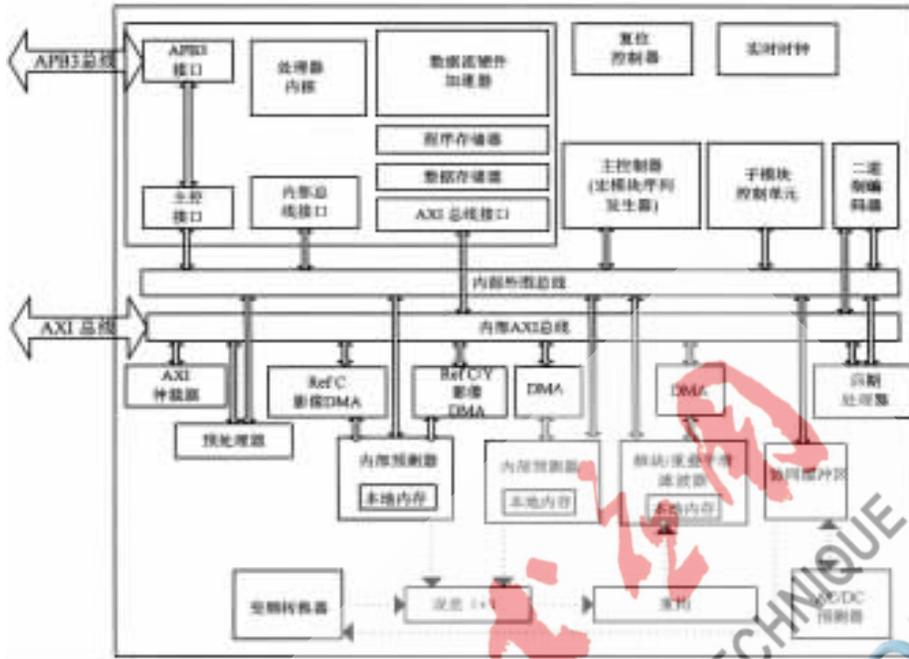


图1 VPU内部框图

部硬件模块并实现视频编解码操作。

1.3 视频解码方案

i.MX51 开发板具有支持多种操作系统的特性,例如 Android 等实时 OS。为了能够获得较好的推广应用,采用比较流行的 Android 系统。

在 Android 中创建自己的解码函数库,通过 JNI 穿越系统的应用框架层直接访问库函数,才能实现 Android 系统下对硬件的直接调用和控制。首先,使用 C 语言对 VPU 的 API 组织调用,实现视频的解码;然后,生成相应的动态库,并提供给 JNI 相应的应用接口;最后,在 Java 类中使用 C 实现视频解码的本地方法。

2 VPU 解码的内部逻辑

2.1 VPU 的初始化

VPU 中位处理器用于处理比特流数据,位处理器是一个高度优化的,它还可控制 VPU 与主处理器之间的通信^[4]。位处理器固件被分为两部分来进行视频的解码:一部分是引导代码,这些代码是主处理器通过 IP 总线下载到位处理器内部存储器的,启动代码大小为 1 KB,这些启动代码被写在外存储器一块区域中,固件区域的基地址由 VPU 的 API 写入;另一部分是进行解码处理的固件包,在开始解码前,固件先被写进外部存储器启动代码之后的一块连续的区域中,运行过程中,位处理器自行下载固件,根据不同的解码标准下载到相应的内部存储器中,它是通过 AXI 总线进行加载的,VPU 初始化过程如图 2 所示。

2.2 VPU 与应用程序的交互

图 3 给出了位处理器和 VPU 视频处理核心模块,并且指明了 VPU 如何与应用软体进行交互。从根本上说,



图2 VPU初始化过程

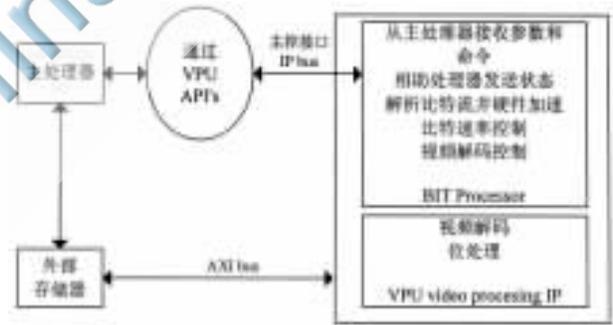


图3 VPU与应用软体交互过程

在框架水准之上,主处理器是通过 API 与主处理器进行通信的。

2.3 VPU 的硬件抽象层

图 4 为 VPU 的硬件抽象层的详细内容。VPU 的解码器的寄存器头文件 vpu_reg.h 中定义了内存映射地址;头文件 vpu_io.h 中定义了 I/O 操作的各种定义,声明了 I/O 操作的各种函数,并在 vpu_io.c 中实现了其头文件中的各个函数;vpu_lib.c 是 VPU 的库函数,实现了 VPU 编解码的所有操作;vpu_lib.h 中定义了 VPU 函数库中的数据类型、数据结构,并声明了各个函数;vpu_util.c 中是与 VPU 中位处理器相关的各种操作。

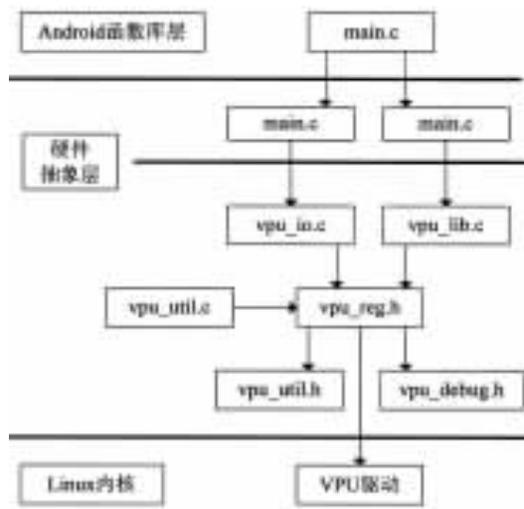


图4 VPU 硬件抽象

3 VPU 硬件解码流程分析

3.1 VPU 进行视频解码的流程

通过 VPU 解码的基本流程包括以下步骤：

- (1) 调用函数 Init() 来初始化 VPU；
- (2) 使用函数 DecOpen() 打开一个解码器实例；
- (3) 为了能够提供合适数量的字节流，使用 DecGet-
BitstreamBuffer() 获取字节流缓冲地址；
- (4) 在改变解码器的输入流之后，使用 DecUpdateBit-
streamBuffer() 通知传输到字节流缓冲区的字节数量；
- (5) 在进行一个图片的解码操作之前，使用 DecGe-
tInitialInfo() 来为解码操作获取重要的参数，如图片大小、
帧缓冲等；
- (6) 使用返回的帧缓冲大小配置帧缓冲区合适的大
小，并且使用函数 DecRegisterFrameBuffer() 把这个数据传
递给 i.MX51 VPU。
- (7) 使用 DecStartOneFrame() 开始图片解码器操作，一
张一张地进行图片解码；
- (8) 等待完成图片解码器操作的中断事件；
- (9) 使用 DecGetOutputInfo() 检查解码器操作的结果；
- (10) 播放第 n 帧后，使用 DecClrDispFlag() 清除播放
缓冲标志；
- (11) 如果还有更多的比特需要解码，返回到步骤(7)
继续执行，否则就执行下一步；
- (12) 使用 DecClose() 关闭实例，终止操作；
- (13) 调用 UnInit() 释放系统资源。

3.2 数据处理与控制

VPU 拥有一个专用的路径进行主处理器与 VPU 之间的数据或信息的交流，即共享存储器。共享存储器通过 ABMA 主总线被访问，使用这个存储器进行比特流和数据帧的交换。通过使用 VPU 寄存器中的主控寄存器来实现主处理器与 VPU 之间专一的信息交流路径。所有在主处理器与 VPU 之间的命令和响应都是通过这

些寄存器来交换的。数据转换与命令及响应相关的信息也是通过主控寄存器来交换的。从主处理器可以访问 VPU 主控接口的所有寄存器。一些主控寄存器用于交换实际的命令和响应，另一些寄存器则用来向主处理器提供 VPU 内部状态信息。

主应用程序通过 API 发送命令和相应的参数给 VPU 来控制 VPU。从 VPU 接收到一个中断之后，发送要求的操作已经完成的信息。如图 5 所示为应用程序数据处理的过程。

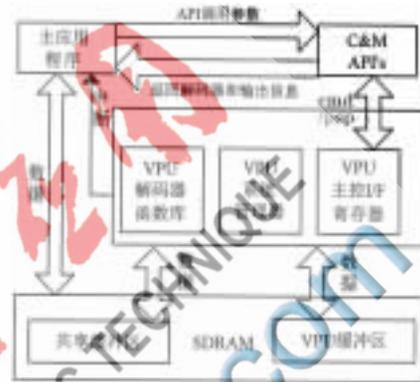


图5 数据处理与控制模型

每个 API 的定义包括请求命令和输入输出数据结构。从 API 给出的命令经常被写进一个专用的 I/O 寄存器，但是输入和输出数据结构是通过一套包括输入参数和输出结果的 I/O 命令寄存器传输的。

所有的像素数据或者数据流处理被主处理器执行或者通过在 SDRAM 中的共享内存被 VPU 执行。为了保证主处理器与 VPU 之间的安全性，所需信息被存放在主控寄存器中。这些事务一般都是单向的，VPU 或主处理器写数据，其他设备在一个单数据缓冲区读取数据。

4 视频硬件解码在 Android 下的应用

4.1 MPEG-4 在 i.MX51 上的硬件解码实现

本文以 MPEG-4 视频流为例实现其硬件解码。图 6 为 MPEG4 的解码流程。

(1) 初始化 VPU

用 BIT Code Download 命令下载 BITProcessor 固件到存储器；设置初始化参数用于对 BIT Processor 进行一般性设置，设置工作缓冲区基址、BIT Code 存储器地址、比特流缓冲区控制等；命令 BIT Run Start 运行 BIT 处理器初始化 VPU。

(2) 创建并初始化一个 MPEG-4 解码进程

设置 SEQ_INIT 参数，这一过程通过调用 MPEG4_initial() 进行解码初始化，包括打开文件、设置硬件参数(如 codec)、解码参数初始化(如波特率设置)、申请地址空间(如配置基地址)；运行 SEQ_INIT 命令，开始一个 MPEG-4 解码进程。如果出现 Wait BusyFlag=0，则等待 BIT 处理器完成 SEQ_INIT 命令的执行，以进行接下来的处理。

(3) 运行 MPEG-4 解码进程

《微型机与应用》2013 年 第 32 卷 第 10 期

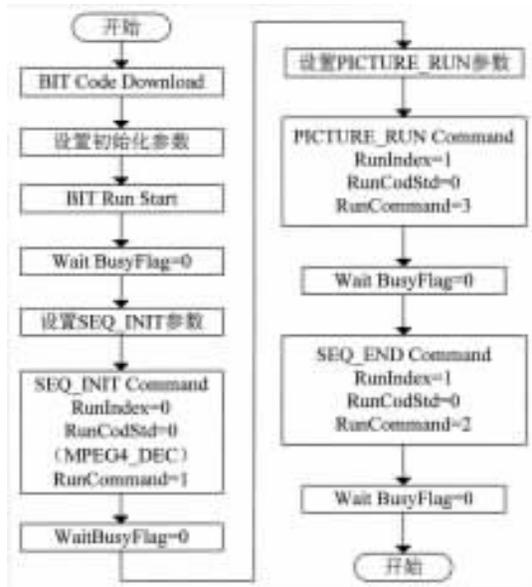


图6 VPU中MPEG4解码流程

设置 PICTURE_RUN 参数配置帧源地址和目标地址;运行 PICTURE_RUN 命令,通过调用 MPEG4_continue() 启动 MPEG-4 的解码进程。以帧为单位完成读输入、解码和写输出的工作。返回 1 代表解码正常,可对其本身进行再调用,实现下一帧的解码;返回 0 则代表文件结束或者出现异常。如果出现 Wait BusyFlag=0,则等待 BIT 处理器完成 PICTURE_RUN 命令的执行。它也意味着结束了一帧数据的解码,将解码后的数据发送到图像处理单元进行后处理并播放。

(4)继续执行第(3)步,如果比特流缓冲区是空的,则在运行下一帧解码之前,主处理器应该下载新的比特流到比特流缓冲区。

(5)运行 SEQ_END 命令调用 MPEG4_stop(), 编码完成终止解码进程。对硬件进行设置,并释放内存空间。

一般而言,不同编码标准的解码处理流程是相似的,尽管不同的编码标准对应的固件版本有小幅改进,但其具体的执行过程是由 VPU 驱动来完成的。

4.2 Android 平台使用 VPU 硬件解码

之前的解码是用 C 语言编译通过的,而开发板 i.MX51 上运行的是 Android 系统,需要在 Android 系统上对解码实现,本文采用的是 JNI 技术。

首先,编写 Mikefile 文件并编译;接着,在终端中进入工程目录,运行 ndk-build 命令进行编译,编译通过后即可生成动态库文件 libvpu.so;最后,在 Java 文件代码中对本地库进行调用。使用 public native String stringFromJNI() 申明本地方法;使用 System.loadLibrary() 来加 C 动态库。至此,对本地库的调用即完成。

本文深入地分析和研究了 i.MX51 开发板的功能特性,详细介绍了其硬件解码模块的内部结构和使用方法,并具体说明了其应用程序接口函数。在此基础上,探讨了其一般使用流程,并用 MPEG4 的具体解码过程进行了说明,最后在 Android 系统下实现。实验结果表明其运行情况良好。

参考文献

- [1] 李强申. Linux 视频硬件解码技术与应用研究[D]. 北京: 北京邮电大学, 2009.
- [2] 王知航. 基于 i.MX51 芯片和 Android 平台平板电脑电源管理的研究与应用[D]. 西安: 西安电子科技大学, 2011.
- [3] i.MX51: 应用处理器开发评估方案[J]. 世界电子元器件, 2010(3): 22-23.
- [4] Freescale 公司. i.MX51 多媒体应用处理器参考手册[S]. 2010.

(收稿日期: 2013-02-21)

作者简介:

周时伟, 男, 1985 年生, 硕士研究生, 主要研究方向: 嵌入式技术。

谢维波, 男, 1964 年生, 教授, 博士, 硕士生导师, 主要研究方向: 嵌入式技术、数字信号处理。

郭利全, 男, 1987 年生, 硕士研究生, 主要研究方向: 嵌入式技术。