

# 达芬奇 TMS320DM6446 的视频软件设计\*

谢文滨, 叶水生

(南昌航空大学 信息工程学院, 江西 南昌 330063)

**摘要:** 针对视频监控系统对实时性和带宽的高要求, 提出以达芬奇异构双核处理器 TMS320DM6446 为基础的解决方案。双核间的通信和协作是基于编解码器引擎机制来实现的。ARM 子系统负责 I/O 端口控制、算法调度、图形用户界面、网络传输等操作, DSP 子系统则实现 H.264 视频编解码。经过研究, 实现了视频软件设计。该方案实时性较好、部署方便, 在视频监控领域具有较好的应用前景。

**关键词:** 达芬奇; TMS320DM6446; 编解码器引擎; H.264

中图分类号: TP273+.5

文献标识码: A

文章编号: 1674-7720(2013)07-0080-03

## Design of Davinci TMS320DM6446 video software

Xie Wenbin, Ye Shuisheng

(School of Information Engineering, Nanchang Hangkong University, Nanchang 330063, China)

**Abstract:** For the real-time and bandwidth demands of video surveillance system, proposed a resolution based on DaVinci heterogeneous dual-core processor TMS320DM6446. And the Communication and collaboration between the dual-core is based on the codec engine mechanism. The ARM subsystem is responsible for the I/O port controlling, the operation of the algorithm scheduling, graphical user interface, network transmission, and so on. The DSP subsystem implements the H.264 video encoding and decoding. This paper implemented the video software system after the study. The resolution has a good real-time, and easily deployable, and has good application prospects in the field of video surveillance.

**Key words:** DaVinci; TMS320DM6446; codec engine; H.264

随着计算机多媒体技术、网络通信技术、微电子技术、数字信号处理技术的高速发展, 嵌入式数字视频监控系统正迅速深入到学校、银行、商场、机场、道路交通等领域的安防监控。然而, 视频数据量庞大, 如此广泛的应用, 也就需要压缩率更高的算法。H.264 是目前最先进的视频压缩标准, 具有高压压缩率的同时还具有更强的网络适应性, 它提供了网络抽象层, 使得 H.264 的文件更容易在不同网络上传输 (例如互联网、WCDMA、CDMA2000 等)。为了应对强劲的市场需求, 德州仪器 (TI) 公司推出达芬奇<sup>[1]</sup> (DaVinci) 技术的 TMS320DM6446 处理器, 它是一种专门针对数字视频应用、基于信号处理的解决方案, 提供了专业的集成处理器、软件、工具和支持, 以简化设计过程, 加速产品创新, 深受开发者青睐。然而, 它的开发却非常复杂, 使得许多开发者都未能

尽可能地发挥其硬件作用。为此, 本设计采用了基于达芬奇技术的 TMS320DM6446 处理器为开发平台, 介绍了达芬奇平台的双核通信机制, 并以 H.264 进行压缩编码, 设计实现视频编码软件系统。

### 1 TMS320DM6446 双核通信

TMS320DM6446 集成了 TMS320C64x + DSP 核和 ARM926EJ-S 核。在 ARM 端基于 MontaVista Linux、相关驱动和应用程序来管理芯片与外设的交互, DSP 作为 ARM 的协处理器, 主要是基于 DSP/BIOS (TI 公司的实时操作系统内核) 来处理复杂的音视频编解码相关的算法。高集成度带来了开发的复杂性, ARM 和 DSP 间通过 DSPLink 完成通信, 同时需要编解码引擎 Codec Engine 和编解码服务器 Codec Server 协助完成。

#### 1.1 Codec Engine

Codec Engine 是处理器间通信的桥梁, 是介于 ARM

\* 基金项目: 国家自然科学基金 (41101426)

侧的应用程序和 DSP 侧的算法之间的软件模块,可以通过调用一组 API 集合来调用和运行符合 xDAIS 标准的算法。Codec Engine 包括核心引擎(Core Engine)API 和 VISA (Video, Image, Speech, Audio)API。其中的 VISA API 通过 stub (ARM 端)来访问核心引擎 SPIs (System Programming Interface)和 skeleton (DSP 端),skeleton 访问核心引擎 SPIs 和 VISA SPIs,VISA SPIs 则最终调用具体的算法。因此,Codec Engine 的工作是通过完成 VISA API 的任务来体现的。整体结构如图 1 所示。

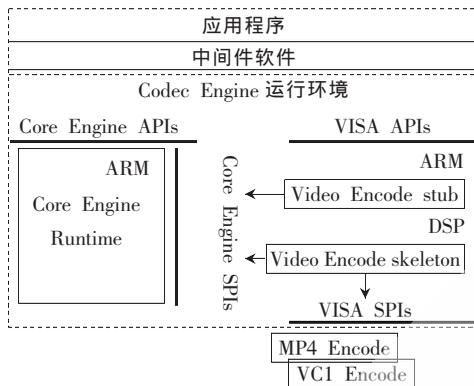


图 1 Codec Engine 结构图

核心引擎操作如下:

```
static string engine_name="videoenc"; //定义引擎名称
Engine_Handle ce; //引擎的句柄
Engine_Error errorcode; //返回引擎打开的状态信息
ce=Engine_open(engine_name,NULL,&Engine_Error);
//打开引擎
```

打开一个编解码引擎之后,就可以创建、控制、处理、删除具体的算法实例。VISA 接口支持 4 种数据源的处理,分别是视频、图像、语音和音频。

VISA API 应用如下:

```
Engine_Handle ce; //引擎的句柄
static string alg_name="h264enc"; //定义编码模块名称
VIDENC_Handle h264Handle; //编码器句柄
H264Handle=VIDENC_create(ce,alg_name,NULL);
```

## 1.2 Codec Server

Codec Server 是一个二进制文件,集成了编解码器、框架组件以及相关的系统代码,可以使用 DSP/BIOS 作为内核运行在 DSP 端。Codec Server 类似于一个网页服务器,包括了对客户请求(ARM 端应用程序对 DSP 端音视频算法的调度)进行响应的相关 DSP/BIOS 任务线程,能够用于创建编解码器,同时也提供处理器性能信息。

Codec Engine 和 Codec Server 间的通信就是 RPC(远过程调用)原理在双核上的实现。ARM 端当成客户机,访问 DSP 服务器的算法;而 DSP 端被当成服务器,响应 ARM 端应用程序的请求。

## 2 系统总体结构

系统基于达芬奇平台数字媒体处理器 TMS320DM6446

进行设计,同时结合 128MB DDR2 SDRAM 内存、视频解码器 TVP5150、CCD 摄像头、10M/100M 标准以太网芯片。系统框图如图 2 所示。

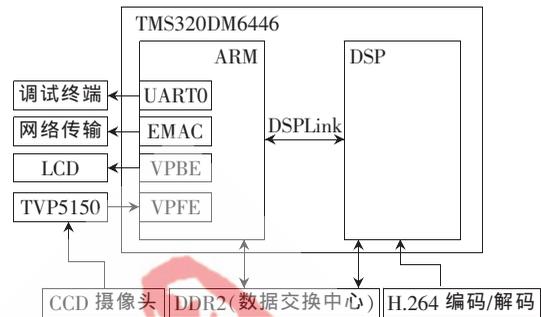


图 2 系统框图

在 ARM 端,使用 CCD 摄像头实时采集模拟视频信号,传入视频解码器 TVP5150 内进行 A/D 转换为数字视频信号,这部分的采集工作由视频前端 VPFE 负责。为了使这些数据能被 DSP 处理,VPFE 需要将 RGB 格式的原始图像转换为 YUV 格式<sup>[2]</sup>。然后使用 DSPLink 模块与 DSP 端进行通信,并通过 Codec Engine 调度 DSP 端的 H.264 编码算法,从而得到已编码的视频数据。基于 TCP/IP 协议,将编码过的数据通过以太网接口 EMAC 传输至远程管理软件管理。同时为了验证 H.264 编码的正确性和编码的质量,调用 DSP 端的 H.264 解码算法对已编码视频数据进行解码操作,最后通过 ARM 侧的视频后端 VPBE 将解码过的视频数据送到 LCD 进行显示。VPBE 能将 YUV 格式的图像转换为 LCD 支持的 NTSC 或者 PAL 格式。UART0 接口主要用于系统调试,DDR2 则是视频流数据的交换中心。

## 3 软件方案设计

本设计中,以 MontaVista Linux 操作系统作为软件开发平台,基于 Linux 下视频设备驱动程序规范 V4L2<sup>[3]</sup>(Video for Linux Two)、帧缓冲机制(Frame Buffer)和多线程技术进行视频应用程序的设计。多线程是一种多任务、并发的工作方式,能够提高程序响应及 CPU 的使用率,并能改善程序的结构。本系统软件设计为 5 个线程,分别为主线程、显示线程、采集线程、视频线程和控制线程。

### 3.1 主线程

主线程用于执行必要的初始化,解析用户程序提供的命令行参数,并根据这些参数的值产生其他线程。初始化任务主要包括视频标准检测,使用了帧缓冲设备驱动程序的 ioctl(FBIO\_GETSTD);用户命令行参数解析;Codec Engine 初始化;创建显示、采集和视频三个线程;调用控制线程的函数 ctrlThrFxn(),从而主线程转变为控制线程。控制线程负责用户接口,可用于响应键盘输入的命令;还能在 OSD(On-Screen Display,达芬奇平台的在屏显示技术)上显示和更新文字、图形,使用 setOsdTransparency()函数设置 OSD 窗口的透明度;同时能够在 OSD 上绘制按钮并对文字着色。具体的流程如

图3所示。

### 3.2 采集线程

ARM端的采集线程基于V4L2设备驱动程序来获取实时的视频缓冲数据,并传给视频线程处理。采集线程的流程如下:

(1) 打开视频采集设备

```
fd=open ("/dev/video0", O_RDWR | O_NONBLOCK, 0);
```

(2) 初始化设备

首先,查询设备的性能,获取视频采集设备的capability,辨别是否具有视频输入特性。

```
struct v4l2_capability cap;
ret=ioctl (fd, VIDIOC_QUERYCAP, &cap);
```

然后,选择视频输入的功能。

```
int input;
```

```
input=TVP5146_AMUX_SVIDEO;
```

```
ret=ioctl (fd, VIDIOC_S_INPUT, &input);
```

设置视频制式和帧格式。

```
struct v4l2_format fmt;
fmt.type=V4L2_BUF_TYPE_VIDEO_CAPTURE;
fmt.fmt.pix.pixelformat=V4L2_PIX_FMT_UYVY;
fmt.fmt.pix.width=IMAGE_WIDTH;
fmt.fmt.pix.height=IMAGE_HEIGHT;
fmt.fmt.pix.field=V4L2_FIELD_INTERLACED;
ret=ioctl (fd, VIDIOC_S_FMT, &fmt);
```

(3) 设定数据传输方式

采用内存映射模式。在这种模式下,应用程序和驱动程序之间只有指向视频数据buffer的指针被交换,数据本身不会被复制,提高了效率。

(4) 实际的采集操作

V4L2中有两个队列:输入和输出<sup>[4]</sup>。先把映射好的内存片段入队,接着开始采集视频数据并进入read循环,此时应用程序等待输出队列被填满时可以将内存片段出队,出队的的数据就是摄像头采集的视频数据。将出队的内存片段处理完后重新入队列尾,这样可以循环采集。入队和出队操作使用V4L2中的VIDIOC\_QBUF和VIDIOC\_DQBUF的ioctl()函数调用。

```
struct v4l2_buffer v4l2buf;
ioctl (fd, VIDIOC_DQBUF, &v4l2buf);
//从采集的数据中获取一个帧缓冲
ioctl (fd, VIDIOC_QBUF, &v4l2buf);
//将采集的帧缓冲发回到设备驱动中
```

同时,应用程序可以调用VIDIOC\_STREAMON的ioctl()来启动视频采集,VIDIOC\_STREAMOFF的ioctl()则用来停止视频采集。

```
type=V4L2_BUF_TYPE_VIDEO_CAPTURE;
```

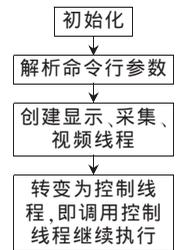


图3 主线程流程图

```
ret=ioctl (fd, VIDIOC_STREAMON, &type);
```

```
ret=ioctl (fd, VIDIOC_STREAMOFF, &type);
```

(5) 关闭设备

```
close (fd);
```

### 3.3 视频线程及其交互

视频线程基于Codec Engine机制调用DSP端的视频编码算法对该数据进行H.264编码,经过预测、量化、熵编码等一系列的操作,大幅减小视频数据的存储容量,从而降低视频传输对网络带宽的需求。已编码的视频数据可以存储在本地硬盘或者基于TCP/IP协议发送至远程管理软件管理。同时为了验证视频编码算法的正确性与质量,调用DSP端H.264解码算法对已编码数据进行解码操作,并且送入显示线程,使用frame buffer机制将其显示在LCD屏幕上。详细的交互流程如图4所示。

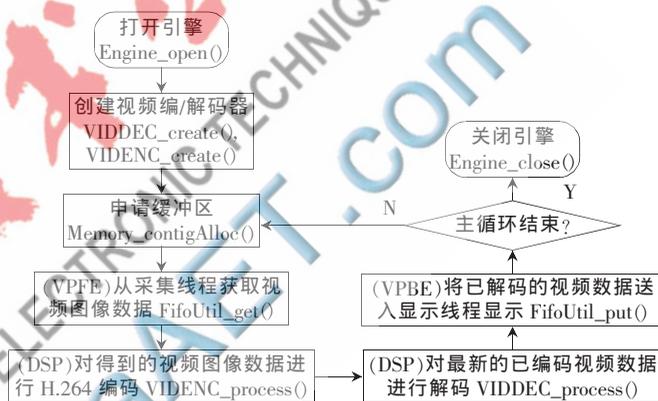


图4 视频线程交互流程图

### 3.4 实际效果

采用PAL制式,分辨率为720×480,帧率25 f/s。当目标移动速度过快时,视频的清晰度会较大程度降低。实际视频效果如图5所示。



图5 效果图

结合达芬奇TMS320DM6446中ARM和DSP处理器各自的优势,提出了基于H.264压缩算法的视频编码软件系统的设计方案,实现视频采集、视频编解码、视频显示和网络传输等功能,在以后的工作中将加入行为分析和追踪等功能,以使系统更加智能。本设计可应用于学校、交通、商业、银行等多个领域,具有较好的前景。

参考文献

[1] 王铭铭,方千山,颜佳泉,等.图像处理 and 达芬奇技术在

- 纺纱断线检测中的应用 [J]. 微型机与应用, 2012, 31 (17): 44-45.
- [2] HAN Peicen, YE Zhaohui, YANG Shiyuan. The Design and Implementation of Network Video Surveillance System Based on Davinci Chips [C]. Qingdao China:Advances in Information Technology and Education, 2011, 201: 296-302.
- [3] Bill Dirks. Video for Linux Two[EB/OL]. (2003-6-26)http://www.thedirks.org/v412.
- [4] 赵勇,袁誉乐,丁锐.DAVINCI 技术原理与应用指南[M].

南京:东南大学出版社, 2008.

(收稿日期: 2012-12-09)

作者简介:

谢文滨,男,1987年生,硕士研究生,主要研究方向:嵌入式系统。

叶水生,男,1957年生,教授,硕士生导师,主要研究方向:嵌入式系统、数据库。

