

doi: 10.3969/j.issn.1006-1576.2012.04.013

## 基于 DaVinci 技术的 H.264 解码系统

代健美<sup>1</sup>, 耿华芳<sup>2</sup>, 刘作学<sup>1</sup>

(1. 装备学院信息装备系, 北京 101416;

2. 武汉军械士官学校光电仪器与指挥控制系, 武汉 430075)

**摘要:** 为便于高效率视频压缩和高可靠性传输, 在基于 DaVinci 技术的 DM6446 平台上设计实现 H.264 解码系统。综合运用 H.264 视频编解码技术、视频传输技术、网络通信技术和嵌入式系统设计技术, 对系统的软硬件设计进行详细论述。结果表明: 该系统可完成多分辨率、可编码率的 H.264 视频数据的实时接收、解码和显示, 而且系统配置灵活, 使用方便, 具有较强的抗信道抖动特性。

**关键词:** Davinci; DM6446; H.264; 解码

**中图分类号:** TJ02 **文献标志码:** A

## H.264 Decoder System Based on DaVinci Technology

Dai Jianmei<sup>1</sup>, Geng Huafang<sup>2</sup>, Liu Zuoxue<sup>1</sup>

(1. Dept. of Information Equipment, Academy of Equipment, Beijing 101416, China;

2. Dept. of Optoelectronics Apparatus & Command Control, Wuhan Ordnance Petty Officer Institute of PLA, Wuhan 430075, China)

**Abstract:** Realize a H.264 decoder system based on DaVinci DM6446 platform for a high efficient video compression and a high reliable video transmission. The system hardware and software framework is discussed in detail integrating and applying the technique of H.264 video coding and recoding, video transmission, network communication and embedded system design. The result shows that the system can receive, decode and display multiple resolution and variable bit rate of H.264 video data real-time, and is flexible to configuring, convenient to use and anti-channel jittered.

**Key words:** Davinci; DM6446; H.264; decoder

### 0 引言

H.264/AVC 是 ITU-T 和 ISO/IEC 联合推出的视频压缩编码标准, 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, 在数字媒体领域有着非常广阔的应用前景。H.264 标准支持帧间编码和帧内编码, 其视频编译码结构采用视频编码层 (video coding layer, VCL) 和网络适配层 (network abstraction layer, NAL) 2 层设计, 不仅提高了视频编码压缩效率, 也增强了网络的适应性; H.264 算法还具有多参考帧模式预测和高精度运动估计等功能, 能够有效改善视频传输性能<sup>[1]</sup>。

基于 TI DaVinci 技术的 TMS320DM6446 (以下简称 DM6446) 是具有 ARM+DSP 双核结构的高性能处理平台, 利用其 DSP 端内置的 H.264 编解码软核, 用户可通过 API 函数调用实现高效、稳定的 H.264 编解码; 当然, DSP 端也可以运行用户定制优化的符合 DM6446 xDM 标准的 H.264 编解码算法库, 具有很好的扩展性和灵活性。

因此, 笔者基于 DM6446 平台, 综合运用 H.264 视频编解码技术、视频传输技术、网络通信技术和嵌入式系统设计技术实现了 H.264 解码系统, 可实现多分辨率视频图像的自适应解码、本地存储和实时显示功能。

### 1 硬件设计与系统定制

参考 DM6446 评估板的原理样图, 根据系统功能, 去除了与解码无关的视频输入模数转换电路、NOR FLASH 接口电路、JTAG 调试接口电路、MSP430 单片机相关电路以及 SPI 等外设接口电路; 保留了 DDR2 接口电路 (用于大容量数据的交换和程序的运行)、NAND FLASH 接口电路 (用于 Bootloader、kernel 和文件系统的存储)、串口和网络接口电路 (用于程序下载、在线调试以及实时配置)、USB 接口电路 (用于外接无线网卡和 U 盘, 以实现数据的无线传输和本地存储) 和音频电路等; 为降低电路板尺寸, 提高系统的集成度, 增强电源的可靠性, 修改了电源电路; 修改了模拟视频输出电

收稿日期: 2011-10-16; 修回日期: 2011-11-07

作者简介: 代健美 (1981—), 男, 吉林人, 硕士, 讲师, 从事军事通信技术、无线通信系统、嵌入式系统设计研究。

路。笔者首先对设计的电源电路和模拟视频输出电路进行分析，然后就系统定制情况进行介绍。

系统硬件结构原理框图如图 1 所示。

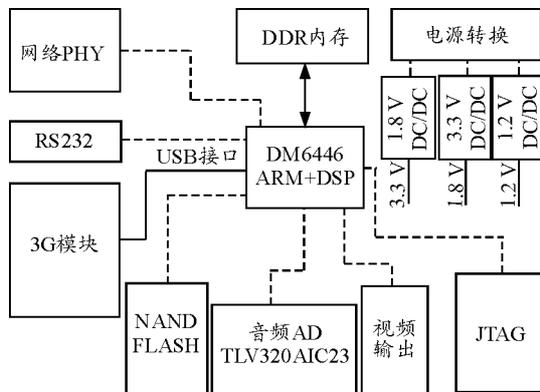


图 1 硬件结构原理框图

### 1.1 电源电路设计

系统电源包括 DM6446 工作电源和外围器件工作电源，其中，DM6446 需要内核与 I/O 2 种电源，而内核电源又分为 CVDD(1.2 V)和 CVDDDSP(1.2 V)，I/O 电源分为 DVDD18(1.8 V)，DVDDR2(1.8 V)，DVDD33(3.3 V)，这些电源有严格的启动时序<sup>[2]</sup>。为保证系统的可靠性，在设计时不仅要考虑电压的精度、稳定性和外围电路的复杂度，还要考虑低功耗、上掉电时序等问题。

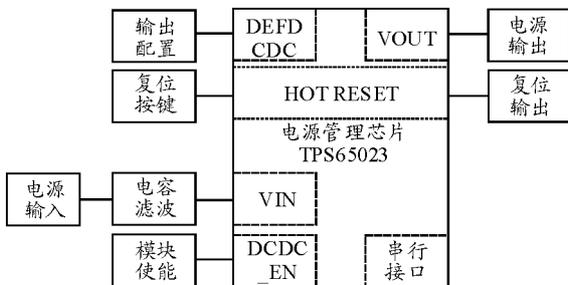


图 2 电源电路原理框图

笔者利用 TPS65023 电源管理芯片构建了电源电路，其原理框图如图 2 所示，5 V 直流电压作为输入电压，利用 100 kΩ 电阻上拉使能输出，根据系统负载情况，将 DEFDCDC1 配置为高，DEFDCDC2、DEFDCDC3 配置为低，实现 1.2 V/1 A，1.8 V/1.2 A 和 3.3 V/1.5 A 等 3 种输出电压，芯片内部逻辑电路保证了 3 种电压的时序要求，利用芯片的 Hot Reset 功能实现系统的软复位。还可以通过 I2C 接口对 TPS65023 进行配置，实现输出电源的动态管理，基于系统设计复杂度和实际功能需求的考虑，该接口悬空。

### 1.2 模拟视频输出电路设计

DM6446 的视频输出接口支持 1 路数字 LCD 接口和 4 路模拟视频 DAC 输出接口。因系统要求视频数据能直接接入电视的 CVBS 接口并显示，故笔者选择 1 路模拟视频 DAC 输出配合阻抗匹配电路实现。由于 DM6446 的模拟视频 DAC 输出设计驱动负载阻抗为 500 Ω<sup>[3]</sup>，而电视机的典型阻抗为 75 Ω，为保证视频信息的正确、不失真，笔者设计了基于 RC 低通滤波器和 OPA357 视频放大器的阻抗匹配电路及视频输出电路，模拟视频输出电路原理框图如图 3，图 4 和图 5 分别显示的是阻抗为 500 Ω 和 75 Ω 的低通滤波器电路原理图。



图 3 模拟视频输出电路原理框图

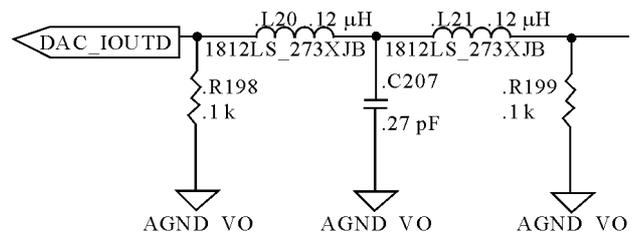


图 4 500 Ω 的低通滤波器电路原理图

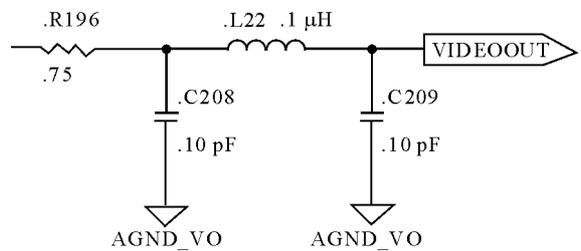


图 5 75 Ω 的低通滤波器电路原理图

### 1.3 系统定制

系统软件包括底层软件和应用软件 2 部分。

底层软件包括引导加载程序 Bootloader、内核和文件系统等 3 部分：1) Bootloader 使用 UBL.bin 和 uboot.bin 2 个文件，UBL.bin 是 TI 开发的针对 DM6446 硬件特点的引导程序，其指定了用于程序引导的 DM6446 内部 RAM 地址；uboot.bin 在 uboot-1.2.0 标准源代码的基础上改进生成，主要增加了对 LARGE PAGESIZE 的 NAND FLASH 的支持，二者通过串口下载；2) 内核文件基于开放源代码的嵌入式 Linux2.6.18 操作系统编译生成，裁减了并口、鼠标、硬盘和无线网络等驱动，保留了 NAND FLASH、mtd 分区、NFS 网络文件系统、yaffs2 文

件系统、USB 主设备和 framebuffer 等驱动, 生成的 uImage 内核文件通过网络 tftp 方式下载; 3) 文件系统使用的是对 NAND FLASH 支持较好、启动速度较快、容错性较强的 yaffs2 文件系统, 通过 NFS 进行烧写。

应用软件用标准 C 代码实现, 交叉编译器版本为 arm\_v5t\_le-GCC 4.3。

## 2 应用软件设计

系统主要实现了 H.264 视频数据的无线网络接收, H.264 视频解码与播放, 以及视频信息 OSD 显示等。此外, 系统还设置了基于文本的数据库, 主要存储视频编码发送端 (Server) 的 IP 地址、地理位置、视频格式等信息, 可通过按键选择数据库内存储的任意 Server 进行解码和显示; 同时, 设计了串口应用程序, 既可对系统数据库进行实时显示, 也可对数据库进行增加、修改和删除等配置。

### 2.1 无线网络接收程序设计

视频数据通过板卡外接的 USB 3G 无线网卡 (静态专网网卡) 接收, 该网卡内部集成了 3G 网络连接端口协议, 并嵌入了 TCP/IP 协议、POP3/SMTP 协议, 在系统中注册为 PPP0 网络设备, 具有固定的 IP、MAC 地址等信息, 可以使用标准的网络接收程序。

因 3G 网络误码率较高、抖动性大, 为保证视频数据的可靠性, 使用多 TCP 传输的方式, 主要接收过程如图 6。网络接收程序为每个 TCP 连接单独创建一个新的线程, TCP 连接数初始设定为 3, 也可根据具体网络环境通过串口进行实时配置, 每个线程收到的数据单独存储在一个环形缓冲区——接收缓存区 (RBUFFERi) 中。其工作过程为:

1) 首先建立 TCP Socket, 得到 Socket 句柄号 fd, 如果 fd 大于 0, 说明 Socket 建立成功, 否则返回;

2) 然后, 向 Server 端发起 connect 连接, 如果成功, 获取到一个新的句柄 nfd, 如果 nfd 为 0, 说明连接成功, 否则返回;

3) 连接成功后, 利用 recv 函数接收视频数据, 视频解码是以帧为单位进行发送的, 但由于采用的是 TCP 传输方式, 到达的数据并不一定是完整的帧, 为了正确恢复视频数据帧, 在发送时, 对视频数据进行了类 RTP 封装, 增加了每次传输数据的帧序号、包序号、时间戳、帧长度等信息;

4) 收到数据后, 首先判断是否为完整的帧, 如果是, 将该帧解封装, 将纯视频数据顺序压入 RBUFFERi, 否则, 继续接收, 直到收到完整的下一帧数据。

需要说明的是, 由于使用多 TCP 方式对同一个 Server 视频源进行接收, 因此各个 RBUFFERi 的数据帧序号不是连续的; 另外, 各 TCP 线程是基于能力来接收数据的, 每个 RBUFFERi 中的帧数并不是平均分配的。为正确解码, 程序对 RBUFFERi 的数据帧进行了重新排序并存入另一个全局环形缓冲区——解码缓冲区 (DBUFFER), 此时, 存储在 DBUFFER 的数据从 RBUFFER 弹出, 释放内存资源; 当解码程序从 DBUFFER 读取数据并解码时, 该数据从 DBUFFER 中弹出, 内存资源被释放。

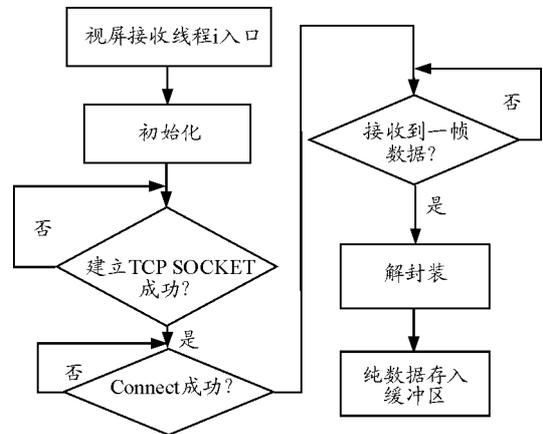


图 6 网络接收程序框图

### 2.2 视频解码程序设计

视频数据的解码由 DM6446 DSP 内置的 H.264 解码器完成, 通过调用其在 ARM linux 端的 API 函数实现<sup>[4-5]</sup>。这些 API 函数可完成 H.264 解码器的例化, 参数的设定, 解码处理和解码器实例的销毁等功能, 函数原型如下:

1) VIDDEC\_create(Engine\_Handle e, String name, VIDDEC\_Params \*params);

此函数完成 H.264 解码器实例的创建, e 表示已打开的引擎的句柄; name 表示将要创建的视频解码器类型的标识符; params 表示创建的参数; 实例创建成功返回一非 NULL 值, 否则返回 NULL;

2) VIDDEC\_process(VIDDEC\_Handle handle, XDM\_BufDesc \*inBufs, XDM\_BufDesc \*outBufs, VIDDEC\_InArgs \*inArgs, VIDDEC\_OutArgs \*outArgs);

此函数完成一帧数据的解码处理, handle 表示

已经创建的视频解码器实例的句柄；inBufs 表示输入 H.264 视频缓冲区的地址；outBufs 表示解码后视频数据缓冲区的地址；inArgs 和 outArgs 分别表示输入输出参数，包括缓冲区大小、个数等信息；解码处理成功返回 0，否则返回-1。

3) VIDDEC\_control(VIDDEC\_Handle handle, VIDDEC\_Cmd id, VIDDEC\_DynamicParams \*params, VIDDEC\_Status \*status);

此函数完成解码参数的设定，包括视频数据的分辨率、码速率、帧率、数据格式、色度和亮度信息等；参数设定成功返回 0，否则返回-1。

4) VIDDEC\_delete(VIDDEC\_Handle handle);

此函数完成 1 个解码器实例的删除工作，释放资源。

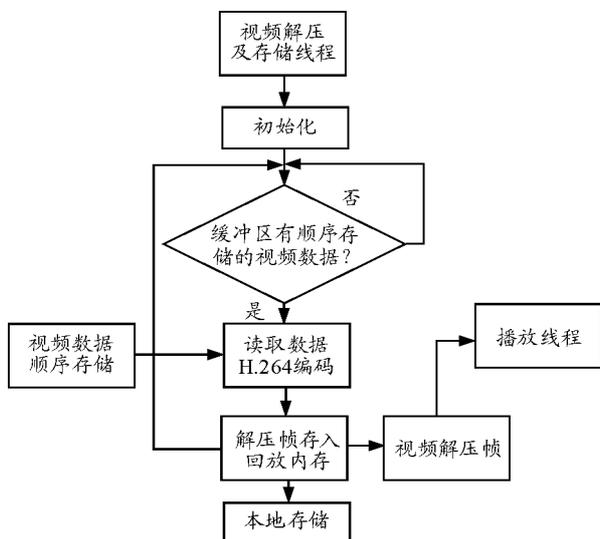


图 7 视频解码程序框图

基于上述 API 函数，设计了视频解码的软件流程，如图 7 所示。首先是初始化过程，包括解码引擎的打开，连续缓冲区的设定等；然后读取 DBUFFER 的信息，如果 DBUFFER 中有可用数据，则拷贝数据并解码；解压后的数据送入 PIPE 管道等待播放线程的读取，也可根据需要以视频文件的形式存储至 U 盘。

### 2.3 播放显示程序设计

为保证视频数据能够以 25 帧/s 实时播放，防止由于解码延时造成帧丢失，视频播放与视频解码分别由不同的线程实现，数据的交互通过管道 (PIPE) 完成。

在本系统中，需要播放显示的内容包含 2 部分：一是 YUV 视频数据显示；二是属性信息显示。包括 DM6446 在内的 DaVinci 平台提供了层次清晰的显示设备——FBDev，该设备分为 4 层，在操作系统中分别表示为/dev/fb/0、1、2、3。笔者将 YUV 视频数据显示在/dev/fb/3 上，显示时，程序会根据视频分辨率的不同而调用 Resizer 硬件缩放模块，从而实现视频信息的全屏显示；包括帧率、数据速率、地理位置信息和播放时间等内容的属性信息绘制在/dev/fb/0 上，可根据实际情况动态更新。

### 3 结语

使用结果证明，该系统可完成多分辨率 (D1、CIF)、可变码率 (0.256~2 Mbit/s) 的 H.264 视频数据的实时接收、解码和显示。同时，多 TCP 接收方法的使用降低了 3G 信道抖动对数据传输的影响，在可以接受的延时 (D1 分辨率、512 kbit/s 下最大延时为 3 s；CIF 分辨率、256 kbit/s 无延时) 条件下，增强视频播放的流畅性；串口配置功能提升了系统使用的灵活性。

### 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] 代健美, 耿华芳. 基于 TPS54310 的 SOC 电源电路设计[J]. 电源技术应用, 2008(5): 10-14.
- [3] TI Instruments, TMS320DM6446 数据手册 [S]. <http://www.ti.com> 修订版, 2006: 170.
- [4] 张云川, 等. 基于 ARM 的便携式视频解码终端设计与实现[J]. 计算机工程, 2009(2): 232-234.
- [5] 卢宁, 等. 基于达芬奇技术的 H.264 视频编解码器的研究与实现[J]. 内蒙古大学学报, 2010(7): 410-416.