基于DaVinciDM 6446处理器的DVS的设计与实现①

侯建华, 蓝发财
(中南民族大学 电子信息工程学院, 武汉 430074)

摘 要 设计了在Linux操作系统上实现H.264视频编码的数字视频服务器(DVS)监控系统, 该监控系统处理器采用TI公司高性能DaVinciDM 6446, 内嵌ARM和DSP双核结构, 在ARM端完成驱动开发, 如视频芯片SAA7114驱动的开发, 网卡芯片KS721BL驱动的开发; 在DSP端实现H.264实时编码算法. 实时监控测试表明: 视频图像质量良好.

关键词 视频编码; 达芬奇; 视频监控服务器; 驱动程序
中图分类号 DTN124 文献标识码 A 文章编号 1672-4321(2009)01-0072-04

Development and Implementation of DVS Based on the DaVinciDM 6446 Processor

Hou Jianhua, Lan Facai
(College of Electronic and Information Engineering, South-Central University for Nationalities, Wuhan 430074, China)

Abstract The paper introduces the H.264 encoder implementation of digital video systems (DVS) in the Linux operating system. TI high performance DaVinciDM 6446 digital media processor is adopted with embedded ARM and C64x+ dsp double core. Driver development is realized on the ARM side such as SAA7114 video decoder driver, and real time H.264 video encoder on the DSP side. Real surveillance tests show the quality of visual image is satisfactory.

Keywords video encoder; DaVinci; DVS; driver program

随着经济的发展和技术的进步, 市场对高性能视频业务的需求不断增长, 如视频监控、视频会议、可视电话的产品. TI DaVinci技术能够提供处理器、开发工具、软件和系统设计为数字视频开发的理想平台, 使数字产品大大缩短上市时间[1,2].

本文基于TI DaVinciDM 6446的ARM 926EJS+DSP 64X+的双核处理器结构, 在ARM 926EJS端上实现系统构建, 如驱动开发、网络传输; 在C64x+DSP端上实现和优化H.264编码器, 实现DVS网络视频服务器系统. 主要给出了在ARM端对SAA7114芯片驱动开发和DSP端的H.264视频编码优化的实现过程.

1 系统硬件设计

DVS监控系统分为图像采集、压缩编码、网络传输、客户端4部分. 图1是DVS监控系统框架图, 由摄像头采集图像通过SAA7114芯片转换数字信号, 然后把码流送到DM 6446处理器完成编码, 编码以后的码流通过网卡芯片KS721BL传送到客户端, 完成整个DVS监控系统.

图像采集采用Philips公司的高性能视频解码器SAA 7114作为图像采集单元, 该芯片输入支持PAL NTSC SECAM视频信号和最多6路的模拟

① 收稿日期 2009-01-11
作者简介 侯建华(1964-), 男, 博士, 教授, 研究方向: 模式识别、图像处理, E-mail: zil@scuec.edu.cn
基金项目 湖北省自然科学基金资助项目(2007ABA106)

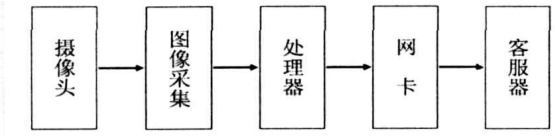


图1　体系框架图

Fig. 1　System frame

输入, 通过模数转换、数字滤波等模块, 将模拟视频信号转变为YUV格式的数字视频信号, 并输出行场同步和时钟信号, 如最常用的ITU 601 656格式的输出格式[1]. 该芯片还要对视频图像进行缩放、图像的垂直消隐解码和剪切, 这些操作通过ARM的I2C对SAA7114进行配置来实现.

压缩编码: DaVinci处理器有着强大的特性和丰富的资源, C64x+ DSP核采用VelociT 1.2架构, 提供高达256 bit的指令读取带宽以及128 bit的数字存取带宽, 8个功能单元通过各自独立的数据存取流寄存. 视频处理子系统前端提供了CCD和CMOS图像接口、BT. 601 BT. 656数字、YCbCr 4∶2∶2的8位、16位接口等; 后端则提供了硬件OSD、4个54 MHz DAC, YUV或RGB数字输出等[2].

网络传输: 采用Micrel公司的KS721BL网卡芯片, 该芯片具有低功耗、快速传输的特点, 配有2.5V的核电源、外围3.3V电压, 能自动配置10 Mbit/s或100Mbit/s和全双通或半双通的通信模式.

2　系统软件设计

2.1　操作系统

目前市场上数字监控产品竞争日益激烈, 各大操作系统也日益完善, 主流有Windows、Linux、Sysmbian、Vxworks等, Linux系统是自由、开源的, 具有内核稳定、功能强大、可移植、可升级、进入门槛低的特点, 因此本文采用基于2.6.10内核的Linux作为操作系统.

Linux通过设备驱动程序为应用程序提供了统一抽象的接口, 从而隐藏了大量不同设备之间的区别和细节. 在Linux中所有对硬件设备的操作和普通文件一样, 利用标准的系统调用在设备上进行打开、关闭、读取和写入操作. 系统中每个设备由"设备特殊文件"来代表. 每个由相同设备驱动程序控制的设备具有相同的主设备号, 而次设备号则用来区分同类设备中的不同设备. 设备特殊文件的虚拟文件系统索引节点中包含设备号信息. 如果通过系统调用访问设备, 则内核可通过该虚拟文件系统索引节点中的设备号信息调用适当的设备驱动程序[3,4].

2.2　SAA7114的驱动方程

SAA7114的驱动方程完成视频设备打开、设备注册、视频格式大小的配置, 设备与DSP驱动的通信, 以及最后设备的关闭[3,4]. 通过module_init(saa7114_init)模块来加载saa7114_init, 是驱动函数的入口.

```
static int __init saa7114_init(void)
{
return i2c_add_driver(&i2c_driver_saa7114);
}
```

i2c_add_driver()用于注册设备驱动程序的i2c_driver数据结构. 由此加载i2c_driver的结体i2c_driver_saa7114.

```
static struct i2c_driver i2c_driver_saa7114 =
{
.owner = THIS_MODULE,
.name = "saa7114",
.id = I2C_DRIVERID_SAA7114,
.flags = I2C_DF_NOTIFY,
.attach_adapter = saa7114_attach_adapter,
.detach_client = saa7114_detach_client,
.command = saa7114_command,
};
```

其中driver为驱动名, id为驱动号. attach_adapter依附i2c_adapter函数指针. detach_client为脱离i2c_client函数指针. command为实现针对设备的控制命令指针, 类似ioctl函数.

saa7114_probe()是实现saa7114_attach_adapter的函数. 可以简单的通过调用i2c_probe()函数来实现.

i2c_probe(adapter, &addr_data, &saa7114_detect_client)函数, adapter为内核指针数组adapters[]的一个元素, 代表当前被扫描的i2c总线. addr_data在ltc3445.h中由I2C_CLIENT_INSMOD宏创建的静态二维数组, 表示使用该驱动程序的所有设备的所有可能地址的集合. saa7114_detect_client函数来完成驱动的设置和注册完成. 首先通过i2c_check_functionality来检测设备什么支持, 然后对设备地址, 驱动名, 句柄设置, 最后通过i2c_attach_client(client)来加载到i2c总线.

脱离函数saa7114_detach_client, 通过i2c_detach_client(client)来卸载函数函数指针. 控制函

数saa7114 command,这个函数实现了针对设备的控制命令,这里主要是留出了V4L2的API函数接口,通过应用层来对接口函数调用,通过接口函数对芯片控制,完成各种寄存器是设置. 如通过DECODER GET STATUS来获得视频输入的格式,NTSC、PAL、SECAM 模式 通过DECODER SET NORM 来获得视频模式NTSC、PAL、SECAM. 通DECODER SET PICTURE来设置图像亮度色度,如视频图像的对比亮度的寄存器0x0AH控制、色度的寄存器通过0x0CH 寄存器进行控制.

控制函数也是与DSP驱动实现通信的桥梁. 通过davinci vpfe.c中里面的vpfe doioctl(struct inode * inode, struct file * file, unsigned int cmd, void * arg)来实现saa7114芯片的设置,如视频的采集格式、大小的设置、码流的设置,最终完成对视频图像采集需求.

2.3 网络传输

网络传输通过TCP、UDP、RTP协议来实现H.264编码以后的码流传输. 其中TCP是面向连接的传输控制协议,为应用进程提供可靠的、端到端的、面向连接的字节流通信协议. UDP是无连接的用户数据报传输协议,与TCP相比,UDP的报文相对简单的通信协议. RTP是一种提供一对一或一对多的实时传输协议,一般位于UDP协议之上,来保证完善的通信传输的机制. 网络的协议取决于路由选择、网络拥塞程度和视频质量[5].

3 编码优化和系统测试

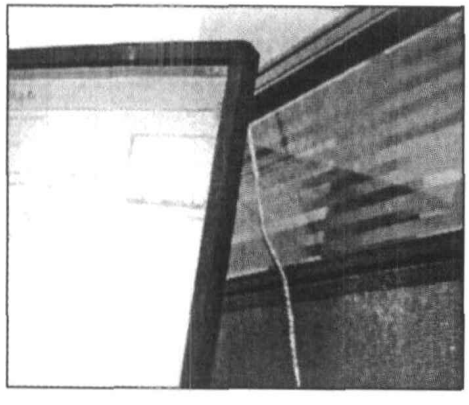
由于H.264编码算法有一系列更高效的压缩编码技术,编码效率、实现复杂度和网络适应能力等因素[5],本系统视频编码采用H.264算法。不过H.264编码算法实现相对复杂,为了在TIDM6446上实现H.264编码器实时功能,需要对代码进行大量的优化,首先是算法级优化,然后是C编译器的编译优化、C代码的修改优化,最后手工优化.

算法级优化包括运动估计算法的选择,搜索算法中的三步法、六边形搜索算法、钻石算法. 三步法虽然简单,但容易陷入局部最小点; 预测运动矢量好坏对搜索算法复杂度和准确性有很大的影响,常用的有PMVFAST算法、APDZS算法.

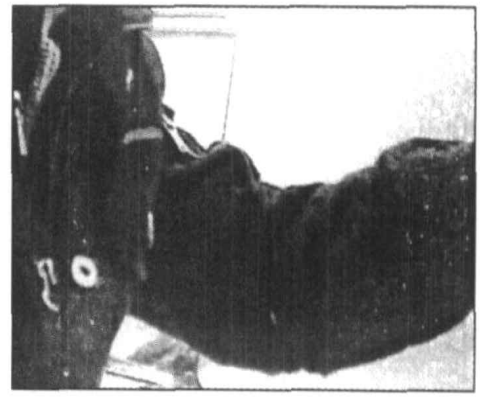
C编译器的编译优化选择,如优化级别-o1、-o2、-o3,其中-o3对文件级别进行最强的优化,不过编译比较条件苛刻,容易出错. -pm选项对程序级别进行优化,所以将文件一起来优化,去掉未使用的函数、变量. -ms0选项不使用冗余进行优化,减少程序的大小.

C代码的修改优化,如关键字的使用,如nassert、restrict、MUST_ITERATE. 进一步线性汇编优化,如打包指令ADDSP、LDDW、SUBABS4、DOTPU4、ADDSUB2、DPACK2、SHLMB、SWAP2. 通过合理使用关键词和打包指令是CPU流水线并行工作,提高程序的执行速度.

在数据的采集传输方面合理利用硬件资源,如Cache, DMA的利用,在Cache主要通过DSPLINK配置和堆栈的设置,DMA的优化主要通过DSP驱动中合理设置,通过EDMA 3通道从片外取数据到片内,并且与CPU可并行工作,从而大大提高编码的速度,满足实时的H.264编码的要求[2,6].



(a)



(b)

图2 实时视频图像

Fig 2 Real-time video image

我们通过TI DaVinciDM 6446 微处理器开发平台上对H. 264 编码以后码流提取. 实际测试视频大小720×576, PAL 格式, 25 帧/s 到30 帧/s, 满足实时监控的要求. 图2 中图(a)、(b) 是实时视频图像, 从图像质量效果来, 与海康威视, 南望科技视频监控产品相当, 可满足DVS 系统的客户端实时视频监控需求.

4　结语

文本针对TI DaVinciDM 6446 微处理器开发平台, 在Linux 操作系统中完成驱动的开发、系统的构建、H. 264 实时编码、应用层的开发, 完成DVS 系统框架. 但作为监控产品其可靠性, 稳定行, 图像的清晰有更高的要求, 友好的用户平台和控制系统, 是今后改善的重点和研发的重点.

参　考　文　献

[1] 陈　东, 杨兆选. 一种嵌入式视频交通监控系统的设计与实现[J]. 计算机测量与控制, 2006, 14(9): 1 200-1 203.

[2] 应　翔. 基于DaVinci处理器的H. 264 视频编码器软件设计和优化实现[D]. 杭州: 浙江大学, 2007.

[3] Corber J. Linux Device Driver [M]. 3rd ed. California: O'Reilly, 2005.

[4] 陈晓宇, 毕笃颜. 基于Linux 的MPEG-4 视频压缩卡驱动程序[J]. 计算机工程, 2007, 33(8): 270-272.

[5] 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: 235-242.

[6] 陈航锋. 基于DaVinci处理器的H. 264 视频解码器软件设计和优化实现[D]. 杭州: 浙江大学, 2007.