

基于 ARM 和 DSP 的多通道数字硬盘录像机的研究与设计

Design of Multi-channel Digital Harddisk Recorder Based on ARM and DSP

李健 黄伟志 (天津工业大学信息与通信工程学院,天津 300160)

摘要

介绍了一种基于嵌入式微处理器 S3C2510 和专用多媒体处理芯片 TMS320DM642,结合其它一些外围器件所开发的高性价比的 16 路数字硬盘录像机系统的硬件设计。讨论了系统总体设计方案,介绍了视音频的编解码、硬盘存储和网络传输等各功能模块及关键接口电路设计;并给出了软件设计及远程控制通信的建立。

关键词: S3C2510, DM642, DVR, 双核, 数字硬盘录像机

Abstract

This paper introduces the design of a cost-effective 16-channel digital harddisk recorder system which based on embedded microprocessor S3C2510 and exclusive multimedia processing chip TMS320DM642, combined with some other external device. Discusses the system design program and introduces the design of the various functional modules, such as video code, disk storage, network transmission, and the vital interface circuits; and gives the software design.

Keywords: S3C2510, DM642, DVR, dual-core, digital harddisk recode

嵌入式数字视频录像机(Digital Video Recorder, DVR)是基于嵌入式处理器和嵌入式实时操作系统的嵌入式系统。它采用专用芯片对视频图像进行压缩存储及解压回放,嵌入式操作系统主要完成整机的控制及管理。目前,DVR已集硬盘录像机、视频控制器、远程网络传输、云台控制、图像移动侦测、报警联动等功能于一体,可以方便地实现网络监控及分控。

本文介绍针对现在市场对于 DVR 系统高性价比的要求,在 TI 专用多媒体芯片 TMS320DM642(以下简称 DM642)和三星公司的嵌入式处理器 S3C2510A 基础上,结合其它一些外围器件开发的高性价比的 16 路数字硬盘录像机系统。

1 系统总体设计及原理

1.1 系统总体方案设计

本方案设计的高性能 DVR 系统可以支持 16 路 CIF/4 路 DI 的视频输入,16 路音频输入,1 路实时监控输出和 1 路回放输出,可以 24 小时不间断地进行有声视频的采集、回放与存储。本系统主要由以下几个模块组成:DM642 模块、视频和音频模块、系统控制、存储、传输模块、电源模块等,系统结构如图 1 所示。

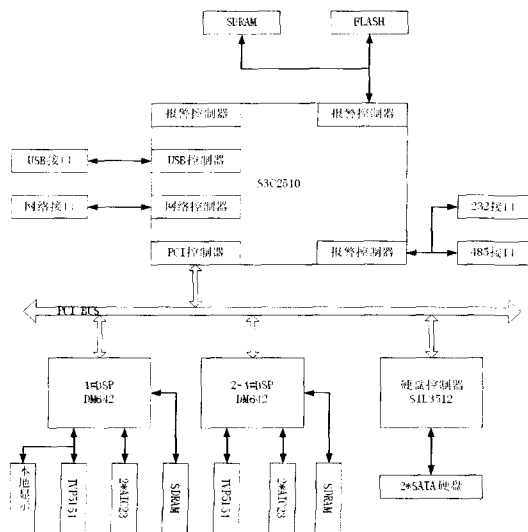


图 1 系统框图

系统的工作过程如下:系统上电或复位后,S3C2510A 从 Flash 加载程序,完成对芯片的初始化和外围硬件的配置等工作,之后便开始进行图像采集。DM642 通过 I2C 口对系统中的其他芯片进行控制,从摄像头采集到的模拟视频信号经过视频解码器转换为数字视频信号,送入 DM642 的视频通道(VP 端口);同步采集到的模拟音频信号经过音频编解码器模/数转换后,送入 DM642 的音频通道(McASP 端口)。DM642 将接收到的数字视频信号和数字音频信号用 H.264 标准编码压缩,编码后的音视频数据可以通过主控部分存储在 SATA 硬盘中或者通过网络进行传输。

1.2 视频和音频模块

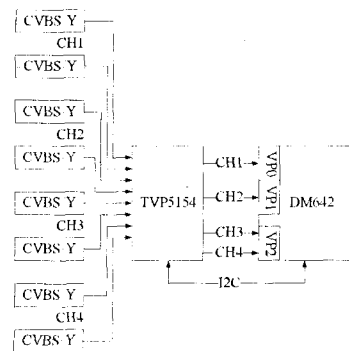
设计中将 DM642 的 VP0、VP1 的 A 通道以及 VP2 的 A、B 通道都配置为视频采集模式,可同时采集 4 路视频信号。若要求进行本地回放,则将 VP0 端口的 A 通道设置为回放模式,此时最多可采集 3 路视频信号。另外,将 VP0 和 VP1 的 B 通道配置成 McASP 功能,用于音频处理。DM642 与视频、音频编解码芯片之间的数据都经过这几个专用的端口进行传输,但是对这些芯片的控制则是通过 I²C 总线实现的。

1.2.1 模拟视频输入模块

采用模拟摄像头进行视频数据采集,需要使用视频解码芯片将采集到的模拟数据进行数字化,然后送给 DM642 的视频端口进行处理。这里选用了 TI 推出的 4 通道低功耗视频解码器 TVP5154 进行 4 路视频输入采集。TVP514 与 DM642 接口见图 2。

1.2.2 视频输出部分

设计中一片 DSP 芯片的 VP0 口 A 通道进行视频回放。对采集到的数据进行回放时需要先将数字视频信号重新转化为模拟信号,才能在监视器上播放。这种转换由 Philips 公司生产的 SAA7105 来完成,图 2 TVP514 与 DM642 接口电路



其可输出 PAL/NTSC 制式的复合视频信号、S 端子视频信号和 RGB 视频信号。视频回放部分的电路连接图如图 3 所示。

图 3 中,DM642 的 VPO 端口向 SAA7105 的 PD 口输出 BT.656 格式的数字视频信号,经过 SAA7105 芯片内部数据管理模块可输出 PAL/NTSC 制式的复合视频信号、S 端子视频信号和 RGB 视频信号;可由 CVBS(复合视频信号)或者 Y、C(S 一端子信号)引脚输出或者用 VGA 口输出。SAA7105 的时钟信号由 DM642 的 VPOCLK0 提供,而 DM642 的 VPOCLK0 则是来自于 TVP5154 的 SCLK 引脚,这就使得视频的采集和回放达到同步要求。DM642 通过 I²C 接口对 SAA7105 芯片内部寄存器进行配置,实现对该芯片的控制。

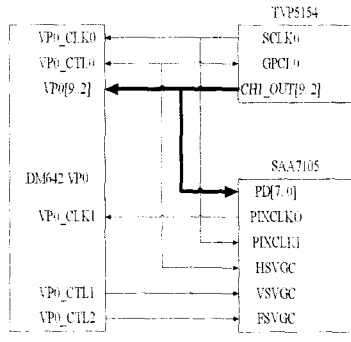


图 3 视频回放模块连接电路

1.2.3 音频模块

系统中的音频编解码芯片采用的是 TI 公司的 TLV320A-IC23B 芯片,它包含 3 个接口:①串行控制接口,与 DM642 的 I²C 接口连接;②模拟音频接口,用于接收来自 MICIN/LINEIN 的模拟音频信号,或输出 LINE-OUT 的模拟音频信号;③数字音频接口,用于和 DM642 的 McASP 端口进行数据传。

在本方案中,音频的采集与图像的采集、音频的播放与图像的播放必须保持同步。因而,每片 DSP 芯片上的 4 片 AIC23B 的时钟输入是由 TVP5154 的 SCLK 经过 PLL1708 适当分频后的 SCLKO3(384fs)提供;另外为了得到合适的位时钟和帧同步信号配合,AHCLKx 与 AHCLKr 分别与 PLL1708 的 SCLKO2 (256fs)连接。

1.3 系统控制、存储、传输模块

本方案采用 ARM+DSP 的双核结构,使 DSP 专注于视音频数据处理的任任务,而将事件处理和控任务交给另一 CPU 处理。S3C2510 是三星公司推出的一款面向网络通信的高性能嵌入式处理器,内置 ARM940T 核和指令/数据 Cache,集成了 Ethernet、PCI、UART、I²C、USB、DMA 等丰富的接口资源,系统时钟可达 133MHz,非常适合网络流媒体数据处理。同时,DM642 主要负责视音频数据压缩及编解码工作,并通过 PCI 总线与 S3C2510 交换数据和控命令;S3C2510 作为主控制器,运行 μC-Linux 实时操作系统,完成音视频编解码后数据的存储传输控制,以及对各种其他外围设备的控,并可将编码后的音视频数据存储在 SATA 硬盘中或者通过服务器任务,将压缩编码后的音视频数据以 IP 包的形式发送到网络上,由远程客户端 PC 接收和解压播放,实现音视频的远程实时监控。

数字硬盘录像机要进行长时间、不间断的视频监控,就必须有足够大的外部存储设备。由于 S3C2510 没有硬盘接口,而且其 PCI 总线最多只能接 5 个外部设备,就必须设计硬盘控电路。本方案中采用 SIL3512 来外扩两个 SATA 接口,其通过 PCI 总线实现与系统的交互响应。

1.4 电源模块

本方案中由于系统结构比较复杂、芯片数量多,因此所需的电源电压也比较多。系统主电源为+3.3V,由 ATX 电源直接提供,此外 ATX 电源还为系统提供+5V 电压输入。同时,DM642 电源及 S3C2510A 的电源要分开处理,此处不仅考虑到内核所需的电压不同,还考虑到彼此之间的电源噪声的干扰,即使是同一部分的

3.3V 电源也要分开设计,提高系统的稳定性及可靠性。

此处,要着重强调下 DM642 的供电系统。DM642 在加电过程中,应当保证内核电源先上电,最晚也应当与 I/O 电源一起加。关闭电源时,先关闭内核电源,再关闭 I/O 电源。设计中采用 2 片 TI 公司专为高性能 DSP、FPGA、ASIC 和微处理器的应用而设计的电源芯片 TPS54310,分别给 DM642 提供 CVDD 和 DVDD 电压。在电路连接上将 TPS54310(1)的 PWRGD 引脚和 TPS54310(2)的 SS/EN 引脚相连。当(1)的输出电压高于 1.2V 时,芯片(2)开始工作;当这个值达到稳定的+1.4V 后,PWRGD 引脚输出高电平送到芯片(2)的 SS/EN 引脚。这就保证了 CPU 内核的上电时间早于 I/O 的上电时间。如图 4 所示。

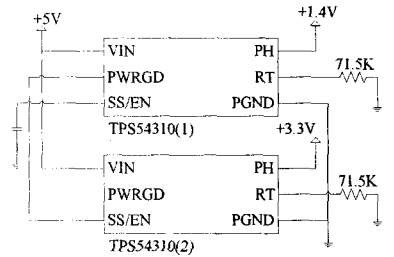


图 4 电源设计模块

2 系统的软件设计

2.1 系统的软件设计

该系统的软件主要包括三个部分:启动程序,内核与根文件系统,专业芯片驱动及应用。为了开发和升级的方便,我们将根文件系统作成 RAMDISK 的格式。据此我们将 FlashROM 分为 3 个区:1 Cramfs,2 Boot-Loader,3 Kernel+RAMDISK。本系统的启动程序用的是 DENX 的 U-boot-1.1.0。它的主要工作是初始化硬件,为加载操作系统准备必要的环境及其参数,同时可以与开发主机通讯,下载程序到 SDRAM 和 Flash。通过修改它的一些配置文件,可以完成对目标系统 Memory Controller, memory map 以及 I/O、PCI、Flash、Ethernet controller、Serial 等硬件设施的初始化。

对于芯片驱动程序的编写,一些通用设备驱动,如以太网卡驱动已经在内核中,对于系统的一些专业芯片的驱动由于其特殊性,将其和应用做成 Cramfs 文件系统格式,在目标板的操作系统启动时以 module 的形式进行加载,方便修改和升级。编写驱动程序可以按照 Linux 下编写驱动程序的规则来编写。编写的驱动程序应该具有以下功能:①对设备的初始化和释放;②数据从内核传到硬件和从硬件读取数据;③读取应用程序传递给设备文件的数据和回应应用程序请求的数据;④检测和处理设备出现的错误。

设备驱动程序的实质就是中断处理。Linux 中断处理程序分为上半部和下半部。上半部即一般的中断服务程序,由硬件中断触发,一般运行在关中断的方式下,应当尽可能的短小,处理尽可能的快;下半部运行在开中断和任务串行化的环境下,处理需要较长时间的任务。驱动程序上半部在处理完实时性很强的任务后,用 Queue-task 函数将下半部处理函数挂入立即队列,用 mark-bh 函数来激活立即队列,下半部就可以最优先的被执行。

2.2 应用程序的设计

应用程序的设计可以采用多线程或者多进程的方式。多线程的优点在于线程比进程要小,可以使应用更轻量,线程间通讯方便,缺点就在于由于线程使用同一个地址空间,如果一个线程出了问题,将可能影响到整个系统;多进程各自占有一份内存空间,因此可以增强系统的健壮性,但是多进程增加了系统的开销,同时进程间通讯较复杂。结合我们系统的实际,考虑到各个

(下转第 53 页)

```

//合成数据信息
...
//发送数据信息
WriteFile (pPara ->hcomm,pcdrq ->drq.Buffer,outputlong,
&writenlong,NULL);
}
DWORD WINAPI SerialComm::Completepro(LPVOID lpPara)
{
//取得当前数据请求结构体
pCurrentDrq temp = (CurrentDrq*) lpPara;
WaitForSingleObject(temp->timeoutevnet,MAXTIMEOUT);
//当前数据请求设定为非运行状态
temp->processing = false;
//调用完成例程
if(temp->drq.OnFinish != NULL)
temp->drq.OnFinish(*(hexdrq*)temp,temp->timeout,
temp->drq.CompletionContext);
return 0;
}

```

2.4 串口读取及协议响应

串口读取函数由串口读取接收到的数据,并将接收到的数据传递至协议处理函数 OnCMD 中按照通讯协议进行相应。当按照协议一次通讯成功完成后应调用数据请求完成函数 DrqComplete,将请求设定为非超时状态并设置超时事件完成本次通讯。

```

void SerialComm::ReadPort (phexParameter para,pCurrentDrq pcdrq)
{
DWORD nByteRead=0;
ReadFile (para ->hcomm,pcdrq ->inputbuff,MAXNUM,
&nByteRead,NULL);
if(nByteRead>0)OnCMD(para,pcdrq,nByteRead);
PurgeComm(para->hcomm,PURGE_RXCLEAR);
}
void SerialComm::OnCMD(phexParameter para,pCurrentDrq pcdrq,DWORD nread)
{

```

```

//对接收到的数据按照协议处理
...
//通讯成功后调用 DrqComplete 函数结束当前数据请求
DrqComplete(pcdrq);
}
void SerialComm::DrqComplete(pCurrentDrq pcdrq)
{
pcdrq->timeout=false;
SetEvent(pcdrq->timeoutevnet);
}

```

2.5 其他部分程序设计

开启关闭串口程序: 开启串口程序主要应用 CreateFile 取得串口句柄,SetupComm、GetCommState、SetCommState 等 API 函数设置串口,调用 SetCommTimeouts 设定读串口超时时间,否则在读串口函数中将持续等待串口数据,此外还应启动串口主线程开始获取数据请求。关闭串口程序完成主线程挂起及串口句柄的关闭工作。

构造函数及析构函数: 在构造函数中完成链表的初始化,创建主线程,创建同步事件。而析构函数则主要完成主线程关闭及链表的清空等清理操作。

3 结束语

针对不同通讯协议只需将数据请求等结构体稍作调整,按协议合成数据请求帧并添加协议处理程序即可实现与下位机的通信。该程序已应用于与 51 及 DSP 等试验设备的通讯系统中,通讯的实时性及效率较好,有一定推广价值。

参考文献

- [1] 龚建伟,熊光明.Visual C++/Turbo C 串口通讯编程实践[M].北京:电子工业出版社,2007
- [2] 张帆,石彩成.Windows 驱动开发技术详解[M].北京:电子工业出版社,2008

[收稿日期 2009.4.17]

(上接第 50 页)

线程通讯的重要性,我们采取多进程多线程的方式,在软件上增加一个与主进程并行的守护进程,在硬件上设置看门狗,以增强系统的健壮性。

3 远程控制与通信链路的建立

本设计采用 PPP 串行通信接口协议方式。PPP(Point-to-Point Protocol,点对点协议)协议中包含 3 个部分:在串行链路上封装 IP 数据报的方法;建立、配置及测试数据链路的链路控制协议(LCP);不同网络层协议的网络控制协议(NCP)。PPP 具有很多优势;支持循环冗余检测、支持通信双方进行 IP 地址动态协商、对 TCP 和 IP 报文进行压缩、认证协议支持(CHAP 和 PAP)等。

PPP 的实现可以通过 2 个后台任务来完成,协议控制任务和写任务。协议控制任务控制各种 PPP 的控制协议,包括 LCP、NCP、CHAP 和 PAP。它用来处理连接的建立、连接方式的协商、连接用户的认证以及连接中止。写任务用来控制 PPP 设备的数据发送。数据报的发送过程,就是通过写任务往串行接口设备写数据的过程,当有数据报准备就绪,PPP 驱动通过信号灯激活写任务,使之完成对串行接口设备的数据发送过程。PPP 接收端程序通过在串行通信设备驱动中加入“hook”程序来实现。在串行通信设备接收到 1 个数据之后,串行设备的中断服务程序(ISR)

调用 PPP 的 ISR。当 1 个正确的 PPP 数据帧接收之后,PPP 的 ISR 通过调度程序调用 PPP 输入程序,然后 PPP 输入程序从串行设备的数据缓存中将整个 PPP 数据帧读出,根据 PPP 的数据帧规则进行处理,也就是分别放入 IP 输入队列或者协议控制任务的输入队列。

4 结束语

DM642 具有高速的处理能力以及 S3C2510A 出色的对外接口能力,这就使其设计出的产品在图像质量、硬件成本、灵活性及产品更新等方面都优于专门的视频编解码芯片。在数据保存方面,系统不仅可以选择本地保存方式,还可以通过网络保存到服务器,或者通过 PCI 口将数据保存到主机中的硬盘上。这种易于扩展性使用户在实际使用过程中有了更多的选择。

参考文献

- [1] S3C2510 32-BIT RISC MICROPROCESSOR USER'S MANUAL. Samsung Electronics.2002
- [2] Texas Instruments Incorporated.TMS320DM642Video/Imaging Fixed-Point Digital Signal processor Data Manual (SPRS200) [S]
- [3] 孟柯,李凤亭,马惠敏.用于视频监控的便携式数字录像机设计[J].电视技术,2002(7)

[收稿日期:2009.4.22]