

基于 FPGA 的视频监控系统

邵应昭,任爱锋,初秀琴

(西安电子科技大学 电子工程学院,陕西 西安 710071)

摘要: 设计并实现了一个基于 FPGA 的视频监控系统。在 FPGA 中设计 I²C 总线配置模块对视频芯片 ADV7181 进行合理的配置,介绍了视频信号的处理过程,包括 ITUR656 视频解码、视频插值、解交织和色度空间变换。处理后的视频一路通过 VGA 接口在本地显示器上显示,另外一路经过压缩后通过网络传输到网络终端,从而实现网络远程监控。

关键词: FPGA;视频监控;Nios II;SOPC

Video monitor system based on FPGA

SHAO Ying Zhao,REN Ai Feng,CHU Xiu Qin

(School of Electronic Engineering, Xidian University, Xi'an 710071, China)

Abstract: A video monitoring system is designed based on FPGA. Firstly, an I²C-config module is designed to configure ADV7181 properly. Then the transaction processes of the video are introduced in detail, including the ITUR656 decode module, interpolation module, de-interlacing module and color space convert module. The processed video is divided into two paths. One path is displayed in CRT through the VGA interface, the other is transmitted to network using the SOPC system which takes the Nios II as core. So it can realize the function of network video monitor.

Key words: FPGA; Video Monitor; Nios II; SOPC

1 系统总体方案

基于 FPGA 的视频监控系统整体框图如图 1 所示。框图按照视频数据流的处理过程划分,各部分功能如下:

- (1) 视频采集部分:将摄像头采集进来的模拟视频数据转换为数字视频数据,并获取相应的视频控制信号。
- (2) 视频格式转换部分:将采集模块输出的视频数据转换成需要的视频格式,包括 ITUR656 解码、插值和色彩空间变换。
- (3) 视频压缩部分:由于视频数据量大,需要进行压缩,以利于传输。
- (4) 视频缓存部分:将压缩好的视频数据存储在外部的 SRAM 中。
- (5) 网络传输部分:将压缩好的视频数据传输到网络中。
- (6) VGA 视频显示部分:设计 VGA 时序,将摄像头采集进来的视频通过 VGA 接口在本地 CRT 显示器上显示。
- (7) Nios II 控制部分:当网络远端有视频监控请求时,Nios II 系统控制视频压缩、视频缓存和网络传输三部分



图 1 系统整体框图

协调工作,将压缩好的视频数据传输给网络远端。

2 视频采集与处理

2.1 ADV7181 解码过程

ADV7181 作为视频解码芯片,通过 SCLK、SDA 两根串行线,用 I²C 总线对 ADV7181 的寄存器进行配置,配置完成后,芯片对输入的模拟视频进行视频解码,输出符合 CCIR656 标准的 YCrCb 4:2:2 视频数据。

ADV7181 输出的同步时钟 LLC 为 27MHz。行同步信号(HS)一个周期包括 1 716 个时钟周期。其中,EAV 占 4 个时钟周期,HS Begin 和 HS End 占 2 个时钟周期,SAV 占 4 个时钟周期,有效数据占 1 440 个时钟周期,H BLANK 占 266 个时钟周期。

ADV7181 输出的视频信号包括奇场和偶场,第 1 行到第 263 行为第一场,其中从第 10 行到第 263 行为第

一场的有效信号;第 264 行到第 525 行为第二场,其中从第 273 行到第 525 行为第二场有效信号。

2.2 视频采集芯片 ADV7181 配置

2.2.1 ADV7181 控制寄存器

要使 ADV7181 工作,需要通过 I²C 总线来对其寄存器进行合理的配置。ADV7181 内部共有 250 个控制寄存器,可通过配置这些控制寄存器来设定和实现 ADV7181 的各种功能。本设计对 ADV7181 的功能要求是采集 NTSC 制式的模拟视频信号,以 27MHz 的时钟进行 A/D 转换,并输出 8 位宽的格式为 ITUR656 YCrCb 4:2:2 的数字信号。在此要求下,参照 ADV7181 数据手册,只需要对 ADV7181 的 40 个寄存器进行配置,其余的寄存器均为系统复位以后的缺省值。

2.2.2 I²C 程序设计

(1)I²C 控制器设计

先用 Verilog HDL 设计一个 I²C 控制器,实现 I²C 写数据。控制器每次传输 24 位数据,前 8 位是从设备地址 SLAVE_ADDR(0x40),接下来 8 位是从设备寄存器地址 SUB_ADDR,最后 8 位是数据。控制器使用 33 个时钟周期完成 1 次 24 位数据的传输。第 1 个时钟周期用于初始化控制器,第 2、3 个周期用于启动传输,第 4~30 周期用于传输数据(其中包括 24 位数据和 3 个 ACK),最后 3 个周期用于停止传输。

I²C 的工作频率不能太高,如果太高,则 ADV7181 接收到数据后来不及响应,从而导致不能正确判断 ACK。本系统的 I²C 总线的时钟频率采用 20kHz,由系统时钟 50MHz 分频得到。

(2)I²C 配置模块

系统用 Verilog HDL 设计一个配置模块,将要配置的寄存器数据存储在查找表 LUT_DATA 中,调用上面的 I²C 控制器模块,完成 ADV7181 的配置。每个寄存器配置分三步,并用 mSetup_ST 表示当前进行到哪一步。第一步准备数据,将 8 位从设备地址与存储在 LUT_DATA 中的寄存器地址合并为 24 位数据 mI²C_DATA,并将 mI²C_GO 设置为 1,启动 I²C 传输;第二步检测传输结束信号,如果检测到传输结束(mI²C_END=1),但 ACK 信号不正常,重新发送数据;如果检测到传输结束且 ACK 信号正常,则进入第三步,将寄存器索引加 1,准备下一个数据的传输。

2.3 视频数据处理

ADV7181 经寄存器配置后,其输出包括:8 位与 CCIR656 标准兼容的 YCrCb 4:2:2 视频数据、行同步信号 TD_HS、场同步信号 TD_VS 以及 27MHz 行锁定时钟。FPGA 获取这些数据后进行相关的后续处理。

2.3.1 ITUR656 解码

ITUR656 解码主要完成以下工作:

- (1)串行数据转化为并行数据;
- (2)对 YCrCb(4:2:2)视频数据进行插值,得到 YCrCb(4:4:4)视频数据;

(3)分频得到 13.5MHz 的像素时钟。

实现步骤:

第 1 步:根据 SAV 字节,判断有效数据的到来。

由 ITUR656 标准可知,当连续三个字节为 FF0000,而第四个字节的第四位为 0 时,接着到来的就是有效数据。本系统使用移位寄存器 R1、R2、R3,将连续的三个字节数据存放在 R1、R2、R3 中。当 R3、R2 和 R1 的值为 FF0000,即可知为 SAV 或 EAV 的到来,接着判断第四个字节数据的第四位是否为 0,如为 0 即表示有效数据开始(START 信号)。

第 2 步:插值,得到 YCrCb(4:4:4)视频数据和 13.5MHz 的像素时钟。

一行 4:2:2 的 YCrCb 的视频数据中,以 27MHz 时钟采样,则 Y 有 720 个,Cb、Cr 各 360 个。也即 Y 的采样时钟为 13.5MHz,Cb、Cr 的采样时钟为 6.75MHz。本文设计中,在 4 个 27MHz CLOCK 中 Y(代码中的 YY)被改变了 2 次,而 Cb、Cr(代码中的 CCB、CCR)被赋值一次。读的时候读两次就变 4:4:4,而像素时钟(代码中的 Ypix_CLOCK)为 13.5MHz。

第 3 步:由于视频采集芯片输出的数据是隔行扫描的,因此,需要对获取到的 YCrCb(4:4:4)进行解交织处理。

本系统中将一行数据读取两次,变成两行,这样,一场就变成两场,从而实现解交织。为将一行变成两行,本系统将 TD_HS 进行倍频,生成 HSX2。倍频的实现是通过在 TD_HS 的起始和中间插入两个低脉冲来实现的。

生成 HSX2 后,将(4:4:4)的 Y、Cb、Cr 以 13.5MHz 的时钟分别写入到三个 1KB 的双口 RAM 中,同时,以 27MHz 的时钟读取出来。这样,就相当于重复一行的数据,一行变成两行了。

2.3.2 色度空间变换

色彩空间变换公式如式(1)所示:

$$\begin{aligned} R &= 1.164(Y-16)+1.596(Cr-128) \\ G &= 1.164(Y-16)-0.813(Cr-128)-0.392(Cb-128) \\ B &= 1.164(Y-16)+2.017(Cb-128) \end{aligned} \quad (1)$$

由式(1)可知,色度空间变换涉及到小数运算。在 HDL 描述语言中,对于小数的运算,如果采取行为描述,通常是将小数乘以一个数 A 转化为整数,所乘的这个数 A 必须取 2 的幂次。这样,计算完成后,将结果除以这个数 A。由于 A 为 2 的整数次幂,所以除以 A 就可转换为对被除数的移位操作。

可以将色彩空间变换公式变换成如下式(2)形式:

$$\begin{aligned} R_{int} &= 1024 \times R = [1.164(Y-16)+1.596(Cr-128)] \times 1024 \\ G_{int} &= 1024 \times G = [1.164(Y-16)-0.813(Cr-128)- \\ &\quad 0.392(Cb-128)] \times 1024 \end{aligned} \quad (2)$$

$$B_{int} = 1024 \times B = [1.164(Y-16)+2.017(Cb-128)] \times 1024$$

这样,设计程序计算出公式(2)的 R_{int}、G_{int}、B_{int}。最后将计算结果 R_{int}、G_{int}、B_{int} 均除以 1024,即右移 10 位,即可得到 R、G、B 的值。

3 视频 VGA 显示及网络传输

3.1 视频的 VGA 显示

本系统使用的视频显示器分辨率为 640×480, 刷新率 60Hz 模式, 像素时钟为 25MHz 模式。VGA 水平时序中, 每行包括 800 像素点, 其中 640 点为有效显示区, 160 点为行消隐区(非显示区)。行消隐区包括: 行同步信号 HS 每行有一个脉冲, 该脉冲的低电平宽度为 96 个脉冲, 后沿 45 个脉冲, 前沿 13 个脉冲。另外, 有效时间包括 6 个列过扫描边界列, 在后沿和前沿中各加 3 列; VGA 垂直时序中, 每场有 525 行, 其中 480 行为有效显示行, 45 行为场消隐区。场消隐区包括: 场同步信号每场有一个脉冲, 该脉冲的低电平宽度为 2 行, 后沿 30 行, 前沿 9 行; 有效时间包括的 4 行过扫描边界行, 后沿和前沿各加 2 行。

应该注意的是, 在上电期间, ADV7181 的输出信号不是稳定的。所以, 需要设计一个稳定检测模块, 即将 ADV7181 的输出延迟一段时间, 待 ADV7181 输出稳定后, 再做 VGA 的显示。

3.2 网络传输的平台

构建好片上系统后, 网络传输部分工作原理如图 2 所示。

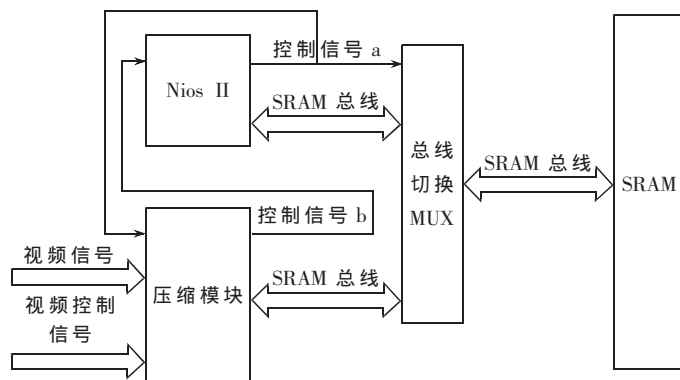


图 2 网络传输部分工作原理

图 2 中网络传输部分工作过程为: 系统上电后, Nios II 等待网络远端的视频请求。当接收到远端请求后, Nios II 通过 IO 口发出控制信号 a 给视频压缩模块。视频压缩模块在收到 Nios II 的控制信号 a 后, 压缩当前视频信号的一帧。同时, 总线切换模块也在控制信号 a 下, 将压缩模块输出的数据总线连接到 SRAM 总线上。这样, 就将压缩好的一帧视频数据存储到 SRAM 中。视频压缩模块在压缩完一帧图像后, 发送控制信号 b 给 Nios II, Nios II 在发送控制信号 a 后, 就一直侦听压缩模块的控制信号 b, 一旦接收到压缩模块的控制信号 b, 马上改变控制信号 a, 压缩模块收到控制信号 a 时, 停止压缩, 同时, 总线切换模块将 SRAM 的总线切换与 Nios II 模块相连。这样, Nios II 就可以将压缩好的图像数据通过网络传输出去。一旦传输完成后, 又给出控制信号 a, 进行下一帧的压缩, 一直反复进行下去。

(1) 压缩模块

由于视频数据量非常大, 而网络的带宽是有限的, 所以, 需要对视频图像进行压缩。本系统只是采用简单的压缩算法, 将一帧图像数据隔一个像素点取一个像素点, 隔一行视频数据取一行视频数据, 这样可以将一帧图像压缩为原来的 1/4。

需要注意的是, 压缩模块在收到 Nios II 的控制信号后, 开始压缩。但此时可能并不是一帧图像的开始。所以, 需要等待当前帧完成后, 再开始压缩到来的完整帧。

(2) 总线切换模块

该模块很简单, 只需要设计一个选择器, 控制信号 a 作为选择器的使能信号, 将 SRAM 总线在压缩模块和 Nios II 模块之间切换。

(3) Nios II 网络传输软件

构建好各个模块后, 在 Nios II 软件开发环境 IDE 下进行网络传输软件的开发。在本系统中, 需要加载 $\mu\text{C}/\text{OS-II}$ 操作系统和 LWIP 网络协议栈。Nios II 软件开发环境 IDE 非常简单、方便, 只需要在创建工程时勾选上相应的选项即可。

4 接收端显示程序设计

客户端视频显示应用程序利用 LabWindows/CVI 软件开发, 此应用程序利用套接字编程, 它作为客户机与视频监控系统进行通信。

接收端主要完成以下几个功能:

(1) 开始接收视频: 向系统发送视频请求, 发送完后, 等待视频数据的到来并且在实时窗口中显示接收到的视频。

(2) 保存视频: 将接收到的数据保存在硬盘上。

(3) 停止视频保存: 停止保存接收的视频。

(4) 回放视频: 当停止视频保存后, 从硬盘上读取原来保存过的视频文件, 在回放窗口中显示。

(5) 退出: 停止视频请求, 关闭软件。

本文实现的视频监控系统具有以下特点: 首先, 采用单片 FPGA 加外围电路构建系统, 系统设计简单, 成本低。其次, 对于视频的处理, 用快速、高效的硬件描述语言实现, 处理效率高, 而对于网络传输等复杂高层次的功能则采用嵌入软核处理器实现。二者结合开发, 可使系统开发成本降低, 设计周期缩短。

参考文献

- [1] 刘富强. 数字视频监控系统开发与应用. 北京: 机械工业出版社, 2003.
- [2] 王建, 李方伟. 视频监控系统的现状和发展趋势. 单片机与嵌入式系统应用, 2006, (7): 25-27.
- [3] 孙景琪. 视频技术与应用 电视原理、遥控系统、电视广播系统. 北京: 北京工业大学出版社, 2004.
- [4] 徐光辉, 程东旭, 黄如, 等. 基于 FPGA 的嵌入式开发与应用. 北京: 电子工业出版社, 2006.
- [5] 任爱锋, 初秀琴, 常存, 等. 基于 FPGA 的嵌入式系统设计. 西安: 西安电子科技大学出版社, 2004.