

基于 FPGA 的视频监控系统

作者：邵应昭 导师：任爱锋
(西安电子科技大学工程学院, 西安, 710071)

摘要: 随着网络、通信、多媒体技术以及微电子技术的迅速发展, 视频监控系统得到了越来越广泛的应用。

本文设计并实现了一个基于 FPGA 的视频监控系统。论文首先在 FPGA 中设计 I2C 总线配置模块对视频 ADC 芯片 ADV7181 进行合理的配置, 然后详细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。经过处理后的视频, 分成两路。一路通过 VGA 接口, 按照设计好的 VGA 时序, 在本地显示器上显示视频, 从而实现本地视频监控, 且给出了本地视频监控显示效果图; 另外一路经过压缩后在 SRAM 中缓存, 利用 Nios II 软核为核心的 SOPC 控制, 将缓存中压缩好的视频数据通过网络传输到网络终端, 从而实现网络远程监控。论文的最后, 简单介绍了 LabWindows 软件开发环境, 用 LabWindows 软件开发了接收端的显示程序, 并给出了接收端远程视频监控显示效果图。

本视频监控系统在局域网内进行了测试验证, 结果表明, 系统能实现本地视频监控和网络视频监控的功能。

关键词: FPGA 视频监控 Nios II SOPC

Video monitor system based on FPGA

Abstract: As the rapid development of the network, communication, medium and the technique of micro-electronics, the Video Monitoring system is applied more and more widely.

This paper designs and realizes a Video Monitoring system based on FPGA. Firstly, the paper designs a I2C-config module to configure the ADC chip ADV7181 properly. Then introduces the transaction process of the video 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 that the video passes through the VGA-timing block and is displayed in CRT through the VGA interface. So it can realize the function of local-area video monitor and also has given out the result displayed in local-area. The other path is that the video passes through the compress module and is saved in SRAM. Then the compressed video in

the SRAM 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. In the end, paper gives briefly introduction of the software LabWindows and designs the displaying software of the network receiver, and also gives out the result displayed in the network receiver.

This Video Monitoring system has been tested and verified in the Lan. The result shows that the system can realize the function of local-area and network video monitor.

Key words: FPGA Video Monitor Nios II SOPC

0 引言

随着网络带宽、计算机处理能力和存储设备的迅速提高,以及各种实用视频信息处理技术的出现,视频监控进入了全数字化的网络时代,称为第三代视频监控系统,即全数字视频监控系统或网络数字视频监控。第三代视频监控系统以网络为依托,以数字视频的压缩、存储、播放为核心,以智能实用的图像分析为特色,引发了视频监控行业的技术革命。

第三代视频监控系统使监控达到一个新的高度,其具有良好的成本优势和健壮的可扩展性。利用现有的网络资源,不需要为新建监控系统铺设光缆、增加设备,轻而易举地实现远程视频监控;系统扩展能力强,只要有网络的地方增加监控点设备就可以扩展新的监控点;维护费用低、系统功能强大、利用灵活、全数字化保存和检索。在网络中的每一台计算机,只要安装了客户端软件,给予相应的权限就可成为监控工作站。

2 系统方案设计

2.1 视频信号的制式[4]

目前,世界上主要有三种不同的电视系统:NTSC制式、PAL制式和SECAM制式。而在视频监控系统中,选用不同制式的摄像头,后续的处理也会相应地不同。

2.2 色彩空间模型

对于数字电子多媒体领域来说,我们经常接触到的色彩空间的概念,主要是RGB, YUV这两种。

(1) YUV模型

YUV模型,实际上很多时候,我们是把它和YIQ/YCrCb模型混为一谈的。

实际上，YUV 模型用于 PAL 制式的电视系统，Y 表示亮度，UV 并非任何单词的缩写；YIQ 模型与 YUV 模型类似，用于 NTSC 制式的电视系统。YCbCr 颜色空间是由 YUV 颜色空间派生的一种颜色空间，主要用于数字电视系统中。

常用到的 YUV 格式有以下几种：

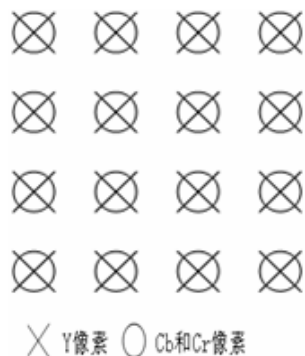


图 2.1 YUV 4:4:4 样例位置

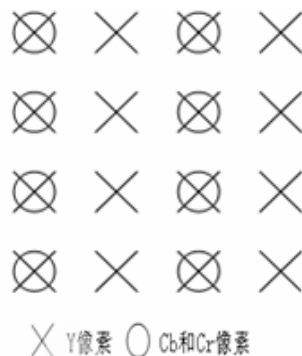


图 2.2 YUV 4:2:2 样例位置



图 2.3 YUV 4:1:1 样例位置



图 2.4 YUV 4:2:0 样例位置

(2) RGB 模型

RGB 色彩模式是工业界的一种颜色标准，是通过对红(R)、绿(G)、蓝(B)三个颜色通道的变化以及它们相互之间的叠加来得到各式各样的颜色的，RGB 即是代表红、绿、蓝三个通道的颜色，这个标准几乎包括了人类视力所能感知的所有颜色，是目前运用最广的颜色系统之一。

(3) 色彩空间的变换

在视频处理中，经常需要将视频数据在这两种色彩空间之间转换。转换关系如式 (2-1)：

$$\begin{bmatrix} Y \\ Cr \\ Cb \end{bmatrix} = \begin{bmatrix} 0.299 & 0.588 & 0.114 \\ 0.500 & -0.4187 & -0.0813 \\ -0.1687 & -0.3313 & 0.500 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} \quad \text{式 (2-1)}$$

2.3 系统总体方案

本文实现的视频监控系统是基于 FPGA 的。主要分为两部分：第一部分，利用硬件描述语言实现视频采集，视频 VGA 显示，视频压缩，视频缓冲存储。第二部分，在 FPGA 中嵌入 Nios II 软核，通过 Nios II 软核控制，将缓冲区的压缩

好的视频数据传输到网络上。

2.3.1 系统功能框图

系统整体框图如图 2.5 所示：

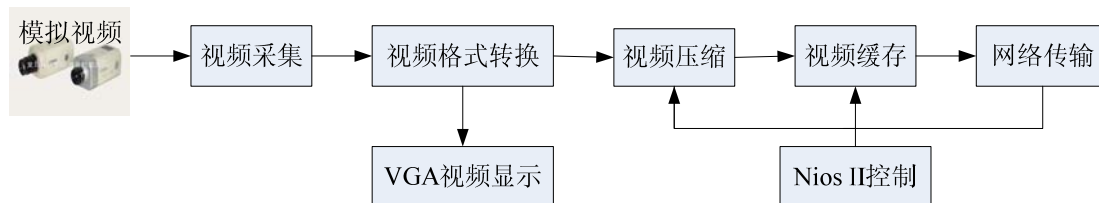


图 2.5 系统整体框图

上述框图是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 控制系统控制视频压缩、视频缓存和网络传输三部分协调工作，将压缩好的视频数据传输给网络远端。

2.3.2 系统硬件结构图

系统硬件结构如图 2.6 所示：

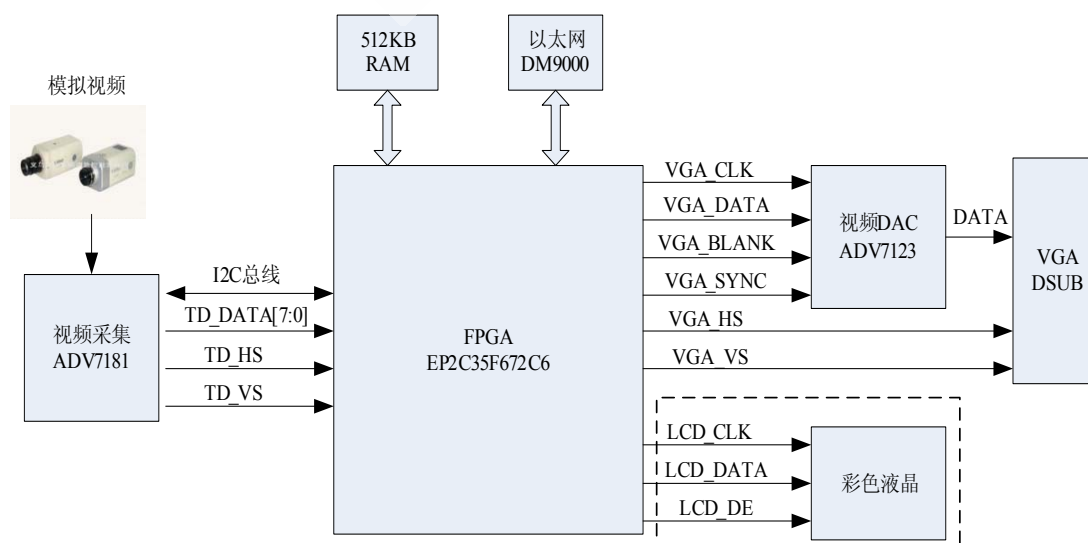


图 2.6 系统硬件结构图

3 视频采集与视频处理

3.1 视频采集芯片ADV7181 功能特性

3.1.1 ADV7181 解码过程

ADV7181 的视频解码过程框图如图 3.1 所示。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。

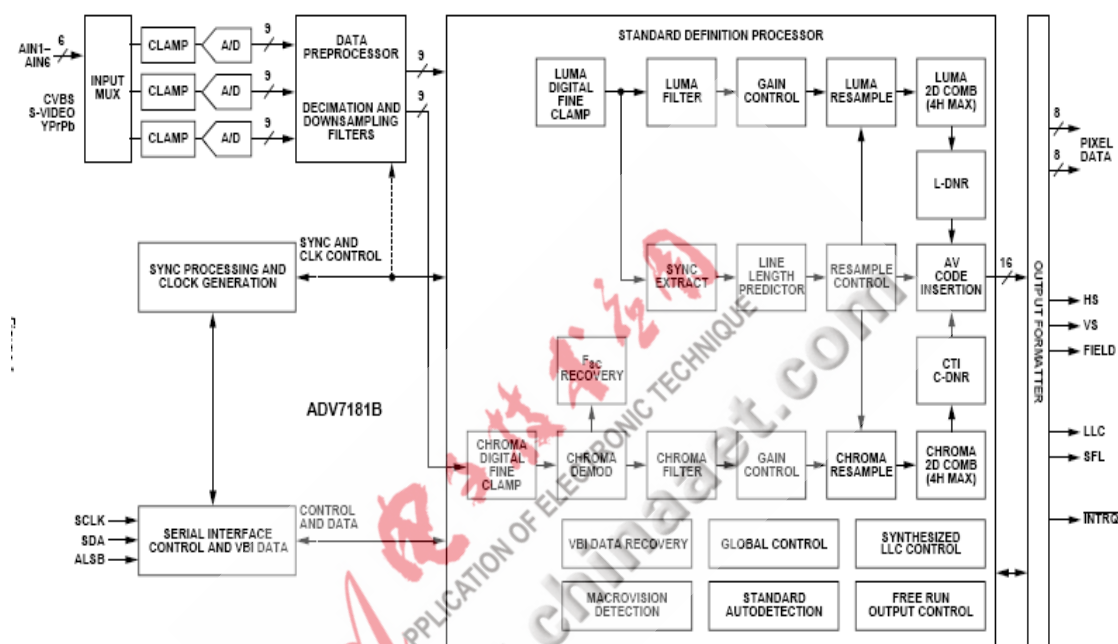


图 3.1 ADV7181 的视频解码过程框图

3.1.3 ADV7181 典型时序

ADV7181 对视频信号进行采样解码后, 得到与 CCIR656 标准兼容的 YCrCb 4:2:2 格式的输出编码。要对 YCrCb 4:2:2 格式的数字图像数据进行后续处理, 就需要了解 ADV7181 输出信号的时序关系。

(1) 图 3.2 为 NTSC 制式摄像头输入, ADV7181 输出的同步时钟 LLC 与数据信号及行同步信号之间的时序关系图。

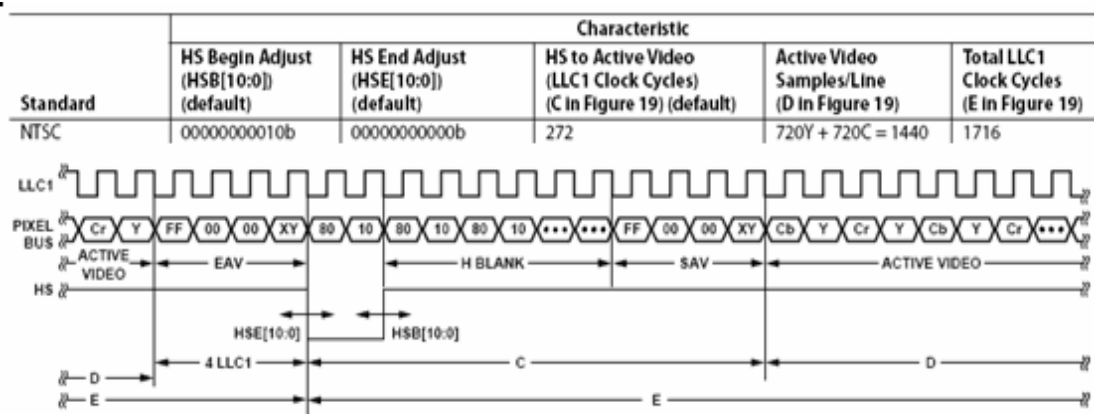


图 3.2 LLC 与数据信号的时序关系及行同步信号时序关系

其中，输出的同步时钟 LLC 为 27MHz。行同步信号（HS）一个周期包括 1716（E）个时钟周期。其中，EAV 占 4 个时钟周期、HS Begin 和 HS End 间占 2 个时钟周期、SAV 占 4 个时钟周期、有效数据占 1440 个时钟周期、H BLANK 占 266 个时钟周期。

(2) 图 3.3 为 NTSC 制式摄像头输入，ADV7181 输出的数据信号与行、场同步信号之间的时序关系图。

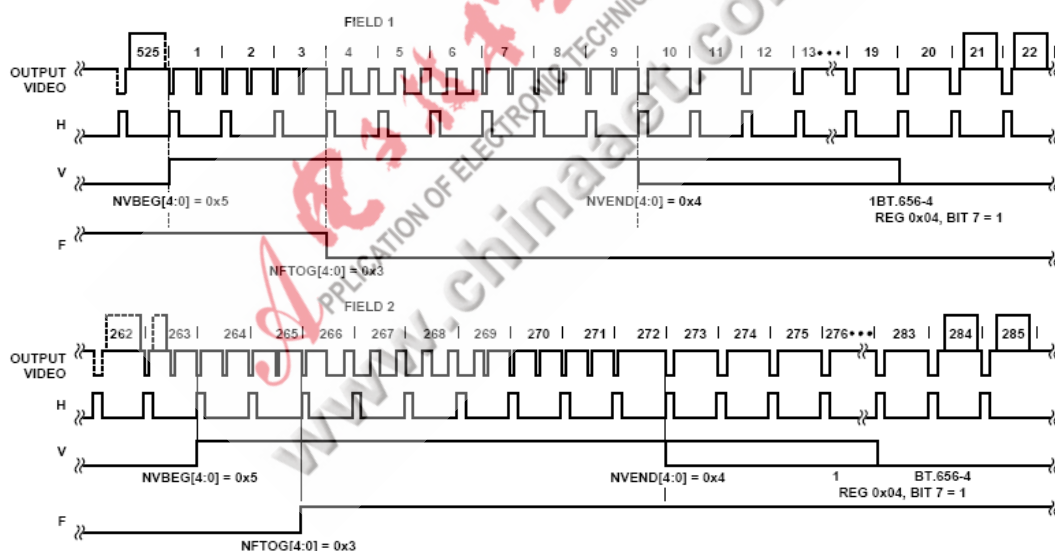


图 3.3 数据信号和行、场同步信号之间时序关系

由图可知，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 10 行到 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 273 行到 525 行为第二场有效信号。

3.2 视频采集芯片 ADV7181 配置

3.2.1 ADV7181 控制寄存器

要使 ADV7181 工作，需要通过 I2C 总线来对它的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。这 40 个寄存器的地址和配置参数值如表 3.1 所示。

表 3.1 寄存器的地址和配置参数值

寄存器地址 (H)	00	04	08	0A	0E	0F	10	11	15	17
参数值(H)	50	02	80	00	20	00	00	00	00	41
寄存器地址 (H)	2B	2C	2D	2E	2F	30	31	32	33	37
参数值(H)	00	8C	F2	EE	F4	D2	12	81	84	A0
寄存器地址 (H)	3A	50	51	52	53	54	58	77	7C	7D
参数值(H)	16	04	20	18	00	00	ED	C5	93	00
寄存器地址 (H)	C3	C4	D0	D5	D7	E4	E5	E6	E7	EA
参数值(H)	05	80	48	A0	EA	3E	80	03	85	0F

3.2.2 I2C程序设计

按照 3.2.2 节 I2C 总线的特点,在 FPGA 中设计 I2C 配置模块,对 3.2.1 节所述 ADV7181 的 40 个寄存器进行配置。I2C 总线完整的数据传输时序如图 3.7 所

示。

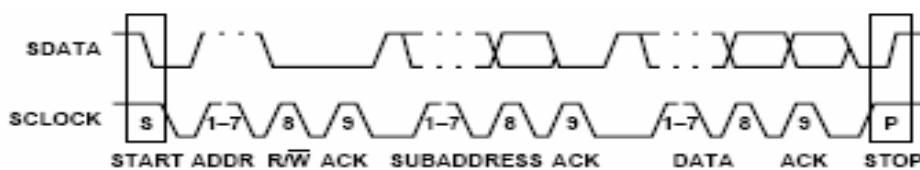


图 3.7 I2C 总线完整的数据传输时序

(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。控制器中，使用 33 个时钟周期完成 1 次传输 24 位数据。第一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，会导致 ADV7181 接收到数据后来不及响应，导致不能正确判断 ACK。本系统的 I2C 总线的时钟采用 20KHz，是由系统时钟 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中 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 位从地址设备地址与存储在 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。

图 3.8 为 I2C 配置模块在 QuartusII 中的顶层框图。设计好后，只需要将 OSC_50、I2C_SCLK 和 I2C_SDAT 锁定到对应的引脚上去，该模块即可对 ADV7181 完成配置。

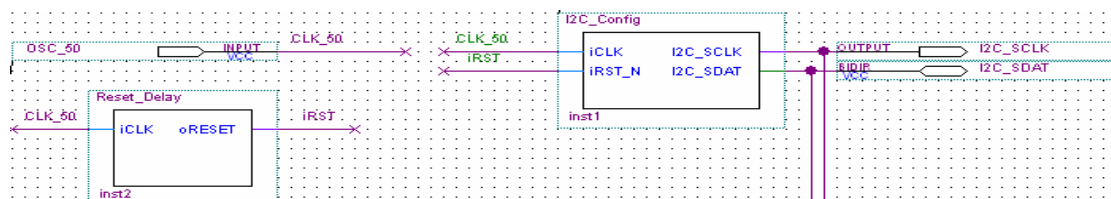


图 3.8 I2C 配置顶层框图

3.3 视频数据处理

视频采集芯片 ADV7181 经 3.2 节寄存器配置后,从视频采集芯片输出有: 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。

3.3.1 获取的视频流格式介绍

视频采集芯片输出的数据为 8 位与 CCIR656 标准兼容的 YCrCb4:2:2 视频数据。

CCIR656 标准的视频流特点:

- (1) 8 位或 10 位的串行数据;
- (2) 行、场同步信号嵌入在数据流中,并不需要单独的行、场同步线;
- (3) SAV CODE (start of active video) 后,有效视频数据开始,并且以 Cb 开始,一直接着 CbYCrYCbYCr 延续下去;
- (4) 27MHz 采样时钟。

CCIR656 标准对于 525 行 / 60 场和 625 行 / 50 场的视频流格式略有不同,本系统输出的是 525 行 / 60 场,它的行数据流格式如图 3.9 所示,垂直奇偶场消隐如图 3.10 所示。

从图 3.9 可知,525/60 场的 CCIR656 标准视频流格式中,一行包括 1716 个时钟周期。其中,SAV CODE (start of active video) 和 EAV CODE (end of active video) 各占 4 个时钟周期,BLANKING 占 268 个时钟周期,有效视频占 1440 个时钟周期。

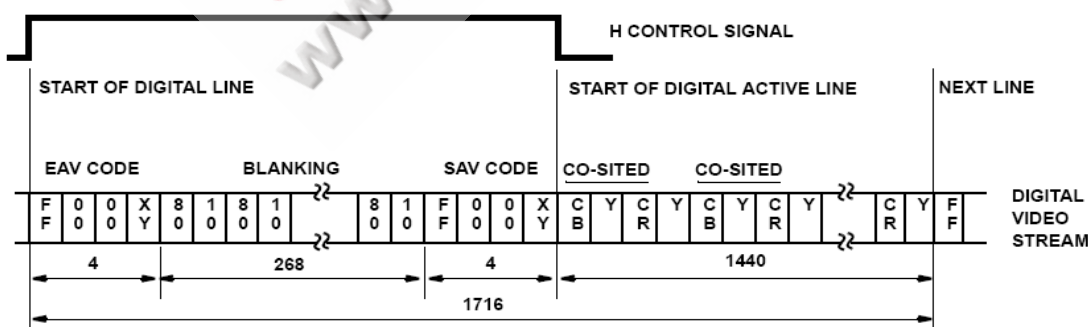


图 3.9 525/60 场的 CCIR656 标准的行数据流格式

SAV CODE 为有效视频数据开始标志, EAV CODE 为有效数据结束标志。并且 SAV CODE 和 EAV CODE 的判断取决于字节 XY。XY 的各位如表 3.2 所示。

表 3.2 状态各位表示的状态

数据位	D7	D6	D5	D4	D3	D2	D1	D0
状态	1	F	V	H	P3	P2	P1	P0

其中，F=0 为 1 场，F=1 为 2 场；V=1 为垂直消隐；H =0 为 SAV；H=1 为 EAV；P3~P0 是保留位， $P3=V \oplus H$ ， $P2=F \oplus H$ ， $P1=F \oplus V$ ， $P0=F \oplus V \oplus H$ 。

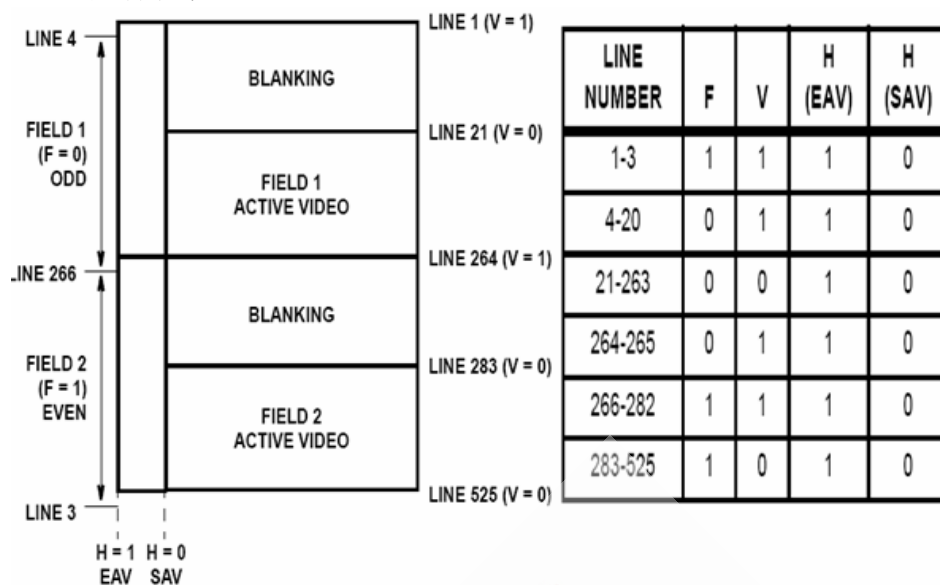


图 3.10 525/60 场的 CCIR656 标准垂直奇偶场消隐图

3.3.2 ITUR 656 解码

由 3.3.1 节，视频采集芯片输出进入 FPGA 的是 CCIR656 标准的串行 YCrCb (4:2:2) 视频数据。在 FPGA 中，我们对视频数据进行 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) 并对 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。

ITUR656 解码主要分两个大部分，一部分为时序设计，另一部分为结果仿真部分。

一 时序设计部分

第 1 步：根据 SAV 字节，判断有效数据的到来。

由图 3.10 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了。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，接着的判断第四个字节的数据的第四位为 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。

由 3.3.1 节知，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 6.25MHz。我们设计，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：由于从视频采集芯片出来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。

本系统中并未从传统意义上的解交织，而是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。要将一行变成两行，我们将 TD_HS 进行倍频，生成 HSX2。本系统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。

生成了 HSX2 后，将 (4:4:4) 的 Y, Cb, Cr 以 13.5MHz 的时钟分别写入到三个 1K 字节的双口 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。

二 仿真结果

上述时序设计，在 Quartus II 下进行了功能仿真。而对于 HSX2，用逻辑分析仪进行了观察。图 3.11 为 SAV 有效数据开始判断的仿真图。图 3.12 为 TD_HS 和 HSX2 时序关系图。

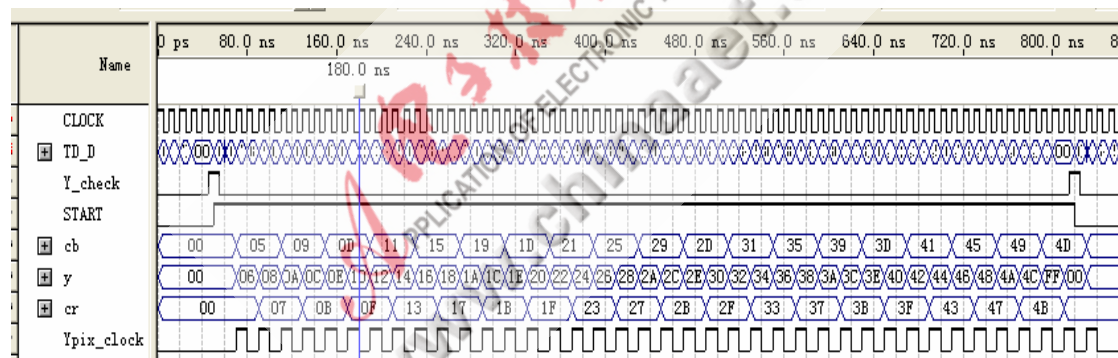


图 3.11 SAV 有效数据开始判断的仿真图

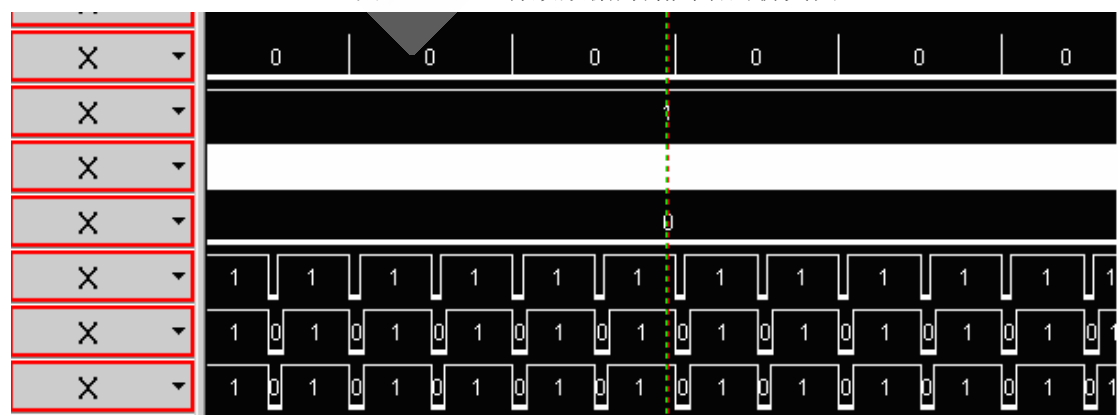


图 3.12 TD_HS 和 HSX2 时序图

3.3.2 色度空间变换

色彩空间变换公式为：

$$R = 1.164(Y-16) + 1.596(Cr-128)$$

$$G = 1.164(Y-16) - 0.813(Cr-128) - 0.392(Cb-128)$$

$$B = 1.164(Y-16) + 2.017(Cb-128)$$

由上面的公式可知，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 就可转换为将被除数的移位操作实现。

我们可以将色彩空间变换公式变换成如下：

$$R_int = 1024 \times R = [1.164(Y-16) + 1.596(Cr-128)] \times 1024$$

$$G_int = 1024 \times G = [1.164(Y-16) - 0.813(Cr-128) - 0.392(Cb-128)] \times 1024 \quad \text{式(2-3)}$$

$$B_int = 1024 \times B = [1.164(Y-16) + 2.017(Cb-128)] \times 1024$$

这样，设计程序计算出公式 (2-3) 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。

4 视频VGA显示及网络传输

4.1 视频的VGA显示

4.1.1 VGA典型时序

在 VGA 中，水平同步脉冲在光栅扫描线需要回到水平开始位置也就是屏幕的左边的时候插入，垂直同步脉冲在光栅扫描线需要回到垂直开始位置也就是屏幕的上方的时候插入。复合同步脉冲是水平同步脉冲与垂直同步信号的组合。RGB 为像素数据，在没有图像投射到屏幕时插入消隐信号，当消隐有效时，RGB 信号无效。

VGA 的典型时序如图 4.3 所示。其中上面部分为水平时序，下面部分是垂直场时序。

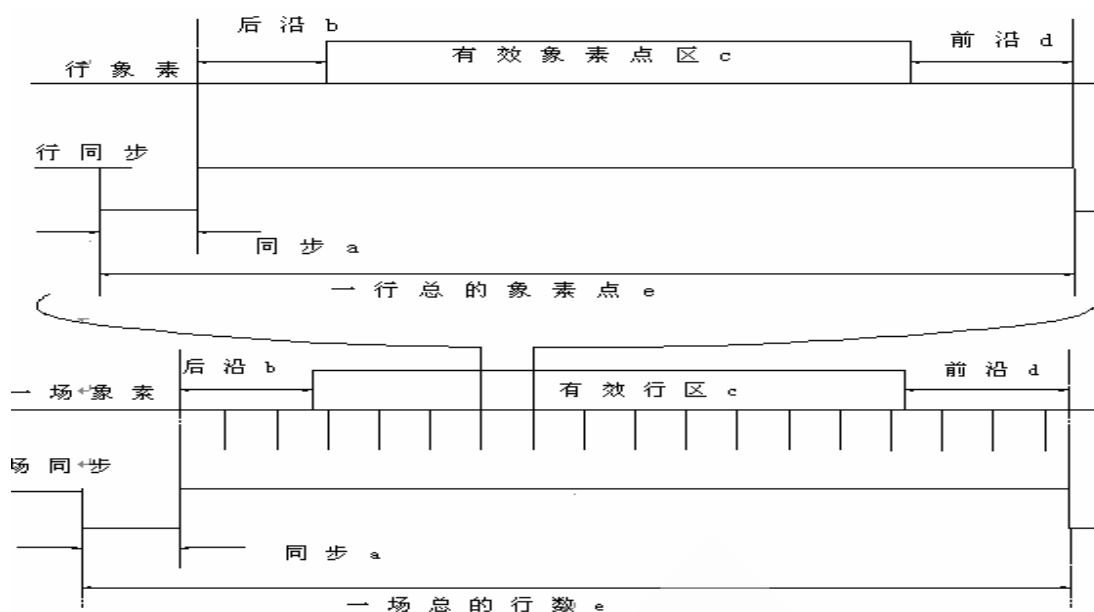


图 4.3 VGA 的典型时序图

4.1.3 VGA时序设计

本系统使用的是 640×480 ，刷新率 60Hz 模式，像素时钟为 25MHz 模式。由 4.1.2 可知，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即 96 个脉冲，后沿 45 个脉冲，前沿 13 个脉冲。另外，有效时间包括的 6 个列过扫描边界列，在后沿和前沿中各加 3 个；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，我们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。

仿真结果：

在 Quartus II 中对 VGA 时序进行了仿真，仿真结果如图 4.4 所示。由图中的仿真结果我们可以看出，行同步信号和场同步信号符合时序要求。

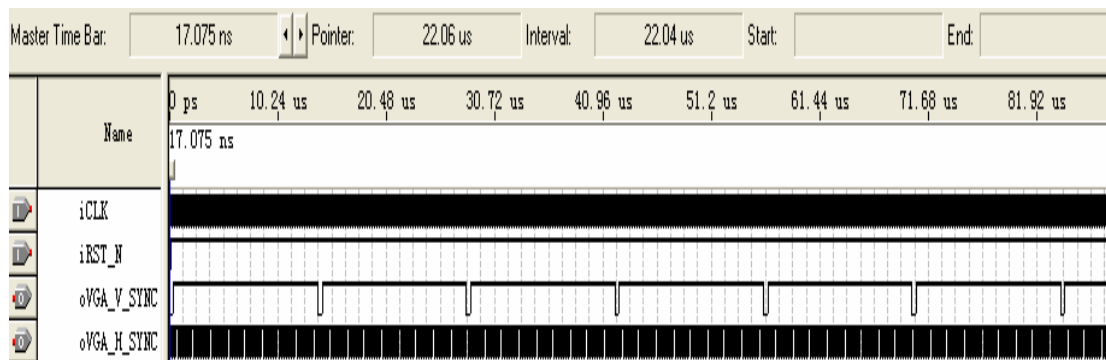


图 4.4 VGA 时序仿真图

4.1.4 VGA监控效果

系统外接 NTSC 制式摄像头，上电后立即采集视频数据。视频数据经过上面章节阐述的处理，通过 VGA 接口输出给显示器。在显示器上显示的结果如图 4.5 所示。



图 4.5 VGA 监控效果图

4.2 网络传输的平台

4.2.1 视频数据网路传输设计

一 构建片上系统

如图 4.7 所示，在 SOPC builder 中，添加 CPU、SRAM 接口、SDRAM 接口、FLASH 接口、LCD 接口、JTAG 接口和网络控制器接口 DM9000A 构建一个片上系统。

系统中各部分作用：

- (1) SRAM：用于缓存压缩好的视频数据；
- (2) SDRAM：程序存储区；
- (3) FLASH：固化程序；

- (4) DM9000A: 用户自定义网络组件, 用于网络传输;
- (5) LCD_16207: 用于显示系统设置的 IP 地址;
- (6) PIO: 用于控制压缩视频与传输视频之间。

Use	Module Name	Description	Input Clock	Base	End	IRQ
<input checked="" type="checkbox"/>	cpu_0	Nios II Processor - Altera Corpo...	clk	0x00480000	0x004807FF	
<input checked="" type="checkbox"/>	tri_state_bridge_0	Avalon Tristate Bridge	clk			
<input checked="" type="checkbox"/>	jtag_uart_0	JTAG UART	clk	0x00480840	0x00480847	0
<input checked="" type="checkbox"/>	sdrām_0	SDRAM Controller	clk	0x00800000	0x00FFFFFF	
<input checked="" type="checkbox"/>	SRAM	SRAM_16Bits_512K	clk	0x00400000	0x0047FFFF	
<input checked="" type="checkbox"/>	cfi_flash_0	Flash Memory (Common Flash Inter...		0x000000...	0x003FFFFFF	
<input checked="" type="checkbox"/>	DM9000A	DM9000A	clk	0x00480848	0x0048084F	1
<input checked="" type="checkbox"/>	pio	PIO (Parallel I/O)	clk	0x00480820	0x0048082F	2
<input checked="" type="checkbox"/>	out_pio	PIO (Parallel I/O)	clk	0x00480830	0x0048083F	
<input checked="" type="checkbox"/>	sysid	System ID Peripheral	clk	0x00480850	0x00480857	
<input checked="" type="checkbox"/>	timer_0	Interval timer	clk	0x00480800	0x0048081F	3
<input checked="" type="checkbox"/>	lcd_16207_0	Character LCD (16x2, Optrex 16207)	clk	0x00480860	0x0048086F	

图 4.7 片上系统包含的接口

在 SOPC Builder 中, 添加上述的接口, 生成顶层框图块如图 4.8 所示。

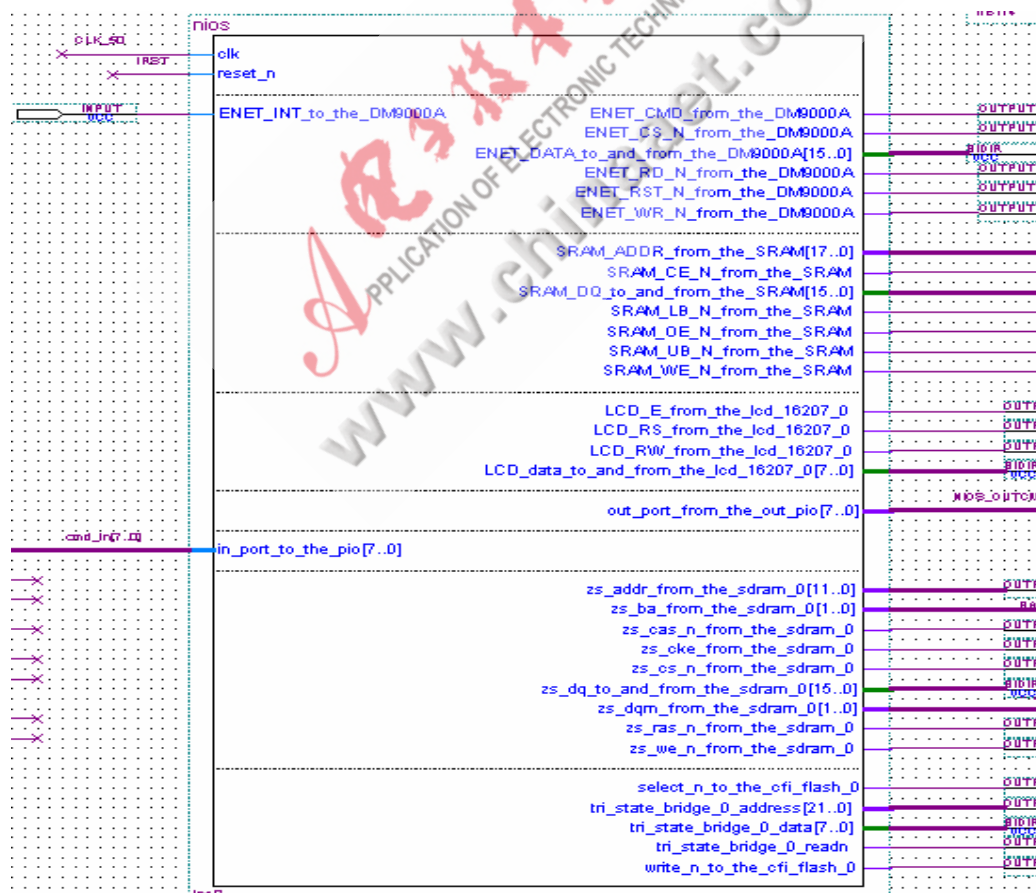


图 4.8 片上系统的顶层框图

二 工作原理

构建好片上系统后, 网络传输部分工作原理如图 4.9 所示。

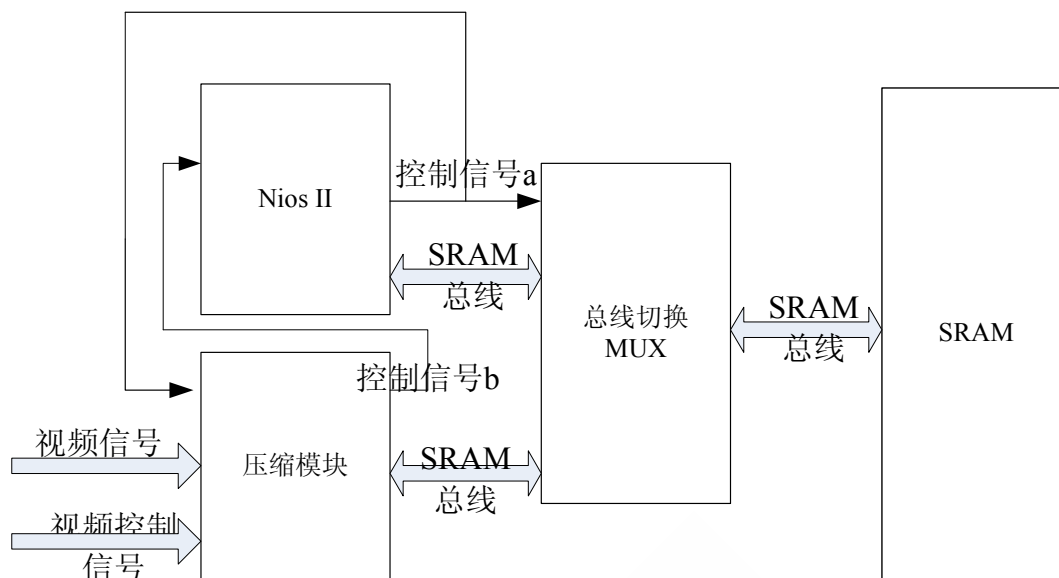


图 4.9 网络传输部分工作原理

如图 4.9 所示，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 总线上去。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。

三 各模块介绍

(1) 压缩模块部分

由于视频数据量非常的大，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样视频数据，这样可以将一帧图像压缩为原来的 1/4。

注意的一点，压缩模块在收取到 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) 总线切换模块部分

这个模块很简单，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 网络传输软件部分

构建好各个模块后，在 Nios II 软件开发环境 IDE 下进行网络传输的软件开发。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 下非常的简单、方便，只需要在创建工程的时候勾上相应的选项就可以了。

网络传输软件流程如图 4.10 所示：

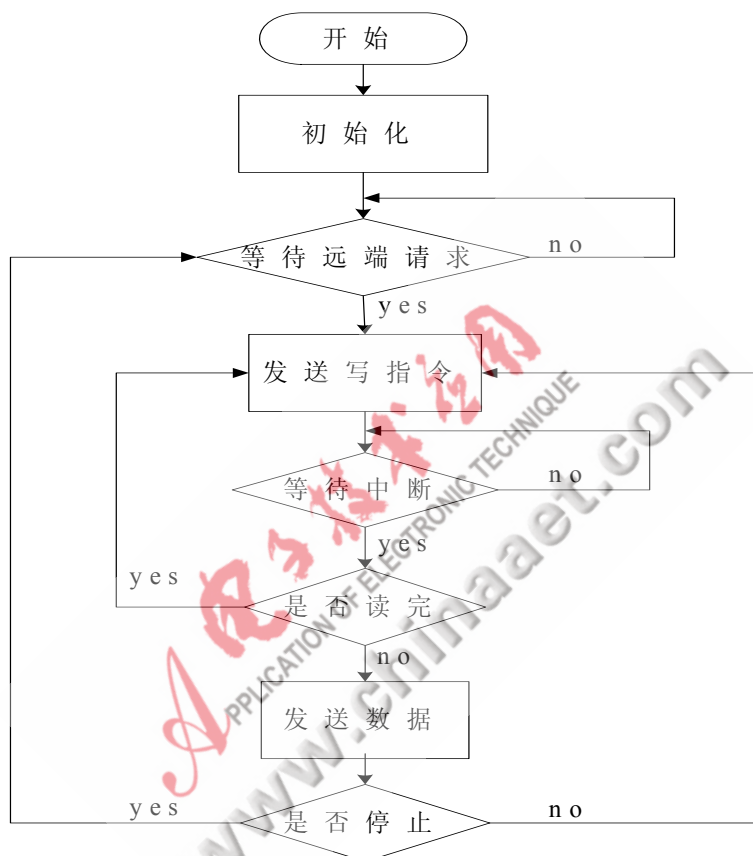


图 4.10 软件流程图

5 接收端显示程序设计

5.1 接收端显示程序设计

客户端视频显示应用程序是利用 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。

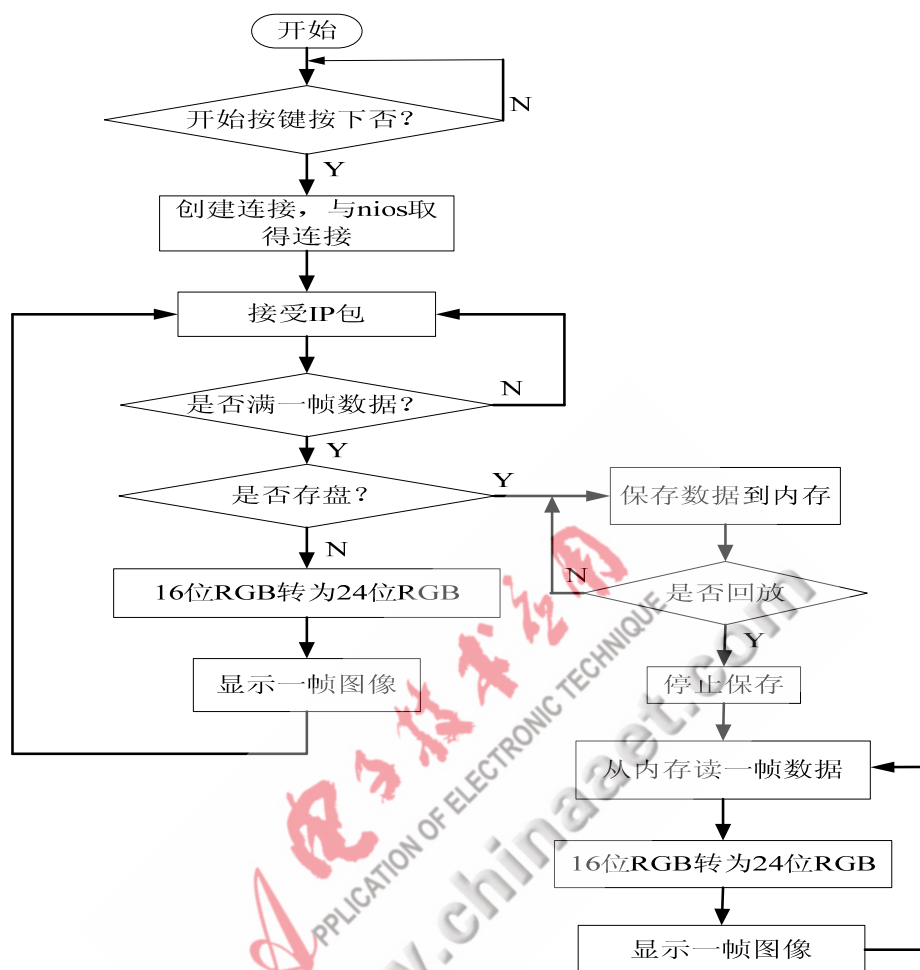


图 5.1 上位机应用程序的流程图

5.2 接收端视频监控效果图

整个系统在局域网中进行了验证，接收端在安装显示软件后，运行软件，监控显示效果图如图 5.2 所示。点击“开始”，开始接收并在实时窗口显示视频；点击“保存”，开始将接收到的视频数据保存在硬盘上；点击“停止”，停止保存视频数据，此时再点击“回放”，即可在回放窗口中显示刚才保存过的视频。



图 5.2 为远程监控显示效果图

结束语

本文中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。这二者结合开发，使得系统开发成本降低，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
- [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] 姚庆栋 毕厚杰等. 图像编码基础.北京：清华大学出版社.2006.8
- [4] 孙景琪编著. 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，2003
- [5] 许光辉，程东旭等. 基于 FPGA 的嵌入式开发与应用.北京：电子工业出版社，2006.9
- [6] 陆绍强.FPGA 将逐渐取代 ASIC 和 ASSP.<http://www.hseda.com>.
- [7] ANALOG DEVICES. Multiformat SDTV Video Decoder ADV7181B Data Sheet.May 2005.
- [8] 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，2004.10
- [9] 吴继华.将低成本FPGA用于视频和图像处理.
http://hi.baidu.com/cpu_/blog/item/3405c6f84829750ed9f9fdcc.html .
- [10] ANALOG DEVICES.Triple 10-Bit High Speed Video DAC ADV7123 Data

Sheet.May 2005.

- [11] DAVICOM Semiconductor, inc. Ethernet Controller with General Processor interface DM9000A Data Sheet. May 2005.

原创性声明

郑重声明：此篇题为《基于 FPGA 的视频监控系统》的论文，是作者在导师指导下，与西安电子科技大学攻读硕士学位期间，进行研究工作的成果，除了文中标注部分，论文中不包含其他人已经发表或撰写的研究成果。

电子设计题目： 基于 FPGA 的视频监控系统

电子设计作者签名： 邵应昭

日期： 二 00 八年三月二十七

作者：

邵应昭：男，1983，陕西户县，西安电子科技大学在读研究生，研究方向为基于 FPGA 和 DSP 的图像视频压缩技术

联系方式：

地址：西安电子科技大学 144#

邮编：710071

Email: yingzhaoshao@yahoo.com.cn 13201736029

电子技术应用
APPLICATION OF ELECTRONIC TECHNIQUE
www.chinaaet.com