

基于 ZYNQ 的远程图像采集系统设计

李庆春¹, 李 祺², 刘彦君³, 赵 越⁴

(1. 海军装备部装备项目管理中心, 北京 100071; 2. 北京机电工程研究所, 北京 100074;
3. 成都普汇科技有限公司, 成都 611731; 4. 电子科技大学 自动化学院, 成都 611731)

摘要: 针对传统图像采集系统远程图像传输延时长和数据丢失的缺点, 设计了一种基于 ZYNQ 芯片开发的实时视频采集与图像传输系统; 系统具有两个采集通道, 模拟视频信号通过 BNC (bayonet neill-concelman) 信号线接入设备并经过 ADC (analog-to-digital converter) 完成信号的数字化; 利用主控芯片内部的 FPGA 资源部署并行处理单元完成对数字化图像数据的低时延处理; 通过驱动片内 AXI (advanced eXtensible interface) 总线以 DMA (direct memory access) 的方式将数据传输至 DDR3 存储器中; 利用芯片内部的双核 ARM Cortex-A9 处理器高性能, 在采集设备上移植嵌入式 Linux 系统, 搭建 Gstreamer 流媒体应用服务器端, 实现整个采集系统复杂任务调度和图像数据远程网络传输; 与传统单 ARM 或 DSP 处理器的图像采集系统相比, 该系统具有 FPGA 的并行处理能力和高带宽的内部互联总线的优势, 提高了图像数据处理速度, 降低了图像数据由采集端到存储器的传输延时, 提供了稳定远程图像传输功能, 经实验测试该系统实现了每秒 25 帧的视频信号输出, 与前端 ADC 的采集速率保持一致, 整个采集展示的过程中视频画面连续且稳定。

关键词: ZYNQ 芯片; FPGA; AXI 总线; Linux 系统; Gstreamer 流媒体应用

Design of Remote Image Acquisition System Based on ZYNQ

LI Qingchun¹, LI Qi², LIU Yanjun³, ZHAO Yue⁴

(1. Equipment Project Management Center, Navy Armaments Department, Beijing 100071, China;

2. Beijing Institute of Mechanical and Electronic Engineering, Beijing 100074, China;

3. Chengdu Jinghui Technology Co., Ltd., Chengdu 611731, China;

4. School of Automation Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China)

Abstract: Aiming at the shortcomings of long remote image transmission delay and loss data of traditional image acquisition system, a real-time system of video acquisition and image transmission based on ZYNQ chip is designed. The system has two acquisition channels, and the analog video signal is connected to the device through the bayonet neill-concelman (BNC) signal line and is digitized by the analog to digital converter (ADC). The FPGA resources in the main control chip are used to deploy a parallel processing unit to complete the low-delay processing of digital image data, and the data is transmitted to the DDR3 memory in the form of DMA by driving the on-chip AXI bus; The dual-core ARM Cortex-A9 processor inside the chip is used to migrate the embedded Linux system to the acquisition device, and the Gstreamer streaming media application server is built to realize the complex task scheduling of the entire acquisition system and the remote network transmission of image data. Compared with the image acquisition system of traditional single ARM or DSP processor, the system has the parallel processing capability of FPGA and the high-bandwidth internal interconnection bus, which improves the image data processing speed, reduces the transmission delay of image data from the acquisition end to the memory, and provides the stable remote image transmission function. The experimental test shows that the system realizes the video signal output of 25 frames per second (FPS), which is consistent with the acquisition rate of the front-end ADC. The video picture is continuous and stable during the whole process of collection and display.

Keywords: ZYNQ chip; FPGA; AXI bus; Linux system; Gstreamer streaming application

0 引言

图像的采集与存储设备已经广泛应用于工业生产、医疗卫生、航空航天等领域^[1], 提升图像数据处理速度与降低图像传输延迟对图像采集终端设备有着重大的意义。在特定的场景下, 如远程手术、航天探测器、无人机电力巡检等, 对图像远程传输的实时性和稳定性有着很高的要求, 而在图像采集、传输及处理过程中需要大量的时间。例如,

由模数转换器转换得到的尺寸为 $1\ 728 \times 625$ 的 8 bit 的数字化图像数据大小为 1.03 MB, 当传感器高帧率、连续的采集图像时需要具备较高的传输带宽才能保证图像数据的完整性。除此之外, 为了有效利用图像信息还需要对数据进行处理。传统的 DSP 或 ARM 处理器具有优秀的控制能力^[2], 但是取样速率较低、指令串行执行且系统使用浮点, 很难对数据量大、像素相关性大、频带较宽的图像数据进行直接处理^[3]。

收稿日期: 2022-01-10; 修回日期: 2022-03-28。

作者简介: 李庆春(1995-), 男, 北京人, 大学本科, 助理工程师, 主要从事装备状态管理和保障技术方向的研究。

引用格式: 李庆春, 李 祺, 刘彦君, 等. 基于 ZYNQ 的远程图像采集系统设计[J]. 计算机测量与控制, 2022, 30(10): 174-180.

目前 FPGA 器件具有的数据并发处理、流水线技术、边接收边处理、高速接口等特性同图像数据传输及处理的需求十分契合, 但是在外设控制能力方面有所不足。为了构建一个图像传输延迟低、可远程网络图传、可编程的图像采集系统, 本文采用 Xilinx 公司推出的 ZYNQ 高性能芯片, ZYNQ 芯片内部融合了 ARM 处理器和 FPGA, 处理器与 FPGA 之间通过高速的 AXI 总线互联, 这些特性能够实现处理速度、控制能力、传输速率的最大化, 同时 ARM 处理器还能借助 Linux 系统搭载 Gstreamer 流媒体应用将处理好的图像数据推送到网络实现远程图像传输。

1 系统结构及原理

在本图像采集系统中, ZYNQ 芯片的 PS (processing system) 端主要完成系统控制, 数据的存储和传输以及后处理, PL (programmable logic) 端则用于对 ADC 输入数据的前处理和输出数据至 DAC。芯片内部 PS 端与 PL 端之间通过 AXI 总线进行数据交互, 设计中用到 AXI4 和 AXI4_Lite 总线, 其中 AXI4 总线的数据位宽为 64 bit, 用于传输数据量大、传输延时要求高的图像数据; AXI4_Lite 总线的数据位宽为 32 bit, 用于控制数据的传输。

根据图像编解码板的控制及传输要求, 需要大量的控制接口和低时延的数据处理能力, 由于 PS 端 IO 口数量有限、ARM 处理器串行执行的特性无法实现高速的数据处理, 所以采用 PL 端丰富的 IO 进行接口扩展, 利用 PL 端的可编程逻辑资源部署并行处理算法完成大量数据处理的方式不仅可以降低成本而且易于系统功能实现。在系统接口方面, 整个设计能够满足 2 路 GJB1188A B 类视频信号输入、1 路 GJB1188A B 类的静态图像信号输出、1 路 GJB1188A B 类的静态图像同步信号输出。

除此之外, 该系统还配备了连接显示设备的 HDMI 接口、可外接键盘鼠标的 USB 接口、标准的千兆以太网接口以及可用于工程调试的串口等。丰富的接口在实现核心功能的基础上增加了系统的灵活性。图像采集系统的整体结构如图 1 所示。

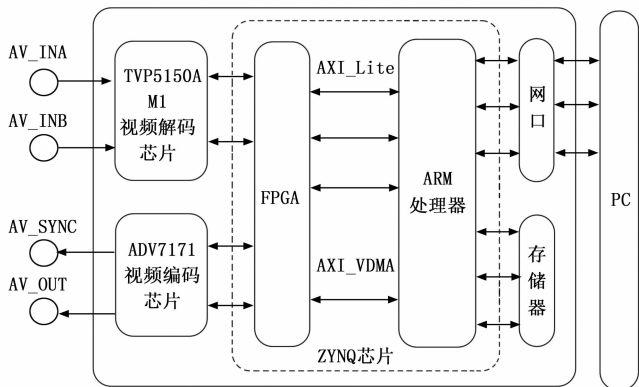


图 1 系统结构框图

2 系统硬件设计

ZYNQ 相对于分立的 ARM+FPGA 架构而言, 单一芯

片能够节省 PCB (printed circuit board) 面积、减小布局布线难度。除此之外, 单芯片融合 ARM 和 FPGA 可以利用高效的片内通信总线进行数据传输, 高带宽的总线对图像数据传输效率有较大的提升。

系统前端输入的 PAL 制式的模拟视频信号经过 A/D 转换为 ITU-R BT.656 标准的数字视频信号, 其中包含大量的消隐信号和辅助信号等一些冗余信号, 如果不对这些冗余信号进行处理而直接传输, 传输的数据量过大导致传输效率低, 延时比较高, 无法满足图像处理实时性的要求, 并且后端输出的视频信号的帧率也无法达到图像采集前端每秒 25 帧的速度, 这些冗余信号的消除不会对图像的质量产生影响, 因此, 需要对视频信号进行分离处理, 只保留信号的有效数据部分进行传输, 等待传输完成, 再将视频信号的有效数据和消隐信号、辅助信号进行复合, 恢复原视频画面, 这种处理方式可以有效地提高数据传输效率, 降低系统延时, 提高视频的显示帧率。如何提取视频信号中的有效数据以及选择硬件还是软件的方式来对 BT.656 视频信号进行解码成为了本系统需要解决的难点。

图像数据传输的低延时是图像采集系统的基本要求之一, 图像处理的速度必须很快, 直接使用 CPU 进行图像处理, 虽然处理起来方便灵活, 但是由于需要处理相当大的数据量, 会极大地消耗 CPU 的运算能力, CPU 的占用率比较高, 而频繁对 DDR 存储器的访问也会增加数据传输和解析的时间, 不能满足系统对于图像数据传输低时延的需求。FPGA 的实时流水线运算可以满足图像数据高效性处理的要求^[4]。为了提高图像传输的实时性, 本系统采用硬件解码的方式, 即利用 FPGA 部署硬件运算电路来完成视频的解码工作^[5], 首先将视频信号中的 Y (亮度值) 和 Cb、Cr (色度值) 准确地分离出来, 然后再将 YCbCr4:2:2 格式转换为 YCbCr4:2:0 进行传输。经过处理器性能、FPGA 资源消耗、开发成本等各方面的评估, 决定采用型号为 XC7Z020 的 ZYNQ 芯片并围绕该款芯片设计外围电路。

利用性能优异的 ZYNQ 芯片构建的系统硬件部分主要划分为:

1) 图像采集通道: 整个采集框架支持 2 路 PAL 格式的视频信号输入, 接入系统的模拟视频信号经过 TVP5150AM1 芯片解码输出 ITU-R BT.656 格式的数字视频数据^[6]。视频解码芯片的工作状态由 ZYNQ 芯片内部的 FPGA 部分中的视频解码芯片配置模块通过 I²C 总线进行配置。

2) 图像处理模块: ITU-R BT.656 格式的数字视频数据通过解码芯片外部的 LVDS (low-voltage differential signaling) 接口传输至 FPGA, FPGA 根据图像格式对数据进行解析和处理。图像处理模块包括通道选择模块、数据解析模块、数据处理模块和数据重组模块。

通道选择模块, 受控于处理器指令, 选择一路图像数据输出至后续模块处理。

数据解析模块，用于对选择的一路图像数据解析，根据 ITU-R BT. 656 格式的构成特点解析出图像数据的有效数据、消隐信号标志、数据错误标志、图像帧的头部和尾部信号、行同步信号、场同步信号、复合同步信号。其中数据错误标志位指示当前解析的图像数据是否包含完整的信息，当信息不完整时，当前图像帧被丢弃，数据解析模块等待下一帧图像数据输入。完整无误的图像数据伴随着解析出的标志信息传入后级模块。

数据处理模块，用于剔除图像数据中的无效数据，只保留有效的 Y 和 Cb、Cr 数据，Y、Cb、Cr 的比例为 4:2:2，因为图像数据中 Y 分量对图像的质量影响较大，而 Cb、Cr 分量对图像数据的质量影响较小，所以通过进一步压缩 Cb、Cr 部分的有效数据来减少数据的传输量，提高数据的传输效率，将处理后的 Cb、Cr 和 Y 数据利用 FIFO (first input first output) 缓存后同步输出至后级模块处理。

数据重组模块，用于将图像有效数据 Y、Cb、Cr 重新组合成 YCbCr 4:2:0 的数据格式。

3) 图像传输模块：利用 AXI_DMA IP 核将处理完成的图像数据传输至 DDR3 存储器中。

4) 图像数据生成模块：利用 FPGA 逻辑资源生成自定义的图像数据，图像数据可以上传至处理器端用作图像数据传输链路的测试；也可以经过编码芯片可输出 PAL 制式的模拟视频数据，可以用作视频源。

5) 图像输出通道：利用图像数据生成模块产生的数字图像数据通过 ADV7171 视频编码芯片转换成对应的模拟视频信号，同时根据图像的构成格式识别图像的行、场同步信号并输出。

3 系统关键部分的实现

3.1 图像采集通道

该图像采集系统的前端信号输入是 PAL 制式的模拟视频信号，目前 NTSC 和 PAL 属于全球两大主要的电视广播制式^[7]，NTSC 电视标准用于美、日等国家和地区。而 PAL 电视标准则广泛用于中国、欧洲等国家和地区。PAL 制式电视标准每秒 25 帧画面，625 条电视扫描线，奇场在前，偶场在后，画面的宽高比为 4:3，标准的数字化 PAL 电视标准每一帧有 625 行，有效数据共有 576 行，每行包含 720 个像素点，分辨率为 720 * 576，色彩位深为 24 比特^[8]。

为了实现数字化处理，本系统需要对 PAL 制式的模拟视频信号做数字化处理，通过图像解码芯片 (TVP5150AM1) 进行模数转换^[9]，其转换后的格式为 ITU-R BT. 656 标准的数字视频信号。PL 端与图像解码芯片 (TVP5150AM1) 之间采用 27 MHz/s 串行接口的数字传输接口标准。

ITU-R BT. 656 标准是将一个视频序列分成 N 帧，PAL 制式转换后的 BT. 656 标准有 625 行，底场有效数据也是 288 行，其余行则是为了标记和区分两种场的垂直消隐信号。采集图像的时候采用隔行扫描 (interlaced scan-

ning)，每一帧一般有两个场 (field)，一个叫顶场 (top field)，一个叫底场 (bottom field)，由于隔行扫描，也可以将顶场和底场称为偶场和奇场。BT. 656 标准 625 列数据格式如图 2 所示。

行消隐	场消隐	第 1 行 V=1	行数	F	V	H(EAV)	H(SAV)
	偶数场有效视频数据	第 23 行 V=0	0-12	0	1	1	0
		第 314 行 V=1	23-310	0	0	1	0
	场消隐	第 336 行 V=0	311-312	0	1	1	0
	奇数场有效视频数据	第 624 行 V=0	313-335	1	1	1	0
		第 336 行 V=0	336-623	1	0	1	0
场消隐	第 336 行 V=0	624-625	1	1	1	0	

图 2 BT. 656 标准 625 列数据格式示意图

BT. 656 标准的每一行主要由以下 4 个部分组成：行 = 结束码 (EAV) + 水平消隐 (Horizontal Vertical Blanking) + 起始码 (SAV) + 有效数据 (Active Video)，如图 3 所示。

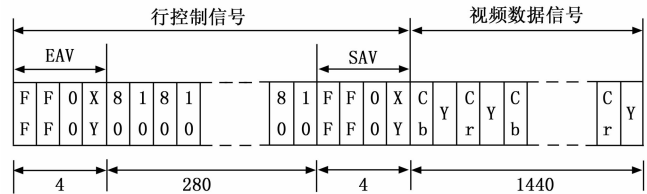


图 3 656 格式行数据组成

每一行数据信号被编码成 8bit 的形式，其中包括辅助信号 (SAV、EAV)、行消隐信号、有效视频信号。SAV 表示视频行数据的开始，EAV 表示视频行数据的结束，SAV 和 EAV 均由 4 个字节构成，前 3 个字节 FF 00 00 为固定头，是 SAV 和 EAV 的数据标志位，第 4 个字节“XY”为辅助信号的信息位，其编码格式如图 4 所示。

bit 7	1
bit 6	F
bit 5	V
bit 4	H
bit 3	P3=V XOR H
bit 2	P2=F XOR H
bit 1	P1=F XOR V
bit 0	P0=F XOR V XOR H

图 4 辅助信号信息位编码格式

XY 的最高位 (bit7) 为固定数据 1；F 标记场信息，传输偶数场时为 0，传输奇数场时为 1；V 标记消隐信息，传输消隐数据时为 1，传输有效视频数据时为 0；H 标记 EAV 还是 SAV，EAV 为 1，SAV 为 0；P0~P3 为保护位，其值由 F、V、H 计算得到，计算方法如下：P3 = V (XOR) H，P2 = F (XOR) H，P1 = F (XOR) V，P0 = F (XOR) V (XOR) H^[10]。每行对应不同的 EAV、SAV 如表 1 所示。

表 1 656 列数据对应的行辅助信号

行数	F	V	EAV	SAV
1—22	0	1	0xb6	0xab
23—310	0	0	0x9d	0x80
311—312	0	1	0xb6	0xab
313—335	1	1	0xf1	0xec
336—623	1	0	0xda	0xc7
624—625	1	1	0xf1	0xec

消隐行数据则由 80 10 组成, 共 280 byte, 本设计在 PL 端可根据辅助信号的变化, 剔除消隐行的无用数据, 用以减小一帧有用信号的传输时间及软件处理无用数据的时间。对于有效视频信号 (Valid data), 其排列顺序如图 5 所示。

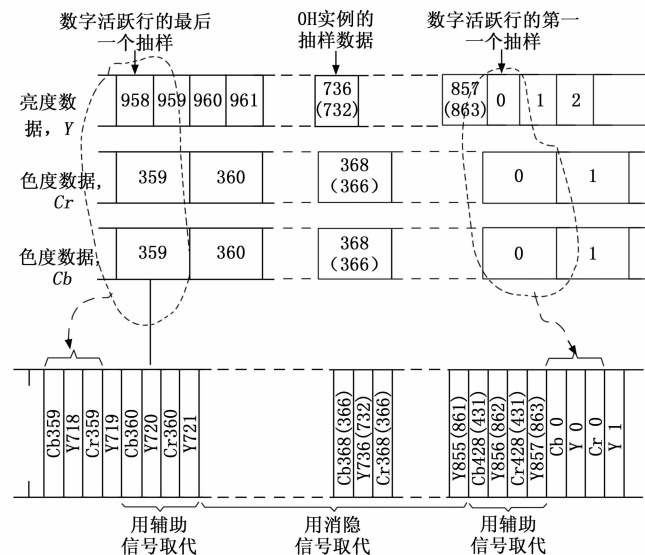


图 5 有效视频信号数据构成

其中 Y 表示明亮度 (Luminance 或 Luma), 也就是灰阶值; 而 Cb 、 Cr 则用来表示色度, 其中 Cb 反映了 RGB 输入信号蓝色部分与 RGB 信号亮度值之间的差异^[11]。 Cr 反映的是 RGB 输入信号红色部分与 RGB 信号亮度值间的差异^[12]。 Y 、 Cb 、 Cr 的比例为 $4:2:2$, 从图像的像素点上来理解, 就是每一个像素点有一个单独的 Y 值, 而相邻的两个像素点的 Cb 和 Cr 数据是一样的^[13]。

本设计将 625 行 BT. 656 标准的数字视频信号的一帧数据在 PL 端根据辅助信号的变化, 去除行消隐信号, 将有用数据重新排列处理后, 经 DMA 传输至存储器, 软件将数据通过奇偶穿插后播放器就能把图像正确地显示出来。

3.2 图像数据预处理

外部的 PAL 制式电视广播信号是模拟视频信号^[14], 为了实现 ZYNQ 数据处理, 还需要进行 A/D 转换, 模拟视频信号经过视频解码芯片 TVP5150AM1 后被转换为 ITU-R BT. 656 标准的数字视频信号^[15], 但是此时的视频信号包含大量的列消隐信号、行消隐信号、辅助信号等一些无用信号, 无法直接传输至上位机做数据处理及图像显示。最终的显示还是需要我们将 ITU-R BT. 656 视频标准中的行场

数据重新排列, 提取有效行中的灰度及色度信息, 即将图像数据中的 Y 、 Cb 、 Cr 准确地分离出来, 并按奇场偶场重新排列, 才能供软件播放使用^[16]。因此, 视频解码工作在本系统中尤为重要, 目前可以实现这种视频解码的方式主要分为两种: 软件解码和硬件解码。

该设计起初应用软件解码, 硬件仅负责数据传输功能, 在此架构下, 软件的大量循环重复操作且需要处理的数据量巨大, 不仅增加了系统功耗及 CPU 的占用率, 频繁对 DDR 的访问也增加了数据传输和解析的时间。最终只能实现每秒 3 至 5 帧的视频播放, 远远低于图像采集前端每秒 25 帧的速度。通过 FPGA 对图像预处理需要将 BT. 656 标准的数字视频信号做解析, 根据每行的 XY 控制字解析出场同步信号 (F)、垂直同步信号 (V)、水平同步信号 (H)。通过提取每帧图像中的有效数据, 拼成逐行图像。

相邻每两行为一组, 从原来 “ $CbYCrY\dots$ ” 顺序的数据中提取每两行的所有 Y 数据, Cb 数据、 Cr 数据。按照第一行 720 字节 Y 数据 (连续 180 个 100 MHz 时钟周期的 dma_tvalid), 第二行 720 字节 Y 数据, 360 字节预处理后的 Cb 数据, 360 字节预处理后的 Cr 数据 (第二行 Y 和 Cb 、 Cr 一起有连续 360 个 100 MHz 时钟周期的 dma_tvalid) 排列, 之后就可以这样为一组, 每帧图像共发送 288 组这样的数据通过 FIFO 缓存后发送给 DMA 进行传输。

为了进一步提高数据传输效率, 保证系统的实时性, 还需要对数字视频数据作进一步压缩处理, 将 $YCbCr=4:2:2$ 格式转换为 $YCbCr=4:2:0$ 格式。

取样方式的描述可以通过一条线被扫描时所呈现的亮度值和色度值之间的比率来表示。描述形式为 $4:X:Y$, X 和 Y 表示每两个色度通道中数量的相对值。 $4:1:1$ 表示每条扫描线每 4 个亮度值和 1 个色度值相对应, $4:2:2$ 表示每 4 个亮度值和 2 个色度值相对应, $4:4:4$ 表示色度值不进行二次取样。 $4:2:0$ 表示每 4 个亮度值和 1 个色度值相对应, 但是它的取样方式不是连续的, 这意味着对于第一个色度元素有两个取样值, 而对于第二个色度元素则不进行取样, 这种取样方式不能产生完整的彩色图像。实际工程中, $4:2:0$ 表示每条扫描线有两个色度取样, 采用隔行的方式进行取样。

虽然将 $4:2:2$ 转换为 $4:2:0$ 可能减少细节处颜色的饱和度, 但通常不会减少大个物体内的颜色饱和度。具体处理过程为: 保留原图像中的 Y 数据, 将逐行图像数据中对应位置的 Cb 和 Cr 相加并除以 2, 得到新的 Cb 和 Cr , 然后就可以将数据按两行为一组, 每帧图像共发送 288 组这样的数据通过 FIFO 缓存后发送给 DMA。这种处理方式将一帧图像的数据量进一步压缩, 有效提高了数据传输效率。

4 系统软件设计

4.1 软件设计思路及编程方法

系统软件分为嵌入式软件 and 上位机软件两个部分, 其中嵌入式软件运行在 Linux 操作系统上, 首先需要在设备上移植 Linux 操作系统, 嵌入式软件分为应用层、系统层、

底层驱动 3 个部分^[17]，应用层的视频服务应用通过操作系统 API 与系统层交互，系统层通过设备驱动程序驱动 FPGA 等外设，从而实现对外部信号以及外部设备的控制。嵌入式软件结构如图 6 所示。

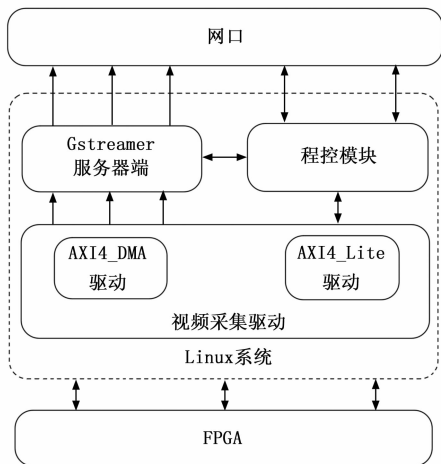


图 6 嵌入式软件架构图

嵌入式软件架构的应用层是整个嵌入式软件提供给上位机的接口，在应用程序下包含程控模块和 Gstreamer 流媒体应用的服务器端。程控模块主要对上位机下发的指令进行解析，根据指令调用功能函数完成对其他模块的控制，例如调用 AXI4_LITE 驱动实现对 FPGA 的控制，或者调用 AXI4_DMA 驱动把 FPGA 处理好的视频数据传输到存储器中。Gstreamer 流媒体应用则是将存储器中的视频数据通过管道技术经由千兆网口推送至上位机的 Gstreamer 客户端。

系统软件中的上位机软件是在 Windows XP 系统下开发的，软件主要包含网络连接模块、人机交互界面、Gstreamer 客户端、视频显示窗口。网络连接模块采用 TCP/IP 通信协议实现上位机与设备之间的网络连接；人机交互界面支持对远程设备的参数配置及功能的控制；Gstreamer 客户端通过管道技术访问服务器端的流媒体数据；视频显示窗口呈现远程读取的视频画面。

上位机软件架构如图 7 所示。

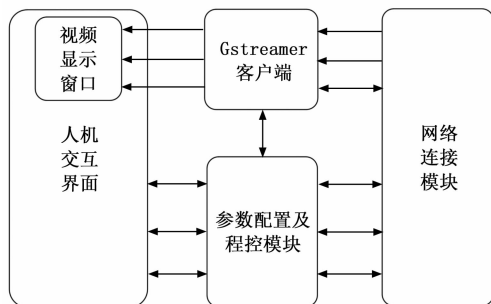


图 7 上位机软件架构图

4.2 驱动软件的实现

设计中采用了 Xilinx 公司的 ZYNQ 芯片，其芯片内部

融合了双核 ARM Cortex-A9 处理器和 Xilinx 7 系列 FPGA，ARM 与 FPGA 通过 AXI 总线互联。视频数据和外设控制数据的交互都经由 AXI 总线，由于视频数据的传输速率、传输带宽及传输延迟均比控制数据要求高，所以对于视频数据采用 DMA 传输的方式^[18]，而控制信号采用自定义 AXI4_Lite 总线进行传输。数据传输是该采集系统能否实现视频数据稳定展示的关键，其中 FPGA 到存储器的数据传输和存储器到网口的数据传输主要通过嵌入式软件完成。

FPGA 对于 ARM 而言是外设，ARM 同 FPGA 完成数据交互需要对两者互联的总线进行驱动，对于本系统需要开发带 DMA 功能的 AXI 总线驱动和自定义通信协议的 AXI4_Lite 总线驱动。介于嵌入式软件功能都是在 Linux 系统上进行开发的，所以这两种 AXI 总线的驱动软件开发需要符合 Linux 驱动框架。

4.2.1 AXI4_Lite 驱动软件设计

AXI4-Lite 是简化版的 AXI4 接口，用于较少数据量的存储映射通信，该总线为外设提供单个数据传输，主要用于访问一些低速外设中的寄存器。本设计中控制数据的交互就是通过在 ZYNQ 芯片内提供的通用 AXI 接口连接具有 AXI 通信协议的自定义功能 IP 核实现的。完成总线互联后，总线上挂载的 IP 核具有一段 I/O 地址空间，通过对该地址空间进行读写操作完成对 IP 核的控制。

AXI4-Lite 总线是按照字节流进行读写操作，它属于 Linux 驱动中的字符设备。由于 AXI4_Lite 总线驱动需要实现的功能只有简单的读写操作，因此按照字符设备驱动框架完善驱动程序中的 open ()、write ()、read ()、release () 等函数，通过获取设备树中描述的 IP 核 I/O 地址空间完成物理地址到虚拟地址的映射，最后指定驱动的出口函数和入口函数完成 AXI4_Lite 的驱动编写。加载编译完成的驱动后，在 Linux 系统的“/dev”目录下生成一个相应的文件^[19]，通过对文件的读写实现对总线的读写。

4.2.2 AXI4_DMA 驱动软件设计

设计中采用 Xilinx 提供的 AXI4_DMA IP 核将经过 FPGA 处理完成后的视频数据通过 DMA 的方式从 FPGA 传输到存储器中。AXI4_DMA 对于 Linux 系统而言也是字符设备，相对于 AXI4_Lite 总线来说驱动 AXI4_DMA 更加复杂，需要处理 DMA 的开启、停止、数据等待、读取数据空、中断异常等情况。进行数据传输时，需要对 DMA 进行初始化，访问 DMA 控制寄存器，配置目的地址和待传输字节数。传输完成后，DMA 状态寄存器中的中断状态通知 DMA 通道当前 DMA 传输已完成。连接到 ARM 处理器的 AXI4_DMA IP 核主要有 3 个数据通道：1) S_AXI_LITE, AXI4 类型接口，用于初始化与配置 DMA；2) S_AXIS_S2MM, AXI4-Stream 类型接口，DMA 通过此端口接收缓存 FIFO 输入的流式数据，并在 DMA 内将流式数据转换为存储映射数据；3) M_AXI_S2MM, AXI4 类型接口，DMA 通过此端口将存储映射数据传输到存储器中。

AXI4_DMA 的驱动软件除了实现基本的读写功能外, 还需要通过 `ioctl()` 函数实现对 DMA 通道传输的控制, 读取当前可用的 DMA 通道数目, 把每个通道的通道号、通道类型、通道方向等相关信息保存到分配的数组中。AXI4_DMA IP 核如图 8 所示。

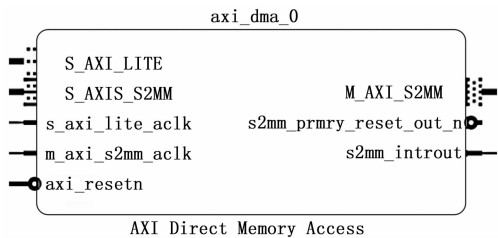


图 8 AXI4_DMA IP 核示意图

在 Linux 系统中, 应用层访问外设接口是通过系统调用从用户空间进入系统内核, 然后通过驱动程序中的 `copy_to_user()` 和 `copy_from_user()` 两个函数实现数据的交互, 应用程序获取外设数据需要从驱动层拷贝。当上位机需要显示视频数据时, 应用程序还需将数据拷贝至 Gstreamer 流媒体应用。在执行视频显示的功能中, 同一份视频数据需要进行两次拷贝, 这会降低整个传输链路的效率, 增加 CPU 的负荷和传输延迟。为了解决多次拷贝的问题, 在编写 AXI4_DMA 驱动时直接将 DMA 传输的地址所在的存储空间通过 `EXPORT_SYMBOL` 内核模块函数导出, 该段存储空间用作共享内存使用。

共享内存的方式能够实现数据的零拷贝传输^[20], 但是多个驱动模块并发的访问同一段内存时会导致读写冲突的问题, 因此设计中将该段共享内存虚拟成循环队列这种数据结构, 并在导出地址空间的同时也导出写指针, 访问共享内存的其他驱动程序拥有特有的读指针, 通过读写指针的位置差异可以判断共享内存空间中是否存在数据, 有效地解决了多个驱动模块对共享资源的并发访问。共享内存如图 9 所示。

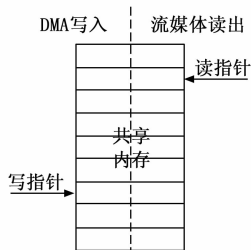


图 9 共享内存示意图

4.3 程控模块软件设计

程控模块主要包含网络服务器端、指令解析模块、DMA 传输控制、Gstreamer 流媒体调用、外设控制。程控模块是整个设计的调度中心, 整个系统运行时上位机软件同设备通过 TCP/IP 网络协议建立通信, 上位机软件在参数设置或者视频调用请求时通过网络接口发送指令至程控模块, 程控模块通过解析指令执行相应的功能函数, 完成

系统自检、输入通道选择、视频数据传输到 Gstreamer 流媒体应用服务器端等操作。

4.4 软件实现流程

软件实现流程如图 10 所示。

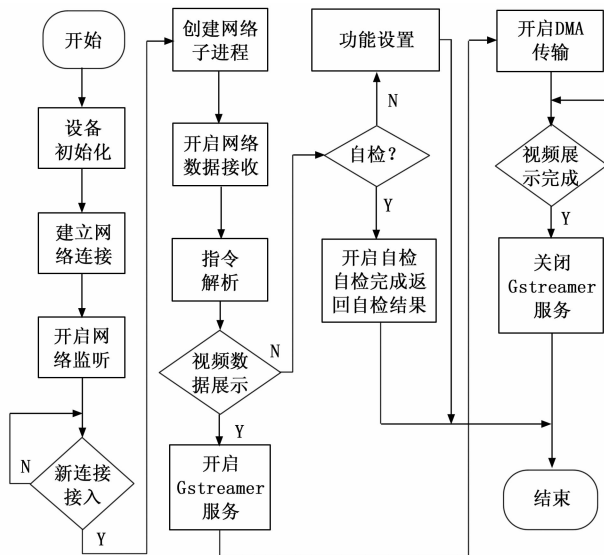


图 10 嵌入式软件的总体设计流程及模块控制

5 实验结果与分析

利用 Xilinx 公司的 Vivado 软件完成 FPGA 侧的开发并导出硬件平台文件。使用 Petalinux 软件构建与硬件平台适配的嵌入式 Linux 系统, 系统移植完成后将驱动软件及应用程序导入嵌入式 Linux 系统中。完成上述操作后对设备重新加电, 进入嵌入式系统后加载驱动程序并运行应用程序, 首先实现对设备的各个模块进行初始化操作、开启网络监听, 然后等待远程客户端接入。

功能测试: 将远程监视设备和图像采集设备接入同一子网下进行网络传输及视频展示测试, 测试步骤如下:

- 1) 将设计的图像采集设备与计算机通过网线相连, 直流电源与采集设备通过 12 V 电源线相连, 设置直流电源电流为 3 A 并输出;
- 2) 在 PC 中启动上位机软件, 设置图像采集设备的 IP 地址, 点击“打开连接”;
- 3) 将视频源与图像采集设备的输入通道相连并输出图像数据。通过上位机软件配置需要查看的采集通道, 并开启视频数据的传输。

上位机软件自带视频显示窗口, 当数据传输关闭时, 显示窗口默认为黑屏状态, 配置菜单栏为不可配置状态。软件状态如图 11 所示。

当上位机软件与图像采集设备建立网络连接且采集设备完成自检给上位机软件反馈自检成功信号后, 人机交互界面的其他控制窗口切换到可配置状态。配置采集设备打开通道 1, 并开启远程采集, 实验结果如图 12 所示。

显示窗口左上方显示采集数据的展示时长及图像展示帧数的累加, 整个视频采集展示的过程是十分流畅, 刷新

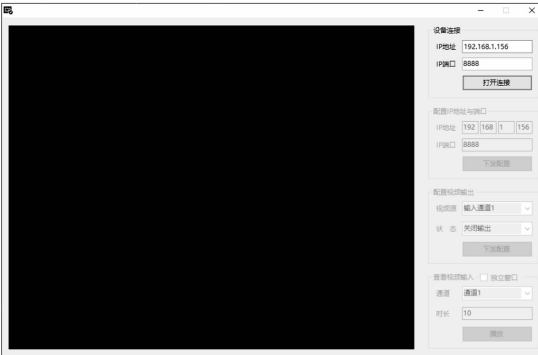


图 11 图像采集系统人机交互界面

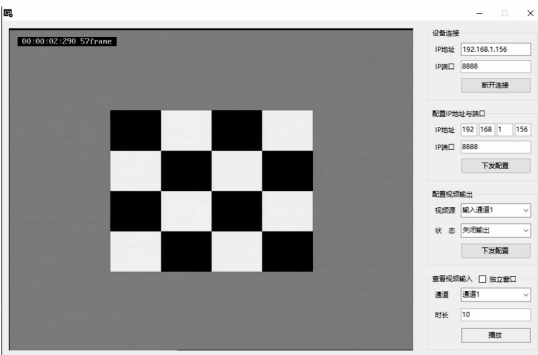


图 12 图像采集远程显示效果

率每秒 25 帧，完全与前端 ADC 采集速率一致。

6 结束语

本设计通过利用 ARM 和 FPGA 融合的 ZYNQ 芯片在软硬件方面对图像的采集流程进行了优化，通过多次测试验证，整个图像采集系统的传输延时显著降低，图像数据的远程网络传输稳定高。本设计能够应用在需要远程图传且对延时有要求图像处理系统中。由于本设计器件选型原因，单使用 FPGA 内部的分布式 RAM (random access memory) 资源无法缓存大量的原始图像数据，因此需要通过运算减少一些对图像影响较小的图像数据量来满足传输过程中的缓存要求。后期设计可以在 ZYNQ 芯片 PL 侧连接一定容量的 DDR3 存储器对整个系统进一步优化。

参考文献:

[1] 汝兴海, 任勇峰, 李辉景, 等. 基于 Camera Link 的高速图像传输模块及时序优化设计 [J]. 计算机测量与控制, 2015, 23

[15] 张坤鳌, 李俊, 方欣. 基于 STM32 的凝结水液位监控系统的设计与实现 [J]. 计算机测量与控制, 2016, 24 (8): 79-82, 136.

[16] 张亚娟, 刘寒冰. 基于 PLC 和组态软件的大棚环境监测系统设计 [J]. 计算机测量与控制, 2015, 23 (6): 1983-1985, 1989.

[17] 李启丙. 基于组态软件的 PLC 控制系统全软件仿真 [J]. 机

(7): 2546-2548.

[2] 姬生毅. 基于 Zynq 的嵌入式数字图像传输系统的设计与实现 [D]. 西安: 西安电子科技大学, 2014: 7-15.

[3] 林振钰, 张志杰, 刘佳琪. 基于 ZYNQ 的高清图像显示及检测系统设计 [J]. 计算机测量与控制, 2021, 29 (2): 30-34.

[4] 李科. 基于 FPGA 的数字图像处理 [N]. 科技资讯, 2012-2-2 (2).

[5] 周瑾, 叶锡恩, 叶志全, 等. 基于 MPEG-4 硬件解码的视频终端设计 [J]. 宁波大学学报 (理工版), 2008, 21 (4): 457-461.

[6] 蒋陈铭, 史小军. 基于 TVP5150 的低功耗视频解码模块 [J]. 电子工程师, 2008, 34 (4): 46-48.

[7] 孙兴国. 基于 NTSC 和 PAL 制式的电视信号解码设计方案 [D]. 北京: 北京工业大学, 2007.

[8] 钱兵. 基于 PC104-plus 的机载视频数据采集系统设计 [D]. 成都: 成都理工大学, 2014.

[9] 肖星. 基于立体视频的远程控制系统的研究与实现 [D]. 天津: 天津工业大学, 2011.

[10] 刘潇, 高峻. 采用 FPGA 通过 BT.656 接口实现传输 4 路视频流的方法 [J]. 电子工程师, 2006, 32 (4): 8-10.

[11] 薛宏伟. 软件视频会议中的视频图像技术研究 [D]. 太原: 中北大学, 2009.

[12] 余燕飞. 视频质量检测技术研究 [D]. 合肥: 中国科学技术大学, 2012.

[13] 张晓宁. 基于 USB 的数据采集及处理方法的研究 [D]. 哈尔滨: 哈尔滨工业大学, 2006.

[14] 戴鹏飞. 视频压缩与信号复接系统设计 [D]. 合肥: 中国科学技术大学, 2010.

[15] 耿洁, 陈家福, 徐宁, 等. ITU-R BT.656 视频解码译码的 FPGA 实现 [J]. 仪器仪表用户, 2007, 14 (2): 98-99.

[16] 关春燕, 章雪挺, 刘敬彪. 基于 BT.656 的电视视频硬件解码的分析与实现 [J]. 电子器件, 2010, 33 (3): 366-369.

[17] 孟凡飞. 基于 Gstreamer 的嵌入式流媒体系统的研究与设计 [D]. 广州: 华南理工大学, 2010.

[18] 江寅. 基于 Intel SoC 处理器视频采集系统的设计与实现 [D]. 成都: 西南交通大学, 2014.

[19] 刘永林. 基于 ARM9 的嵌入式检测系统控制平台设计 [D]. 太原: 中北大学, 2012.

[20] 王祥. 基于 FPGA 的 PCI-E 数据采集系统的设计与实现 [D]. 成都: 电子科技大学, 2012.

电工程技术, 2021, 50 (4): 135-137.

[18] 王海涛. 基于易控组态软件全虚拟 PLC 的实现及应用 [J]. 商用汽车, 2020 (8): 86-90.

[19] 白仲花, 常涛, 杨军, 等. 基于易控软件实现工厂自动化监控 [J]. 信息技术与标准化, 2020 (12): 64-67.

[20] 高洪青. 基于故障确认的一体化测试性试验验证平台 [J]. 计算机测量与控制, 2019, 27 (11): 249-251, 279.