

基于 Zynq-7000 的视频处理系统框架设计

李 朗¹, 张索非², 杨 浩¹

(1. 南京邮电大学 电子科学与工程学院, 江苏 南京 210003;

2. 南京邮电大学 物联网学院, 江苏 南京 210003)

摘 要:随着嵌入式机器视觉技术的不断发展,对视频数据带宽、处理速度和数据精度的要求越来越高。针对当前基于 ARM 及 DSP 的嵌入式图像处理系统存在的采集速度慢、带宽有限等缺点,提出了一个基于内部集成 ARM+FPGA 结构的 Zynq-7000 芯片且能够实现软硬件协同工作的视频处理系统框架。该系统框架采用 FPGA 并行处理算法实现图像预处理环节,采用 ARM 处理器进行程序流程控制以及实现复杂的图像识别算法,通过 DDR 缓存以实现软件与硬件的协同工作,在保证处理速度与带宽的基础上兼顾了系统功能的灵活性,并且降低了系统功耗。测试验证结果表明,所提出的系统框架不仅可以提供高速彩色图像采集和预处理的方案,同时还可以实现高性能的中端图像处理算法,并能够将结果快速送达显示端显示。

关键词: Zynq-7000; 图像采集; 图像处理; 软硬件协同

中图分类号: TP302

文献标识码: A

文章编号: 1673-629X(2017)05-0192-04

doi: 10.3969/j.issn.1673-629X.2017.05.040

Design of Video Processing System Framework with Zynq-7000

LI Lang¹, ZHANG Suo-fei², YANG Hao¹

(1. College of Electronic Science and Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China;

2. College of Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

Abstract: With the continuous development of embedded machine vision technology, the requirement of video data bandwidth, processing speed and data accuracy has become higher and higher. Current embedded image processing system are mainly based on ARM with DSP framework. Such designs normally come with deficiencies as low acquisition speed and limited processing bandwidth. To address this problem, Zynq-7000 has been adopted to realize a video processing framework by synergy of FPGA as hardware and ARM as software. FPGA acts as an implementation of parallelized image preprocessing algorithm as well as ARM processing unit to control the program flow and to implement the complex image processing algorithms are integrated in this framework. A DDR buffer has been exploited to coordinate the hardware and software. Such synergy ensures the processing efficiency and bandwidth of system, while considering the flexibility of system simultaneously. The experimental results show that not only can the design provide a high speed color image acquisition and preprocessing, but also can implement a high performance image processing algorithm with video display of graphic results.

Key words: Zynq-7000; image acquisition; image processing; synergy of software and hardware

0 引言

视频采集和处理系统在视频监控、自动化生产等领域有着广泛的应用需求。而越来越广泛的应用对视频流处理的分辨率、清晰度以及实时性等指标也提出了更高的要求。传统的数字信号处理(Digital Signal Process, DSP)加片上处理器芯片(如ARM)构成的解

决方案^[1]逐渐显现出带宽过窄、处理性能不足等缺陷。而单纯以现场可编程门阵列(Field Programmable Gate Array, FPGA)作为解决方案^[2],虽然在处理速度上可以达到要求,却无法像传统的嵌入式开发一样以较短的周期进行复杂的图像算法流程实现,从而使整个系统的功能受到了限制。以工业自动化生产以及机器人

收稿日期: 2016-05-26

修回日期: 2016-09-09

网络出版时间: 2017-03-13

基金项目: 国家自然科学基金面上项目(41471329)

作者简介: 李 朗(1990-),男,硕士研究生,研究方向为嵌入式系统、图像处理;杨 浩,博士,副教授,研究方向为物联网、机器视觉。

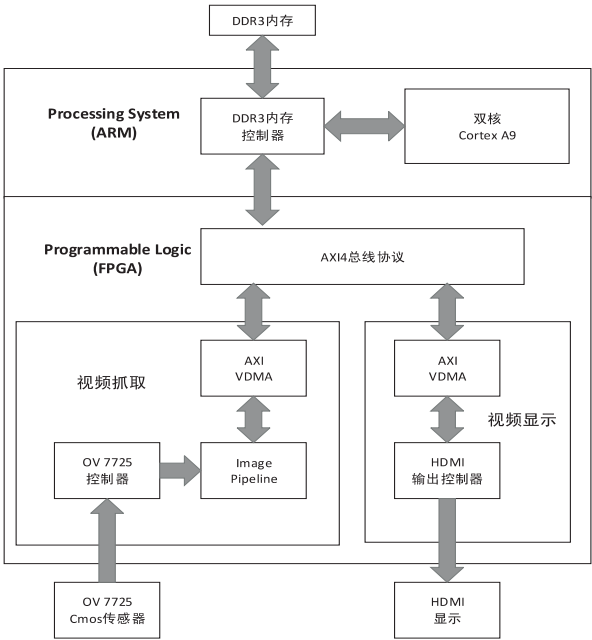
网络出版地址: <http://kns.cnki.net/kcms/detail/61.1450.TP.20170313.1546.062.html>

等视频处理需求密集的行业为例,现代视频采集处理系统通常由视频采集前端、图像预处理以及图像信息提取三个部分构成。其中视频采集以及预处理部分需要系统高速处理视频流信号以满足高清视频格式的要求,而图像信息提取则要求系统能够调用图像特征提取、匹配等功能,将图像信号转换为高级语义信息。三部分功能的结合对于系统的软硬件性能以及通信带宽都提出了更高的要求。

针对新兴的应用需求,基于 Xilinx 公司推出的 Zynq-7000^[3]平台来构建彩色视频采集和显示系统。Zynq 基于可扩展处理平台(Embedded Processing Platform,EPP)结构,将双精度浮点引擎的 ARM Cortex-A9 双核处理器与低功耗可编程逻辑紧密结合,提供了强大的灵活性、可配置性和系统性能。平台主要由处理系统(Processing System,PS)和可编程逻辑(Programmable Logic,PL)两大功能模块构成。其中,PS 主要包括应用处理单元(Application Processing Unit,APU)、存储器接口、I/O 外设;PL 则采用 FPGA 技术实现扩展功能以满足特定的功能要求。两部分以高速片上总线 AXI(Advanced eXtensible Interface)互联,保证整个系统的处理带宽。鉴于以上优点,Zynq 解决方案在工业视觉以及视频监控领域获得了越来越多的关注^[4-6]。以 Zynq 平台为基础,以 DDR3 内存为图像缓存设计视频采集、硬件预处理、软件处理的程序流程,实现了软硬件协同的视频信号采集、处理以及显示功能。

1 系统硬件总体框架

系统硬件框图如图 1 所示,包含采集、缓存、处理以及显示四个模块,分别由 PS 和 PL 部分协同完成,



万方数据 系统硬件结构框图

而各部分通过 AXI 总线内部互联^[7]。

(1)图像采集部分在 PL 端完成,OV7725 CMOS 摄像头使用排线和 ZYNQ-7000 相连,信号通过 Verilog HDL 描述的控制器进入系统。PS 通过 IIC 协议来配置和初始化 OV7725 摄像头,采集到的图像数据转化为 AXI-Stream 总线型数据以便使用 Xilinx 的 Video IP 核进行图像校正增强的预处理。OV7725 摄像头采用常见的 30 万像素格式采集图片(640×480),帧率为 60 fps/s,每像素通信带宽 32 bit,理论上所需的最小 IO 吞吐量为 562.5 Mbit/s,而系统的高速 IO 接口最高达到 10 G/s 带宽,完全满足了系统的要求。

(2)PL 部分采集图像通过 AXI-VDMA(视频直接内存访问接口)存入 DDR3 进行缓存,AXI 总线协议包括 9 个物理接口,分别为 AXI-GP0~AXI-GP3,AXI-HP0~AXI-HP3,AXI-ACP。其中两个 AXI-GP 为主机接口,虽然宽带性能较低,却具有发起读写的权限。通过把 PL 映射到 DDR3 中的相应地址,ARM 就可以利用该接口访问 PL,如同读写自己的内部存储器。其余接口则为被动接口,允许 PL 部分以高速主动读写。

(3)图像处理功能由 PL 和 PS 协作完成,PL 使用一系列 IP 核对原始图像进行预处理,经过 Video Pipeline 转化为 32 位 ARGB 格式的数据流。数据流通过双端口 AXI VDMA 存入 DDR,PS 调用该缓存地址图片并使用软件算法进行信息提取,实现软硬件协同处理功能。

(4)处于 PL 中的视频显示部分通过一个双端口的 AXI-VDMA 从 DDR 中读取含有软件处理结果的渲染后图片,并将图片送入 HDMI 输出控制器以实现图像显示。

2 系统数据处理流程

视频流数据经过系统各环节处理的流程如图 2 所示。采集图像首先通过阵列插值、颜色校正以及亮度校正等一系列基础预处理模块获得比较理想的显示质量,然后通过高层次综合(High-Level Synthesis,HLS)的 IP 模块进行处理。HLS 为 Xilinx 推出的新型开发工具,可以由 C/C++标准规范直接综合至寄存器传输级别(Register Transfer Level,RTL)模型,以较短的周期实现硬件开发。通常一些并行化程度高的像素级算法操作可以放入该模块进行高速实现。

经过图像预处理后,图片被缓存于 Buffer1 中,PS 部分从缓存中读取图片并进行软件算法处理,在输出结果的同时将渲染后的结果图像送至 Buffer2 进行缓存。而显示部分读取图片以后将 RGB 通道图片转为 YUV 通道图片,最终送至 HDMI 显示控制器进行显

示。图像的采集和显示都由 PL 部分实现,保证了整个流程的处理速度。而图像算法一方面由 HLS 通过硬件实现预处理,另一方面通过 PS 软件实现后续算

法。其中软件实现部分可以使用 OpenCV^[8]等开源库构建标准 C/C++规范流程,在保证系统灵活性和功能的同时,兼顾了执行效率和开发周期。

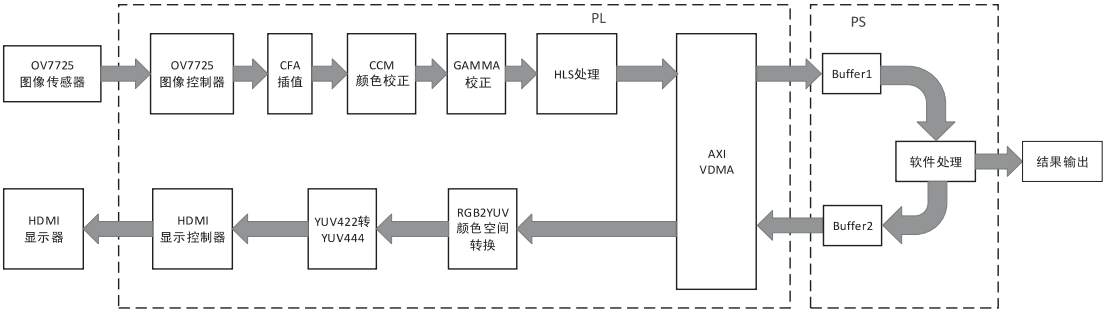


图 2 系统处理流程图

3 驱动设计及系统移植

3.1 嵌入式 Linux 系统搭建

嵌入式 Linux 是将 Linux 内核进行剪裁和修改,使之能够运行在嵌入式系统上的一个小型系统。Xilinx 公司提供了一个 Linux 平台的嵌入式 Linux 开发环境 Petalinux^[9]。Petalinux 工具不仅提供了 Linux 内核 (Kernel)、U-boot、文件系统以及嵌入式应用程序的配置和编译方法,还整合了同步 Vivado FPGA 硬件设计工程到 Petalinux 中的方法,方便了软硬件协同设计。在以 Zynq 为核心的硬件系统中,U-boot、Linux 内核、驱动程序以及应用程序的编译工作都要借助整合在 Petalinux 中的 arm-xilinx-linux-gnueabi-gcc 交叉编译器来完成。该交叉编译器可将各系统组件程序进行预处理、编译、汇编和链接,最终生成能够运行在 Zynq 体系结构中的二进制可执行文件。

3.2 驱动程序设计

设备驱动程序是建立在硬件与软件之间互联的一种接口程序。Linux 系统将存储器和外设大致分为三类,分别为字符设备、块设备和网络设备。在 PL 中实现了视频采集和视频显示的各个 IP 核模块,在驱动中则要为每一个 IP 核添加对应的驱动程序。PS 与该模块交互的方式主要有三种^[10]:通过 PS 部分的专用串行接口直接访问,作为 AXI 总线外设供 PS 部分访问,作为 IP 核虚拟成独立的逻辑器件单独访问。该系统采用第二种方式访问,打开 IP 核的 AXI Lite 总线接口,使得 PS 能通过 AXI Lite 总线接口来配置 IP 核寄存器。

由于 OV7725 摄像头采集的图像格式为 Bayer,图像数据中每一个像素点只有 R、G、B 中的一个颜色分量,还需要通过彩色滤波阵列 (Color Filter Array, CFA) 插值算法对信号进行插值处理,即在每个通道上使用相邻像素点的对应值对当前像素进行估计。现以 CFA IP 核为例说明驱动程序的设计方法。

由于 Linux 内核中的 V4L2 驱动框架为视频设备提供了供上层访问的统一接口^[11],对于 CFA IP 核的驱动设计的主要思路是将 CFA 作为一个视频处理模块整合进 V4L2 的框架之下。V4L2 驱动框架支持视频输入输出设备、VBI 设备以及 Audio 设备等,通常这些设备要通过多个 I2C 总线或 SPI 总线实现和 CPU 的通讯。将设备挂载至总线作为子设备 (Sub-devices) 即可使用 V4L2 框架,将 CFA 模块挂载为 V4L2_Subdev 并驱动设备的工作流程如图 3 所示。

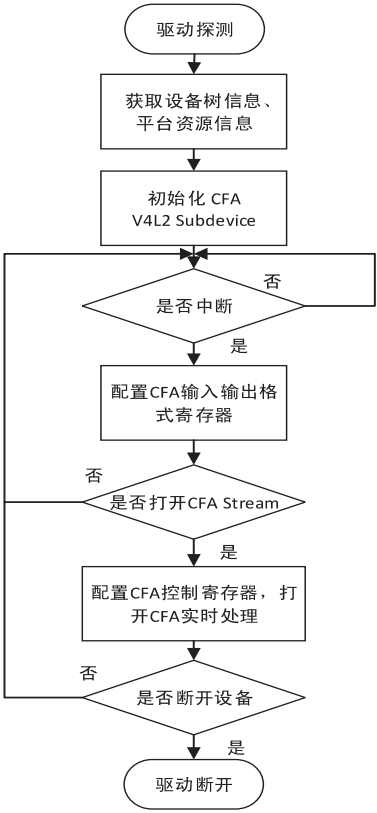


图 3 CFA 驱动工作流程图

具体设计流程如下:

- (1) 构建 CFA Device 结构体。
- (2) 基于 Platform 虚拟总线设计设备初始化和退出函数,并关联相应设备树资源。

- (3) 构建 CFA 的视频操作函数、V4L2 内部处理函数、Media 操作函数。
- (4) 根据 Sysfs 文件系统结构设计设备属性读写函数,创建属性组并注册相应的 V4L2-Subdev 设备。
- (5) 驱动完成调试后加入 Linux 内核编译,以内嵌模式加入内核,并最终生成系统镜像 uImage。

4 测试结果分析

平台基于 OV7725 摄像头运行测试并进行性能分析,每一帧 640×480 分辨率图像通过 PL 进行预处理存入 DDR,应用层软件通过 V4L2 驱动框架调出图片进行软件处理,处理后结果图片经过颜色空间转换变为 YUV 格式送至 HDMI 显示,图 4 为整个系统的实物图及运行示例。

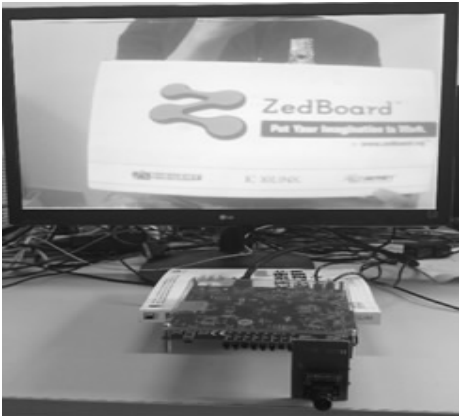


图 4 视频处理系统平台实物图

系统实现了一种经典的 Harris 角点检测算法^[12]。首先将输入图像转换为单通道灰度图像,后使用 Harris 方法检测角点,最终检测结果渲染至原彩色图像进行显示,检测结果参见图 5。其中图 5(b)是在 PS 部分与应用层以 OpenCV 方案实现角点检测,而图 5(c)则是在 PL 部分使用 HLS^[13-14]方案实现检测。可以发现两种方法最终的检测结果基本一致。而从表 1 中的算法执行效率对比中可以发现,软硬件协同处理一帧图像只需要 3 458 μ s,与纯 ARM 实现角点检测算法相比效率提高了 15.2 倍。

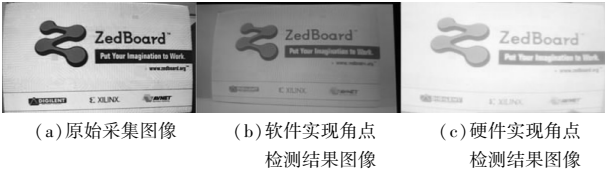


图 5 检测结果

表 1 图像处理算法的执行速度比较

图像处理算法	ARM+FPGA	纯 ARM	调高倍数
	处理时间/ μ s	处理时间/ μ s	
角点检测	3 458	52 631	15.2

万方数据

5 结束语

针对高分辨率、高带宽下 ARM 及 DSP 芯片处理视频数据效率低的问题,提出了一套基于 Zynq-7000 平台使用软硬件协同处理的视频采集、处理及显示方法,并结合 CMOS 摄像头有效地验证了设计的正确性。实验结果表明,所设计的基于 Zynq-7000 的视频采集处理框架具备了完整的图像采集、存储、处理和显示功能。同时,平台可以根据实际需求在 PL 及 PS 部分继续添加各种图像处理环节,实现更为复杂的功能,为后续的视频处理系统实现提供了基础。

参考文献:

[1] 肖政宏,韩秋凤,朱丽群. 基于 ARM 和 DSP 的远程视频监控系统的设计与实现[J]. 计算机工程与科学,2006,28(9):53-55.

[2] 祝长锋,肖铁军. 基于 FPGA 的视频图像采集系统的设计[J]. 计算机工程与设计,2008,29(17):4404-4407.

[3] 何 宾. Xilinx All Programmable Zynq-7000 SoC 设计指南[M]. 北京:清华大学出版社,2013.

[4] 刘 宏,符意德. 基于 Zynq 芯片的图像处理系统平台设计[J]. 计算机与现代化,2015(8):43-47.

[5] 杨晓安,罗 杰,苏 豪,等. 基于 Zynq-7000 高速图像采集与实时处理系统[J]. 电子科技,2014,27(7):151-154.

[6] 孟繁星. 基于 ZYNQ-7000 的视频图像处理系统设计[J]. 信息通信,2015(8):60-62.

[7] 陆佳华,潘祖龙,彭竞宇. 嵌入式系统软硬件协同设计实战指南[M]. 北京:机械工业出版社,2014.

[8] Bradski G, Kaehler A. Learning OpenCV: computer vision in C++ with the OpenCV library[M]. [s. l.]: O'Reilly Media, Inc., 2013.

[9] Xilinx Inc. Xilinx PetaLinux SDK user guide: getting started guide[EB/OL]. 2013-04-22. http://www.xilinx.com/support/documentation/sw_manuals/petalinux2013_04/ug977-petalinux-getting-started.pdf.

[10] 叶 琴,谢捷如. 基于 SoC 和嵌入式 Linux 的数据采集系统设计[J]. 计算机技术与发展,2015,25(8):203-207.

[11] 张 辉. 基于 V4L2 的嵌入式视频驱动程序开发与实现[D]. 合肥:安徽大学,2010.

[12] Harris C. A combined corner and edge detector[C]//Alvey vision conference. [s. l.]:[s. n.], 1988:147-151.

[13] Li C, Cheng W S, Fan J Z, et al. Parallel image processing technology of surface detection system[J]. Semiconductor Photonics and Technology, 2008,14(4):217-223.

[14] Colodro-Conde C, Toledo-Moreo F J, Toledo-Moreo R, et al. A practical evaluation of the performance of the impulse Co-Developer HLS tool for implementing large-kernel 2-D filters[J]. Journal of Real-Time Image Processing, 2014,9(1):79-83.