

国内图书分类号: TH7412
国际图书分类号: 681.2.08

学校代码: 10213
密级: 公开

工学硕士学位论文

基于FPGA的Camera Link相机图像采集及处理技术研究

硕 士 研 究 生: 吴振锋

导 师: 刘国栋 教授

申 请 学 位: 工学硕士

学 科: 仪器科学与技术

所 在 单 位: 电气工程及自动化学院

答 辩 日 期: 2013 年 6 月

授予学位单位: 哈尔滨工业大学

Classified Index: TH741.1
U.D.C: 681.2.08

Dissertation for the Degree of Master in Engineering

FPGA-Based Technology Research on Camera's Image Acquisition and Processing with Camera Link Interface

Candidate:	Wu Zhenfeng
Supervisor:	Prof. Liu Guodong
Academic Degree Applied for:	Master of Engineering
Speciality:	Discipline of Instrumentation Science and Technology
Affiliation:	School of Electrical Engineering and Automation
Date of Defence:	June, 2013
Degree-Conferring-Institution:	Harbin Institute of Technology

摘 要

二维激光位移传感器是一种基于线结构光的测量系统。该系统中线结构光光条图像的实时采集和处理是制约其测量效率的主要瓶颈，因此对于其图像实时采集和处理技术的研究至关重要。

二维激光位移传感器中图像的获取往往采用 Camera Link 接口的工业相机。这是由于 Camera Link 接口具有抗干扰能力强、传输速率快、传输距离远及开放性好等诸多优点，不但可以为二维激光位移传感器的图像采集提供稳定的图像数据，而且能方便地通过对相机的曝光控制等来适应不同的测量环境。

随着 FPGA 技术的不断发展，基于 FPGA 的图像采集处理系统逐渐成为研究的热点。由于 FPGA 强大的并行处理能力，其在图像预处理领域的优势也越来越明显。因此以 FPGA 为核心器件的图像采集处理系统，可以充分利用 FPGA 的并行处理能力以满足二维激光位移传感器的实时性测量要求。

因此，本文进行了基于 FPGA 的 Camera Link 相机图像采集及处理技术研究，主要工作内容如下：

- 1) 设计了基于 FPGA 的图像采集调试平台；
- 2) 对图像采集调试平台进行了相应的硬件逻辑设计，实现了相机图像的实时采集、存储、VGA 显示及串口通信等功能；
- 3) 对典型的空间域图像增强算法进行了硬件实现，验证了基于 FPGA 的图像采集处理系统在图像预处理领域的巨大的并行处理优势；
- 4) 对激光光条提取算法进行了设计与硬件实现，为二维激光位移传感器图像处理算法的进一步研究奠定了基础。

关键词：FPGA；图像采集；图像处理；硬件实现

Abstract

The two-dimensional laser displacement sensor is a system based on line structured light. And real-time acquisition and processing of the structural light image is the main bottleneck of the system. So it is quite important to research on technology of its real-time image acquisition and processing.

Image acquisition of the two-dimensional laser displacement sensor is always done by industrial camera with Camera Link interface. That's because the Camera Link interface has many advantages such as strong anti-interference ability, fast transmission rate, far transmission distance, total openness and so on. So, cameras with Camera Link interface can not only provide stable image data, but also adapt to different measuring environment through conveniently controlling the camera's exposure level.

With the continuous development of FPGA technology, FPGA-based image acquisition and processing systems are becoming more and more hot. As its powerful parallel processing capabilities, FPGA's advantages in the field of image preprocessing has become increasingly evident. So image acquisition and processing systems with FPGA being core component can take full advantage of parallel processing capabilities of FPGA to meet the real-time measuring requirement of the two-dimensional laser displacement sensor.

Thus, this paper conducted a FPGA-based technology research on camera's image acquisition and processing with Camera Link interface. The main work contents are as follows:

- 1) Design of image acquisition and debugging platform based on FPGA.
- 2) Design of hardware logic corresponding to above-mentioned platform, realizing camera image's real-time acquisition, store, VGA display, serial communication and etc.
- 3) Hardware implementation of typical spatial domain image enhancement algorithm, verifying FPGA's parallel processing capabilities' adaptability for image preprocessing algorithms.
- 4) Design and hardware implementation of laser stripe extraction algorithms, providing a reference for further study of image processing algorithms in research of two-dimensional laser displacement sensor.

Keywords: FPGA, Image Acquisition, Image Processing, Hardware Implementation

目 录

摘 要	I
ABSTRACT	II
第 1 章 绪 论	1
1.1 课题背景及研究意义	1
1.2 基于 FPGA 的图像采集及处理技术研究现状	2
1.3 本文研究的主要内容	4
第 2 章 图像采集调试平台电路设计	5
2.1 平台总体设计	5
2.2 Camera Link 接口电路设计	6
2.3 存储电路设计	7
2.4 VGA 显示接口电路设计	9
2.5 RS-232 串口通信电路设计	11
2.6 FPGA 芯片选择及配置下载电路设计	12
2.7 电源方案设计	13
2.8 本章小结	13
第 3 章 图像采集调试平台的硬件逻辑设计	15
3.1 图像采集调试平台介绍	15
3.2 硬件逻辑顶层设计	15
3.3 PLL 时钟分频模块	16
3.4 图像采集模块	17
3.5 SRAM 读写控制模块	20
3.6 图像存储控制模块	23
3.7 VGA 显示控制模块	24
3.8 RS-232 串口通信模块	26
3.8.1 串口接收子模块	27
3.8.2 串口发送子模块	28
3.9 本章小结	30
第 4 章 典型空间域图像增强算法的硬件实现	31
4.1 3×3 滤波窗口实现	31
4.2 中值滤波	33
4.2.1 算法原理	33

4.2.2 算法的硬件实现	33
4.2.3 硬件算法性能分析	34
4.3 拉普拉斯滤波	35
4.3.1 算法原理	35
4.3.2 算法的硬件实现	36
4.3.3 硬件算法性能分析	38
4.4 Sobel 滤波	39
4.4.1 算法原理	39
4.4.2 算法的硬件实现	39
4.4.3 硬件算法性能分析	40
4.5 本章小结	41
第 5 章 激光光条中心提取算法的硬件实现	42
5.1 激光光源及光束特性	42
5.2 传统极值法	42
5.2.1 算法原理	42
5.2.2 算法的硬件实现	43
5.2.3 算法硬件实现效果分析	44
5.3 改进的极值法	45
5.3.1 算法原理	45
5.3.2 算法的硬件实现	45
5.3.3 算法硬件实现效果分析	46
5.4 灰度质心法	46
5.4.1 算法原理	46
5.4.2 算法的硬件实现	47
5.4.3 算法硬件实现效果分析	48
5.5 本章小结	48
结 论	50
参考文献	51
附录 A	54
附录 B	55
哈尔滨工业大学学位论文原创性声明和使用权限	56
致 谢	57
个人简历	58

第 1 章 绪 论

1.1 课题背景及研究意义

本课题的研究背景是二维激光位移传感器的研制项目。二维激光位移传感器是一种基于线结构光的嵌入式测量系统，其测量原理如图 1-1 所示。激光器①通过将线激光束②打在被测表面⑦上，由面阵 CCD 相机⑥采集激光光条图像⑤，最后通过对激光光条图像处理得到被测表面的二维形貌信息。显然，激光光条图像的实时采集和处理是实现二维激光位移传感器高精度、高效率测量的关键。系统每次测量时需要处理整幅的图像数据，进行包括图像预处理、光条提取、位移计算等大量运算，如此之大的运算量和系统实时性要求之间的矛盾成为了制约二维激光位移传感器研制的主要瓶颈。

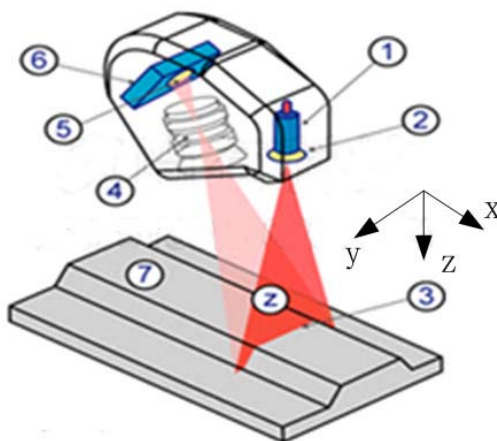


图 1-1 二维激光位移传感器测量原理示意图

Camera Link 是一种高速数据传输协议，其支持的最高数据传输率可达 680MB/s。为满足实时性图像采集的要求，系统设计时采用了 Camera Link 接口的工业面阵 CCD 相机。为适应不同的传输速率和距离需求，数字图像传输领域里先后出现过 IEEE1394, USB2.0, DCOM3, RS-644(LVDS), Camera Link 和千兆以太网等传输协议。其中，Camera Link 接口协议由于抗干扰能力强、传输速率快、接口形式标准、传输距离远等优点，正在逐步地取代原有的一些接口成为市场的主流^[1]。

以往的基于 PC 和图像采集卡的图像处理系统，并不能满足实时性、便携性等方面的要求^[2]。然而，随着 FPGA 技术的不断成熟，基于 FPGA 的图像采集及处理技术成为研究的热点之一^[3]。这是由于 FPGA 的逻辑资源并行性特

点,其支持数据的并行运算及多级流水线技术,可以大大提高数据处理的速度。对于需要大量重复运算但过程相对简单的图像预处理算法,FPGA 具有其他技术手段不可比拟的优势,可实现多模块的并行处理,因此在嵌入式图像采集与处理领域中得到了迅速发展^[4]。

综上所述,研究基于 FPGA 的 Camera Link 相机图像采集与处理技术,不仅顺应技术发展的潮流,而且有助于解决二维激光位移传感器研制过程中大量的数据运算与实时性要求之间的矛盾。

1.2 基于 FPGA 的图像采集及处理技术研究现状

自第一台计算机问世以来,伴随着计算机技术不断进步,数字图像采集处理成为一个迅速发展的学科^[5,6]。而且,随着数字图像处理技术的发展,各种各样的图像采集处理系统的相继出现。总体上看,当前主流的数字图像采集处理研究方向有以下几大类^[7]: 1) 以传统 PC 为核心的图像采集处理技术; 2) 基于专用的数字信号处理器 DSP 的图像采集及处理技术; 3) 基于专用集成电路 ASIC 的图像采集卡; 4) FPGA 和 DSP 相结合的图像采集处理技术; 5) 基于 FPGA 的硬件图像采集处理技术。

这几大类图像处理技术各有优缺点。以 PC 为核心的图像采集处理技术最为成熟,拥有丰富的资源,但往往伴随着系统体积大、便携性差及速度不高的问题。基于 DSP 的图像处理技术可以满足便携性要求,而且其速度也越来越快,但毕竟 DSP 系统内部硬件算法模块有限,当需要处理的数据量很大且速度要求严格时,其显得力不从心。基于 ASIC 的图像采集卡技术拥有无可比拟的处理速度优势,但有着研发周期长、成本高的缺点。以 FPGA 和 DSP 结合的图像采集处理系统,可以充分发挥 FPGA 并行计算的速度优势和 DSP 易于实现高难度算法的优点,属于一种软硬件结合的技术,因此存在软硬件功能划分、FPGA 与 DSP 之间的接口衔接等问题。

随着 FPGA 制造工艺的不断提高,FPGA 向着大容量、高速度的方向不断前进。大容量、高速度的 FPGA 的出现及 SOPC 技术的不断发展,为图像采集和处理提供了一种新的解决方案^[8,9]。基于 FPGA 的图像处理技术适用于处理数据量大、重复性高但算法复杂度相对简单的场合,但是随着基于 FPGA 的硬件算法 IP 核越来越成熟、广泛,以 FPGA 为核心的图像采集处理系统应用越来越受欢迎。另外随着 FPGA 的制造工艺和集成技术的发展,采用 FPGA 的数字硬件系统越来越凸显其在实时图像处理领域的强大优越性^[10]。

国外的众多大学和科研机构从 20 世纪五十年代便开始了基于 FPGA 的图像采集及处理研究^[11]。

墨西哥 CINVESTAV-IPN 大学采用 FPGA+NIOS 软核的体系结构,实现了基于 Sobel 算子的图像分割。其图像处理速度在 40MHz 时钟下达到了每秒 43 帧 640×480 的图像。

乌克兰大学于 1996 年设计了基于 FPGA 的卡尔曼滤波,并提出了在 FPGA 内部配置一个基于卡尔曼滤波的协处理器,其卡尔曼滤波算法的时间周期达到了 0.4013 μs [12]。

英国帝国理工学院于 2005 年提出了一种基于累积直方图的硬件中值滤波算法,为大窗口的中值滤波提供了一种更易于实现的方法,因为其占用逻辑资源取决于输入采样数据,而不是滤波窗口大小 [13]。

德国蒂宾根大学设计了一个小型的 FPGA 板作为 PCI 系统的协处理器。该小型 FPGA 协处理器板适用于可视化 3D 数据集的算法加速,其将大量的短整数和位操作运算从 CPU 转移到 FPGA 上运行 [14]。

爱尔兰贝尔法斯特女王大学于 2006 年提出了一种三维灰度结构代码用于图像分割。硬件平台采用 Virtex-II 系列 FPGA,两个独立的 DDR-SDRAM 及两个独立的 ZBT-SRAM。其硬件图像分割算法可以在 2 秒内处理一副 512³ 的三维图像 [15]。

国内对基于 FPGA 的图像采集处理技术研究虽然起步晚,但发展迅猛,出现了许多有代表性的研究成果。

2007 年西南交通大学设计了基于 FPGA 和 USB2.0 的视频图像采集及处理平台 [16]。其在此平台上验证了视频压缩算法,并通过 USB2.0 实现了平台与 PC 机之间的处理后数据传输。

2009 年北京交通大学闰宇航提出了一种改进的粒子滤波算法。其采用了一种新的重采样算法来实现重采样,以二维被动目标跟踪系统为模型进行仿真并将改进的粒子滤波算法在 FPGA 器件上实现 [17,18]。

2010 年哈尔滨工业大学庞业勇以基于 Stratix III 系列 FPGA 的 DE3-340 作为图像处理平台,进行了典型图像预处理算法的 FPGA 实现,同时论述了基于 FPGA 可重构计算的算法硬件设计方法,剖析了全新的软硬件协同设计思想 [11]。

2013 年 2 月,中国科学院长春光学精密机械与物理研究所 Xilinx 公司的 Virtex5 系列 FPGA 与 TI 公司 TMS320C6455 型高速 DSP 为核心处理器,设计了多核架构的应用于周扫式红外预警系统的实时图像处理平台 [19]。

2013 年 3 月,上海大学基于形态学滤波和 Roberts 边沿检测算法提出一组数字电路,在 FPGA 内部实现了抗噪 Roberts 算子边沿检测器。该检测器消耗 Altera-EP4CE115 中 2139 个逻辑单元,每秒可处理 30 帧分辨率为 640×480

的 RGB 图像。

1.3 本文研究的主要内容

根据上述图像采集处理的实时性要求，本文就基于 FPGA 的 Camera Link 相机图像采集及处理技术中一系列问题进行了研究。主要研究内容如下：

- 1) 基于 FPGA 进行图像采集调试平台电路设计。该平台电路包含 Camera Link 接口、SRAM 存储器、VGA 视频接口、RS232 串行通信接口等。
- 2) 针对平台进行硬件逻辑设计，实现包括图像实时采集、存储、VGA 显示及串口通信等功能，为后续图像的实时采集处理提供硬件环境。
- 3) 进行典型空间域图像增强算法的硬件实现，验证 FPGA 的并行处理能力在图像预处理领域的适用性。
- 4) 进行激光光条中心提取算法的硬件实现，为二维激光位移传感器图像处理算法的后续研究奠定基础。

第 2 章 图像采集调试平台电路设计

在开展基于 FPGA 的图像采集及处理技术研究工作之前,首先需要建立一个合理的图像采集调试平台电路。本章进行基于 FPGA 的图像采集调试平台的电路设计工作。本章第一节进行平台的总体设计,后续各节进行各模块的详细设计。

2.1 平台总体设计

图像采集调试平台应实现如下的基本功能: Camera Link 相机输出视频图像数据的采集、存储、处理以及与上位机的通信(上传处理后的数据)。另外,为便于观察图像采集、处理的效果,需要 VGA 显示接口,对采集处理后的图像实时显示。为此,本文设计了如下的图像采集调试平台,如图 2-1 所示。

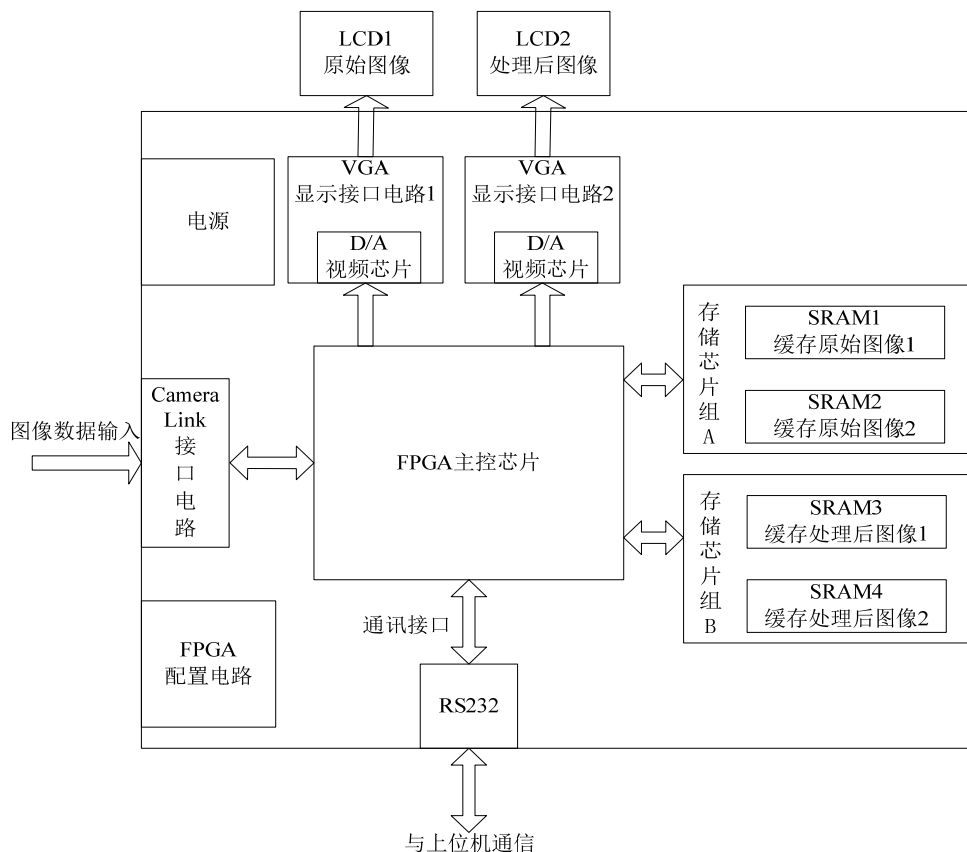


图 2-1 图像采集调试平台

图 2-1 中,相机输出图像数据经 Camera Link 接口电路由 FPGA 采集、并进行预处理后,缓存到存储芯片 SRAM 中,并由 VGA 显示系统进行实时显示。

经过对图像数据的处理得到光条中心数据后,通过通信电路将数据传送给上位机。

图像采集、存储及 VGA 显示的详细过程如下:

- 1) 相机输出第 1 幅图像时,分别缓存第 1 幅原始图像和预处理后的图像到 SRAM1 和 SRAM3;
- 2) 相机输出第 2 幅图像时,分别缓存第 2 幅原始图像和预处理后的图像到 SRAM2 和 SRAM4,与此同时将 SRAM1 和 SRAM3 中的图像数据送给 VGA 接口,对比显示原始图像和处理后的图像;
- 3) 相机输出下一副图像时,重复步骤 1)。

这一过程循环往复。这里用到了 FPGA 设计时常用到的“乒乓操作”原理,即对于相机输出的原始图像和处理后的图像分别使用两片 SRAM 芯片进行图像的交替存储与显示。

2.2 Camera Link 接口电路设计

1) 首先介绍 Camera Link 技术。Camera Link 接口是基于 National Instruments 公司标准的 Channel Link 技术发展而来的^[20]。其基于 LVDS(Low Voltage Differential Signaling, 低电压差分信号) 电平标准,采用一个信号驱动端加上一个信号接收端来分别实现信号的发送与接收。在信号驱动端,28 位并行数据以 7:1 的比例被转化成 4 对差分信号(每对差分信号流包含 7 个 bit 数据);同时,同步时钟信号也被转化成一队差分信号,与上面的 4 对差分数据同步输出。在信号接收端,5 对差分信号流数据则被相应地还原成 28bit 并行数据和—个同步时钟信号,如图 2-2 所示。

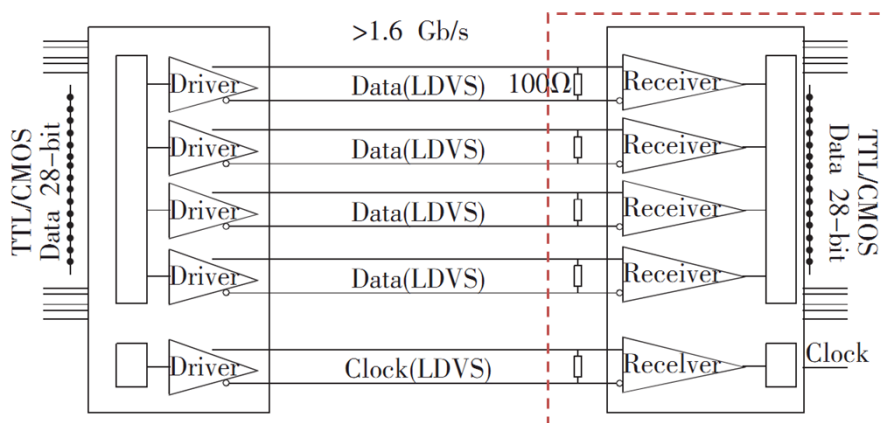


图 2-2 Channel Link 连接示意图

除去上面说的 28bit 的视频数据信号传输, Camera Link 接口还提供相机

控制信号和串行通信信号。这三类信号分别对应一对 Channel Link 转换芯片 (LVDS 与 LVTTTL 电平转换及数据流串并转换)。

本文中 Camera Link 接口设计为信号的接收端设计(红色框里面的), 即把相机输出的 5 对 LVDS 数据流转化为 28 位并行数据接收。

另外, Camera Link 的接口有三种配置 Base、Medium、Full, 主要是解决数据传输量的问题。使用哪种配置模式需要根据相机输出数据量大小来定。不同的配置模式使用的 Channel Link 芯片数目不同。表 2-1 是三种配置模式下所使用的硬件资源情况。

表 2-1 不同配置模式下资源需求

配置模式	支持端口	所需芯片个数	所需连接器个数
基本	A、B、C	1	1
中等	A、B、C、D、E、F	2	2
最高	A、B、C、D、E、F、G、H	3	3

2) 相机输出数据带宽

本文所选用的面阵 CCD 相机型号为 STC-CL33A。该相机有多种工作模式, 默认状态下相机输出图像大小为 $648 \times 494 \times 10\text{bit}$, 频率 60 帧/秒, 每一像素为 10 位数据。时钟频率为 25MHz。故对于视频数据流的接收, 使用 Base 模式(单片 Channel Link 芯片)即可。

3) 芯片选择

相机时钟默认频率 25MHz, 频率不算高, 故可选用较常用的 DS90CR288A(视频数据, 其速率范围为 20MHz~85MHz)及 DS90LV019(串行通信), DS90LV047A(相机控制)。

4) 详细电路设计

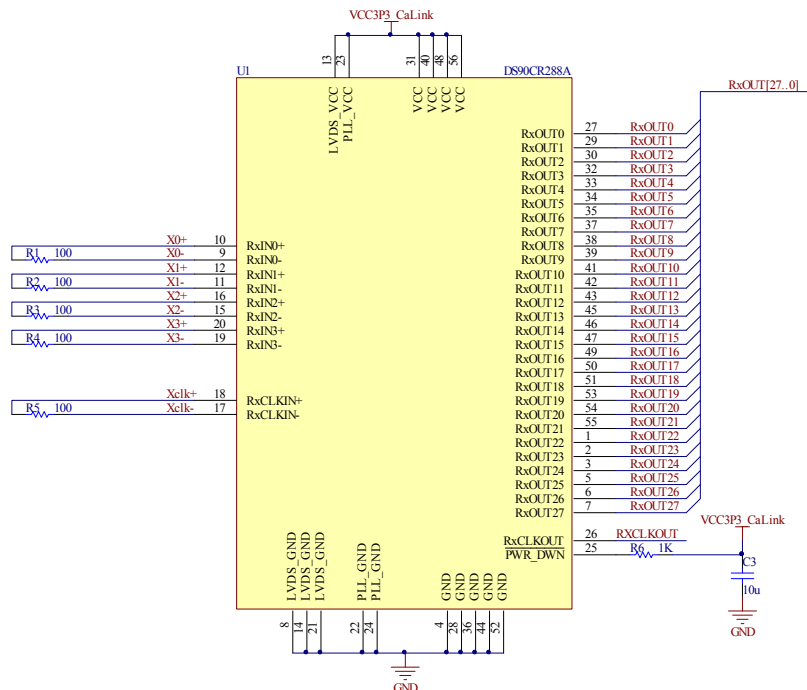
如图 2-3 所示, (a)图是 DS90CR288A 芯片, 负责 28bit 视频数据的串并转换; (b)图是 DS90LV047A 芯片, 负责相机控制; (c)图是 DS90LV019 芯片, 负责与相机通信。值得注意的是, 对于电路图中的差分信号线布线时要根据差分信号的布线原则, 相邻距离越近越好, 这一点不同于普通信号线布线时的“3W”原则。

2.3 存储电路设计

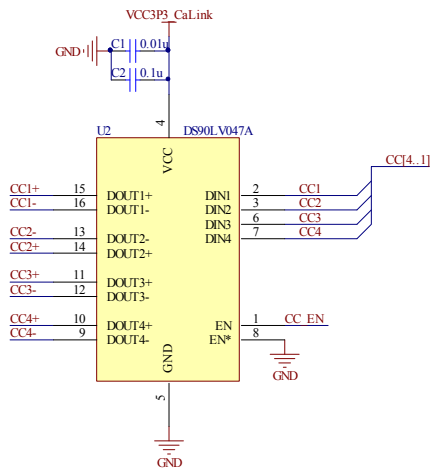
存储电路用于缓存原始图像数据及处理后的图像数据。由于 FPGA 片内存储资源极其有限, 很难满足需要缓存大量图像数据的应用场合。SRAM(Static Random Access Memory), 即静态随机存储器, 因其不需要刷新电路、拥有更

快的读写速度及相对简单的控制时序在嵌入式开发中得到广泛的应用。

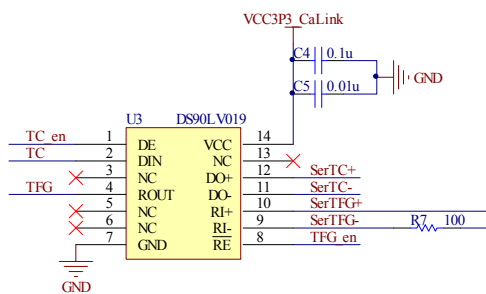
SRAM 存储电路的设计，需要根据应用需求来选择合适的 SRAM 芯片。



(a) 视频数据串并转换接口电路



(b) 相机控制接口电路



(c) 相机串行通信接口电路

图 2-3 Camera Link 接口电路设计

存储容量需求分析如下：相机输出图像大小(默认模式下)为 $494 \times 648 \times 10\text{bit}$ 。SRAM 存储器一般为 8 位或 16 位宽度。为方便设计，对相机图像进行存储时把每一像素(10bit)存储成 16bit(高位补 0)。如此，所需 SRAM 大小为： $494 \times 648 \times 16\text{bit} = 320112 \times 16\text{bit} = 313\text{K} \times 16\text{bit}$ 。因此需要选用 $512\text{K} \times 16\text{bit}$ 规格的 SRAM 芯片。

功能需求分析如下：如果要想实现原始图像的乒乓存储与显示，需要 2 片 SRAM；如果要想实现原始图像和处理后的图像的对比 VGA 显示，则需要 4 片 SRAM 芯片(如图 2-1 所示)。

综上，共需要 4 片大小为 512K×16bit 的 SRAM 芯片。本文选择常用的 IS61WV51216BLL-10T，其特性参数如下所示：

存取速度：10ns；

存储空间：512K×16bit；

供电方式：单电源供电，2.4V~3.6V；

全静态操作：无时钟及刷新要求；

电平标准：TTL 标准输入及输出；

封装类型：48-ball miniBGA 或 44-pin TSOP；

以单片 SRAM 为例，其详细电路设计如图 2-4 所示。

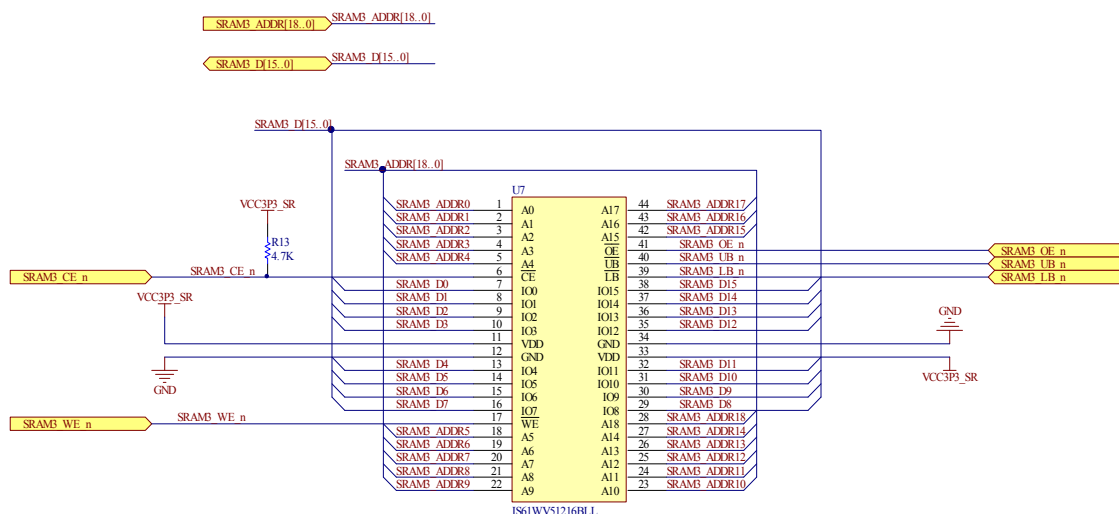


图 2-4 SRAM 存储电路设计

由于 SRAM 芯片引脚为 TTL 电平标准与 FPGA 的 I/O 引脚电平标准一致，SRAM 芯片的数据总线、地址总线及片选、读写使能端等控制线均可以与 FPGA 的 I/O 直接连接。另外值得注意的是，片选端 \overline{CE} 外接的 4.7K 的上拉电阻确保了芯片空闲时片选端口被拉高。

2.4 VGA 显示接口电路设计

VGA(Video Graphics Array, 视频图形阵列)是一种最常见的视频传输标准，由 IBM 在 1987 年提出。VGA 标准支持多种帧频及分辨率格式，并且具有显示速率快、丰富的颜色深度等优点，因此在彩色显示器领域的应用尤其广泛。

VGA 所使用的接口是一种 D 型接口，共 15 个针脚，目前大部分显卡都带有这种类型的接口，如图 2-5 所示。

VGA 接口引脚说明：15 个引脚中，最主要的是 R、G、B 基色信号和场同步 VS、行同步 HS 信号。

由于 R、G、B 基色信号都是模拟信号，而 FPGA 输出为数字信号，故 VGA 接口与 FPGA 之间还需要一个 D/A 转换芯片。本文选择常用的 video DAC 芯片 ADV7123,其输入数据宽度 10 位，刚好等于相机输出数据宽度。



公插头（显示器端）

母插座（PC 显卡端）

VGA 公插头编号

图 2-5 VGA 接口示意图

按照 2.1 节所述，共需要 2 个 VGA 显示接口以进行原始图像和处理后图像的对比显示。以其中一个 VGA 接口为例，其详细电路设计如图 2-6 所示。

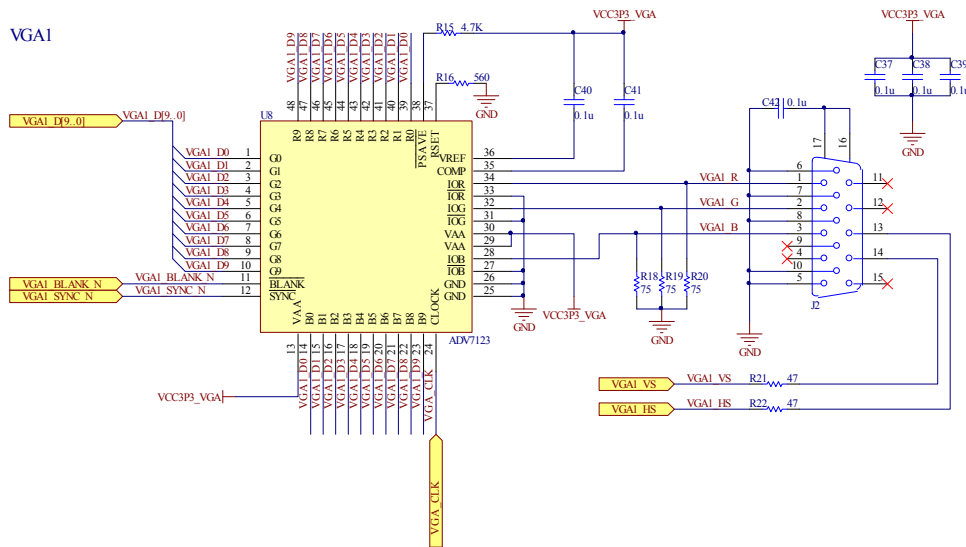


图 2-6 VGA 显示接口电路设计

从图中可以看出，ADV7123 共有 R、G、B 三个颜色的数据输入端口。设计时将这三个数据端口统一接到了一个数据输入端口 VGA_D，这是由于我们所选用的相机为黑白相机。这样相当于提供给 ADV7123 同样大小的三色数字信号，经 ADV7123 数模转换后输出同样大小的 R、G、B 三色模拟信号提供给 VGA 接口，这样在 VGA 显示器上即可合成得到原始的黑白灰度图像。

2.5 RS-232 串口通信电路设计

FPGA 经过图像预处理提取出激光光条信息后，在面对较为复杂的光条曲线拟合、相机标定等算法时显得力不从心；因此需要通信电路将数据发送给上位机进一步处理。鉴于光条数据量不是很大(最大为 $494 \times 10\text{bit}=4940\text{bit}$)，采用典型的且易于实现的 RS232 串口即可满足需求。

RS-232 是一种异步传输标准接口，由电子工业协会 (Electronic Industries Association, EIA) 所制定^[21]。其传送距离最大约 15m，最高速率 20Kb/s。

最新的 RS-232 标准是 RS-232-C，其采用 EIA 电平，其电气特性如下：

在 TXD 和 RXD 上：逻辑 1(MARK)=-3V~-15V；逻辑 0(SPACE)=+3~+15V；在 RTS、CTS、DSR、DTR 和 DCD 等控制线上：信号有效(接通，ON 状态，正电压)=+3V~+25V；信号无效(断开，OFF 状态，负电压)=-3V~-25V。

FPGA 管脚采用 TTL 电平表示逻辑的正负，并不兼容 EIA 电平。故需要专用的电平转换芯片完成信号在 TTL 与 EIA 电平之间的双向转换。电平转换芯片有很多种，本文选择常用的 MAX3232。

另外 RS-232 接口有 9 针与 25 针两种。本文选择 9 针接口，其各引脚定义见表 2-2。

表 2-2 9 针串口引脚定义

引脚名称	管脚定义(描述)	引脚名称	管脚定义(描述)
1) CD	载波检测	6) DSR	数据准备完成
2) RXD	接收数据	7) RTS	发送请求
3) TXD	发送数据	8) CTS	发送清除
4) DTR	数据终端准备完成	9) RI	振铃指示
5) GND	信号地		

详细的电路设计如图 2-7 所示。

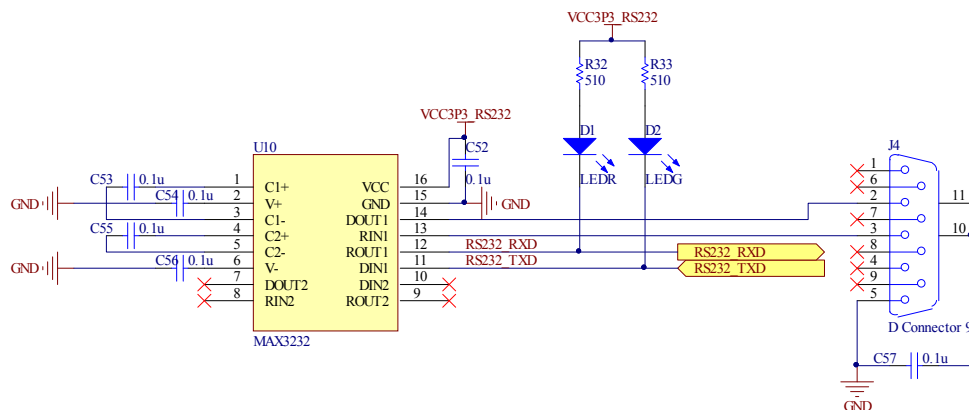


图 2-7 RS232 串口通信电路

为了更加方便地指示通信状态，数据接收端 RXD 和发送端 TXD 与电源之间分别接入了红色、绿色的 LED 指示灯。空闲状态下由于 RXD、TXD 均为高电平，故 LED 处于截止状态。当要接收或发送数据时，由于起始位造成 RXD 或 TXD 变成低电平，LED 导通开始发光，当一个字节的数据发送或接收完毕时，RXD 或 TXD 重新回到高电平，LED 截止。因此，每发送或接收一个字节数据，相应的 LED 闪烁一次。

2.6 FPGA 芯片选择及配置下载电路设计

FPGA 芯片的选择要根据 I/O 引脚、内部逻辑资源、存储资源、乘法器资源甚至功耗需求进行选择。由于课题涉及图像采集、存储及处理等多方面的内容，所需 I/O 引脚数目较大。另外，内部逻辑资源和存储资源的需求取决于图像处理算法的复杂性。此外，成本因素也是需要考虑的重要指标。综合以上几点的考虑，最终选择了 Cyclone III 系列中的 EP3C120F780，其通用 I/O 引脚数目多达 531 个，内部资源如表 2-3 所示。

表 2-3 EP3C120F780 内部资源

器件	逻辑单元	M9K 存储块	RAM Bits	18×18 乘法器	PLL	I/O
EP3C120	119088	432	3981312	288	4	531

Altera 公司的 FPGA 配置方式有多种，根据 FPGA 不同系列又略有不同。Cyclone III 系列 FPGA 为其低成本系列可编程逻辑器件，有诸如 AS、AP、PS、FPP、JTAG 等多种配置方式。除 JTAG 外每种配置方式又区分为标准上电复位时间和快速上电复位时间。本文选择最常用的 JTAG 和 AS 配置方式。

JTAG 配置方式是将程序直接下载至 FPGA，一般用于程序调试。虽然这种方式断电以后程序会丢失，但是充分利用了 FPGA 的无限擦写性。而 AS 配置方式是将程序下载到配置芯片里面，之后 FPGA 每次上电都能自动进行上电配置。结合这两种配置方式的优点，本文采用 JTAG 和 AS 相结合的配置电路，使电路板既能利用 JTAG 接口灵活的对程序进行调试，又能在程序调试完毕时利用 AS 接口永久保存程序。其具体电路如图 2-8 所示。

图中，EPCS64 为 AS 模式下的 FPGA 配置芯片，其存储容量高达 64Mbits。图中两个 5×2 的 header 分别为 AS 与 JTAG 配置时的接口。对 FPGA 配置时根据所选择的配置方式，将 USB Blaster 接入相应的接口即可。值得注意的是，FPGA 的配置方式选择引脚 MSEL3、MSEL2、MSEL1、MSEL0 通过接地或高电平而被连接成了“0010”，即 AS 配置模式。这是由于 JTAG 配置模式优先级高于任何其他配置模式，MSEL 配置方式选择位的接法并不影响 JTAG 配置方

式。

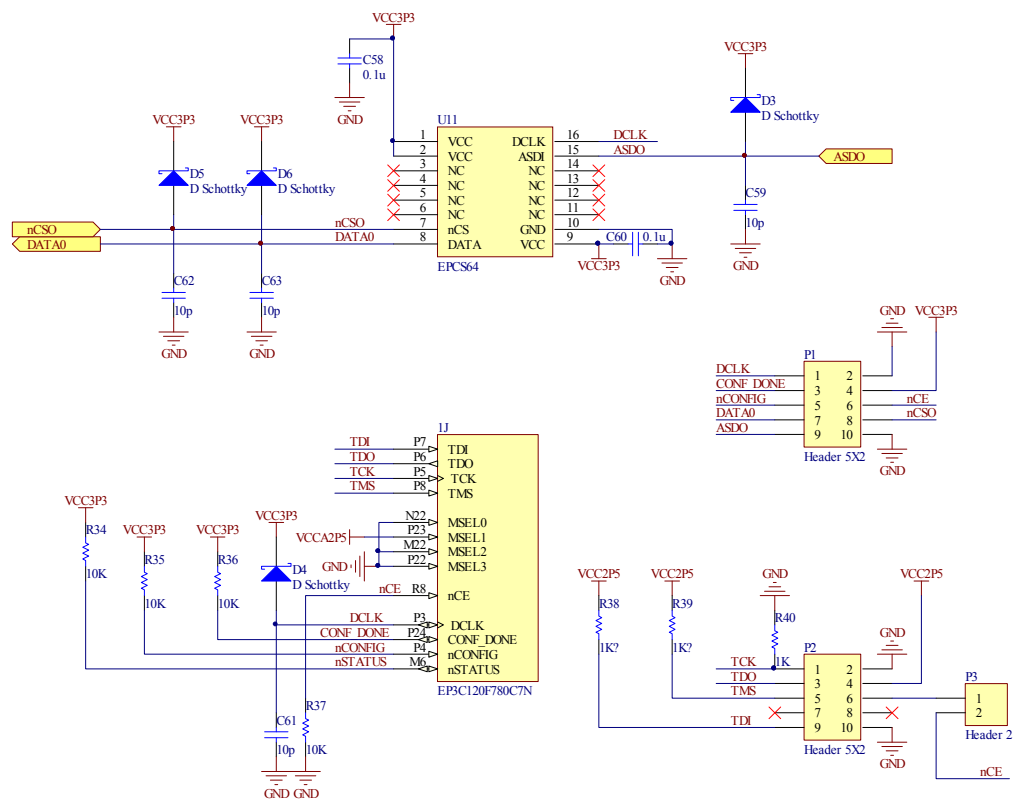


图 2-8 JTAG 与 AS 相结合的 FPGA 配置电路

2.7 电源方案设计

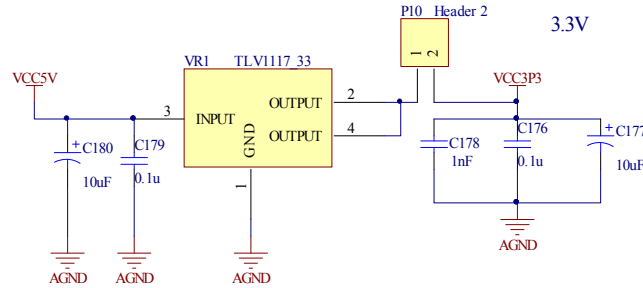
前面所述各模块供电都选择了低压供电方式。统计各个模块芯片所需电源电压，共有 3.3V、2.5V、1.2V 三种电平。由于涉及到的都是高速数字电路，其功耗水平随着工作频率升高而变大，因此需要采用负载能力较强的电源芯片。本文采用了 TLV1117 系列稳压芯片提供 2.5V 和 3.3V 供电。1.2V 电源另外采用 AMS1117 系列。这三款电源芯片输出电流都高达 800mA，均可满足需求。所有的电源芯片都采用常用的外接 5V 电压供电。详细电路设计如图 2-9 所示。

供电时需要考虑对于噪声及电磁干扰的抑制，因此在各个电源入口出均加入了不同容值的滤波电容以消除不同频率的噪声干扰。另外在每个电源的输出处均加入了跳线以作为电源的开关，方便电源模块及其它模块的独立调试。

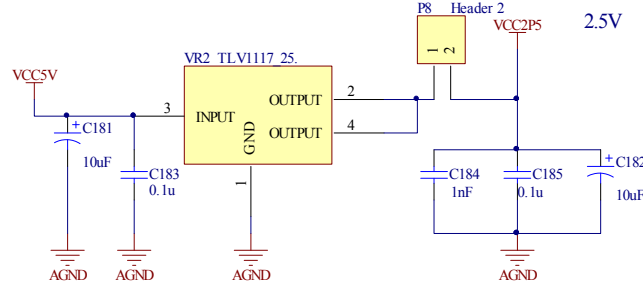
2.8 本章小结

本章完成了图像采集调试平台的电路设计(设计完成的 PCB 电路板见附录

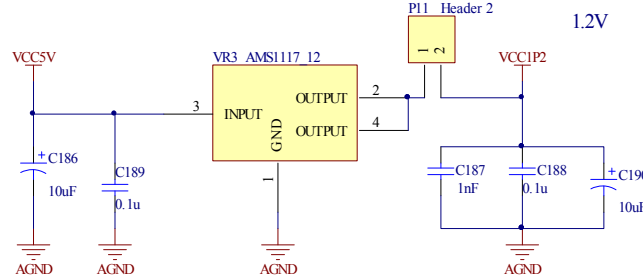
A)。图像采集调试平台包括 Camera Link 接口电路、SRAM 存储电路、VGA 显示接口电路、串口通信电路及 FPGA 配置电路等，为后续的图像实时采集、存储、VGA 实时显示和串口通信等功能的实现奠定了硬件基础。



(a) 3.3V 电源设计



(b) 2.5V 电源设计



(c) 1.2V 电源设计

图 2-9 电源模块电路设计

第 3 章 图像采集调试平台的硬件逻辑设计

本章进行与图像采集调试平台相对应的硬件逻辑设计,以实现相机图像数据的采集、存储、VGA 显示及与上位机的串口通信功能,为之后的图像处理算法提供硬件实现环境。

3.1 图像采集调试平台介绍

上一章设计的硬件电路与 Camera Link 相机、激光器、VGA 显示器等组成完整的图像采集调试平台的示意图如图 3-1 所示,实物平台见附录 B。

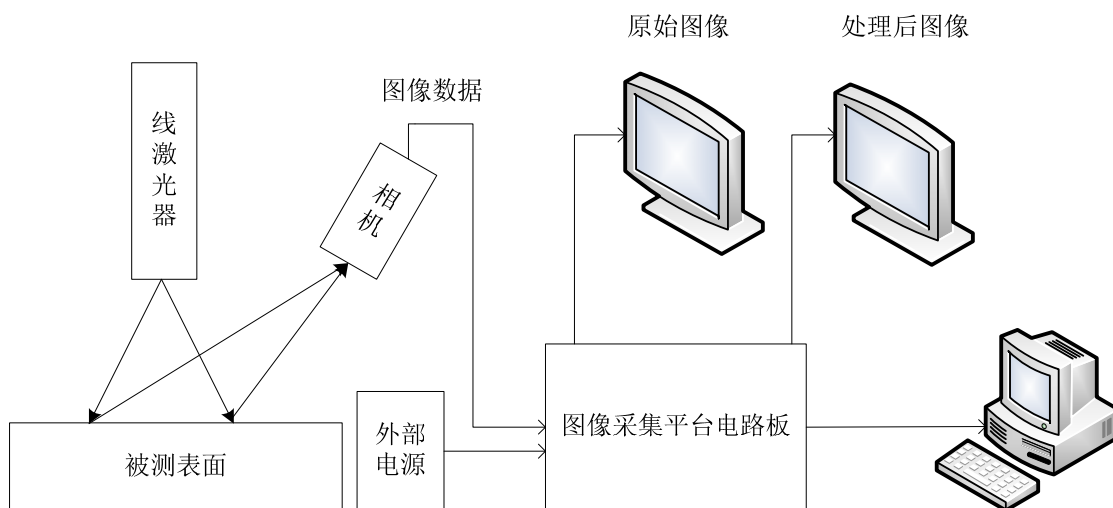


图 3-1 图像采集调试平台示意图

图中,相机实时采集包含激光光条的图像,经过硬件电路处理以后在两个显示器上对比显示原始图像和处理后图像。同时预留的 RS-232 串口负责上位机 PC 与硬件电路板的双向通信。

3.2 硬件逻辑顶层设计

硬件逻辑设计遵循自顶向下的原则,采用 Verilog HDL 语言进行设计。本节阐述硬件逻辑的顶层设计,进行总体功能模块的划分;后续各节对各个功能模块进行详细设计。

根据“按功能划分模块”的思想,整个顶层设计包含 6 大功能模块:PLL 时钟分频模块、图像采集模块、图像存储控制模块、SRAM 读写控制模块、VGA

显示控制模块和 RS232 串口通信模块。顶层硬件逻辑设计原理如图 3-2 所示。

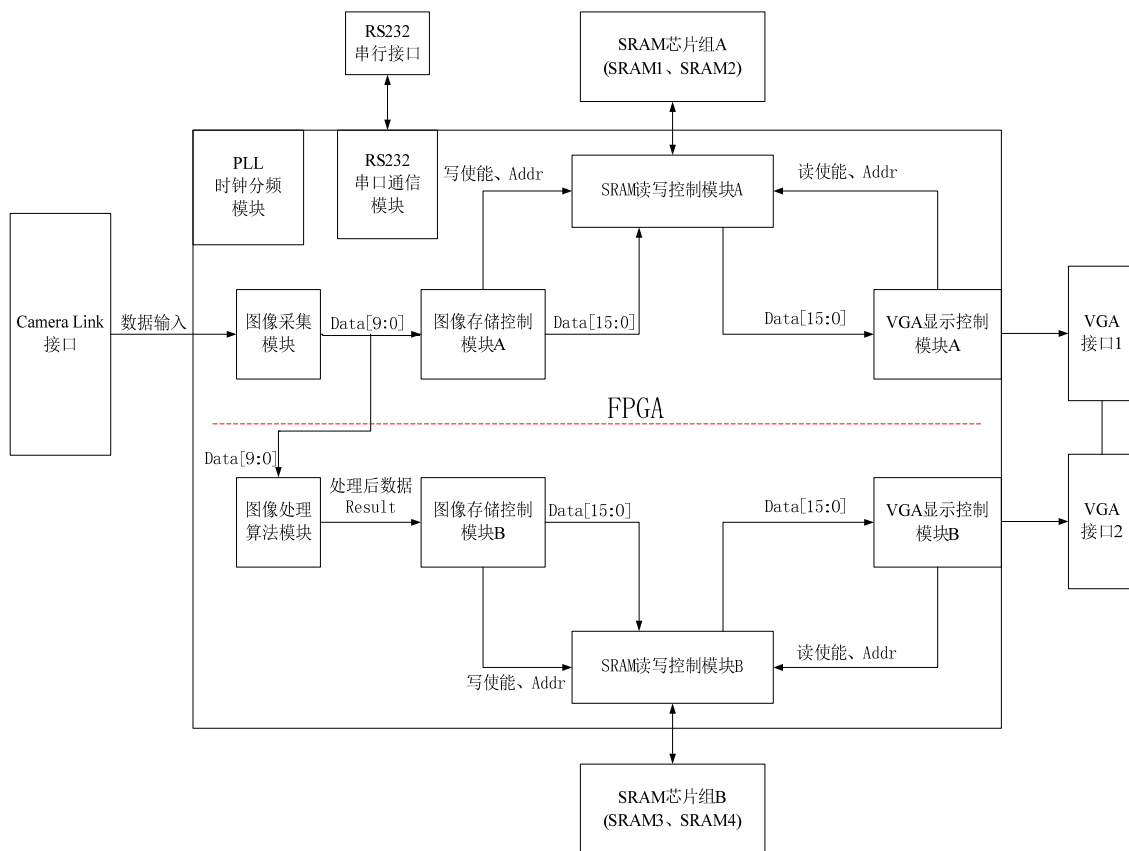


图 3-2 顶层设计原理图

图中，红色虚线上方为相机原始图像的采集、存储、VGA 实时显示及串口通信的硬件逻辑设计；红色虚线下方为处理后图像数据的存储、VGA 实时显示功能的硬件逻辑设计，所以比红色虚线上方的部分多了一个图像处理算法模块，其余部分采用对红色虚线上方硬件逻辑进行“逻辑复制”的方法实现。这样就建立了一个图像处理的硬件实现环境，后续图像处理算法的实现，只需插入到该硬件平台的“图像处理算法模块”部分，即可通过 VGA 显示器实现原始图像和处理后图像的实时对比。

3.3 PLL 时钟分频模块

时钟分频模块负责提供所有逻辑模块的驱动时钟信号。其输入为相机输出的视频时钟，经过 PLL 倍频后，输出其它模块所需要的时钟。PLL 分频模块调用 Altera 公司现成的 IP 核即可。如图 3-3 所示，输入端 $inclk0$ 为相机输出的数据同步时钟，约为 25MHz。输出端 $c0$ 为 $inclk0$ 的 4 倍频，调试时作为在线逻辑分析仪 Signal Tap II 的采样时钟； $c1$ 为 $inclk0$ 的 2 倍频即 50MHz，作

为 VGA 显示的数据同步时钟；c2 为 inclk0 的同频时钟，提供给中心控制模块、数据采集模块等。

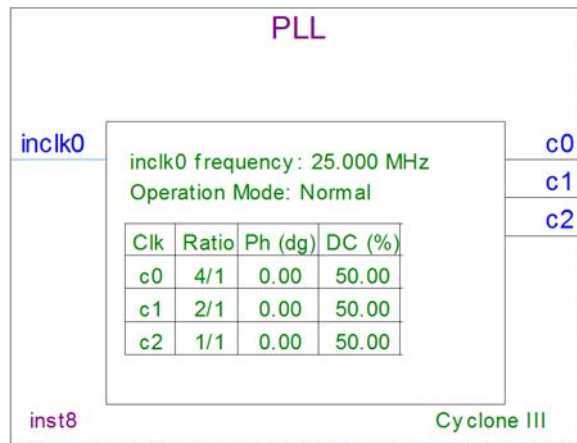


图 3-3 PLL 时钟分频模块

3.4 图像采集模块

图像采集模块完成相机输出图像数据的接收，并输出与图像数据同步的数据有效、行有效、帧有效，以及它们的上升、下降沿信号。相机输出图像数据的顺序如图 3-4 所示，逐行逐个输出像素灰度数据。

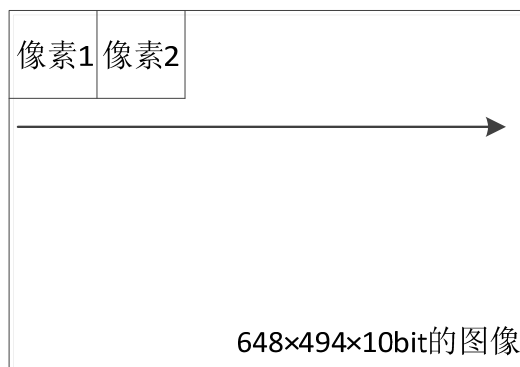
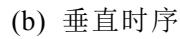
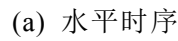


图 3-4 相机输出图像数据顺序

相机输出数据时序分为水平时序和垂直时序，分别如图 3-5(a)和(b)所示。从图中可以看出，相机输出每帧图像以 FVAL 的上升沿为起始，下降沿为结束，每帧图像包含 494 行×648 列有效像素。每行有效数据以 DVAL 的上升沿为起始，以 DVAL 的下降沿为结束。另外，除去有效的 494 行数据，每帧图像包含 6 行的无效行；每个有效行又包含 648 个有效像素数据和 132 个无效数据。



根据以上的时序关系,设计出数据采集模块如图 3-6 所示。其输入输出端口定义如表 3-1 所示。

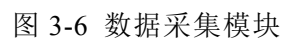
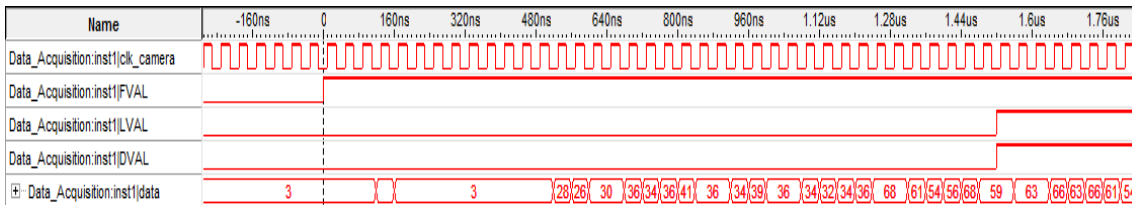


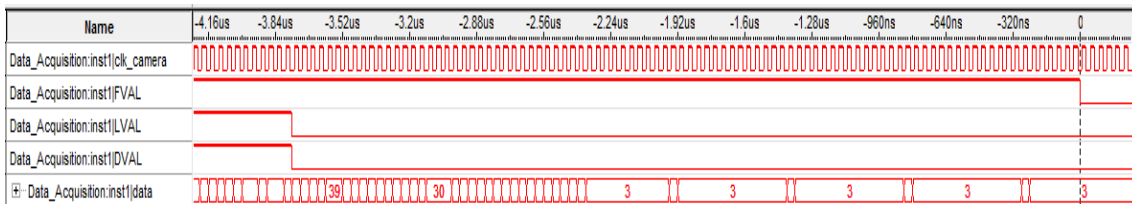
表 3-1 数据采集模块端口定义

输入端	端口定义(描述)	输出端	端口定义(描述)
clk_camera	数据同步时钟	data[9..0]	数据输出
rst_n	复位信号	DVAL	数据有效信号
i_data[9..0]	相机输入 10bit 数据	LVAL	行有效信号
i_DVAL	相机输入数据有效信号	FVAL	帧有效信号
i_LVAL	相机输入行有效信号	row_cnt[9..0]	当前像素行坐标
i_FVAL	相机输入帧有效信号	column_cnt[9..0]	当前像素列坐标
		pos_FVAL	FVAL 上升沿脉冲
		pos_LVAL	LVAL 上升沿脉冲
		pos_DVAL	DVAL 上升沿脉冲
		neg_FVAL	FVAL 下降沿脉冲
		neg_LVAL	LVAL 下降沿脉冲
		neg_DVAL	DVAL 下降沿脉冲
		frame_switch	奇偶帧标志

经过 Signal Tap II 仿真得到的相机输出数据时序如图 3-7 所示。



(a) 帧起始处 FVAL、LVAL、DVAL 与 data 的时序关系



(b) 帧结束处 FVAL、LVAL、DVAL 与 data 的时序关系

图 3-7 相机输出数据时序仿真

(a)图中，在时刻 0 处帧有效信号 FVAL 拉高(一帧的起始)，经过 38 个时钟周期后行有效和数据有效信号 LVAL、DVAL 拉高，数据端 data 开始出现有效的图像灰度数据 59、63、66.....；(b)图中，在时刻 0 处帧有效信号 FVAL

拉低(一帧的结束),在此之前的 94 个时钟周期处行有效和数据有效信号 LVAL、DVAL 同时拉低,表示一帧图像数据输出完毕。

仿真时序图与图 3-5 给出的相机输出时序基本吻合,表明相机输出的图像数据被图像采集模块准确地进行了采集。

3.5 SRAM 读写控制模块

根据 3.1 节所述,每个 SRAM 读写控制模块控制两片 SRAM,交替进行存储和读出操作。因此,SRAM 读写控制模块由两个单 SRAM 芯片读写子模块构成。由于两个 SRAM 读写子模块是采用逻辑复制得到的一模一样的子模块,故本节只阐述单 SRAM 芯片读写子模块的设计。

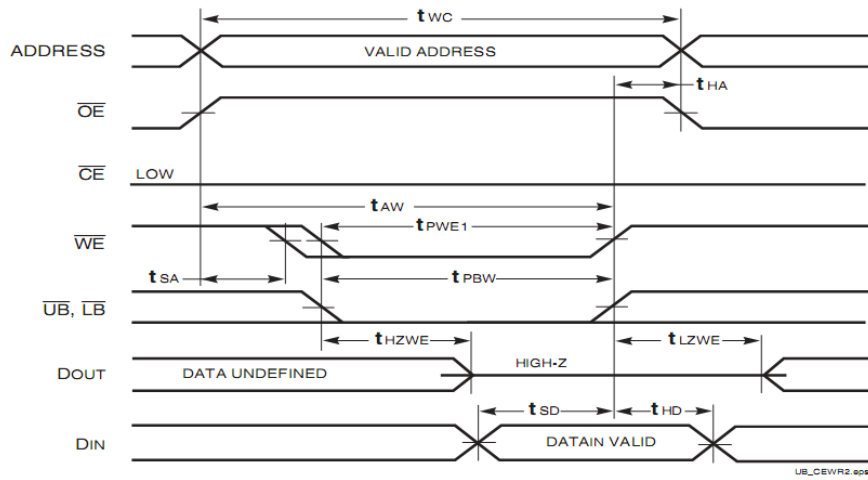
根据在硬件电路设计中选择的 SRAM 芯片 IS61WV51216BLL-10T,其各管脚定义如下表 3-2 所示:

表 3-2 SRAM 芯片管脚定义

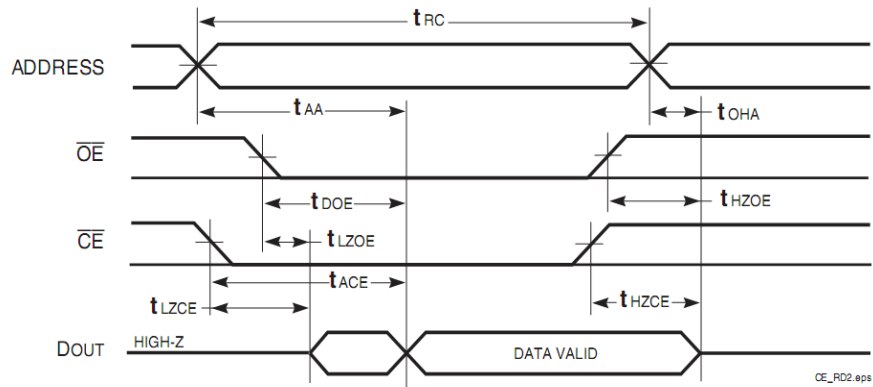
引脚名称	引脚定义(描述)
A0---A18	地址输入
I/O0---I/O15	数据输入/输出
$\overline{\text{CE}}$	片选输入
$\overline{\text{OE}}$	输出使能
$\overline{\text{WE}}$	写使能
$\overline{\text{LB}}$	低字节控制
$\overline{\text{UB}}$	高字节控制
NC	无连接
VDD	电源
GND	地

可以看出,SRAM 引脚包含 5 路控制信号: $\overline{\text{CE}}$ 、 $\overline{\text{OE}}$ 、 $\overline{\text{WE}}$ 、 $\overline{\text{LB}}$ 及 $\overline{\text{UB}}$ 。只有遵守 SRAM 读写时序,才可以准确存取有效的图像数据,其读写时序如图 3-8 所示。

从 SRAM 读写时序图可以看出,SRAM 为异步器件,并没有读和写的同步时钟信号,这并不符合同步设计思想。于是,在设计 SRAM 读写逻辑时应加入读和写时钟信号,以使数据有节奏地进行存取。根据以上时序要求,设计出单 SRAM 芯片读写子模块如图 3-9 所示,其端口定义见表 3-3。



(a) 写 SRAM 时序图



(b) 读 SRAM 时序图

图 3-8 SRAM 读写时序图

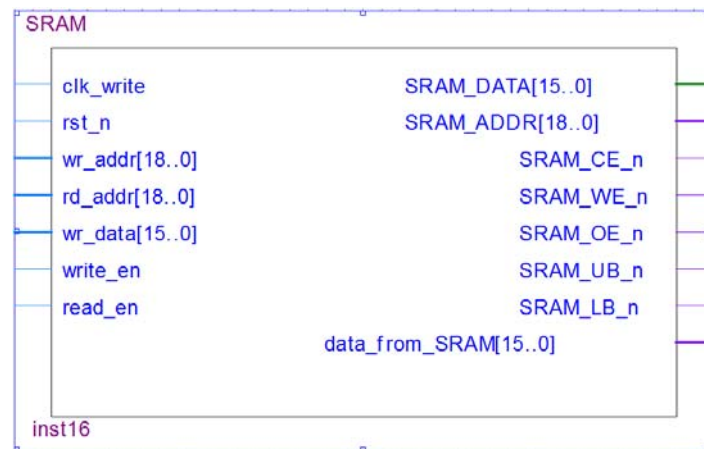
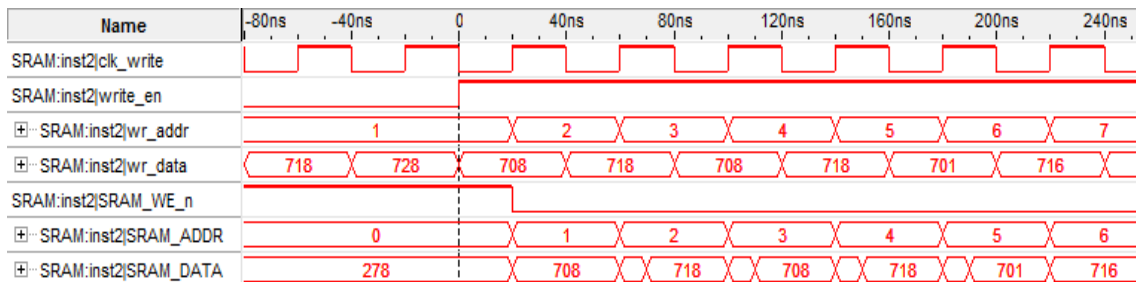


图 3-9 SRAM 读写子模块

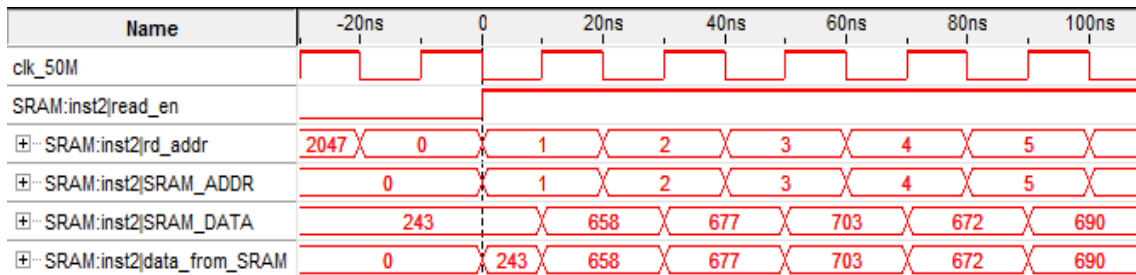
表 3-3 SRAM 读写子模块端口定义

输入端	端口定义(描述)	输出端	端口定义(描述)
clk_write	写操作时钟	SRAM_DATA	SRAM 数据总线
rst_n	复位	SRAM_ADDR	SRAM 地址总线
wr_addr	写操作地址	SRAM_CE_n	SRAM 片选线
rd_addr	读操作地址	SRAM_WE_n	SRAM 写控制线
wr_data	要写入的数据	SRAM_OE_n	SRAM 读控制线
write_en	写使能	SRAM_UB_n	SRAM 高字节控制线
read_en	读使能	SRAM_LB_n	SRAM 低字节控制线
		data_from_SRAM	从 SRAM 读出的数据

经过 Signal Tap II 仿真得到 SRAM 读写时序如图 3-10 所示。



(a) SRAM 写时序



(b) SRAM 读时序

图 3-10 SRAM 读写模块仿真时序图

如图 3-9(a)所示, 输入端在写使能 write_en 拉高后, 相继向地址 1、2、3... 写入 708、718、708...; 同时 SRAM 的地址和数据总线上在 SRAM_WE_n 拉低后相继出现上述数据。同样, 如图 3-9(b)所示, 输入端在读使能 read_en 拉高后, 紧随地址线上给出地址 1、2、3...后, 数据总线上传出数据 658、677、703..., data_from_SRAM 与总线上数据一致。

SRAM 读写模块的仿真结果表明，FPGA 对 SRAM 芯片进行了正确的写入和读出操作。

3.6 图像存储控制模块

图像存储控制模块衔接图像采集模块和 SRAM 读写模块，实现数据带宽转换、提供 SRAM 写入地址、写入数据、写使能及片选等信号。每个图像存储控制模块控制两片 SRAM 之间的乒乓存储操作。其硬件模块图如图 3-11 所示。其端口定义如表 3-4 所示。

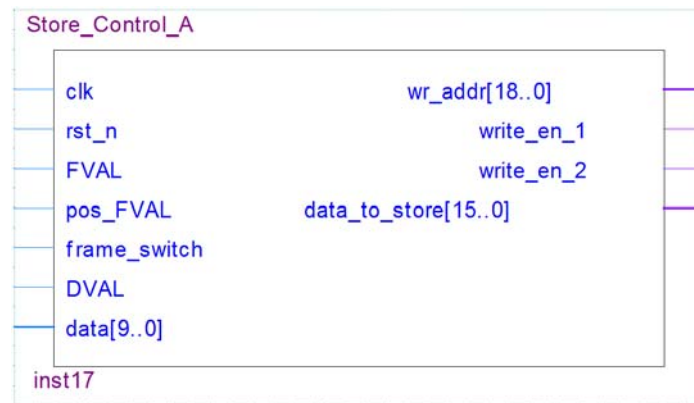


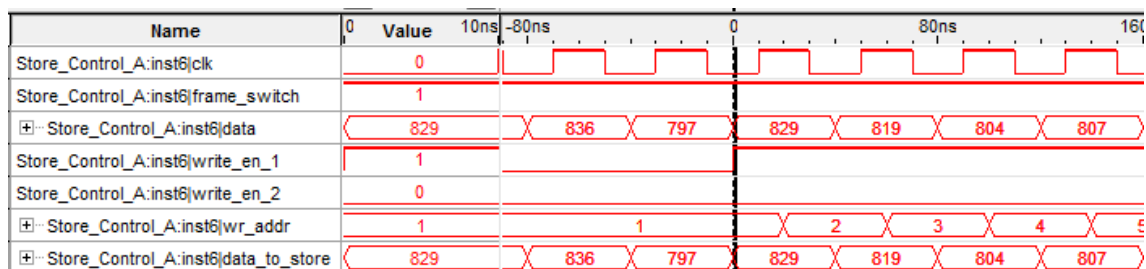
图 3-11 图像存储控制模块

表 3-4 图像存储控制模块端口定义

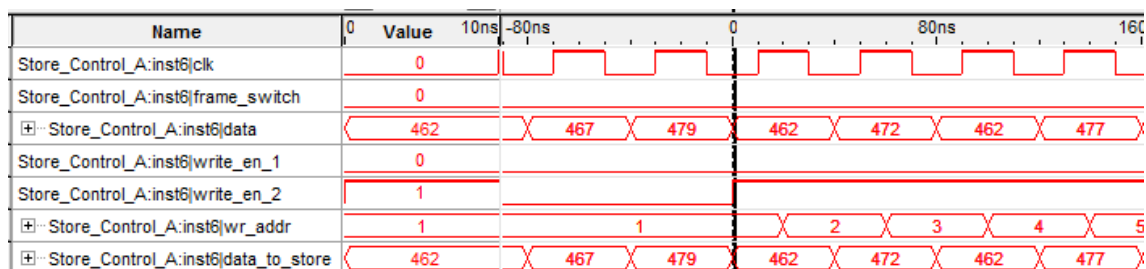
输入端	端口定义(描述)	输出端	端口定义(描述)
clk	时钟	wr_addr	写操作地址
rst_n	复位	write_en_1	SRAM1 写使能
FVAL	帧有效信号	write_en_2	SRAM2 写使能
pos_FVAL	FVAL 上升沿脉冲	data_to_store	待存储的数据
frame_switch	奇偶帧标志		
DVAL	数据有效信号		
data	相机输出数据		

经过 Signal Tap II 仿真得到存储控制模块时序如图 3-12 所示。

(a)图中，奇偶帧标志 frame_switch 为高，在时刻 0 时 SRAM 芯片 1 的写使能 write_en_1 拉高，写地址 wr_addr 由 1 开始递增，数据 829、819、804..... 相继存入 SRAM 芯片 1。同样(b)图中，奇偶帧标志 frame_switch 为低，数据存入了 SRAM 芯片 2。



(a) 奇数帧图像存储控制时序



(b) 偶数帧图像存储控制时序

图 3-12 图像存储控制时序仿真图

3.7 VGA 显示控制模块

VGA 显示空中模块将图像数据从 SRAM 中取出送给 VGA 的数据端口，并提供 VGA 扫描的场、行时序信号。

由于相机输出图像(默认状态下)大小为 648×494 ，欲完整显示相机图像，至少选用 800×600 的分辨率。常用 800×600 分辨率的 VGA 刷新频率有 56Hz、60Hz、72Hz、75Hz 和 85Hz，不同的刷新频率对应不同的时序参数。本文选择了 72Hz 的刷新频率，查阅 VSEA 制定的《Monitor Timing Standard》可得到其详细时序参数如表 3-4 所示。其 VGA 时序如图 3-13 所示。

表 3-4 $800 \times 600 @ 72\text{Hz}$ 的 VGA 时序参数

名称	同步脉冲	后沿	显示脉冲	前沿	帧长
行脉冲	120	64	800	56	1040
列脉冲	6	23	600	37	666

整个 VGA 显示控制模块设计可以细分为 VGA 驱动模块和 VGA 控制模块。其中 VGA 驱动模块负责提供 VGA 接口所需的行、场驱动信号及 R、G、B 基色信号；VGA 控制模块负责衔接 SRAM 读写模块和 VGA 驱动模块，提供 SRAM 读地址、读使能等信号以从 SRAM 读取所需数据，同时完成数据的

带宽转换。根据 VGA 时序要求，设计 VGA 驱动模块和 VGA 控制模块如图 3-14 所示，端口定义见表 3-5。

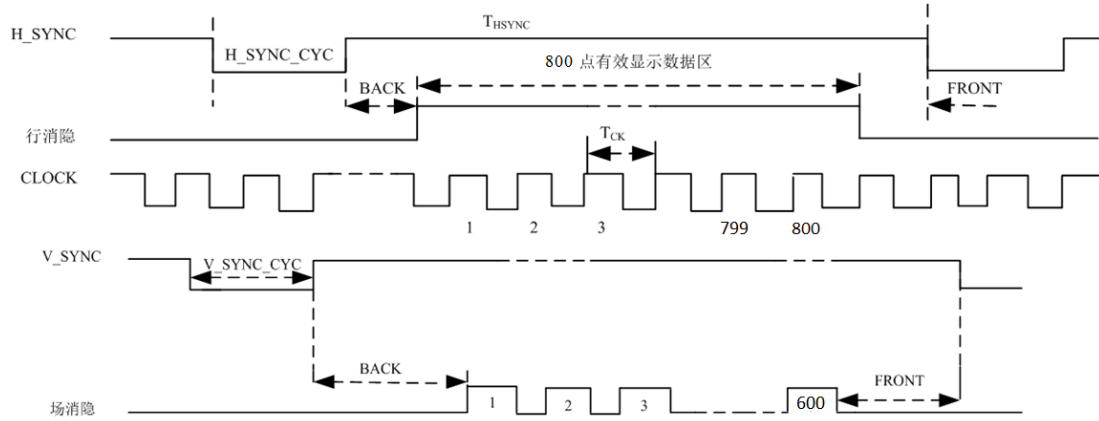
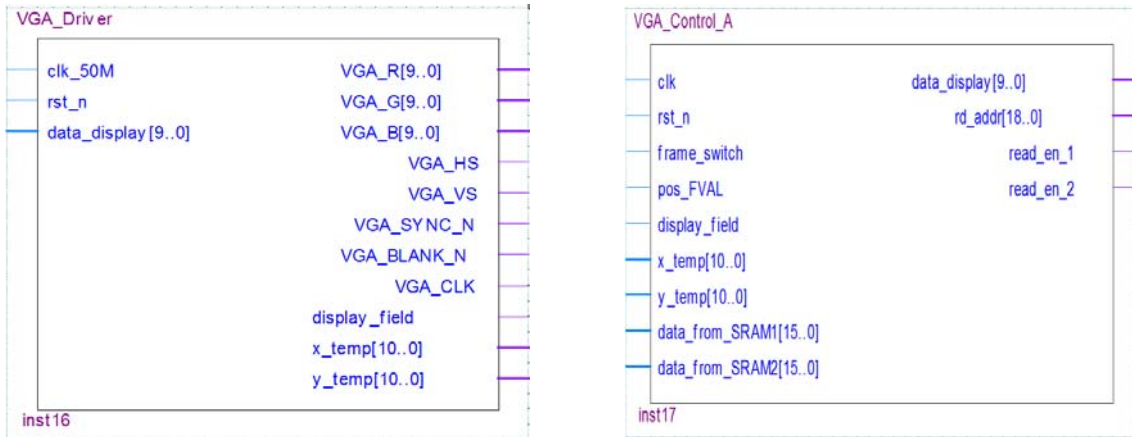


图 3-13 800×600 分辨率的 VGA 扫描时序



(a) VGA 驱动模块

(b) VGA 控制模块

图 3-14 VGA 显示控制模块视图

经过 Signal Tap II 仿真得到 VGA 显示控制模块时序关系如图 3-15 所示。

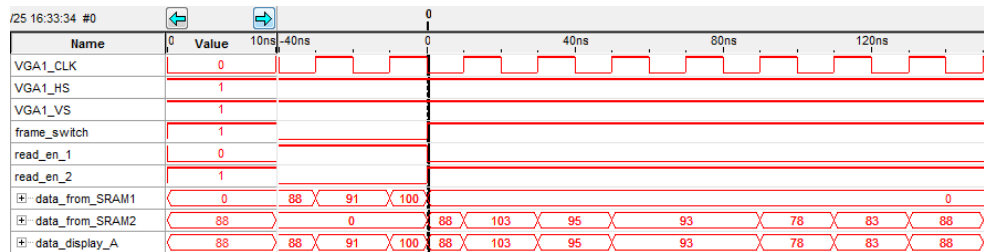


图 3-15 VGA 显示控制时序仿真

如图，在时刻 0 处，奇偶帧切换标志位 frame_switch 拉高，SRAM 芯片 2

的读使能 read_en2 拉高，从 SRAM2 读取数据 88、103、95...,经过数据带宽转换，送给了 data_display。

表 3-5 VGA 显示模块端口定义
(a) VGA 驱动模块端口定义

输入端	端口定义(描述)	输出端	端口定义(描述)
clk_50M	50MHz 时钟	VGA_R	红色基色信号
rst_n	复位	VGA_G	绿色基色信号
data_display	待显示图像数据	VGA_B	蓝色基色信号
		VGA_HS	行同步
		VGA_VS	场同步
		VGA_SYNC_N	同步控制
		VGA_BLANK_N	消隐控制
		VGA_CLK	VGA 接口扫描时钟
		display_field	有效区域标志
		x_temp	当前扫描行位置
		y_temp	当前扫描列位置

(b) VGA 控制模块端口定义

输入端	端口定义(描述)	输出端	端口定义(描述)
clk	时钟	data_display	待显示图像数据
rst_n	复位	rd_addr	读地址
frame_switch	奇偶帧标志	read_en_1	读使能 1
pos_FVAL	FVAL 上升沿	read_en_2	读使能 2
display_field	有效区域标志		
x_temp	当前扫描行位置		
y_temp	当前扫描列位置		
data_from_SRAM1	从 SRAM1 读出的数据		
data_from_SRAM2	从 SRAM2 读出的数据		

3.8 RS-232 串口通信模块

RS232 串口模块用于上位机与调试平台之间的双向通信，所以串口通信模块包含两个独立的子模块：串口接受模块和串口发送模块。

典型的串口数据帧格式包含 1 位起始位，7 或 8 位数据位，1 位奇偶校验

位和 1~2 位停止位^[22]。一般而言，奇偶校验位可以省略。本文设计的串口通信逻辑，采用了 1 位起始位+8 位数据位+1 位停止位的数据帧格式，如图 3-15 所示。

串口传输常用的波特率有 9600bps、19200bps、38400bps、57600bps 及 115200bps。所谓波特率即每秒钟传输的位数。为了适应不同的传输速率需求，在硬件逻辑设计时应加入对波特率进行设置的模块。

3.8.1 串口接收子模块

串口接收数据的过程如下：如图 3-16 所示，为了使采集的数据最为稳定，每次采样都在数据位的中间时刻采集。其中 rs232_rx 是串口接收端电平变化；clk_bps 是波特率采样时钟。对每一帧数据接收时，先检测 rs232_rx 端口的低电平(起始位)并忽略该位，然后开始对之后的连续 8 个 bit 位采样，最后的停止位同样采取忽略。如此，8 个 bit 位组成一帧字节，保存到 rx_data 寄存器并输出。

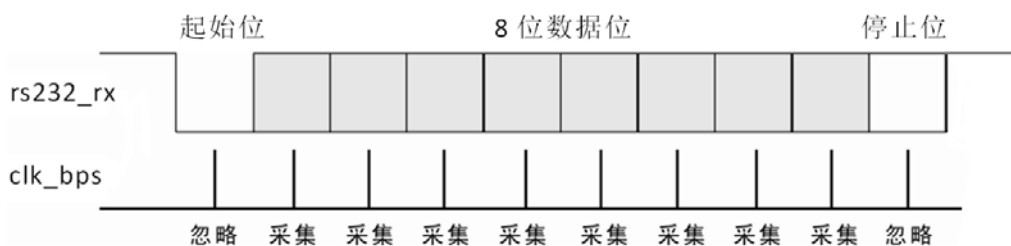


图 3-16 串口接收端的数据采样

设计封装好的串口接收子模块如图 3-17 所示，其端口定义见表 3-6。

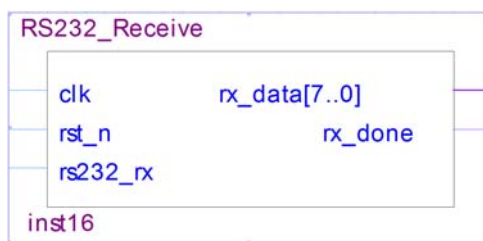


图 3-17 串口接收子模块

表 3-6 串口接收子模块端口定义

输入端	端口定义（描述）	输出端	端口定义（描述）
clk	时钟	rx_data	接收到的数据
rst_n	复位	rx_done	接收一帧完毕(一个周期高电平)
rs232_rx	串口接收端		

串口接收子模块的 RTL(Register-Transfer-Level)视图如图 3-18 所示。从图 3-17 可以看出, 串口接收模块进一步向下细分为波特率发生模块(波特率设置)和数据接收模块。波特率发生模块负责对波特率进行设置, 并产生波特率采样时钟; 数据接收模块在波特率采样时钟的节拍下对数据接收端口进行采样, 并保存输出接收到的数据。

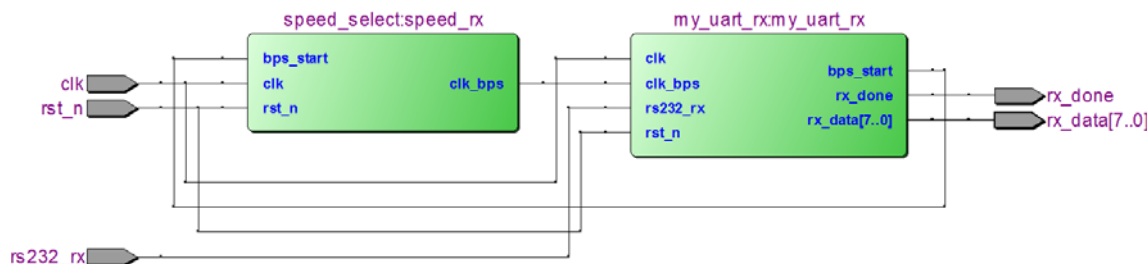
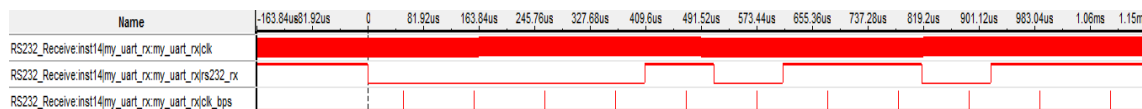


图 3-18 串口接收子模块 RTL 视图

在不同的时钟频率下, 每一种波特率对应不同的参数。本文下面的串口接收和发送实验, 均采用 50MHz 的时钟频率, 波特率初始值设置为 9600bps。

上位机使用串口调试助手, 向调试平台发送 16 进制数“68H”(即 01101000B)。经过 Signal Tap II 仿真验证得到串口接收端接收到数据如图 3-19 所示。图(a)中, 在每一个 clk_bps 高电平脉冲时对 $rs232_rx$ 端上的数据进行采样, 连续采集得到“00001011011”, 取第一个 0 后面 8 位有效数据, 即“00010110”。由于串口数据格式是低位到高位接收(发送), 故接收到的数据为“01101000”, 即“68H”。从(b)图可以看出, 当数据接收完毕时, rx_done 拉高了一个时钟周期的高电平。



(a) 串口接收端数据



(b) 串口接收完毕

图 3-19 串口接收时序仿真图

3.8.2 串口发送子模块

串口数据的发送过程与接收过程类似, 如图 3-20 所示: 发送模块一旦检测到有需要发送的数据, 则开始进行计数定时产生波特率时钟, 在波特率时钟

的节拍下，数据发送端 rs232_tx 逐位发送数据：首先发送起始位 0，之后发送 8 位数据位，最后发送停止位。封装好的串口发送子模块如图 3-21 所示，其端口定义见表 3-7。

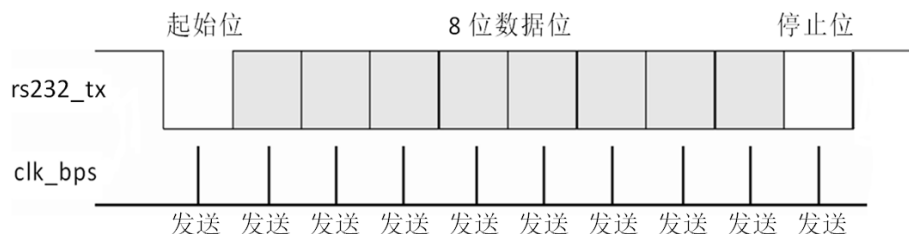


图 3-20 串口发送过程



图 3-21 串口发送子模块

表 3-7 串口发送子模块端口定义

输入端	端口定义(描述)	输出端	端口定义(描述)
clk	时钟	rs232_tx	串口发送端
rst_n	复位	tx_done	发送一帧完毕(一个周期高电平)
data_to_send	待发送数据输入		
data_valid	数据输入有效		

同样，串口发送子模块向下细分为波特率发生模块和数据发送模块，其内部 RTL 视图如图 3-22 所示。

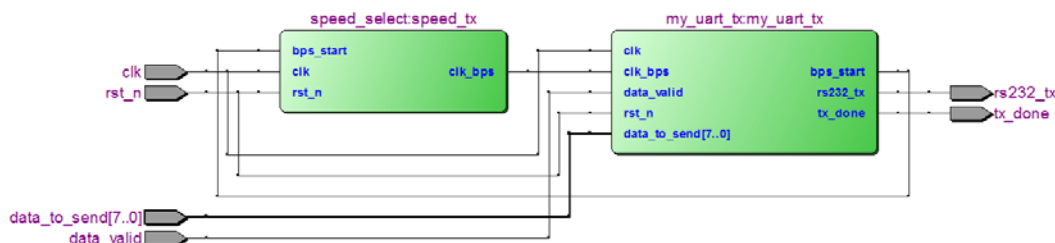


图 3-22 串口发送子模块 RTL 视图

为了验证串口发送子模块，利用之前串口接收模块接收到的数据反发送给

上位机。实验时，通过串口调试助手不断地发送 68，得到串口发送端仿真波形数据如图 3-23 所示。在 0 时刻，data_valid 拉高一个时钟周期，待发送数据“68H”输入，之后数据发送端口 rs232_tx 上相继出现“00001011011”，取第一个 0 之后的 8 位有效数据，即“00010110”。同样，串口发送也是从低位到高位发送，故发送的真实数据位“01101000”即“68H”。

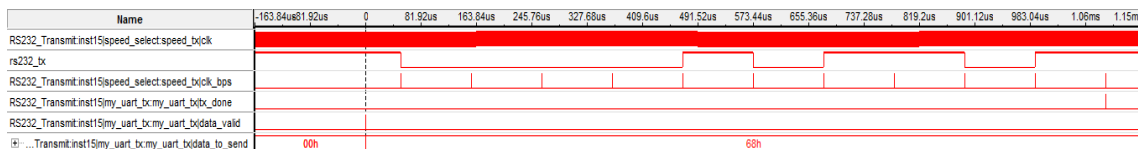


图 3-23 串口发送模块仿真波形

3.9 本章小结

本章在上一章硬件电路平台的基础上，进行了图像采集、存储、VGA 显示和串口通信等的硬件逻辑设计。经过大量的实验和仿真，成功实现了相机图像的采集和 VGA 实时显示，并利用上位机(PC)与调试平台进行了串口通信，为后续的图像处理算法的实现提供了硬件环境。

第 4 章 典型空间域图像增强算法的硬件实现

图像增强方法包括两大类：空间域方法和频域方法^[23,24]。

“频域”处理往往先将图像进行傅里叶变换，对变换后的图像进行一定的处理后经过傅里叶逆变换的到处理结果。由于这种间接的处理方法涉及到复杂的变换函数，硬件实现起来非常困难，本文不做研究。

“空间域”即图像平面本身，这类方法对图像中的像素直接进行处理^[23,25]。空间域图像处理的定义式如下：

$$g(x, y) = T[f(x, y)] \quad (4-1)$$

其中， (x, y) 代表图像中某个像素的坐标位置； $f(x, y)$ 代表输入图像； $g(x, y)$ 代表处理后的图像； T 代表对输入图像的某种处理过程，其定义在 (x, y) 的邻域。

图像增强预处理算法包括图像灰度变换、直方图修正、空域平滑和锐化处理、伪彩色处理等^[26,27]。本章主要研究图像空域平滑和锐化处理算法。无论是空域平滑处理，还是空域锐化处理，都属于空间滤波的范畴。其大致原理如下：设置一个 $m \times n$ 的掩膜，在待处理图像上逐点移动该掩膜，图像中每一点 (x, y) 处的响应为滤波器掩膜作用于滤波掩膜扫过区域的相应像素后的结果。

硬件算法的实现不同于软件算法。硬件上要对一副图像进行处理，需要将待处理数据分次取出，而不能像软件算法一样一次读入整幅图像数据。因此，在实现滤波算法之前，需要做一个滤波窗口来进行数据的读入和输出。当图像灰度数据被组织成滤波窗口的形式后，利用掩膜算子与滤波窗口内的数据进行运算得到输出相应。因此本章将首先介绍 3×3 滤波窗口的硬件实现，然后在此基础上硬件实现典型的空域滤波算法：中值滤波、拉普拉斯滤波和 Sobel 滤波。

4.1 3×3 滤波窗口实现

如图 4-1 所示， 3×3 滤波窗口中是以 $f(x, y)$ 为中心的 9 个像素灰度值，空间滤波操作都是在此窗口的基础上进行的。

如图 4-2 所示，像素数据在时钟节拍下逐个输入 $648 \times 10\text{bit}$ 移位寄存器(由于所用相机输出为 $648 \times 494 \times 10\text{bit}$)。当相机输出的第三行图像数据存入了 Line3，第一行和第二行数据正好分别存入到 Line1 和 Line2。此时，再经过三个时钟周期， 3×3 窗口内正好存入第一组有效像素值，之后 3×3 窗口内数据每

个时钟周期都会刷新一次。

$f(x-1, y-1)$ (R_{11})	$f(x, y-1)$ (R_{12})	$f(x+1, y-1)$ (R_{13})
$f(x-1, y)$ (R_{21})	$f(x, y)$ (R_{22})	$f(x+1, y)$ (R_{23})
$f(x-1, y+1)$ (R_{31})	$f(x, y+1)$ (R_{32})	$f(x+1, y+1)$ (R_{33})

图 4-1 3×3 滤波窗口

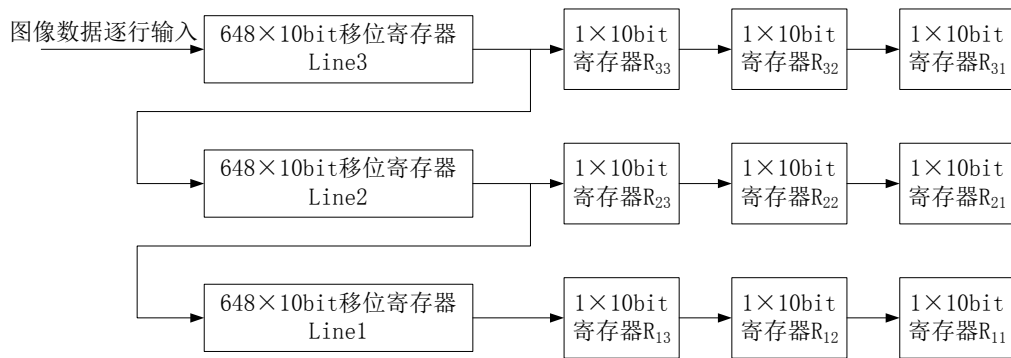


图 4-2 3×3 滤波窗口数据输入

由 Verilog HDL 语言编写 3×3 窗口模块,生成的模块化视图如图 4-3 所示。其中,输入端: clk_camera 是控制时钟, en 是模块使能, rst_n 是复位控制, data[9..0]是 10bit 的像素数据输入;输出端: R11、R12、.....R33 是 3×3 窗口数据。

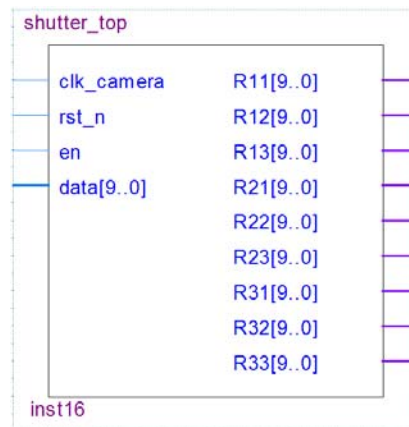


图 4-3 3×3 窗口模块

4.2 中值滤波

4.2.1 算法原理

中值滤波是将像素邻域内灰度的中值代替该像素的值^[28]，它对处理椒盐噪声非常有效，是一种典型的非线性空间滤波器。中值滤波定义式如下：

$$g(x, y) = \text{med}\{f(x-k, y-l)\}, (k, l) \in W \quad (4-2)$$

式中， $g(x, y)$ 为输出灰度值， $f(x-k, y-l)$ 为输入图像灰度值。 W 为掩膜窗口，可以取 3×3 窗口或者 5×5 窗口。本文取 3×3 窗口进行试验。

传统的中值滤波算法基于冒泡法对窗口内所有数据进行比较得到中值输出。如 3×3 窗口的中值滤波，每移动一次窗口需要进行 $9 \times (9-1) \div 2 = 36$ 次比较^[29]。这在硬件上实现起来需要耗费过多时钟和流水级寄存器。

中值滤波的改进算法有多种。例如基于不完全排序思想的冒泡法排序：其对于窗口内像素比较到第 $n/2$ 个像素时停止比较即可得到中值。对于 3×3 窗口需要 $8+7+6+5+4=30$ 次比较^[30]。又如基于统计思想的中值滤波算法：选取窗口内任一灰度值 D_0 ，统计窗口内大于和小于 D_0 的像素个数，分别记为 Large 和 Little 。当 $\text{Large}=\text{Little}$ 时 D_0 即为窗口中的中值，否则选取下一个像素作为 D_0 。但是这种算法过于依赖时钟信号对 Large 和 Little 进行计数，需要消耗过多时钟周期^[31, 32]。

另一种经典的中值滤波改进算法如下：首先对窗口内像素进行行排序，再进行列排序，最后取副对角线上的中值即为窗口中值^[11]。其过程如图 4-4 所示：每次行或列排序需要 3 次比较，再加上一次对角线排序，共需要 $3 \times 3 + 3 \times 3 + 3 = 21$ 次比较。

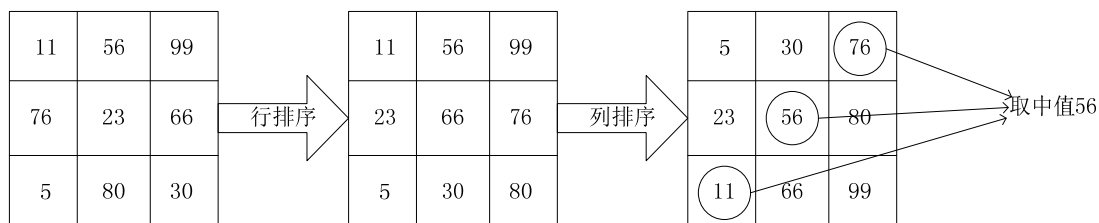
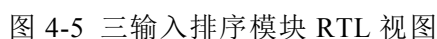


图 4-4 改进的中值滤波排序过程

4.2.2 算法的硬件实现

在硬件算法实现上，首先设计了一个三输入排序模块，其 RTL 视图如图 4-5 所示。

经过一个时钟周期即可得到三个输入值的最大、最小和中值。因此，整个窗口内求取中值的过程仅仅需要三个时钟周期(行排序、列排序和对角线排序各一个时钟周期)。整个排序过程需将三输入排序模块做 7 次相同的例化。从



06 16:28:55 #0		-14.58us										+120ns									
Name	-14.58us	Value	-14.57us	-14.6us	-14.56us	-14.52us	-14.48us	-14.44us	-14.4us	-14.36us	-14.32us	-14.28us	-14.24us	-14.2us	-14.16us	-14.12us					
sequencing:inst6 clk		1																			
sequencing:inst6 R11		44		46	44	56	54	49	51	49				51							
sequencing:inst6 R12		56		44	56	54	49	51		49		51									
sequencing:inst6 R13		54		56	54	49	51		49		51			56							
sequencing:inst6 R21		51		46	51	54		51		54	46		54								
sequencing:inst6 R22		54		51	54		51		54	46		54		56							
sequencing:inst6 R23		51		54		51	54	46		54		59	56								
sequencing:inst6 R31		44		49	44	56	44	49	46	41	44	46	46								
sequencing:inst6 R32		56		44	56	44	49	46	41	44	46	55									
sequencing:inst6 R33		44		56	44	49	46	41	44	46	59	45									
sequencing:inst6 mid_result		49		51	49	46	49		51				45								

图 4-6 中值滤波算法仿真时序图

可以验证算法的正确性：在-14.58 μ s处，3 \times 3窗口数据输入44、56、54、51、54、51、44、56、44，3个时钟后期后（120ns）得到计算结果mid_result=51；其余数据的中值滤波结果均正确。

4.2.3 硬件算法性能分析

经过硬件中值滤波处理后的图像和原始图像的对比效果如图 4-7 所示。

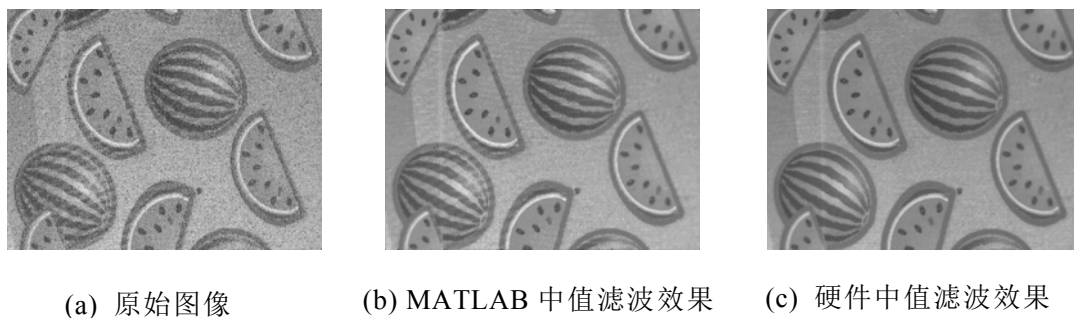


图 4-7 中值滤波对比效果

滤波效果：硬件中值滤波的效果非常明显，原始图像中的椒盐噪声得到了明显的抑制，和采用 MATLAB 处理结果几乎一致。

处理速度：由于硬件和软件算法实现所使用的处理器时钟频率差异巨大，比较硬件和软件算法的处理速度应该通过它们各自消耗的时钟周期来评价，而不是处理时间。本文所选择的中值滤波算法，包括 3 次行排序，3 次列排序和 1 次对角线排序。采用硬件实现时，3 次的行排序和列排序均可以并行执行，分别用 1 个时钟周期完成，总计需要 3 个时钟即可得到一次中值计算；而采用软件实现时，每次排序均需要至少耗费一个时钟，故总计需要耗费 7 个时钟周期。对于整幅 648×494 大小的图像，中值滤波算法采用硬件实现要比软件少耗费 $(7-3) \times 648 \times 494 = 1280448$ 个时钟周期。

4.3 拉普拉斯滤波

4.3.1 算法原理

拉普拉斯滤波是一种重要的图像增强算法，其滤波器的响应与滤波器作用图像的突变方向无关，具有各项同性的滤波特性，因此被广泛用于图像锐化和高频增强等处理中^[33]。

一个二元图像函数的拉普拉斯变换定义为^[34]：

$$\nabla^2 f = \frac{\partial^2 f}{\partial^2 x} + \frac{\partial^2 f}{\partial^2 y} \quad (4-3)$$

在 x 方向和 y 方向上对二阶偏微分采用下列定义：

$$\frac{\partial^2 f}{\partial^2 x} = f(x+1, y) + f(x-1, y) - 2f(x, y) \quad (4-4)$$

$$\frac{\partial^2 f}{\partial^2 y} = f(x, y+1) + f(x, y-1) - 2f(x, y) \quad (4-5)$$

将式(4-4)和(4-5)带入(4-3)可得：

$$\nabla^2 f = [f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1)] - 4f(x, y) \quad (4-6)$$

如果将对角线方向也纳入拉普拉斯变换定义式中，则式(4-6)变为^[35]：

$$\begin{aligned} \nabla^2 f = & [f(x-1, y-1) + f(x, y-1) + f(x+1, y-1) \\ & + f(x-1, y) + f(x+1, y) + f(x-1, y+1) \\ & + f(x, y+1) + f(x+1, y+1)] - 8f(x, y) \end{aligned} \quad (4-7)$$

式(4-6)和(4-7)分别为 4 邻域和 8 邻域的拉普拉斯算子，表示成掩膜形式如图 4-8 所示。下面将就这两种掩膜分别用硬件算法实现，以对比其处理效果。

0	1	0
1	-4	1
0	1	0

(a) 4 邻域拉普拉斯算子掩膜

1	1	1
1	-8	1
1	1	1

(b) 8 邻域拉普拉斯算子掩膜

图 4-8 拉普拉斯算子掩膜

4.3.2 算法的硬件实现

实际应用中，为保证式(4-6)及(4-7)的计算结果为正值，通常给计算结果取绝对值。这时，式(4-6)变为：

$$\nabla^2 f = |f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)| \quad (4-8)$$

式(4-7)变为：

$$\begin{aligned} \nabla^2 f = & |f(x-1, y-1) + f(x, y-1) + f(x+1, y-1) \\ & + f(x-1, y) + f(x+1, y) + f(x-1, y+1) \\ & + f(x, y+1) + f(x+1, y+1) - 8f(x, y)| \end{aligned} \quad (4-9)$$

硬件实现时，为保证计算结果为正值，需要加入比较判断逻辑，若被减数大于减数，正常做减法；如果被减数小于减数，则被减数与减数互换后再进行减法运算。

下面以 4 邻域的拉普拉斯算子为例介绍拉普拉斯算子的硬件实现原理，如图 4-9 所示。数据按照 4.1 节所述组织成 3×3 模板的形式后输入，在第一个时钟沿到来时做乘法和加法运算，并判断乘法结果和加法结果的大小；在第二个时钟沿到来时根据比较结果进行减法运算，并输出结果。整个计算过程共需要 2 个时钟周期，然而采用流水线设计之后，仅仅第一个数据输入时需要耗费 2 个时钟周期得到结果，而之后每个时钟周期都能完成一次计算。

由硬件生成的拉普拉斯算子如图 4-10 所示。其中，输入端 clk 为时钟信号，rst_n 为复位信号，R11, R12, …, R33 为 3×3 模板输入的像素灰度数据；输出端 result 为计算结果。从图 4-10 可以看出输入端数据都是 10bit，而计算结果 result 变成了 16bit，这是由于拉普拉斯模块内部进行了数据的带宽转换，以方便将计算结果存储到外部 SRAM 芯片。

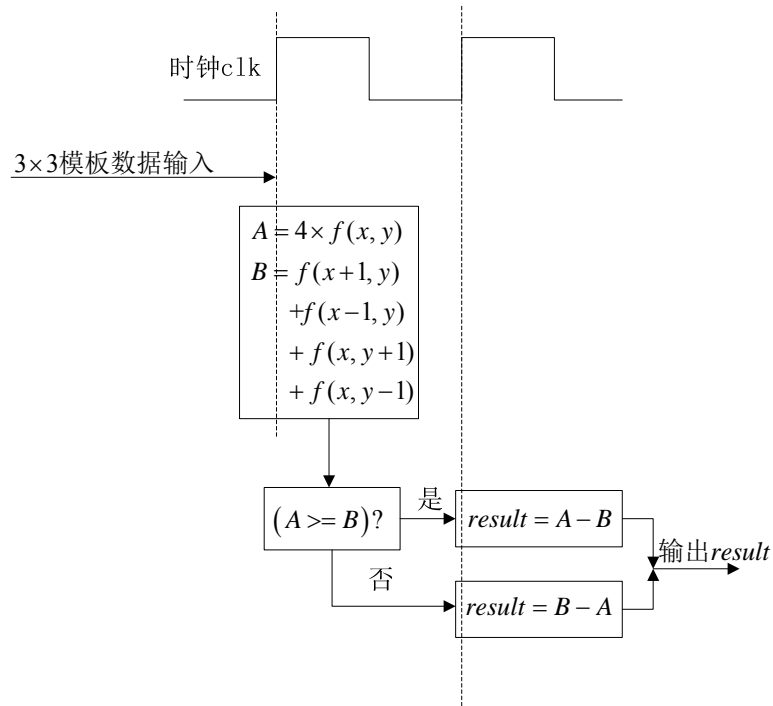


图 4-9 拉普拉斯滤波算法硬件实现原理

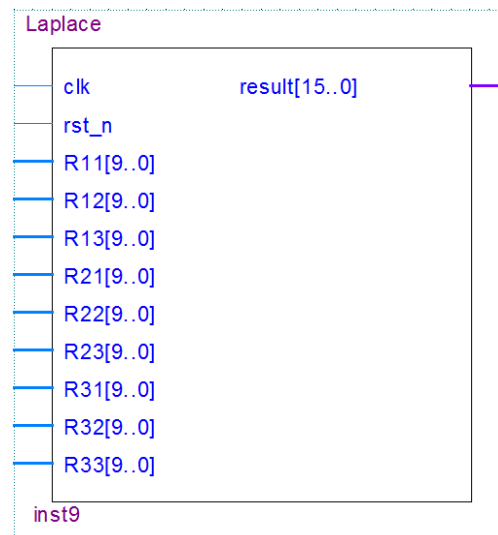


图 4-10 拉普拉斯算子硬件模块

经过 signal Tap II 仿真，得到正确的硬件拉普拉斯算子运算结果如图 4-11 所示。在 37.38 μs 处的第一个时钟周期内，9 个 3 \times 3 模板内的数据同时输入，40ns 后（一个时钟周期）得到计算结果 result=260。可以验证，计算结果是正确的： $|[R12+R21+R23+R32]-4\times R22|=|[26+36+39+287]-4\times 32|=|388-128|=260$ 。同样，

在 40ns 处输入的数据计算结果在 80ns 处得到 result=238。

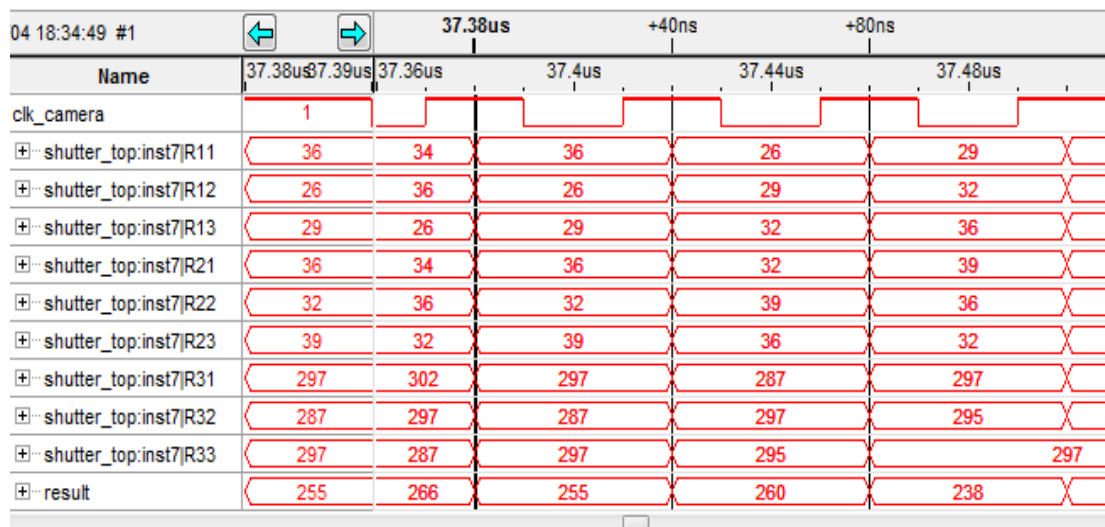


图 4-11 4 邻域拉普拉斯算子的硬件仿真时序

4.3.3 硬件算法性能分析

为对比显示硬件拉普拉斯滤波的效果，分别用 4 邻域和 8 邻域拉普拉斯算子做硬件算法实验。其对比效果如图 4-12 所示。

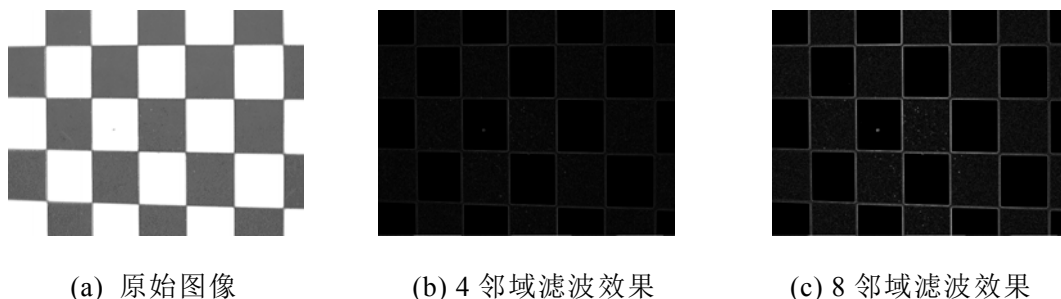


图 4-12 硬件拉普拉斯滤波对比效果

滤波效果：可以看出，无论是 4 邻域还是 8 邻域的拉普拉斯滤波的效果都达到了预期----即突出图像边缘。另外，对比 4 邻域和 8 邻域滤波效果可以看出 8 邻域的拉普拉斯算子对图像的边缘增强效果更为突出。

处理速度：拉普拉斯算法每次计算共需要 1 次乘法、1 次加法和一次求绝对值运算。软件算法实现需要 3 个时钟周期完成；而硬件实现的拉普拉斯滤波算法，由于乘法和加法并行执行，且采用了一级流水线操作，仅需要 1 个时钟周期。对于一副 648×494 大小的图像，采用硬件实现比软件实现节省了 $(3-1) \times 648 \times 494 = 640224$ 个时钟周期。

4.4 Sobel 滤波

4.4.1 算法原理

Sobel 滤波是又一种常用的图像增强算法^[27]，与拉普拉斯滤波基于二阶微分不同，Sobel 滤波是基于二阶微分的滤波算子。在图像处理中，一阶微分的实现往往借助于梯度法^[27,34]。对于一个二维图像，其坐标 (x, y) 处的梯度模值定义为：

$$\nabla f = \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right]^{\frac{1}{2}} \quad (4-10)$$

实际操作时，常用绝对值替代平方与平方根运算以减少计算量^[36]：

$$\nabla f = \left| \frac{\partial f}{\partial x} \right| + \left| \frac{\partial f}{\partial y} \right| \quad (4-11)$$

Sobel 算子是在梯度法的基础上，引入加权因子的滤波算法^[37]。其定义如下：

$$\begin{aligned} \nabla f = & [f(x-1, y+1) + 2f(x, y+1) + f(x+1, y+1)] \\ & - [f(x-1, y-1) + 2f(x, y-1) + f(x+1, y-1)] \\ & + |[f(x+1, y-1) + 2f(x+1, y) + f(x+1, y+1)] \\ & - [f(x-1, y-1) + 2f(x-1, y) + f(x-1, y+1)]| \end{aligned} \quad (4-12)$$

表示为掩膜形式如图 4-13 所示，式(4-12)可表示为两个掩膜分别作用后相加的结果。其中，权值 2 的作用是通过突出中心点的作用达到平滑的目的。

-1	-2	-1
0	0	0
1	2	1

(a) 垂直方向梯度算子

-1	0	1
-2	0	2
-1	0	1

(b) 水平方向梯度算子

图 4-13 Sobel 算子掩膜

4.4.2 算法的硬件实现

以垂直方向的梯度算子掩膜为例，其硬件实现的结构如图 4-14 所示。所有的乘法单元都是并行模块，整个计算过程需要 1 次乘法和 2 次加法运算。由于都是组合逻辑且延迟很小，只需要一个时钟延迟即可得到计算结果。

经过 Signal Tap II 仿真，得到正确的 Sobel 滤波算子硬件实现结果，如图 4-15 所示。可以验证计算结果的正确性：在 $1.7\mu\text{s}$ 时， 3×3 模板数据输入， 40ns (一个时钟周期) 后得到计算结果 $\text{result}=70$ 。同样， 40ns 处输入的模板数据在 80ns 处得到计算结果 $\text{result}=76$ 。

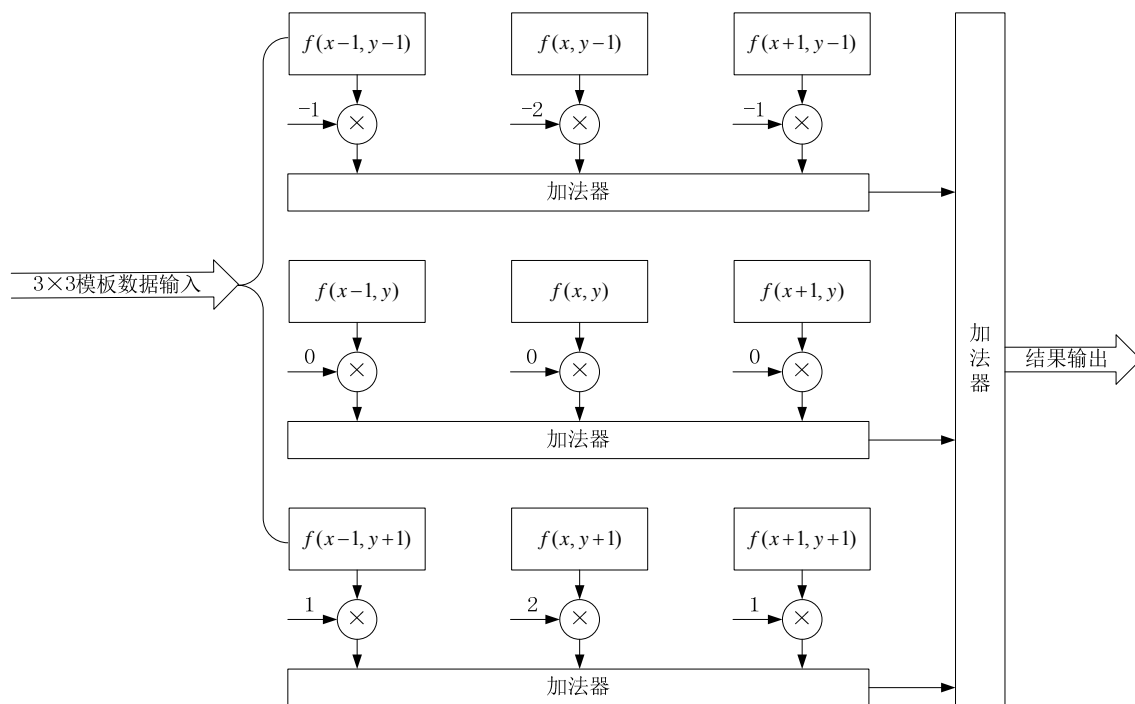


图 4-14 Sobel 算子硬件实现结构

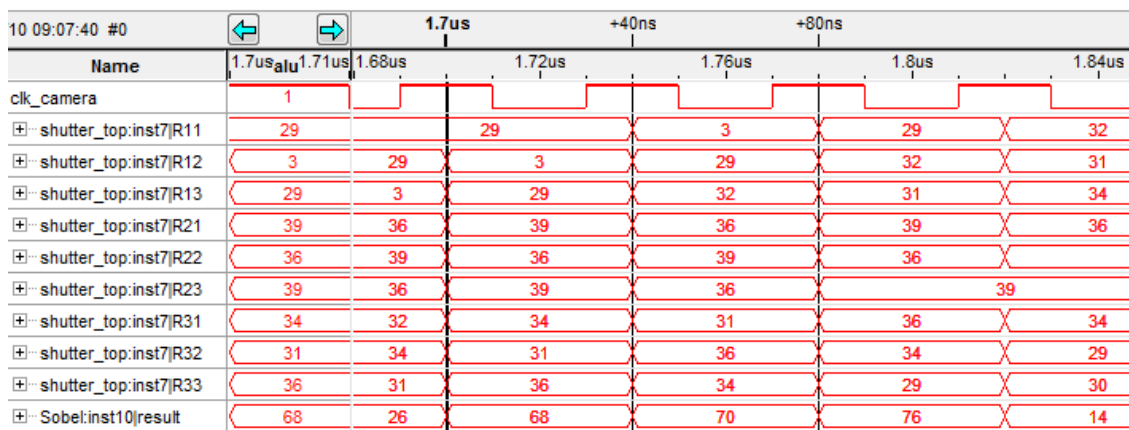


图 4-15 Sobel 算子的硬件仿真时序

4.4.3 硬件算法性能分析

滤波效果：从图 4-16 可以看出，硬件实现的 Sobel 滤波效果明显，达到了几乎可以和软件处理相媲美的程度。另外，对比 Sobel 滤波和拉普拉斯滤波

的结果，验证了 Sobel 滤波相较于拉普拉斯滤波在平滑噪声方面的优越性。

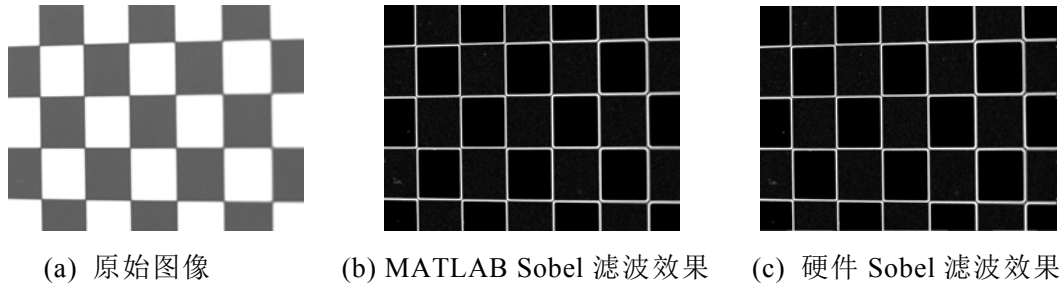


图 4-16 Sobel 滤波结果对比

处理速度：对于 Sobel 滤波的单个方向梯度算子，每次计算需要 9 次乘法、4 次加法运算，而两个方向的梯度算子共需要 18 次乘法、9 次加法，如公式(4-12)所示。采用软件实现 Sobel 滤波的每次计算需要 27 个时钟周期；而采用硬件实现仅需要 1 个时钟周期即可得到计算结果。每次计算硬件实现比软件实现要快 26 个时钟，硬件滤波的优势非常明显。

4.5 本章小结

本章在图像采集调试平台上进行了常见的空间域图像增强算法的硬件实现。经过实验表明，FPGA 对于这些典型的图像增强算法具有不错的实现效果，而且在 25MHz 的时钟频率下处理速度均达到了 60 帧/秒。这表明 FPGA 对于实时性要求高且具有大量重复运算的图像预处理领域的适用性。

第 5 章 激光光条中心提取算法的硬件实现

激光光条的提取是二维激光位移传感器进行下一步位移计算的基础,激光光条的定位精度直接影响整个系统的测量精度。因此本章在前面各章工作的基础上,进一步研究激光光条定位算法的硬件实现问题,为二维激光位移传感器的后续研究工作奠定基础。

5.1 激光光源及光束特性

本文选择的线激光器波长 635nm, 功率 8mW。

理想的线结构光光条在其法线方向呈高斯分布。有众多文献显示^[38,39],当线结构激光投射到物体上时,其光强沿光条法向的分布可以用高斯函数描述^[40]。

$$f_s(x) = \frac{A}{\sqrt{2\pi}\sigma} e^{-\frac{(x-x_c)^2}{2\sigma^2}} \quad (5-1)$$

其中, x_c 为光条中心, x 为偏离光条中心的距离, σ 为高斯函数方差。高斯光束在其法线方向上的辐射照度、场强分布如图 5-1 所示。

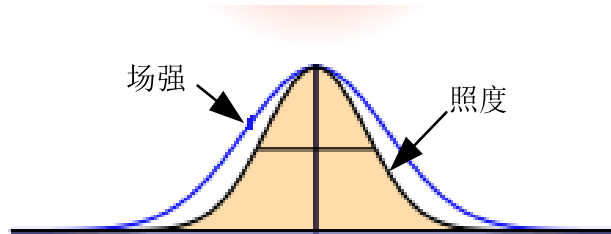


图 5-1 辐射照度(黑色)和场强(蓝色)分布曲线

为了提高光条提取的效果,在相机目镜前加入了 632nm 的滤光片。该滤光片可以很好地滤除背景,使激光光条灰度明显高于背景图像灰度。

5.2 传统极值法

5.2.1 算法原理

极值法是最基本的光条中心提取算法,其利用光条的高斯分布特性,取光条横截面上的灰度最大值作为光条中心。算法原理如下:逐行扫描图像,寻找每行像素灰度最大值 PMAX,如果 PMAX>阈值 TH,则认为该像素为光条中心。否则,该行不存在光条像素。该算法过程简单,易于实现,但是极易受到

噪声干扰,如图 5-2 所示。当真实的光条中心相邻像素处存在脉冲噪声干扰时,如果噪声与像素灰度和高于真实光条中心像素,定位就会产生偏差。

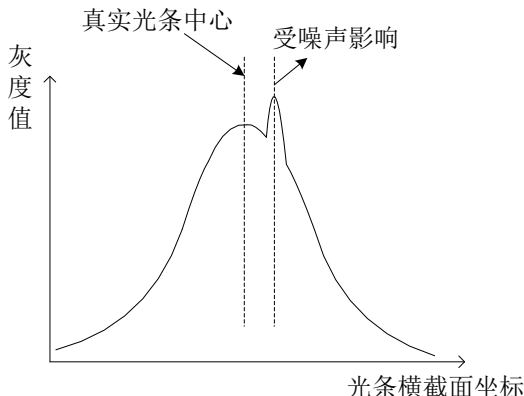


图 5-2 极值法受噪声影响示意图

5.2.2 算法的硬件实现

极值法的硬件实现过程如图 5-3 所示。图像数据逐行输入 $648 \times 10\text{bit}$ 的移位寄存器,同时寻找该行数据的最大值 P_{MAX} 。当该行数据输入完毕时,比较 P_{MAX} 和阈值 TH ,取二者中较大的作为本行光条中心的匹配阈值 Threshold 。当下一行数据到来时,上一行数据从移位寄存器输出,同时判断 P_i 是否匹配 Threshold 。如果二者相等,则输出 1023(10bit 图像灰度最大值),否则输出 0(10bit 图像灰度最小值)。

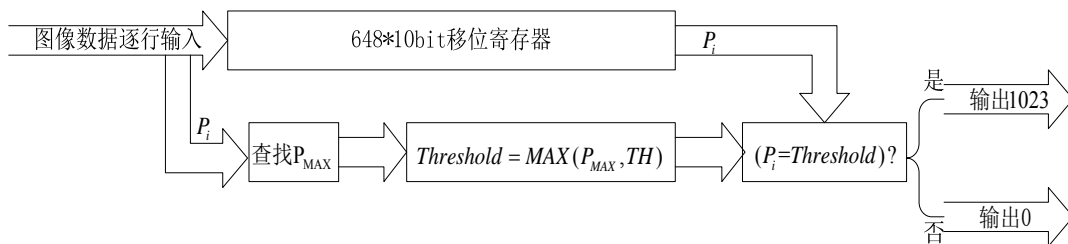
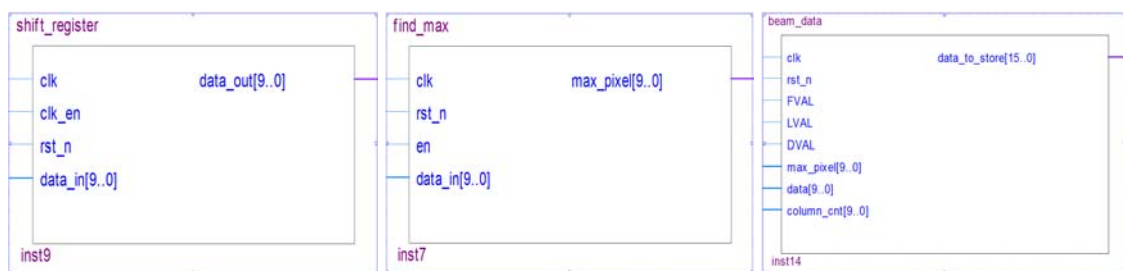


图 5-3 极值法硬件算法原理

按照功能进行模块划分,整个算法实现过程分为三个模块:移位寄存器模块、查找最大值模块和光条中心判定模块,如图 5-4 所示。

移位寄存器模块对输入的图像灰度数据缓存;查找最大值模块在数据输入移位寄存器的同时进行每行最大的灰度值查找,并在每一行结束时将最大值输出给光条中心判定模块;光条中心判定模块通过比较最大灰度值和固定阈值以确定最后的判定阈值 Threshold ,最后将每一个从移位寄存器输出的灰度值与该 Threshold 匹配。若二者相等输出 1023,否则输出 0。



(a) 移位寄存器模块 (b) 查找最大值模块 (c) 光条中心判定模块

图 5-4 极值法的硬件实现模块

5.2.3 算法硬件实现效果分析

根据经验观察值，分别取阈值 $TH=50$ 、 60 、 70 进行试验。原始光条图像和经过极值法提取的光条图像分别如图 5-5 所示。

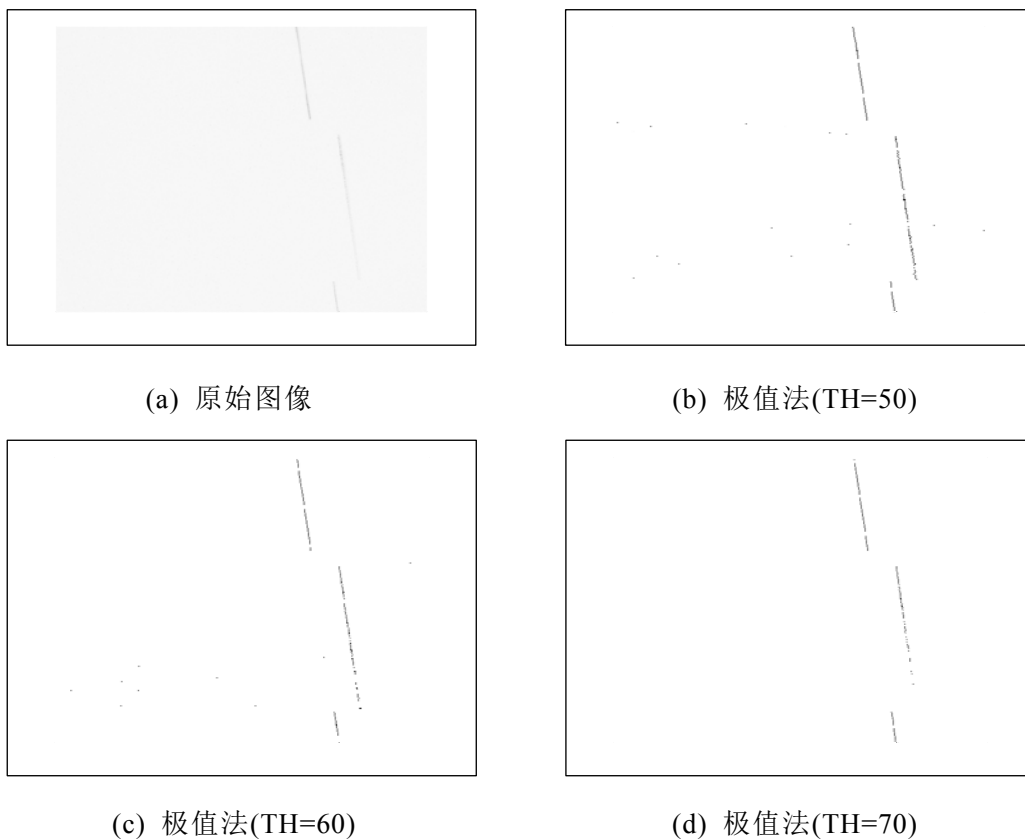

 (a) 原始图像 (b) 极值法($TH=50$)
(c) 极值法($TH=60$) (d) 极值法($TH=70$)

图 5-5 不同阈值下的极值法提取光条效果

对比原始图像和极值法提取光条后的图像可以看出，经极值法提取，激光光条和背景图像的对比度明显增强了。这是由于极值法处理过程中，将背景图像的灰度值都置 0，光条中心灰度值置成最大值 1023 的原因。这个过程类似于图像的二值化。

比较图(b)、(c)、(d)可以发现：当取 $TH=50$ ，激光光条较为完整的保存下来，但是引入了较多的孤立噪声点；当取 $TH=70$ ，虽然几乎可以完全排除噪声点，但是激光光条损失点较为严重；当取 $TH=60$ ，孤立噪声点和激光光条损失同时存在。

显然，这是由于采用 P_{MAX} 和固定阈值 TH 进行比较以判定光条是否存在造成的：在光条灰度较小处或者光条断线处(无光条的区域)，灰度突变的脉冲噪声点容易被误判成光条上的像素。为了改善这一状况，本文提出了改进的极值法。

结论：采用极值法对激光光条提取，可以达到像素级的定位精度，但是极其容易受到脉冲噪声的干扰。

5.3 改进的极值法

5.3.1 算法原理

利用光条的高斯分布特性及脉冲噪声的分布独立性，可以通过检测最大灰度像素邻域的像素灰度，来辅助判断 P_{MAX} 是否是光条中心灰度值。最简单的方法便是给 P_{MAX} 左右两边的像素也设置一个固定阈值，当它们都大于这个阈值时，认为光条中心存在于 P_{MAX} 处。这是由于脉冲噪声分布相对独立，相邻像素连续出现脉冲噪声的概率较小，而光条中心附近的灰度值成高斯分布，相邻像素灰度变化有一定的延续性。

5.3.2 算法的硬件实现

改进的极值法的硬件实现过程如图 5-6 所示：

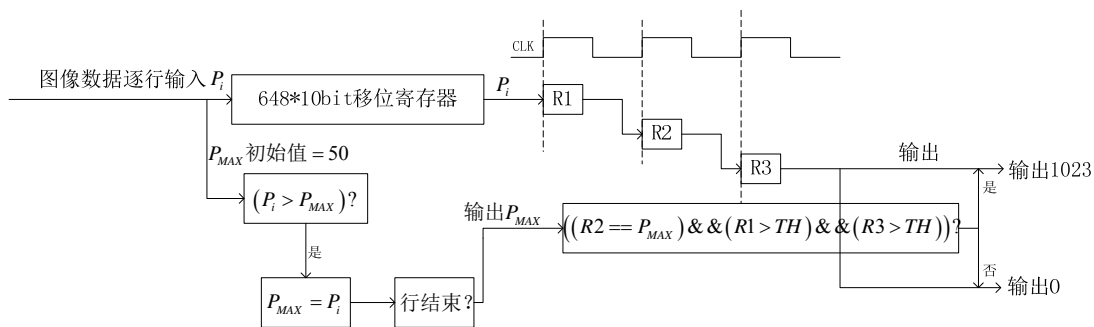


图 5-6 改进的极值法

图像数据 P_i 逐行逐个输入 $648 \times 10\text{bit}$ 移位寄存器，同时查找最大灰度值 P_{MAX} 。当下一行数据到来时，移位寄存器输出 P_i 。 P_i 经过三级缓存 $R1$ ， $R2$ ， $R3$ 的同时，检查匹配条件 $((R2 == P_{MAX}) \&\& (R1 > TH) \&\& (R3 > TH))$ 是否成立。当匹配成功时，输出灰度最大值 1023；否则输出 0。

同样，硬件实现分为移位寄存器模块、查找最大值模块和光条中心判定模块，这里不再赘述。

5.3.3 算法硬件实现效果分析

实验分别选取阈值 $TH=35, 40, 45, 50$ ，得到光条提取效果如图 5-7 所示。从图中可以看出，当选取阈值 $TH=35$ 时，容易引入较多的噪声；选取 $TH=45, 50$ 时，噪声被完全滤除，但是光条损失也较为严重；选取阈值 $TH=40$ 时提取效果最好，不但没有引入噪声，光条也得到较为完整的提取。

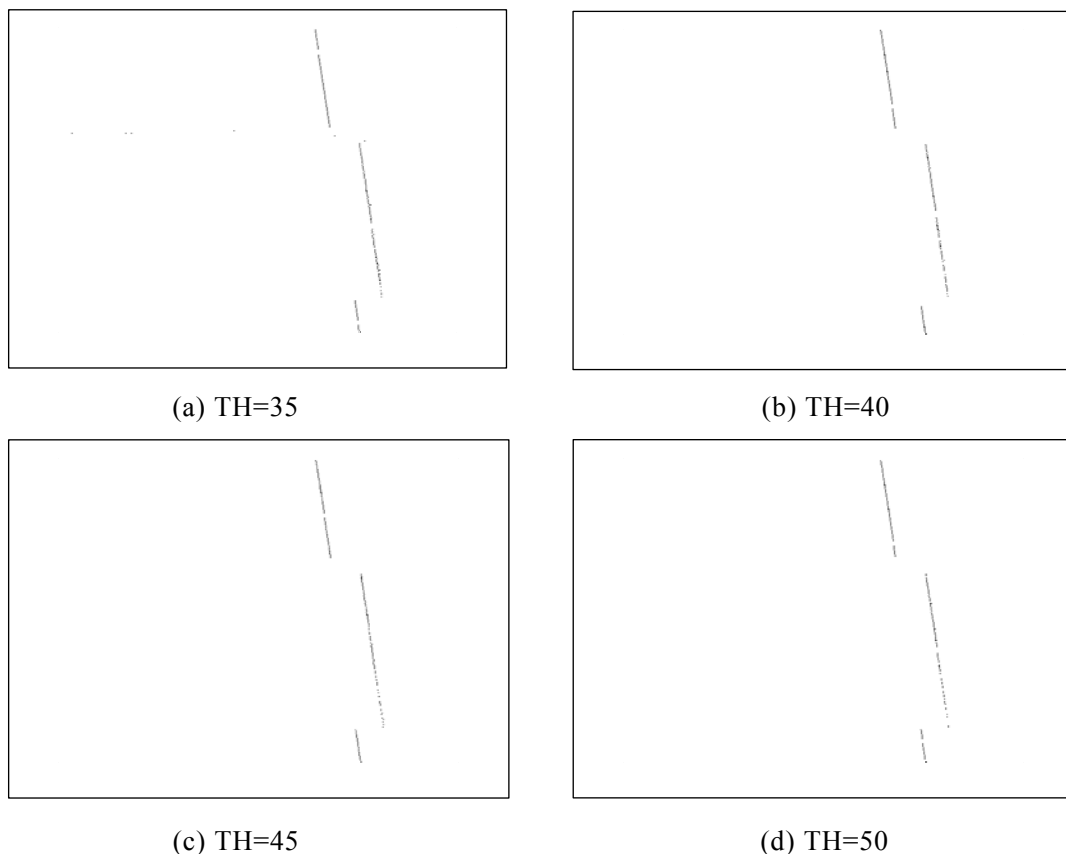


图 5-7 不同阈值下的改进极值法光条提取效果

结论：改进后的极值法通过改变阈值 TH 可以得到比较理想的激光光条数据。

5.4 灰度质心法

5.4.1 算法原理

灰度质心法是把光条横截面的灰度分布的质心作为光条中心点^[41]。该方法是在利用极大值法找到光条上灰度最大像素位置 C_{MAX} 后，取其两边各 k 个点，求这 $2k+1$ 个点的灰度质心 B 作为该行光条质心。设位置 i 处像素灰度为

$G(i)$, $i \in [C_{MAX} - k, C_{MAX} + k]$, 则 B 可以表示为:

$$B = C_{MAX} - k - 1 + \frac{\sum_{C_{MAX}-k}^{C_{MAX}+k} G(i) \times [i - (C_{MAX} - k - 1)]}{\sum_{C_{MAX}-k}^{C_{MAX}+k} G(i)} \quad (5-1)$$

由于同样需要先找到每行光条灰度极大值, 为了提高光条定位精度, 采用改进的极值法里用到的邻域辅助判别方法。

5.4.2 算法的硬件实现

硬件算法实现过程如图 5-8 所示。整个计算过程分解为一次乘法, 两次加法, 一次除法。

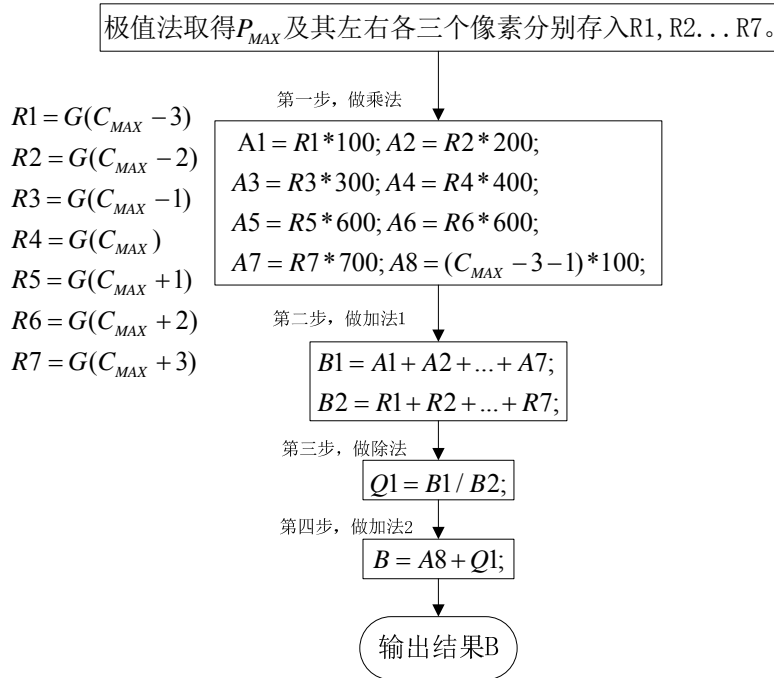


图 5-8 灰度质心法

从式(5-1)可以看到, 运算过程涉及到了除法。硬件上实现除法, 如果采用整数除法, 非常容易实现, 但精度大大折损(损失了小数部分)。FPGA 内部并非不能实现浮点数的除法运算, altera 公司已有现成的浮点数除法 IP 核可供直接调用。但实现浮点数除法时, 仍需将二进制的整数转换成二进制浮点数的标准存储格式, 涉及到了数据格式转换的问题, 增加了问题的复杂程度。为了简化设计过程并保留一定的运算精度, 本文在做除法运算时, 先将分子乘以固定系数 100, 再进行除法运算。如此, 整数除法得到的商的后两位为小数部分。虽然这样做耗费了更多的存储资源, 但大大简化了设计过程, 实验证明是非常有效的方法。

5.4.3 算法硬件实现效果分析

为了验证质心法的提取效果，对同一被测表面连续采集 10 副图像分别进行质心法提取，得到如表 5-1 所示的数据。由于整幅光条图像数据包含 494 个光条中心数据，过于庞大，这里只给出前 5 行的提取数据。采集的原始图像数据大致如图 5-6 所示(10 副图像大致相同，只给出了其中一幅)。

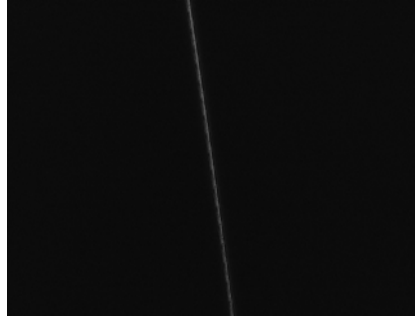


图 5-6 改进的极值法

表 5-1 质心法提取的光条数据 (单位: pixel)

行坐标	1	2	3	4	5
第 1 组	291.96	291.95	291.14	291.06	291.33
第 2 组	291.46	292.35	291.70	291.06	290.96
第 3 组 列	291.21	292.38	291.85	291.45	291.63
第 4 组	292.36	292.01	291.74	292.13	291.23
第 5 组	292.73	292.36	291.13	292.42	291.71
第 6 组 坐	292.94	292.01	292.48	292.35	291.97
第 7 组	292.53	291.59	291.30	292.46	292.03
第 8 组	292.01	293.20	291.60	292.34	292.52
第 9 组 标	293.33	291.95	291.20	292.93	292.50
第 10 组	292.15	292.38	291.85	291.94	292.76

从表中可以看出，相邻组的提取效果基本一致，最大差异 2.22 个像素，基本达到了对于重复性的要求。

结论：质心法进行光条中心提取，达到了预期提取效果，具有良好的重复性。

5.5 本章小结

本章在前面工作的基础上，对二维激光位移传感器中尤为关键的“光条中心提取”算法进行了研究，并进行了相应的硬件算法实现。

其中，采用极值法对激光光条提取，可以达到像素级的定位精度，但是极

其容易受到脉冲噪声的干扰；改进的极值法则对于脉冲噪声有很好地抑制能力；采用质心法进行光条中心提取，达到了预期提取效果，具有良好的重复性。

结 论

本文以二维激光位移传感器的研制为背景,进行了一系列基于 FPGA 的相机图像采集及处理技术的研究工作。

本课题的研究围绕相机图像采集及处理的实时性这一需求,设计了图像采集处理硬件平台,并进行了相应的硬件逻辑设计,成功实现了相机图像的实时采集、存储和 VGA 显示等的功能。在此基础上,本文又进行了典型空间域图像增强算法的硬件实现工作。最后论文进行了二维激光位移传感器图像处理算法中最为关键的光条提取算法的硬件设计与实现,为后续二维激光位移传感器算法的进一步研究奠定了基础。

然而,本文最后一章中激光光条提取算法的硬件实现部分,只是针对某一特定的被测物进行的,算法并不具备良好的环境适应性。这一缺点的克服有赖于硬件上的改进和硬件逻辑算法的进一步研究,例如使用数字可调亮度的激光器、CCD 曝光控制逻辑的设计等。

总结之,本文进行了大量的实验、仿真工作,充分证明了以 FPGA 为核心处理器的图像采集及处理系统是可行的,并且在图像预处理领域具有明显的并行处理优势。

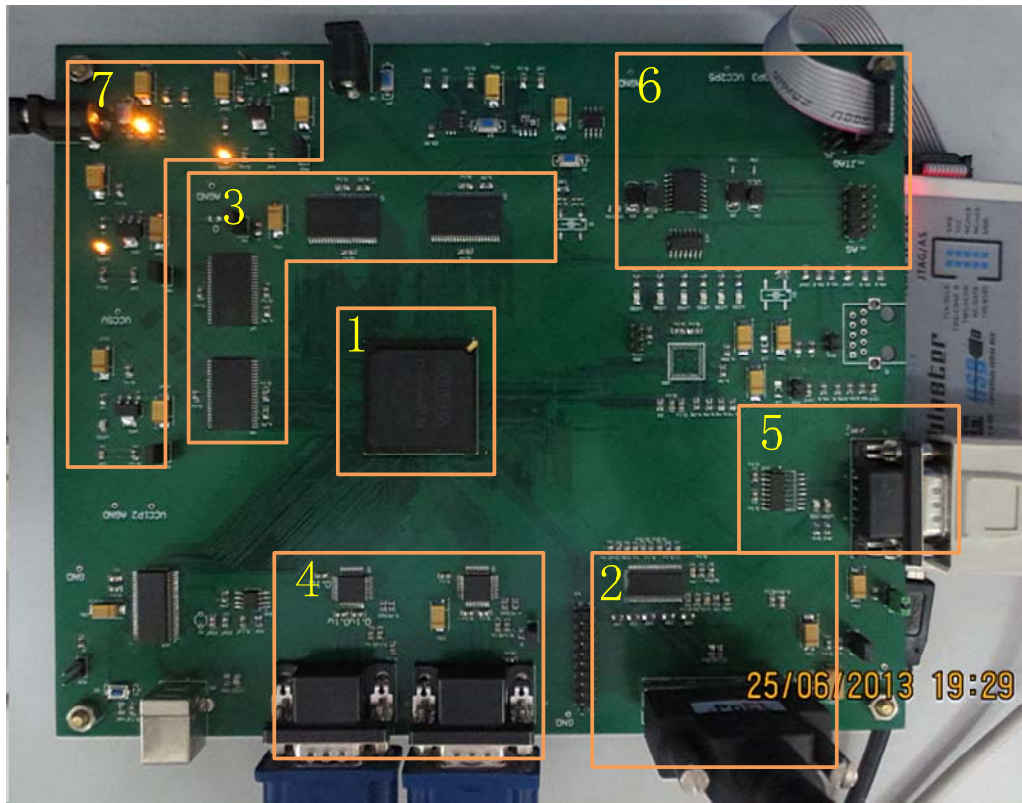
参考文献

- [1] 彭平良. 基于FPGA的嵌入式图像采集卡的研究[D]. 大连 : 大连理工大学, 2008.
- [2] TORRES-HUITZIL C, ARIAS-ESTRADA M. Real-time image processing with a compact FPGA-based systolic architecture[J]. Real-time Imaging, 2004, 10(3): 177-187.
- [3] CROOKES D, BENKRID K, BOURIDANE A, et al. Design and implementation of a high level programming environment for FPGA-based image processing[J]. IEE Proceedings-Vision, Image and Signal Processing, 2000, 147(4): 377-384.
- [4] CLOUTIER J. FPGA-based processor[Z]: Google Patents, 1999.
- [5] 龚涛. 基于FPGA的图像采集处理系统[D]. 武汉 : 华中科技大学, 2005.
- [6] 侯法柱. 基于FPGA的图像采集与处理系统设计[D]. 湖南: 湖南大学, 2010.
- [7] BATLLE J, MARTI J, RIDAO P, et al. A new FPGA/DSP-based parallel architecture for real-time image processing[J]. Real-Time Imaging, 2002, 8(5): 345-356.
- [8] 姜华. 基于FPGA的图像采集处理系统设计与实现[D]. 大连: 大连海事大学, 2009.
- [9] 唐晓燕, 贾国锋, 韩磊. 基于FPGA的视频颜色空间转换电路设计[J]. 电子与电脑, 2006 (08) : 110-112.
- [10] 沈淦松, 叶玉堂, 刘霖, 等. FPGA软硬件协同处理实时图像处理系统[J]. 光电工程, 2012, 39 (10) : 15-18.
- [11] 庞业勇. 基于FPGA的图像处理系统设计方法研究[D]. 哈尔滨: 哈尔滨工业大学, 2010.
- [12] LEE C R, SALCIC Z. High-performance FPGA-based implementation of Kalman filter[J]. Microprocessors and Microsystems, 1997, 21(4): 257-265.
- [13] FAHMY S A, CHEUNG P Y, LUK W. Novel FPGA-based implementation of median and weighted median filters for image processing[Z]: IEEE, 2005: 142-147.

- [14] KNITTEL G. A pci-compatible fpga-coprocessor for 2d/3d image processing[Z]: IEEE, 1996: 136-145.
- [15] DILLINGER P, VOGELBRUCH J F, LEINEN J, et al. FPGA-based real-time image segmentation for medical systems and data processing[J]. Nuclear Science, IEEE Transactions on, 2006, 53(4): 2097-2101.
- [16] 吕炎军. 基于FPGA和USB2.0视频图像采集及处理平台设计[D]. 成都: 西南交通大学, 2007.
- [17] HAUG A J. A tutorial on Bayesian estimation and tracking techniques applicable to nonlinear and non-Gaussian processes[J]. Mirte Technique Report, 2005, 20(1): 39-55.
- [18] 闫宇航. 改进粒子滤波算法在FPGA中的研究与实现[D]. 北京: 北京交通大学, 2009.
- [19] 李一芒, 何昕, 魏仲慧. 红外预警实时图像处理系统设计与实现[J]. 液晶与显示, 2013, 28 (01) : 18-23.
- [20] 罗辉舞, 洪海丽, 任宇飞. 基于CameraLink接口的图像采集控制器的设计与实现_罗辉舞[J]. 电子测量技术, 2010, 33 (07) : 23-39.
- [21] 曾柏杞. 600速全自动生化分析仪控制系统的研究[D]. 湖南: 湖南大学, 2012.
- [22] 王立业. 双电源自动转换器的设计与研究[D]. 河北: 河北工业大学, 2007.
- [23] 冈萨雷斯. 数字图像处理[M]. 北京: 电子工业出版社, 2003: 59.
- [24] 杨柱中, 周激流, 晏祥玉, 等. 基于分数阶微分的图像增强 [J]. 计算机辅助设计与图形学学报, 2008, 20 (03) : 343-348.
- [25] 冯安, 王希常. MATLAB在数字图像增强中的应用 [J]. 信息技术, 2007 (05) : 65-73.
- [26] 张海青. 基于FPGA图像处理系统的关键算法研究及硬件实现[D]. 重庆: 重庆大学, 2010.
- [27] 张小娟. 便携式医用X光机图像清晰化研究[D]. 山西太原: 中北大学, 2009.
- [28] 冈萨雷斯. 数字图像处理[M]. 第2版. 北京: 电子工业出版社, 2003: 97.
- [29] HONG-SWEE L, ZHANG D. Video processing using fpgas in video surveillance systems[J]. DSP Magazine, 2007, 40(4): 157-161.

- [30] 朱捷, 朱小娟, 贺明. 计算机测量与控制[J]. 计算机测量与控制, 2007, 15 (06) : 41-44.
- [31] 李元帅, 张勇, 周国忠, 等. 图像中值滤波硬件算法及其在 FPGA 中的实现 [J]. 计算机应用, 2006, 26 (6) : 62-75.
- [32] 董付国, 原达, 王金鹏. 中值滤波快速算法的进一步思考[J]. 计算机工程与应用, 2007, 43 (26) : 49-64.
- [33] 毛伟民, 赵勋杰, 李明, 等. 拉普拉斯算子的FPGA实现方法[J]. 现代电子技术, 2009 (16) : 132-134.
- [34] 冈萨雷斯. 数字图像处理[M]. 第2版. 北京: 电子工业出版社, 2003: 100-105.
- [35] MASATO I, AKIRA I, RYUJI T. Automatic analysis apparatus with liquid level detection function[Z]: Google Patents, 2002.
- [36] RAY A. A survey of CORDIC algorithms for FPGA based computers[Z]: ACM, 1998: 191-200.
- [37] ZHANG Tao, QUAN Hao-jun, ZHAO Liang, et al. High efficient implementation of image matching algorithm[Z]: IEEE, 2009: 1-5.
- [38] 张益昕. 基于计算机视觉的大尺度三维几何尺寸测量方法及应用[D]. 南京: 南京大学, 2011.
- [39] SHI Quan, XI N, CHEN Yi-fan. Development of an automatic optical measurement system for automotive part surface inspection[Z]: IEEE, 2005: 1557-1562.
- [40] HIROFUMI N, DAISUKE I, SATO K. 3D shape measurement using fixed camera and handheld laser scanner[Z]: IEEE, 2008: 1536-1539.
- [41] 徐静珠. 结构光三维测量中光条中心提取方法及其评价的研究[D]. 南京: 南京大学, 2012.

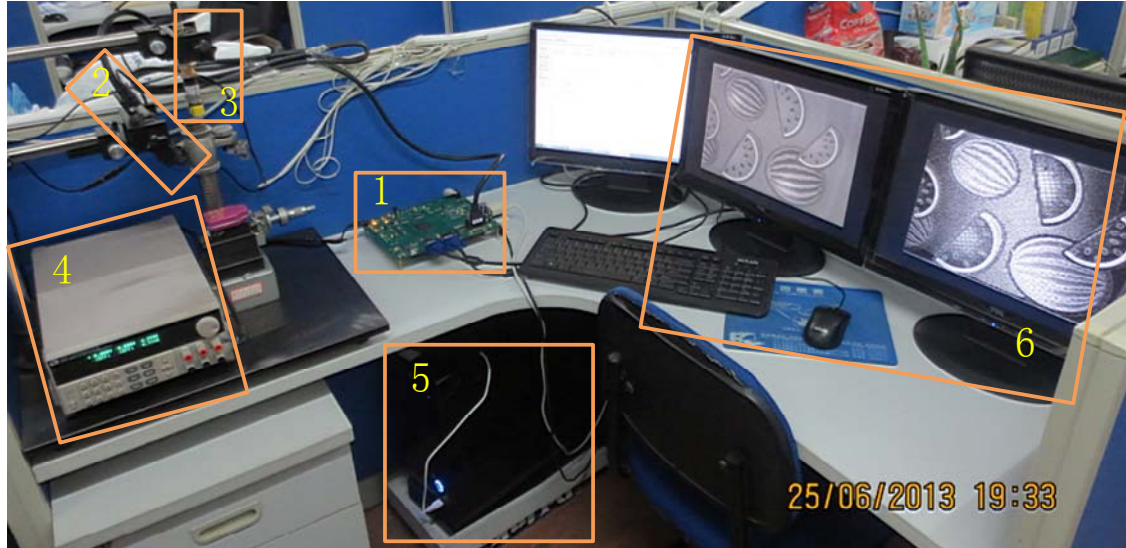
附录 A



图像采集调试平台电路板

注： 1、FPGA 芯片；2、Camera Link 接口电路；3、SRAM 存储电路；4、VGA 接口电路；5、RS-232 串行通信电路；6、FPGA 配置电路；7、电源电路；

附录 B



图像采集调试平台

注：1、硬件电路；2、Camera Link 相机；3、线激光器；4、电源；5、上位机 PC；6、VGA 显示器×2。

哈尔滨工业大学学位论文原创性声明和使用权限

学位论文原创性声明

本人郑重声明：此处所提交的学位论文《基于FPGA的Camera Link相机图像采集及处理技术研究》，是本人在导师指导下，在哈尔滨工业大学攻读学位期间独立进行研究工作所取得的成果，且学位论文中除已标注引用文献的部分外不包含他人完成或已发表的研究成果。对本学位论文的研究工作做出重要贡献的个人和集体，均已在文中以明确方式注明。

作者签名：吴振锋

日期：2013年6月28日

学位论文使用权限

学位论文是研究生在哈尔滨工业大学攻读学位期间完成的成果，知识产权归属哈尔滨工业大学。学位论文的使用权限如下：

(1) 学校可以采用影印、缩印或其他复制手段保存研究生上交的学位论文，并向国家图书馆报送学位论文；(2) 学校可以将学位论文部分或全部内容编入有关数据库进行检索和提供相应阅览服务；(3) 研究生毕业后发表与此学位论文研究成果相关的学术论文和其他成果时，应征得导师同意，且第一署名单位为哈尔滨工业大学。

保密论文在保密期内遵守有关保密规定，解密后适用于此使用权限规定。本人知悉学位论文的使用权限，并将遵守有关规定。

作者签名：吴振锋

日期：2013年6月28日

导师签名：刘国栋

日期：2013年6月28日

致 谢

论文接近尾声，我的学生生涯也将告一段落。两年学习的成果结晶为一本硕士毕业论文。透视这本论文，其中不仅有我付出的努力，更凝结着老师们的谆谆教诲和各位友好人士的关心与帮助，在此，作者谨向各位敬爱的师长、亲爱的同学、尊敬的师傅表示诚挚的谢意！

感谢导师刘国栋教授！刘老师治学严谨、工作勤勉、勇于创新、对学术精益求精，在我准备毕业设计期间，无论在学习中还是生活上均给予了我极大的帮助和关怀。从刘老师身上，我学到了规格严格的科学态度，学到了勤勉扎实的工作作风，学到了锲而不舍的执着精神。本论文的选题、方案的确定、研究工作的开展以及论文的撰写和修改，都是在刘老师的悉心指导下完成的，每一个环节都凝聚着老师大量的心血。刘老师您辛苦了！学生吴振锋永远爱戴您！

感谢实验室的所有老师和师傅！在我两年的研究生生活中，各位老师给予了我数不尽的关怀和帮助。庄志涛老师在我课题遇到难题时给予我精心指点和耐心教诲；刘炳国老师、胡涛老师、甘雨老师、陈凤东老师、王亦工老师在生活中给予我热情关怀；王文峰师傅给予我 PCB 制作方面经验指导；马晶月老师给予我 PCB 投版购置大力支持；刘海波师傅在 PCB 调试中给予我倾情相助；还有，53 所实验室的彭宇老师、刘大同老师及各位师兄同学的精辟点拨。祝各位老师、同学和师傅身体健康、工作愉快、家庭幸福、万事如意。

感谢实验室的师兄师姐和所有同学！感谢你们对于我生活学习工作中方方面面的帮助。冯博师兄在我论文算法部分给予了无私的奉献；卢炳辉师兄、尹哲师姐、许新科师兄、路程师兄给予了周到的服务；田一普师兄、林灿发同学在 FPGA 方面给予了大力帮助；唐烽同学在我课题遇到各种困难时提供了有力支持；万霏、梁功涛等同学给予了各种无形有形的援助。祝实验室所有的同学们学业进步、成就辉煌！

感谢父母的养育之恩，二位亲人是我人生的支撑，是我前进的动力，更是我幸福的源泉。感谢我的姐姐、哥哥等在生活上给予我的关心和支持。感谢女友沈丽云给予我的无微不至的关怀和照顾，感谢你的激励和鞭策，感谢你所有的付出。

最后，感谢在百忙之中为我精心评阅论文的诸位老师，谢谢你们！