

湖北大学

硕士学位论文

基于ARM和FPGA的便携式图像采集系统若干问题的研究

姓名：蔡朝

申请学位级别：硕士

专业：微电子学与固体电子学

指导教师：潘永才

20070501

摘 要

随着数字图像技术和电子技术的不断发展,图像采集系统已经在民用、商用和军用等各个不同的领域得到广泛的应用。发展体积小,功耗低的便携式数字图像采集系统显得非常必要,而且十分迫切。

本文通过研究当前数字图像采集系统的研究成果和发展趋势,设计了一种基于 FPGA 和嵌入式 ARM 组成的便携式数字图像采集系统;并对该系统设计过程中的若干问题进行了深入探讨,论文重点研究了系统的硬件设计和软件实现。

论文主要的研究内容包括:详细介绍了 CMOS 传感器 OV9121;提出了利用 FPGA 实现对 CMOS 传感器的控制;在研究图像噪声的机理后,提出了利用图像中值滤波消除噪声的方法;并研究了在 FPGA 中实现中值滤波的方法;嵌入式 ARM 处理器 LPC2210 的功能和使用方法;ARM 处理器控制液晶显示器 TFT6758 的方法;论文详细讨论了 ARM 处理器和 FPGA 协调工作以及人机接口的实现等相关问题。

论文通过分析 CMOS 传感器 OV9121 的控制方法,使用 VHDL 语言在 FPGA 中实现对 CMOS 传感器的控制和图像数据的采集,设计了大容量的双口 RAM 来缓存数据,并且利用 DSP Builder 在 FPGA 中实现了图像中值滤波的硬件设计。本文研究了 ARM 处理器 LPC2210 的特点,提出了利用 ARM 来控制 FPGA 图像采集的方法,设计了 FPGA 和 ARM 的接口电路和控制方法。编写了 LPC2210 的 BootLoader 程序,利用 ADS1.2 开发软件开发了 TFT6758 图像显示器的驱动程序,在 TFT6758 上显示采集到的图像数据。最后本文对系统中存在的问题和今后努力的方向进行了总结。

关键词: 数字图像采集; FPGA; DSP Builder; ARM; OV9121

ABSTRACT

As the development of digital technology and Electronic technology, image system is used in many application fields, including civil field, commercial field and military field. It is necessary and urgent to develop a small, portable digital acquisition system with low power consumption.

The latest techniques and trend in the field of image system were studied. A system with chip Field Programmable Gate Array (FPGA) and embedded ARM, was introduced. This paper has carried out deeply investigation and discussion on some problem in the system's turn to design process, including the software and hardware.

The paper mainly includes some content: the functioning of OV9121 a CMOS sensor; how to control the sensor using FPGA; a method in dispelling image noise of medium value filter; how to realize the filter in FPGA; the usage and characteristic of embedded CPU LPC2210; how to drive the LCD display TFT6758 for ARM CPU; how the embedded CPU ARM and FPGA to work in harmony and realize the interface.

According to analyzing the CMOS sensor, the VHDL describes the driver of the module in FPGA, and collects image data. To design a buffer area with dual RAM, the hardware of medium value filter was researched by DSPBuilder in FPGA. The ARM processor is used to complete a task to control the FPGA. The interface between ARM processor LPC2210 and FPGA was designed. Write the bootloader, the driver of TFT6758 was designed by ADS1.2, and output the image to LCD display.

Finally, the dissertation summarizes the system and introduces the prospect of the system in the future.

Keywords: image acquisition; FPGA; DSP Builder; ARM; OV9121

湖北大学学位论文原创性声明和使用授权说明

原创性声明

本人郑重声明：所呈交的论文是本人在导师的指导下独立进行研究所取得的研究成果。除了文中特别加以标注引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写的成果作品。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律后果由本人承担。

论文作者签名：蔡翊

日期：2007 年 5 月 21 日

学位论文使用授权说明

本学位论文作者完全了解学校有关保留、使用学位论文的规定，即：

按照学校要求提交学位论文的印刷本和电子版本；学校有权保留学位论文的印刷本和电子版，并提供目录检索与阅览服务；学校可以允许采用影印、缩印、数字化或其它复制手段保存学位论文；在不以赢利为目的的前提下，学校可以公开学位论文的部分或全部内容。（保密论文在解密后遵守此规定）

作者签名：蔡翊

日期：2007.5.22

指导教师签名：(蔡翊)

日期：2007.5.22

1. 前言

1. 1 研究便携式数字图像采集系统的意义

视觉是人类最高级的感知器官，在人类感知中扮演着最重要的角色。人类约有 70 % 的外界信息是通过视觉来获得的，图像作为一种重要的信息来源，是人类认识外部世界的有效手段。

现代电子技术的高速发展，特别是计算机技术的飞速发展，图像的数字化已成为计算机进行图像处理之前的必经的基本步骤。随着计算机技术、传感器技术和数字图像处理技术的迅速发展和相互渗透，进一步促进了数字图像技术的不断进步和广泛应用。多媒体通信、高清晰度电视以及图像处理、模式识别和计算机视觉等众多应用领域都对数字图像的采集与处理提出了越来越高的要求，如何获取高质量的数字式图像信息是非常重要的。由于数字图像采集技术在研究被测对象方面具有非接触、可重复性好以及应用对象广泛等优点，使得图像采集在现代测量和检测领域方面具有十分广阔的市场前景。随着数字技术的不断发展，特别是数字图像传感器、大规模存储系统和彩色显示系统的发展，图像的数字处理技术也得到了飞速的发展。数字图像技术越来越多的应用于医学、生物学、法律、国防等众多行业。^[1-3]

研究功耗和成本低、体积小、便于携带的数字图像采集系统越来越受到多方面的广泛关注。同时，数字图像采集系统又是传感器、数字图像处理、计算机接口技术等学科的交叉和综合，涉及知识面广，其成果应用广泛，具有远大的市场前景和研究价值。

1. 2 国内外研究现状

采用 PC 作为主机的图像采集系统在 80 年代~90 年代开始流行，普遍采用当时的 ISA 总线，虽然 ISA 总线的速度为每秒十几兆，但 ISA 总线标准的发展和改进已经停止，无法满足今后的高速传输数据需求。随着 ISA 总线接口在 PC 机上的消失，这种图像采集系统现在已经被使用 PCI 总线接口方式的图像采集卡所取代。

当前流行的基于 PC 主机的图像采集系统一般是采用 PCI 总线的图像采集卡，PCI 总线速度非常快，达到 33MB/s，可以实现摄像机图像到计算机内存的实时传送。在 Windows 平台上编制图像处理软件，利用高级语言可以开发各种满足需要的模块。由于 PCI 总线的诸多优点和当前 PCI 接口在 PC 机上的普及，这种系统是目前应用最为广泛的图像采集系统。

随着半导体技术的不断发展，另一种能脱离 PC 的图像采集系统也得到了飞速的发展，目前主要有两种实现方式：

一种是采用嵌入式处理器 ARM+数字信号处理器 DSP 的方案，ARM 作为主控制器，控制图像传感器的图像采集、通讯接口、人机界面等，由 DSP 完成数字图像处理算法，对数字图像进行算法级处理，这是目前众多便携式图像采集系统的主要构成方式；

另外一种是采用 DSP+可编程逻辑芯片 FPGA 或 CPLD 的实现方法，FPGA 或 CPLD 能利用硬件描述性语言设计复杂的逻辑，可以作为整个系统的协处理器来完成某些复杂、高速传输数据的逻辑接口，例如图像传感器的控制、SDRAM 存储等，由 DSP 完成整个系统的数字图像信号处理运算，配合高速存储设备，实现大数据量的吞吐，这种架构也是目前便携式图像采集系统的常用实现方式。^[4-7]

随着现代大容量、高密度、高速度 FPGA 的出现，在这些 FPGA 中一般都内嵌有可配置的高速 RAM、PLL 以及硬件乘法器等 DSP 专用 IP 模块，而且在原有逻辑宏单元的基础上嵌入了许多面向 DSP 的专用模块，结合这些硬件资源使 DSP 开发者能十分容易地在一片 FPGA 上实现整个 DSP 系统，同时 FPGA 中能嵌入高速的 ROM 和 RAM 模块，实现高达 10Mb/s 的读写速率（Altera 公司的 Stratix 系列）^[8]，FPGA 的设计非常灵活，通用 DSP 芯片通常只有 1~4 个乘法器，而在 FPGA 中可以配置数十个乘法器（例如 Altera 公司最新的低成本 Cyclone II 系列 FPGA 可以提供多达 150 个 18×18 位的乘法器），^[9]用来实现通用的 DSP 功能。很多 FPGA 供应商提供了专用的开发软件（例如 Altera 公司的 DSP Builder），以 MATLAB 工具箱的形式出现，利用 MATLAB 中的 Simulink 工具进行图像化设计，建立各种数字信号 DSP 模型，完成后利用 DSP Builder 将其转换成硬件描述性语言 VHDL，通过综合、下载，最后得到能实现 DSP 功能的 FPGA 电路。使得 DSP 开发变得简单，而且 FPGA 高速硬件实现算法的特点使得系统处理速度和效率得到最大限度的发挥。基于 FPGA 数字信号处理开发流程见图 1-1 所示。^[9-13]

在研究了很多便携式图像采集系统的结构和特点后，本文选用了直接用 FPGA 作为整个数字图像采集系统的核心，利用 FPGA 完成对图像传感器的控制、数字图像的处理，嵌入式 ARM 处理器作为协处理器，负责控制显示器、人机接口等。

1. 3 本文的研究内容

本文在系统分析了嵌入式系统、FPGA 的发展和图像处理的有关知识后，结合已有的研究成果，提出了以 FPGA 和 ARM 作为系统主体框架的便携式图像采集系统的设计方法，重点研究了系统软硬件构成的若干问题，并通过多种软件模拟了图像采集的软硬

件实现效果，以及在图像采集中遇到各种噪声模型的滤波处理的预期效果。

1. 前言 介绍便携式数字图像采集系统的应用前景、当前国内外研究现状及本文研究的主要内容。

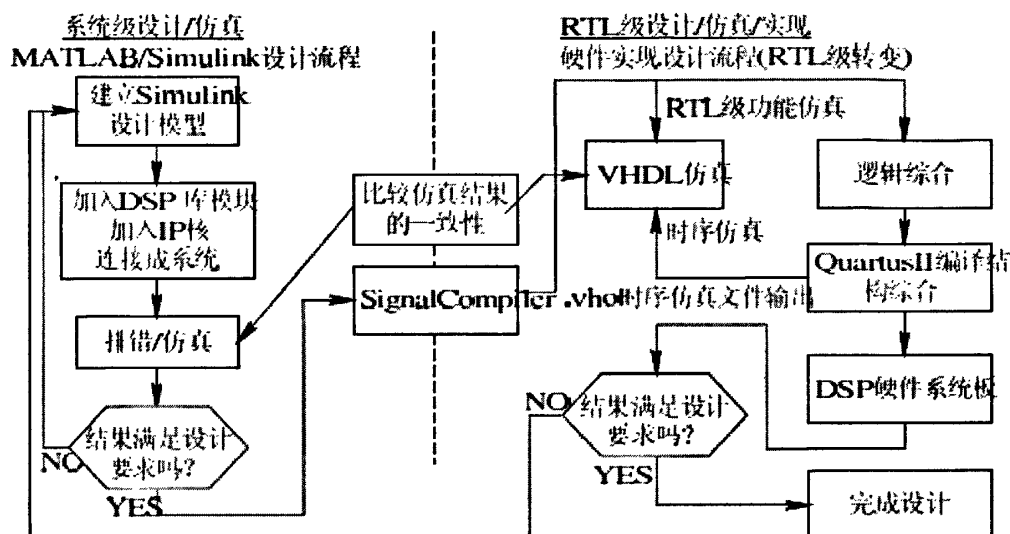


图 1-1 FPGA 数字信号处理开发流程

2. 数字图像采集的相关理论 主要介绍了与数字图像采集相关的理论和技术基础，对数字图像常见噪声，数字图像处理的基本算法中值滤波、平滑滤波等进行了阐述。

3. FPGA 实现数字信号的方法 介绍了目前 FPGA 的开发流程，VHDL 语言的相关特点和 DSP Builder 的使用。

4. 系统硬件模块设计 主要包括本文中所使用的 CMOS 传感器、FPGA 及 ARM 处理器、FPGA 控制 CMOS 传感器模块、FPGA 中值滤波模块的实现、ARM 处理器 LPC2210 硬件模块、TFT6758 和 ARM 的接口模块。

5. 系统软件设计 详细论述了系统的软件设计方法，主要包括 ARM 处理器 LPC2210 引导程序的编写、图像显示模块的驱动、人机接口模块，FPGA 和 ARM 协同工作方法等。

6. 总结及展望 讨论了本系统在进行数字图像采集过程中存在的问题、改进的意见和以后的工作中应努力的方向。

2. 数字图像采集有关理论^[1-3]

2. 1 图像数字化

2. 1. 1 CMOS 传感器的工作原理和结构特点^[14-16]

CMOS (Complementary Metal Oxide Semiconductor, 互补型金属氧化物半导体) 图像传感器诞生于上个世纪八十年代, 时至今日已发展了两代 CMOS 图像传感器, 第一代 CMOS 无源像素传感器, 第二代 CMOS 有源像素传感器。其中以有源像素发展最快, 已由最初的几万像素、几十万像素发展至今的百万像素, 乃至上千万像素的 CMOS 图像传感器。用 CMOS 图像传感器开发的数码相机、微型和超微型摄像机已大批量进入市场。到目前为止, 在开发 CMOS 图像传感器中所采用的先进的关键技术可归纳如下:

- (1) 相关双取样(CDS) 电路技术;
- (2) 微透镜阵列制备技术;
- (3) 彩色滤波器阵列技术;
- (4) 数字信号处理(DSP) 技术;
- (5) 抑制噪声电路技术;
- (6) 模拟数字转换(A/D) 技术;
- (7) 亚微米光刻技术。

CMOS 传感器采用感光元件作为影像捕获的基本手段, 感光元件的核心都是一个感光二极管 (photodiode), 该二极管在接受光线照射之后能够产生输出电流, 而电流的强度则与光照的强度对应, 每个感光元件对应图像传感器中的一个像点, 由于感光元件只能感应光的强度, 无法捕获色彩信息, 因此必须在感光元件上方覆盖彩色滤光片。在这方面, 不同的传感器厂商有不同的解决方案, 最常用的做法是覆盖 RGB 红绿蓝三色滤光片, 以 1: 2: 1 的构成由四个像点构成一个彩色像素 (即红蓝滤光片分别覆盖一个像点, 剩下的两个像点都覆盖绿色滤光片), 采取这种比例的原因是人眼对绿色较为敏感。

CMOS 传感器除了处于核心地位的感光二极管之外, 它还包括放大器与模数转换电路, 传感器中的每一个感光元件都直接整合了放大器和模数转换电路模块, 当感光二极管接受光照、产生模拟电信号之后, 电信号首先被该感光元件中的放大器放大, 然后直接转换成对应的数字信号。

每个像素点的构成为一个感光二极管和三个晶体管，而感光二极管占据的面积只是整个元件的一小部分，造成 CMOS 传感器的开口率远低于 CCD（开口率：有效感光区域与整个感光元件的面积比值），而且 CMOS 感光元件中的放大器属于模拟器件，无法保证每个像素点的放大率都保持严格一致，致使放大后的图像数据无法代表拍摄物体的原貌，体现在最终的输出结果上，就是 CMOS 传感器捕捉到的图像细节丢失情况严重且图像中噪声明显。因而在使用 CMOS 传感器的时候必须使用多种滤波手段对采集到的数字图像数据进行滤除噪声处理。

当然 CMOS 传感器也具有许多优点：CMOS 传感器更容易制造、成本也非常低；CMOS 传感器采用标准的 CMOS 半导体芯片制造技术，很容易实现大批量生产，加之 CMOS 的每个感光元件相互独立，即便有若干个元件出问题，也不会影响传感器的完整性；CMOS 传感器可轻松实现较高的集成度，由于采用半导体工艺制造，厂商可以将时钟发生器、DSP 处理芯片等周边电路与 CMOS 传感器本身整合在一起，从而实现整个图像捕获模块的小型化，也有效降低 OEM 客户的设计难度，同时设计出体积更小的图像捕获装置；CMOS 传感器采用主动式图像采集方式，感光二极管所产生的电荷直接由晶体管放大输出，这种做法虽然导致严重的噪声，但也令 CMOS 传感器拥有超低功耗的优点，在低功耗、小型化图像采集模块等方面 CMOS 传感器都有大量的应用。但由于 CMOS 传感器的结构特点决定了所采集的图像具有大量的噪声，因而 CMOS 传感器所采集的图像必须采用处理方法进行噪声消除。

2. 1. 2 图像的取样与量化

自然界中的图像都是模拟的，模拟图像必须首先经过离散化处理后变成计算机能够识别的点阵图像——数字图像，计算机才能进行处理。这个过程被称为图像的采集。为了把一副模拟图像转换为数字图像，必须在坐标和幅度上都进行取样操作。数字化的坐标值称为取样，数字化幅度值称为量化。

严格的数字图像是一个经过等距离矩形网格采样，对幅度进行等间隔量化的二维函数，因此数字图像实际上就是被量化的二维采样数组。

用 $f(x,y)$ 二维函数形式表示图像，在特定的坐标 (x,y) 处， f 的值或幅度是一个正的标量，其物理意义由图像源决定。将 $f(x,y)$ 通过脉冲相应函数 $h(x,y)$ 转换，得到采样后的图像函数 $g(x,y)$ 见式 (2-1)：

$$g(x,y) = f(x,y) * h(x,y) * s(x,y) \quad (2-1)$$

其中 $s(x, y)$ 见式 (2-2):

$$s(x, y) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \delta(x-m, y-n) \quad (2-2)$$

利用二维单位脉冲 $\delta(x, y)$ 图像函数 $g(x, y)$ 进行采样, 最终完成图像的数字量化。取样和量化的结果是产生一个 M 行和 N 列的矩阵, 一副图像将被分成若干个小区域, 每个小区域对应一个数据, 也就是图像上的一个点, 这样, 整副图像被分为每行若干个像素 (m 个), 和每列若干个像素 (n 个), 及存在 $m \times n$ 个像素点的集合。然后对每个像素点的灰度用 0~255 的数字表示, 这就是量化过程。

2. 1. 3 数字图像采集的常见噪声

图像传感器的工作情况受各种外界因素的影响, 如图像采集的光照环境、传感器本身的质量等。而且图像在传输的过程中, 在信道中又会受到噪声污染。常见的模拟噪声有: 高斯噪声和脉冲噪声 (椒盐噪声)。其中高斯噪声的概率密度随机密度由函数式 (2-3) 给出:

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(z-\mu)^2}{2\sigma^2}} \quad (2-3)$$

其中 z 表示灰度值, μ 表示 z 的平均值或期望值, σ 表示 z 的标准差。标准方差的平方 σ^2 , 称为 z 的方差。高斯噪声在图像中的表现是颗粒状的噪声。

脉冲函数 (椒盐噪声) 的概率密度随机密度由函数式 (2-4) 给出。

$$p(z) = \begin{cases} P_a & z = a \\ P_b & z = b \\ 0 & \text{其他} \end{cases} \quad (2-4)$$

如果 $b > a$, 灰度值 b 在图像中将显示为一个亮点, 相反, a 的值将显示一个暗点。若 p_a 和 p_b 均为 0, 则脉冲噪声称为单极脉冲。若 p_a 和 p_b 均不可能为零, 尤其是它们近似相等时, 脉冲噪声值将类似于随机分布在图像上的胡椒和盐粉微粒, 所以也称其为椒盐噪声。脉冲噪声是正的话, 图像中将出现白点, 脉冲噪声是负的话, 图像中将出现黑点, 图像混入噪声后将出现如图 2-1 所示的情况。



图 2-1 混入了高斯噪声和脉冲噪声的图像

2.2 数字图像降噪处理算法

2.2.1 中值滤波基本原理

中值滤波是由 Turkey 在 1971 年提出的一种基于排序统计理论的有效抑制随机噪声的非线性信号处理技术。中值滤波是统计滤波器中最常见的一种，是一种非线性的空间滤波器，它的响应基于图像滤波器包围的图像区域中像素的排序，然后把数字图像像素灰度的值用该点邻域中各点值的中值替代。其定义为：

已知一组数列 $x_1, x_2, x_3, \dots, x_n$ ，将其按值的大小顺序排列：

$x_{i1} \leq x_{i2} \leq x_{i3} \leq \dots \leq x_{in}$ 则该序列的中值见式 (2-1)。

$$y = \text{median}\{x_1 x_2 \dots x_n\} = \begin{cases} x_{i\left(\frac{n+1}{2}\right)} & (n \text{ 为奇数}) \\ \frac{1}{2} \left[x_{i\frac{n}{2}} + x_{i\left(\frac{n+1}{2}\right)} \right] & (n \text{ 为偶数}) \end{cases} \quad (2-1)$$

例如有一个序列为 (1, 2, 3, 4, 5)，这个序列的中值为 3。

式 2-1 中，若把一个点的特定长度或形状的领域作为窗口，在一维情况下，中值滤波器是一个含有奇数个像素的滑动窗口。窗口正中间的那个像素的值由窗口各像素值的中值来替代，设输入序列为 $\{x_i, i \in I\}$ ， I 为自然数集合或子集，窗口长度为 n ，且令

$u = \frac{n-1}{2}$ ，则滤波器的输出 y 见式 (2-2)。

$$y = \text{median}\{x_i\} = \text{median}\{x_{i-n} \dots x_i \dots x_{i+n}\} \quad (2-2)$$

图 2-2 的是空间滤波的处理机理，假定模板为 3×3 模板。掩模从左到右，从上到下，在待处理的图像中移动。则图像中任一点 (x, y) 的响应可由式 (2-3) 给出。

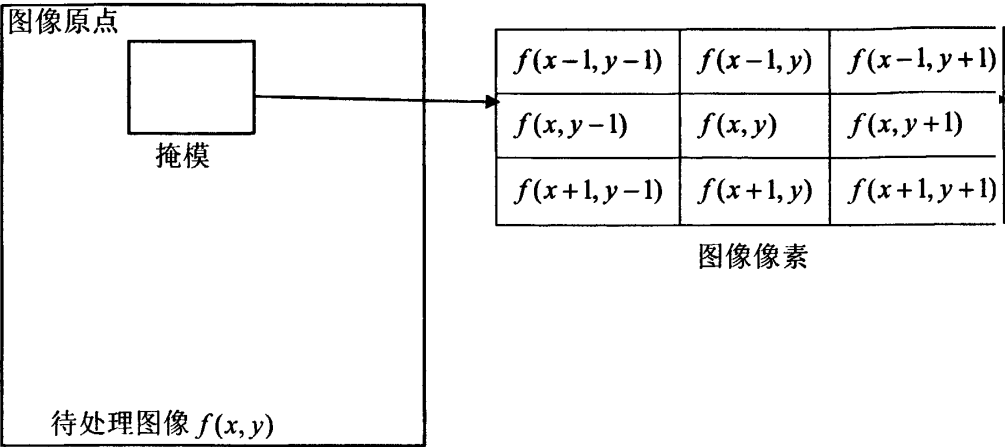


图 2-2 空间滤波的机理

$$R = \omega_1 z_1 + \omega_2 z_2 + \cdots + \omega_9 z_9 = \sum_{i=1}^9 \omega_i z_i \quad (2-3)$$

一个集合中的中值 x 应该是这样来定义的, 在数值集合中有一半的数要小于或等于 x , 也有一半的数要大于 x 。对于一幅图像来说, 要想对图像上的任一点作中值滤波, 首先必须对掩模内的像素点及其邻域进行排序, 然后确定出中值, 并用将中值赋予该像素点, 对于上述的 3×3 的邻域, 其中中值为第 5 个灰度值。

进行空间滤波时有一个重要的问题就是, 当滤波掩模靠近图像边缘的时候发生的情况。对于 3×3 的模板来说, 当掩模中心距离图像的边缘仅为一个像素时, 该掩模至少有一条边与图像边缘重合。如果掩模继续向图像边缘移动, 掩模的行或列就会处于图像平面之外。为了简化起见, 对于边缘的处理, 本文采用的是对于使掩模的中心距离图像边缘为 1 个像素时停止移动 (也可以认为丢弃掉一部分图像), 这样滤波出来的图像会比原图像偏小, 但保证了图像的精度。

中值滤波的优点是运算简单且速度较快, 在滤除随机噪声方面显示出了极好的性能, 但对高斯噪声的消除效果不理想。在某些条件下, 中值滤波方法可以去除噪声, 保护图像边缘, 使图像较好地复原, 它非常适用于一些线性滤波器无法胜任的数字图像处理的应用场合。受噪声污染的图像经过中值滤波后的效果见图 2-3。

2. 2. 2 均值滤波基本原理

均值滤波的机理同样是空间滤波, 是属于一种线性滤波器, 同样对于图 2-2 中的 3×3 窗口, 采用局部平均法, 设窗口中的中心点为 (x,y) , 则该点的灰度值由式 (2-4) 给出。



噪声干扰图像 (a)



中值滤波后 (b)

图 2-3 中值滤波的效果图

$$f(x, y) = \frac{1}{9} \sum_{i=-1}^1 \sum_{j=-1}^1 f(x+i, y+j) \quad (2-4)$$

假设图像中的噪声是随机不相关的，且为加性噪声，在窗口中噪声是独立分布的，通过上式处理后的图像将有效的降低噪声。均值滤波可以平滑图像信号，特别是可以使图像边界变得模糊，均值滤波是以牺牲图像的清晰度来实现消除噪声的。对于图像中的颗粒状的噪声，例如高斯噪声，可以使用均值滤波器去除掉。

3. FPGA 实现数字图像处理方法^[8-10]

3.1 Altera 公司 Cyclone II 系列 FPGA—EP2C35 的特点^[17]

FPGA (Field Programmable Gate Array) 中文名称是现场可编程门阵列, 属于大规模可编程逻辑器件的一种, 《EDN》杂志对可编程逻辑器件作过这样的评价: “可编程逻辑器件是逻辑器件产品中增长最快的领域, 这主要有两个基本原因。可编程逻辑器件不断提高的单片器件逻辑门数量集成了众多功能, 不然的话这些功能只能采用大量分立逻辑和存储器芯片才能实现, 这可以改善最终系统的体积、功耗、性能、可靠性和成本。同样重要的是这样的事实, 在许多情况下只需要数十秒或数分钟的时间, 你就可以在工作站或系统组装线上配置和重新配置这些器件。这一能力提供了强大的灵活性, 支持迅速对最后一分钟设计修改, 以及在设计定型前对各种想法进行原型实验, 同时还可满足在消费者需求和竞争压力下不断缩短的上市时间最终期限要求。” (EDN, “PLD 年度目录”, 2000 年 8 月 17 日)。

可编程逻辑器件能够为用户提供各种逻辑能力、特性、速度和电压特性, 而且可以在任何时候对此类器件进行修改, 以完成多种不同的功能。对于可编程逻辑器件, 设计人员可利用价格低廉的软件工具快速开发、仿真和测试其设计, 然后将设计快速编程到器件中, 并立即在实际运行的电路中对设计进行测试。采用可编程逻辑器件的另一个关键优点是在设计阶段中客户可根据需要修改电路, 直到对设计工作感到满意为止, 这是因为可编程逻辑器件的结构是基于可重写的存储器技术, 即要改变设计, 只需要简单地对器件进行重新编程, 一旦设计完成, 客户可立即投入生产, 只需利用最终软件设计文件简单地对该器件编程即可。

可编程逻辑器件提供了一些重要的优点, 包括:

(1) 在设计过程中为客户提供了更大的灵活性, 因为对于可编程逻辑器件来说, 设计反复只需要简单地改变编程文件就可以了, 而且设计改变的结果可立即在工作器件中看到;

(2) 远程硬件升级。在一个硬件系统已经交给用户使用后, 由于有了可编程逻辑器件, 设备制造商现在可以为已经安装在现场的产品增加新功能或者进行升级。要实现这一点, 只需要通过 Internet 将新的编程文件下载到可编程逻辑器件中就可以在系统中创建出新的硬件逻辑;

(3) 有越来越多的知识产权(IP)核心库的支持。随着各大 FPGA 供应商不断的提供各

种 IP 核，用户可利用这些预定义的模块在可编程逻辑器件内迅速实现系统功能。目前的 IP 核包括：复杂数字信号处理算法（FFT、FIR 等）、存储器控制器（SDRAM 等）、总线接口（PCI、SPI 等）、软核微处理器等模块。此类 IP 核为客户节约了大量时间和费用，缩短电子产品开发周期。随着各 FPGA 供应商继续致力于在可编程逻辑器件中集成更多的功能、降低其成本并提高能够节约开发时间的 IP 核心的可用性，可编程逻辑器件一定会在数字设计人员中进一步普及开。

CycloneII 系列 FPGA 是继 Cyclone 系列低成本 FPGA 在市场上取得成功之后，Altera 公司推出的更低成本的 FPGA，将 FPGA 的密度扩展到了 68, 416 个逻辑单元 (LEs)，从而可以在低成本 FPGA 上实现复杂的数字系统。Cyclone II 系列 FPGA 采用 TSMC(台基电)的 90nm 工艺，与竞争对手的 90nm 工艺 FPGA 相比，性能高出 60%而功耗减低一半，而其价格则几乎可以与 ASIC 产品竞争。优异的性价比使其可以广泛地应用在汽车电子、消费电子、音视频处理、通信以及测试测量等终端产品市场。

Cyclone II 系列 FPGA 支持 Altera 公司的 NIOS II 嵌入式软核处理器。NIOS II 具有灵活的可配置特性而且可以非常容易地实现各种外设的扩展。对于并行事务处理，可以在一个 FPGA 上放置多个 NIOS II 软核，大大提高处理器的效率，也方便多个小组同时开发，进一步加快新产品研发速度。

在数字信号处理方面，Cyclone II FPGA 也具有明显的优势。Cyclone II 系列 FPGA 内置最多可达 150 个 18×18 的硬件乘法器，片上大容量的 M4K RAM 以及经过专门优化的对外部存储器的高速存取特性，使它们非常适合数字信号处理器或数学协处理器的应用场合。Altera 公司提供的数字信号处理器 IP 核以及 DSP Builder 开发软件使数字信号处理产品的开发非常容易。

CycloneII 系列 FPGA 的结构是基于查找表 (Look Up Table, LUT) 结构的由大量的逻辑宏单元 LE (Logic Element) 所构成的可编程逻辑器件。逻辑宏单元是 FPGA 中的最小的单元，每个 LE 都包含有一个 4 输入的 LUT、一个带有同步使能的可编程触发器、一个进位链和一个级联链。其结构见图 3-1 所示。每 16 个 LE 组成一个逻辑阵列块 LAB (Logic Array Block)。

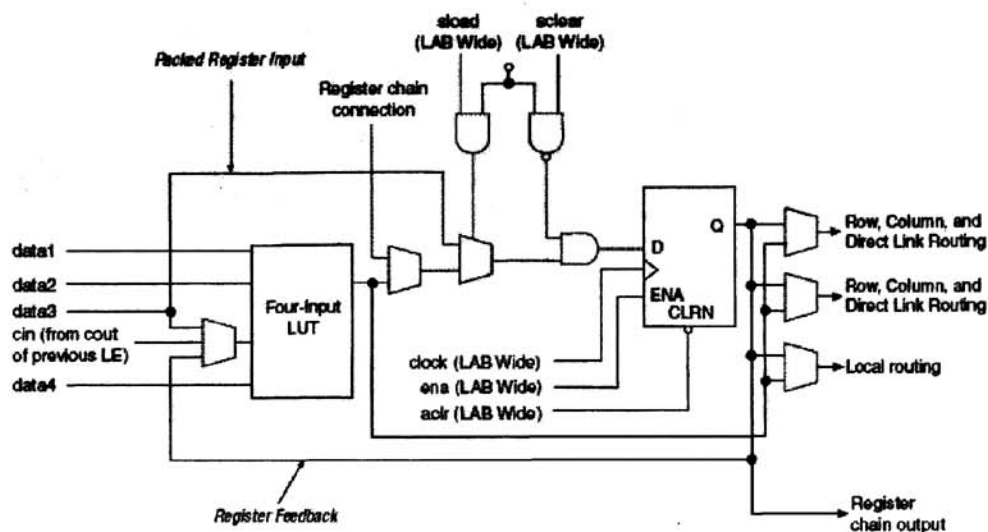


图 3-1 逻辑宏单元 LE 结构图

CycloneII 系列器件主要由以行列形式排列的逻辑阵列块 LAB、嵌入式存储器块 M4K 及嵌入式乘法器组成，锁相环(PLL)为 FPGA 提供时钟，输入输出单元(Input/output elements, IOEs) 提供输入输出接口逻辑。逻辑阵列、嵌入式存储器、嵌入式乘法器、输入输出单元以及锁相环之间可实现各种速度的信号互连。CycloneII 的内部结构图见图 3-2 所示。

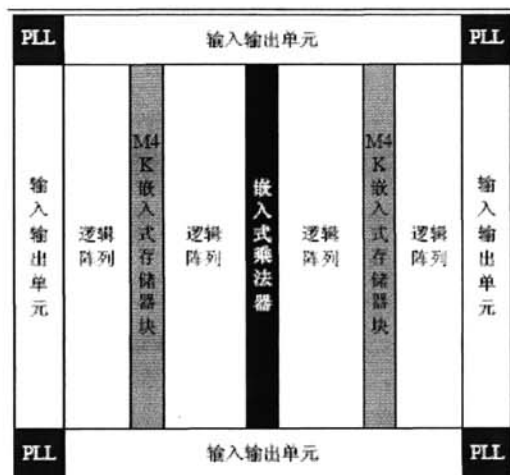


图 3-2 Cyclone II 系列 FPGA 内部结构

CycloneII 系列 FPGA 有片内 PLL，并有最多可达 16 个全局时钟线的全局时钟网络为逻辑阵列块、嵌入式存储器块、嵌入式乘法器和输入输出单元提供时钟。Cyclone II FPGA 的 M4K 嵌入式存储器块由带校验的 4K 位(4096 位)真双口 RAM 组成，可配制

成真双口模式、简单双口模式或单口模式的存储器，位宽最高可达 36 位，存取速度最高可达 260MHz，M4K 嵌入式存储器分布于逻辑阵列块之间。CycloneII 系列 FPGA 的 M4K 嵌入式存储器的容量从 119K 位至 1152K 位不等。每个嵌入式乘法器可以配制成两个 9×9 或一个 18×18 的乘法器，处理速度最高达 250MHz。

3.2 VHDL 语言和 Altera FPGA 的 DSP 开发工具

3.2.1 VHDL 语言

VHDL 的英文全名是 Very-High-Speed Integrated Circuit HardwareDescription Language，诞生于 1982 年。1987 年底，VHDL 被 IEEE 和美国国防部确认为标准硬件描述语言。自 IEEE 公布了 VHDL 的标准版本，IEEE-1076（简称 87 版）之后，各 EDA 公司相继推出了自己的 VHDL 设计环境，或宣布自己的设计工具可以和 VHDL 接口。此后 VHDL 在电子设计领域得到了广泛的接受，并逐步取代了原有的非标准的硬件描述语言。1993 年，IEEE 对 VHDL 进行了修订，从更高的抽象层次和系统描述能力上扩展 VHDL 的内容，公布了新版本的 VHDL，即 IEEE 标准的 1076-1993 版本，（简称 93 版）。现在，VHDL 和 Verilog HDL 作为 IEEE 的工业标准硬件描述语言，得到众多 EDA 公司的支持，在电子工程领域，已成为事实上的通用硬件描述语言。有专家认为，在新的世纪中，VHDL 与 Verilog HDL 语言将承担起大部分的数字系统设计任务。

VHDL 主要用于描述数字系统的结构、行为、功能和接口。除了含有许多具有硬件特征的语句外，VHDL 的语言形式和描述风格与句法是十分类似于一般的计算机高级语言。VHDL 的程序结构特点是将一项工程设计，或称设计实体（可以是一个元件，一个电路模块或一个系统）分成外部（或称可是部分，及端口）和内部（或称不可视部分，即涉及实体的内部功能和算法完成部分）。在对一个设计实体定义了外部界面后，一旦其内部开发完成后，其他的设计就可以直接调用这个实体。这种将设计实体分成内外部分的概念是 VHDL 系统设计的基本点。应用 VHDL 进行工程设计的优点是多方面的。

（1）与其他的硬件描述语言相比，VHDL 具有更强的行为描述能力，从而决定了它成为系统设计领域最佳的硬件描述语言。强大的行为描述能力是避开具体的器件结构，从逻辑行为上描述和设计大规模电子系统的重要保证。

（2）VHDL 丰富的仿真语句和库函数，使得在任何大系统的设计早期就能查验设计系统的功能可行性，随时可对设计进行仿真模拟。

（3）VHDL 语句的行为描述能力和程序结构决定了它具有支持大规模设计的分解和

已有设计的再利用功能。符合市场需求的大规模系统高效、高速的完成必须有多人甚至多个代发组共同并行工作才能实现。

(4) 对于用 VHDL 完成的一个确定的设计, 可以利用 EDA 工具进行逻辑综合和优化, 并自动的把 VHDL 描述设计转变成门级网表。

(5) VHDL 对设计的描述具有相对独立性, 设计者可以不懂硬件的结构, 也不必管理最终设计实现的目标器件是什么, 而进行独立的设计。

3. 2. 2 DSP Builder 开发工具

Altera 为使用 FPGA 进行 DSP 设计提供了专用的开发工具——DSP Builder 和 QuartusII。Altera 开发的 DSP Builder 是连接 Simulink 和 Quartus II 开发软件的 DSP 开发工具。在 DSP Builder 的无缝设计流程中, 设计人员首先在 MATLAB 软件中进行算法设计, 然后在 Simulink 软件中进行系统集成, 最后将设计输出为硬件描述语言(HDL)文件, 以便在 Quartus II 软件中使用。DSP Builder 工具与 SOPC Builder 工具紧密结合, 使用户能够构建整合了 Simulink 设计、Altera 嵌入式处理器以及 IP 内核的系统。对于在使用可编程逻辑设计软件方面缺乏经验的设计人员来说, 该设计流程非常方便、直观。

DSP Builder 的主要特性如下:

- (1) 实现 MATLAB 与 Simulink 软件和 Quartus II 软件的连接;
- (2) 采用 Altera 开发板模块, 可以快速实现设计;
- (3) Hardware In Loop(HIL)模块加速实现 Simulink 系统级协同仿真, 并提供高级调试功能;
- (4) 支持 SignalTap II 逻辑分析仪, 探测 DSP 板上的 Altera 器件信号, 将数据导入 MATLAB 工作空间, 方便进行直观分析;
- (5) 自动启动 Quartus II 软件来对生成的 VHDL 进行编译;
- (6) 支持 Simulink 软件使用的各种定点算法和逻辑操作等。

Altera FPGA 中的 DSP 系统设计需要高级算法设计工具与 HDL 开发工具的良好配合。Altera DSP Builder 将 MATLAB 和 Simulink 系统级设计工具的算法开发、仿真和验证功能与 Quartus II 的基于 Verilog HDL 及 VHDL 语言的设计流程整合在一起, 实现了这些工具的集成。DSP Builder 开发环境界面友好, 帮助设计人员生成 DSP 设计硬件, 从而缩短了 DSP 设计周期。已有的 MATLAB 函数和 Simulink 模块可以和 Altera DSP Builder 模块以及 Altera 知识产权(IP)MegaCore 功能相结合, 将系统级设计实现和 DSP 算法开发相连接。DSP Builder 支持系统、算法和硬件设计共享一个公共开发平台。设

计人员可以使用 DSP Builder 模块迅速生成 Simulink 系统建模硬件。DSP Builder 包括比特和周期精度的 Simulink 模块，涵盖了算法和存储功能等基本操作。可以使用 DSP Builder 模型中的 MegaCore 集成复杂的功能。使用 DSP Builder 完成设计时，首先在 MATLAB/Simulink 软件中建立模型文件(.mdl)，DSP Builder SignalCompiler 模块读取由 DSP Builder 和 MegaCore 模块构建的 Simulink 建模文件(.mdl)，生成 VHDL 文件，利用 FPGA 开发软件进行综合、硬件实施和仿真。DSP Builder 开发流程见图 3-3。

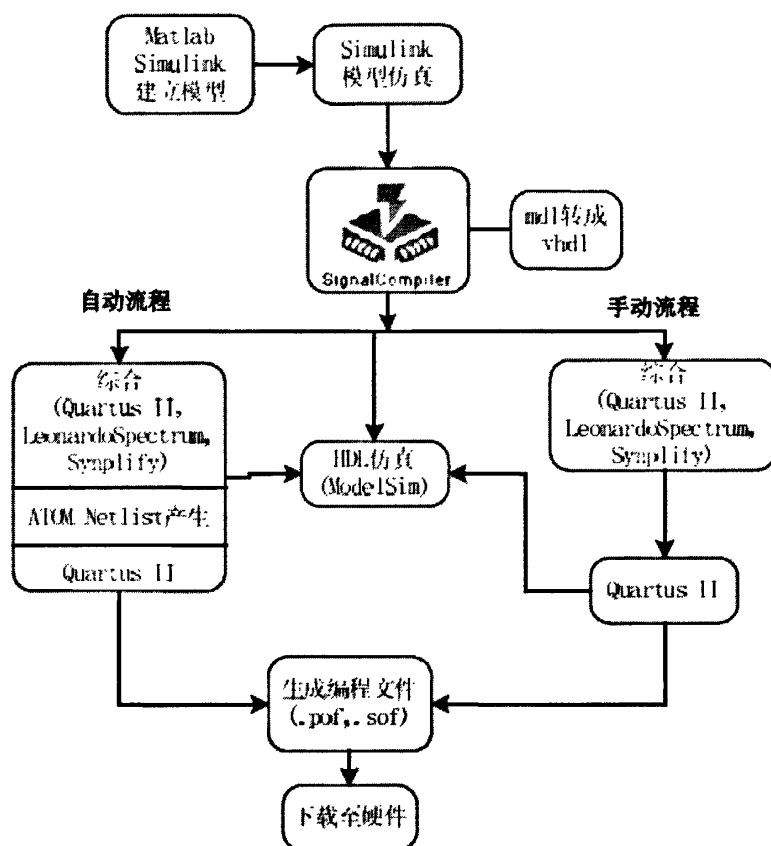


图 3-3 DSP Builder 开发流程

4. 系统硬件模块设计与实现

4. 1 数字图像采集系统组成模块

在研究了大量的便携式图像采集系统后，设计采用了嵌入式处理器 ARM+FPGA 构成的数字图像采集系统，图像采集系统的整体框图如图 4-1 所示，整个系统主要由 ARM 处理器和 FPGA 两个模块构成，两个系统相对独立，互相配合完成图像采集任务。

FPGA 是整个系统的核心部分，负责整个图像采集系统的工作。主要完成以下功能：

- (1)控制 CMOS 图像传感器工作，提供 CMOS 传感器工作时序，并对图像数据进行采集；
- (2)利用 Quartus II 中的 LPM 模块定制一个数据缓冲器（双口 RAM），对采集到的图像数据进行缓冲；
- (3)图像数据在数据缓冲器要能同时进行图像预处理，对图像进行降噪算法；
- (4)将滤波完成的图像数据送入 FIFO 中，并提供图像数据输出接口，可以将图像数据输出。

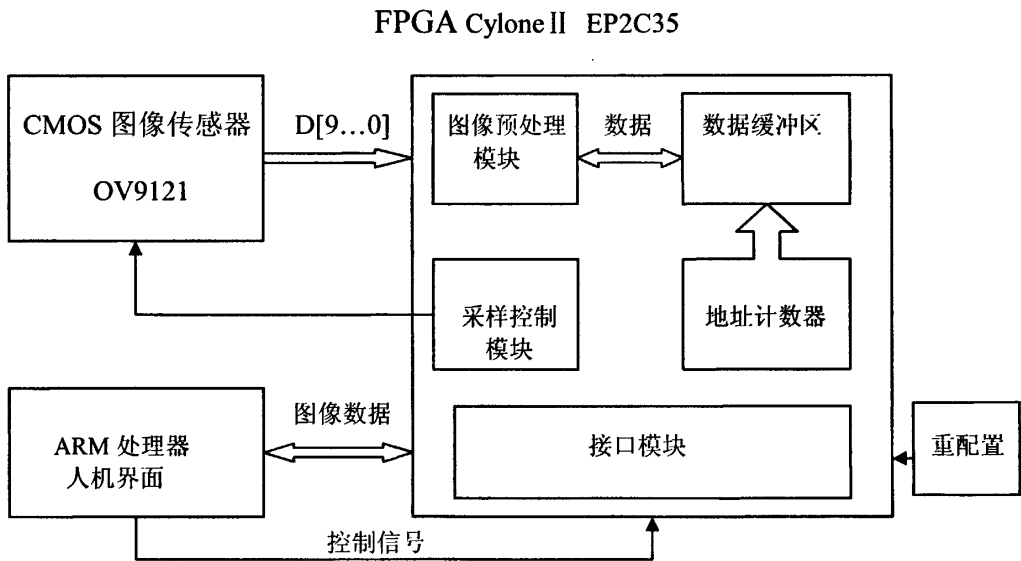


图 4-1 系统整体框图

ARM 处理器作为系统的控制接口和人机界面的实现，主要完成以下任务：

- (1)将 FPGA 采集到的图像数据进行显示，应该设计 ARM 和 LCD 的接口和驱动程序；

(2)负责控制 FPGA 的图像采集启动和停止;

(3)提供和 FPGA 进行数据交换的通道, 可以从 FPGA 中取出数据送往 LCD 显示器等。

系统在完成上述基本功能后还应考虑的问题及实现方法:

(1)FPGA 的上电重配置问题: EP2C35 是基于 LUT 结构 (RAM) 的可编程逻辑器件, 器件掉电后配置信息会完全丢失。FPGA 可以采用多种配置方式, 包括从使用计算机终端通过下载电缆直接下载配置数据的方式, 到利用电路板上的微处理器从存储器空间读取配置数据的配置方式。最通用的方法是使用专用配置器件。本系统中使用 AS (主动串行) 模式, 使用专用的配置芯片 EPCS4, 该芯片容量为 4MB, 支持在系统可编程和多次编程能力。

(2)系统的便携性到底能否达到要求, 包括尺寸和功耗。

(3)FPGA 和 ARM 的工作时序不同, 如何进行同步。

(4)系统电源问题: Cyclone II 兼容 3.3V 和 1.8V 工作电压, ARM 处理器外设 IO 口的工作电压为 3.3V, ARM 的内核和片内外设驱动电压为 1.8V。整个系统的电源必须能提供两种电压。设计采用的电路是使用 SPX1117M3-1.8 和 SPX1117M3-3.3 两个芯片分别产生 1.8V 和 3.3V 电压, 该芯片的工作特点是输出电流大, 输出电压精度高, 稳定性好。

4. 2 各模块工作方式和原理

4. 2. 1 黑白 CMOS 图像传感器 OV9121^[18]

OV9121 是由美国 OmniVision 公司生产的黑白 CMOS 图像传感器,采用最大 135 万像素(1312×1036), 1/2 英寸的 CMOS 传感器,该传感器可以采集静态图像 (SXGA) 和动态的图像 (VGA),常用的静态图像的最大分辨率为 1292×1024 ,动态图像的最大分辨率为 640×480 ,帧速率可达到 30 帧/秒,而且该传感器将 CMOS 光感应核与外围辅助电路集成在一起,采用 SCCB 串行总线进行控制。其内部结构见图 4-2。

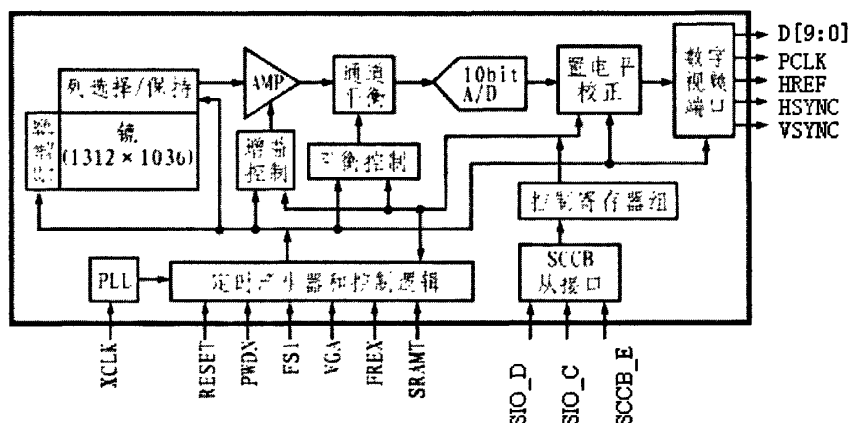


图 4-2 OV9121 内部结构

OV9121 传感器内部有 1312×1036 的 CMOS 传感器阵列,每次采样时可以在这个范围内任选视窗尺寸,即是在 $2 \times 4 \sim 1292 \times 1024$ 之间选择,如图 4-3 所示。

该芯片尺寸为 $6.66 \times 5.32\text{mm}$,采用 48 脚 LCC (Leadless chip carrier) 封装,其中主要使用的控制引脚为:

(1) 像素数据输出口 D[9: 0], OV9121 内部嵌入一个 10 位的 A/D 转换器,可以输出 10 位的图像数据流;

(2) 像素时钟输出信号 PCLK 和水平参考信号 HREF,当 HREF 为高电平时,开始采集图像数据,每个 PCLK 的下降沿输出一个图像数据,输出图像数据的时序见图 4-4 所示;

(3) 复位信号 RESET,该引脚上引入一个不低于 1ms 的高电平可以使传感器复位;

(4) SCCB 串行总线接口 SIO_C、SIO_D 和 SCCB_E 共 3 条线,用来配置 OV9121 内部的所有寄存器,控制其工作方式,控制时序图见图 4-4 所示。3 个引脚互相配合,

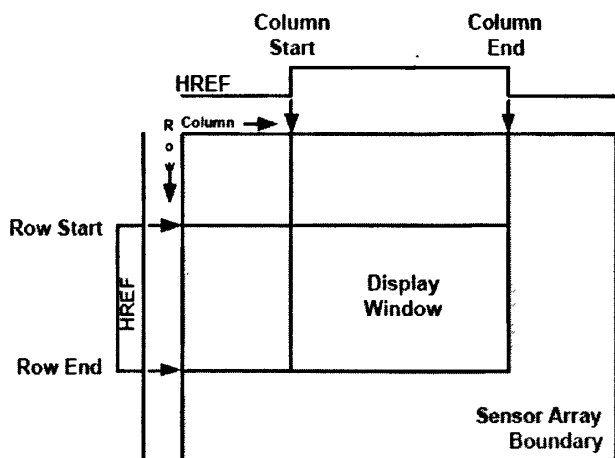


图 4-3 OV9121 视窗输出

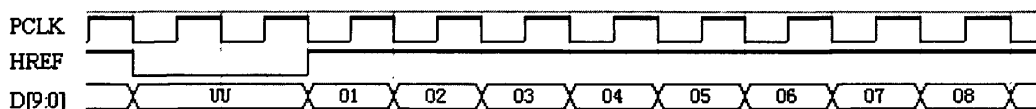


图 4-4 OV9121 静态图像输出时序

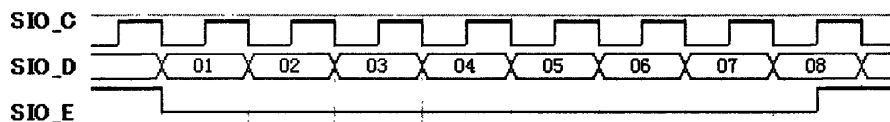


图 4-5 OV9121 SCCB 总线控制时序

其中 SIO_E 是串行总线使能信号，SIO_C 是串行总线时钟信号，SIO_D 是串行总线数据信号。控制总线规定的协议为：当 SIO_E 由高电平变为低电平时，数据传输开始。当 SIO_E 由低电平转化为高电平时，数据传输结束。在数据传输期间，当 SIO_E 始终保持低电平，当 SIO_C 上每出现一个正脉冲时，传感器都将传送一位数据；

(5)XCLK1 和 XCLK2 时钟信号输入，OV9121 可以工作在两种模式下：主模式和从模式，在主模式下，OV9120 作为主导设备，此时 XCLK1 和 XCLK2 上应接外部晶振，当 OV9121 采集到图像后，在 PCLK 的下降沿到来时，系统便可依次将像素值输出，此时外部设备只是被动的接收信号。而在从模式下，OV9120 是作为从属设备，此时 XCLK1 和 XCLK2 不能与外部晶振相连，由 XCLK1 接外部时序信号，可以是控制器输出的时序信号，OV9121 在这个信号的控制下依次发送像素值，及用此信号作为 PCLK。

(6)FREX 电子快门触发端，当 FREX 被置位于 1 时，像素阵列被迅速充电，传感器保持为高以拍摄图像(或物像)。在 FREX 转换到 0 时，视频数据流(datastream) 用逐行读出方式送达到输出端口。

(7) VSYNC 从模式下的垂直同步输入, 当 VSYNC 的上升沿表示一帧图像的到来, 下降沿表示图像数据输出。

CMOS 图像传感器 OV9121 的数据输出特点是逐行传送出传感器阵列曝光数据, 电子曝光控制算法(或系统规则) 则建立在整个图(物) 像亮度基础之上, 对于 SXGA 格式的输出生, 范围则从 2×4 到 1280×1024 , 可以在内部 1312×1036 边界内的任何地方定位, 通过 SCCB 总线写入不同的控制字就可以实现。同时在本系统中, OV9121 应该工作在从模式, 这样方便 FPGA 的控制, 也可以接受 ARM 处理器的控制。

4. 2. 2 FPGA 控制 OV9121 采集和存储数据模块

在本系统中, OV9121 作为系统的图像传感器, 首先在内部将获取的图像采样量化, 然后在外部 FPGA 的控制下输出数字图像, 并将图像数据存入 FPGA 中的缓冲器。FPGA 作为采集系统核心控制逻辑的主控模块, 同时来协调其它各模块的工作。

OV9121 的控制方式是通过 SCCB 总线将控制字送入内部的寄存器中完成的, 其内部共有 54 个寄存器, 不用每次上电都重新设置一遍, 只需要对采集图像的开口位置、窗口大小和工作模式作设定。常用的设定寄存器如下(括号中是寄存器的地址):

CLKRC (11H), 控制 OV9121 的数据输出速度和设定为是否从外部获得时钟信号;

COMH (12H) 设定 OV9121 的工作模式, 可采用默认值;

COMM (32H), 控制寄存器, 和下面的 4 个寄存器一起确定图 4-3 中的传感器的视窗大小;

HREFST (17H) 和 HREFEND (18H), 两个点用来设定视窗的水平尺寸大小, 这两个寄存器只有 8 位, 作为高 8 位分别和 COMM 的 BIT[1: 0]和 BIT[3: 2]一起组合成 10 位数据, 再乘以 2, 最大为 1292 个点;

VSTRT (19H) 和 VEND (1AH), 两个点用来设定视窗的垂直尺寸大小, 这两个寄存器同样只有 8 位, 分别和 COMM 的 BIT[4]和 BIT[5]组合成 9 位数据, 再乘以 4, 最大为 1024 个点;

在本系统中, OV9121 的输出时钟信号直接由 FPGA 来提供, 将 OV9121 设定为工作在从模式且时钟为外部时钟; 采样视窗可以开在任意位置, 但考虑到采样效果, 应该开在比较中间的位置, 同时一副图像的数据量非常大, 因而将传感器的输出定为: 240×128 , 此时一副图像有 30720 个数据, 为 30K 个 10 位数据。在这样的工作方式下控制字分别为: CLKRC 为 40H, COMM 为 0AH, 起始点: VSTRT 为 30H, HREFST 为 30H,

结束点：VEND 为 38H，HREFEND 为 48H。FPGA 和 OV9121 的硬件共有 8 个控制和 10 个数据引脚相连，见表 4-1 所示

表 4-1 FPGA 和 OV9121 硬件连接

FPGA 引脚	OV9121 引脚
PIN A1	RESET
PIN B1	SIO_C
PIN C1	SIO_D
PIN D1	SIO_E
PIN E1	PCLK
PIN F1	HREF
PIN G1	XCLK1
PIN H1	FREX
PIN G1	VSYNC
PIN N1~PIN Y1, PIN AA1, PIN AB1	D[9: 0]

整个采样控制程序用 VHDL 语言编写，工作过程如下：在系统上电后将 OV9120 的 RESET 脚拉至 VCC 高电平，OV9120 将清除全部寄存器，并复位到它们的默认值。然后对 CMOS 图像采集芯片进行初始化，通过 SCCB 串行控制总线接口将控制信号输入，本系统只需对几个不连续寄存器的数据进行更改，系统配置完毕后，在系统控制下进行图像数据的采集。由于没有和 FREX 信号同步的机械快门装置，在采集图像的一副图像过程中，只能通过控制 FREX 信号来控制，这样可能会影响图像数据的完整性。HREF 是水平同步信号，其上升沿表示一系列图像数据的开始。PCLK 输出数据同步信号。HREF 为高电平即可开始有效地数据采集，而 PCLK 下降沿的到来则表明数据的产生，PCLK 每出现一个下降沿，系统便传输一位数据。而 PCLK 下降沿的到来则表明数据的产生，PCLK 每出现一个下降沿，系统便传输一位数据。HREF 为高电平期间，系统共传输 240 位数据。在 VSYNC 为低电平期间，HREF 会出现 128 次高电平。而下一个 VSYNC 信号上升沿的到来则表明分辨率 240 × 128 的图像采集过程的结束。

图像数据的存储利用 Quartus II 中的 LPM 模块，在 FPGA 在生成双口 RAM，由于本系统图像采集的数据要求为 30K，因而生成一个容量为 32K，10 位数据存储的双口 RAM，见图 4-6 所示。

双口 RAM 的功能为：

- 一个数据输入端：Data[9..0]，输入图像数据；

- 写入允许端：wren，高电平有效；
- 写入地址：waddress[15..0]；
- 读出允许：rden，高电平有效；
- 读出地址端：rdaddress[15..0]；
- 输出端：q[9..0]。

在 FPGA 中设计双口 RAM 对 FPGA 的资源消耗比较大，特别是对于 FPGA 中的 memory bits 资源，消耗了 61%，但双口 RAM 不仅为当前的图像采集作缓冲，同时为后续的中值滤波提供了很方便的处理。

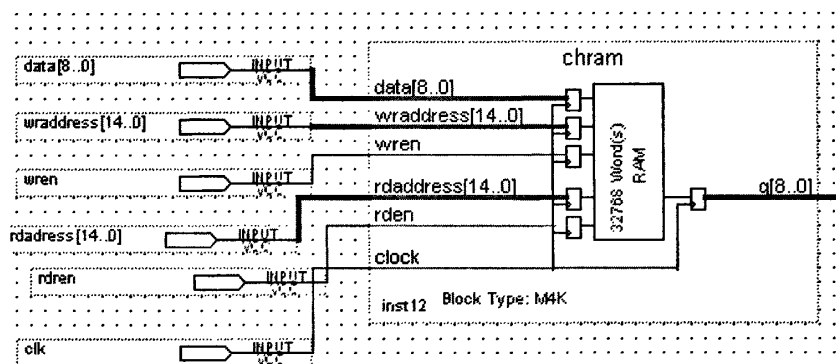


图 4—6 图像缓冲双口 RAM

4.3 FPGA 图像预处理模块^[19-37]

4.3.1 中值滤波模块的设计

通过 2.2.1 对中值滤波原理的分析可知,中值滤波的输出只和相邻的两行、两列图形数据有关,即是只需要通过缓冲 $3 \times 3 = 9$ 个数据,并进行排序就可以得到该模板内的中值数据。因而设计的中值滤波模块的核心是 3 位数据排序模块和 3 位数据中值输出模块,比较方式见图 4-6 所示,一共需要 30 次比较才可以得出最后的中值。其中输入数据以 $x(i, j)$ 为中心的 3 行 3 列数据,其中 A1、A2、A3、B1、B2、B3 为相同的三点排序模块,可以对 3 个输入数据进行排序,自上而下的 3 个箭头表示依次输出最大值,中值,最小值, C 为 3 位数据中值输出模块,可以比较输出 3 个输入数据的中值。

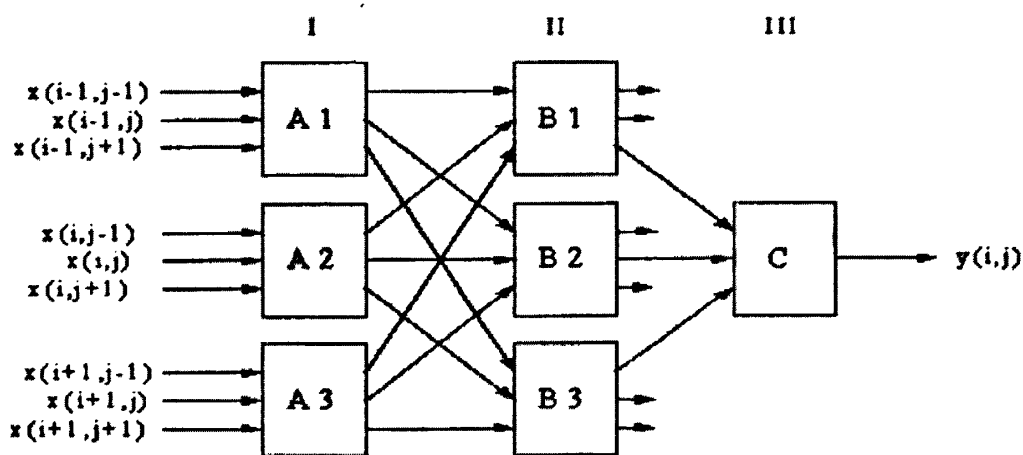


图 4-7 3×3 中值滤波实现

图 4-7 的功能描述如下:比较器 A1、A2、A3 为第一级,分别对各自输入的 3 个数据进行排序,将各自得到的最大值送往 B1 比较器,中值送往 B2,最小值送往 B3, B1、B2、B3 为第二级,运算规则和第一级一样,将 B1 的最小值、B2 的中值和 B3 的最大值送往 C1, C1 是中值比较输出单元,负责输出整个 9 位数据的中值。

利用 Altera DSP Builder 中的仿真模块在 Simulink 来进行设计,其中 3 位数据的排序使用 DSP Builder 中的 If Statement 模块来设计,见图 4-8 (a) 所示,该模块可以编写 IF 语句,判断三个输入数据的大小,并输出是否满足要求的状态。要实现图 4-6 中的比较过程可以使用这样的模块。3 个数据排序共有 6 种情况,利用 6 个该模块即可以实现,最后实现的 3 数据排序模块见图 4-8 (b) 所示,输入数据为 $\text{In}[9:0]a$ 、 $\text{In}[9:0]b$ 、 $\text{In}[9:0]c$,输出为最大值 $\text{Out}[9:0]$ 、中值 $\text{Out1}[9:0]$ 、最小值 $\text{Out2}[9:0]$ 。

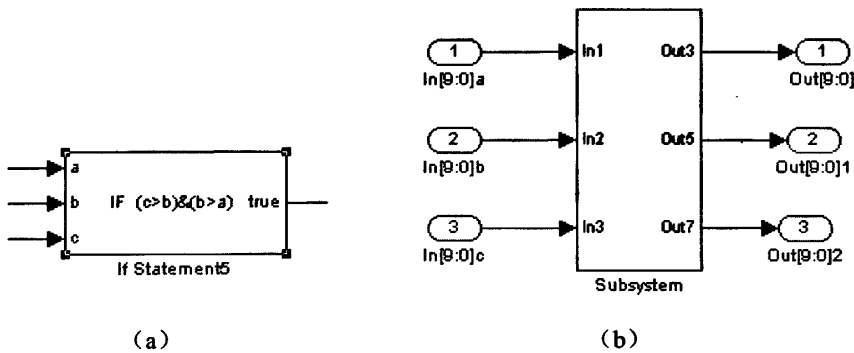


图 4-8 DSP Builder 中的 If Statement 模块和生成的 3 数据排序模块

中值比较输出模块见图 4-8，3 个数比较大小其逻辑真值表见表 4-2。图 4-9 中利用了 3 个比较器 (Comparator、Comparator1, Comparator2)，将 3 个 10 位的输入数据 input, input1, input2 进行两两比较，他们的大小总共有 6 种排序方式，将比较的结果组合成一个 3 位二进制数 sel[2:0]，送入 8 到 1 数据复用器 (n-to-1 Multiplexer) 中，复用器在 sel[2:0] 的控制下选择输出 3 个数据的中值。由于有两种排序方式没有（此时对应的大小是不和逻辑的），所以 2 端和 5 端悬空。

表 4-2 3 位数据比较器输出中值真值表

输入数据大小关系			输出 sel[2:0]	中值
Input<Input1	Input<Input3	Input1<Input3	000	Input1
Input<Input1	Input<Input3	Input1>Input3	001	Input2
Input<Input1	Input>Input3	Input1<Input3	010	不合逻辑
Input<Input1	Input>Input3	Input1>Input3	011	Input
Input>Input1	Input<Input3	Input1<Input3	100	Input
Input>Input1	Input<Input3	Input1>Input3	101	不合逻辑
Input>Input1	Input>Input3	Input1<Input3	110	Input2
Input>Input1	Input>Input3	Input1>Input3	111	Input1

对于一个 3×3 的模板，共有 9 个数据，可以采用 3 个数据一组，求出中值，再将 3 个中值送入上述模板中求出整个模板的中值。

将一共 6 个 3 数据排序模块和一个 3 位数据比较器中值输出模块组成图 4-7 所示的 3×3 中值滤波模块，然后调用 SignalCompiler 来生成 VHDL 语言，调用 Quartus II 进行综合，数据输出见图 4-10 所示，经检查发现中值滤波逻辑基本正常。

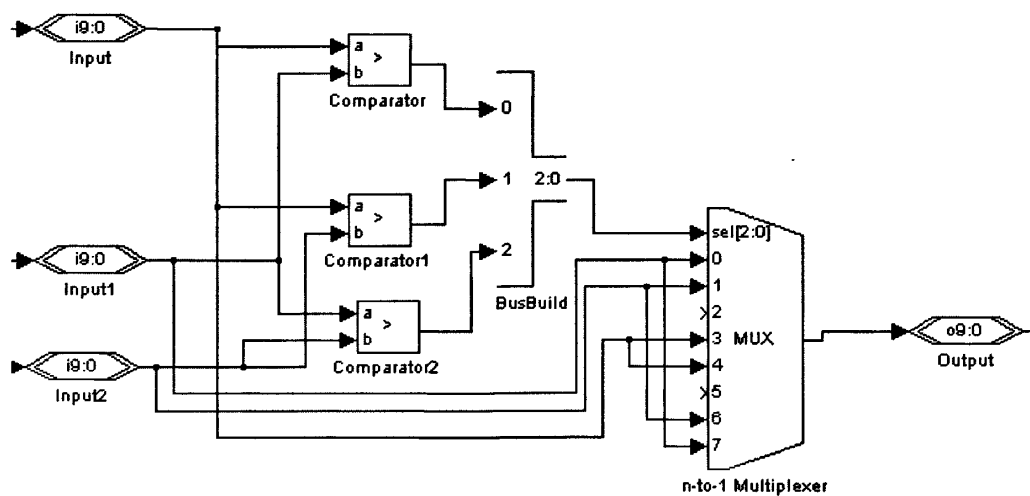


图 4—9 3 位数据比较器中值输出模块











	D1	H 000	025	028	
	D2	H 000	0FF	028	
	D3	H 000	05F	028	
	D4	H 000	089	028	
	D5	H 000	059	028	
	D6	H 000	04E	028	
	D7	H 000	02D	028	
	D8	H 000	0E9	028	
	D9	H 000	0ED	028	
	MED	H 000	05F	028	

图 4—10 中值滤波输出图

4. 3. 2 中值滤波算法在 FPGA 中的实现及图像边缘处理

在数据缓冲区中的图像数据必须对其进行滤波处理，然后才能送入后续的处理模块。在本系统中，利用 OV9120 输出的 VSYNC 信号来作中值滤波模块的启动信号，即在一副 240 ×128 图像采集完成后，启动中值图像滤波模块。

由于系统所采集的 240×128 的图像数据是按行排放的（见 4. 2 分析），以任一点为中心的邻域，假设模板的起始地址为 A0，其地址见表 4—3：

这样为了能取出图像数据送入中值滤波模板内，必须使用一种特殊的地址计数器，该计数器为 15 位计数器（图像数据为 30K，所以需要 2¹⁵ 地址空间），将图像数据按表 4—3 所示输出，然后进行中值滤波。

该地址计数器为 9 进制计数，每计满 9 个脉冲后，计数的初值 A0 自动加一，作为下一个 3×3 模板的初始地址，一直到扫描整个图像区域，滤波完成。

表 4-3 模板地址

A0	A1 = A0 + 240	A2 = A1 + 240
A3 = A0 + 1	A4 = A1 + 1	A5 = A2 + 1
A6 = A3 + 1	A7 = A4 + 1	A8 = A5 + 1

当 A0 加到 239 后, A4=480, A6=241, A6 这个存储单元的数据并不是 A4 的邻域, 而是下一行数据的开始, 如果不进行处理, 中值滤波的结果会出错。这个图像边缘的处理, 常用的方法是在模板扫到这个窗口时, 直接输出 0, 即当 A0 到达图像边缘时 (即 A0=320 的整数倍-2), 直接输出中值为 0, 这样带来的问题是图像的边缘会出现黑线同时图像比原来小。

15 位的计数器可以使用 VHDL 语言来编写, 用 CASE 语句实现, 在 9 个不同的状态下, 计数器输出不同的地址, 将 9 个模板数据依次取出。

...

```

when A0 => addout <= addin; --产生地址 A0
addout1 <= addin; --=A0
addout2 <= addin; --=A0
next_state <= A1;
when A1 => addout <= addin2; --产生地址 A3 = A0 + 240
addout1 <= addin2; --=A0 + 240
addout2 <= addin; --=A0
next_state <= A2;
when A2 => addout <= addin1; --产生地址 A4 = A1 + 1
addout1 <= addin1; --=A1 + 1
addout2 <= addin; --=A0
next_state <= A3;
when A3 => addout <= addin1; --产生地址 A7 = A4 + 1
addout1 <= addin; --=A0
addout2 <= addin2; --=A0 + 240
next_state <= A4;
when A4 => addout <= addin2; --产生地址 A2 = A1 + 240
addout1 <= addin2; --=A1 + 240
addout2 <= addin; --=A0
next_state <= A5;

```

```
when add5 =>addout<= addin1; --产生地址 A5 =A2 + 1
addout1 <= addin1; --=A2 + 1
addout2 <= addin; -- =A0
next_state <= A6;
when add6 =>addout<= addin1; --产生地址 A8 =A5 + 1
addout1 <= addin; --=A0
addout2 <= addin; -- =A0
next_state <= A7;
when A7 =>addout<= addin1; --产生地址 A3 =A0 + 1
addout1 <= addin1; -- =A0 + 1
addout2 <= addin; -- =A0
next_state <= A8;
when A8 =>addout<= addin1; --产生地址 A6 =A3 + 1
addout1 <= addin;
addout2 <= addin;
next_state <= A0; --循环
...
```

由于进行中值滤波时，需要模板内的 9 个数据能同时送到中值滤波模块，计数器输出 9 个数据是一个一个输出的，这样数据将串行送到中值滤波模块，必须在数据输入端加入存储模块来实现，在地址计数器控制下，内部缓冲器输出的数据流，首先送入 3 片小容量的双口 RAM，图 4-10 所示的方法实现。

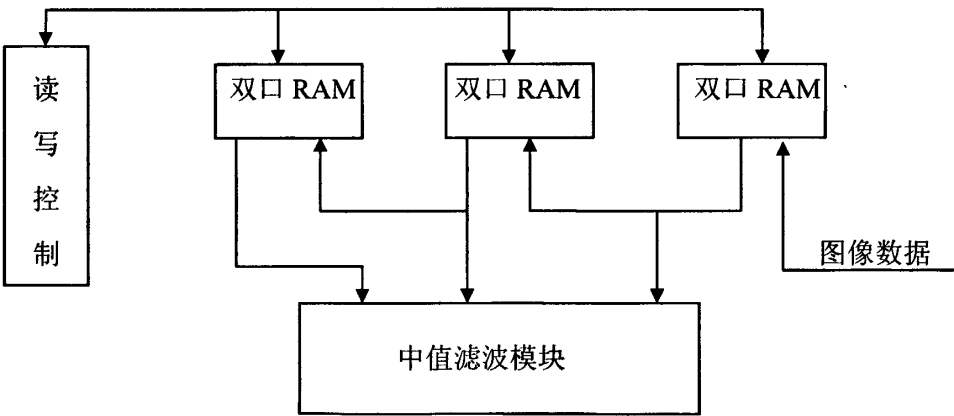


图 4-10 双口 RAM 实现数据并行输入

为了让 ARM 处理器能从 FPGA 上读取中值滤波输出数据，在中值滤波后设计 32K

的 FIFO, 见图 4-11 所示。FIFO 只有一个时钟信号 clock, data[9..0] 是数据输入端, q[9..0] 是数据输出端, wrreq 是写允许, 高电平有效, rdreq 是读允许, 高电平有效。FIFO 最有特色的是 almost_full 输出信号, 该信号可以设定在缓冲了多少数据的时候发出“几乎满”的信号, 这里为了能让 ARM 能方便的取走数据, 设定为 30K, 即缓冲数据到 30K 的时候, 就发出这个信号。

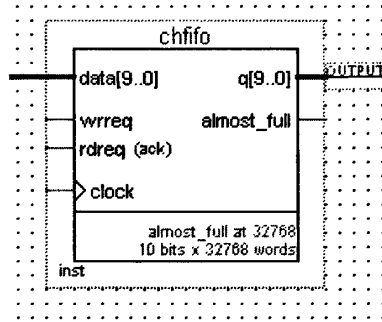


图 4-11 图像输出 FIFO

4.4 嵌入式 ARM 处理器模块^[38-44]

4.4.1 ARM (Advanced RISC Machines)

ARM (Advanced RISC Machines), 既可以认为是一个公司的名字, 也可以认为是对一类微处理器的通称, 还可以认为是一种技术的名字。

1991 年 ARM 公司成立于英国剑桥, 主要出售芯片设计技术的授权。目前, 采用 ARM 技术知识产权 (IP) 核的微处理器, 即我们通常所说的 ARM 微处理器, 已遍及工业控制、消费类电子产品、通信系统、网络系统、无线系统等各类产品市场, 基于 ARM 技术的微处理器应用约占据 32 位 RISC 微处理器 75% 以上的市场份额, ARM 技术正在逐步渗入到我们生活的各个方面。

ARM 公司是专门从事基于 RISC 技术芯片设计开发的公司, 作为知识产权供应商, 本身不直接从事芯片生产, 靠转让设计许可由合作公司生产各具特色的芯片, 世界各大半导体生产商从 ARM 公司购买其设计的 ARM 微处理器核, 根据各自不同的应用领域, 加入适当的外围电路, 从而形成自己的 ARM 微处理器芯片进入市场。目前, 全世界有几十家大的半导体公司都使用 ARM 公司的授权, 因此既使得 ARM 技术获得更多的第三方工具、制造、软件的支持, 又使整个系统成本降低, 使产品更容易进入市场被消费者所接受, 更具有竞争力。

ARM7 系列微处理器为低功耗的 32 位 RISC 处理器, 最适合用于对价位和功耗要求较高的消费类应用。ARM7 微处理器系列具有如下特点:

- 具有嵌入式 ICE-RT 逻辑, 调试开发方便。
- 极低的功耗, 适合对功耗要求较高的应用, 如便携式产品。
- 能够提供 0.9MIPS/MHz 的三级流水线结构。
- 代码密度高并兼容 16 位的 Thumb 指令集。
- 对操作系统的支持广泛, 包括 Windows CE、Linux、Palm OS 等。
- 指令系统与 ARM9 系列、ARM9E 系列和 ARM10E 系列兼容, 便于用户的产品升级换代。
- 主频最高可达 130MIPS, 高速的运算处理能力能胜任绝大多数的复杂应用。

ARM7 系列微处理器的主要应用领域为: 工业控制、Internet 设备、网络和调制解调器设备、移动电话等多种多媒体和嵌入式应用。

ARM7 系列微处理器包括如下几种类型的核: ARM7TDMI、ARM7TDMI-S、

ARM720T、ARM7EJ。其中，ARM7TMDI 是目前使用最广泛的 32 位嵌入式 RISC 处理器，属低端 ARM 处理器核。

TDMI 的基本含义为：

T：支持 16 为压缩指令集 Thumb；

D：支持片上 Debug；

M：内嵌硬件乘法器 (Multiplier)；

I：嵌入式 ICE，支持片上断点和调试点。

4.4.2 ARM 处理器 LPC2210 的特点

Philip 公司的 LPC2210 是一款基于支持实时仿真和嵌入式跟踪的 16/32 位 ARM7TDMIS CPU 的微控制器。芯片采用 144 脚封装，有 16 KB 片内静态 RAM，开放外部总线；通过外部存储器接口可将外部存储器配置成 4 组，每组的容量高达 16 MB，数据宽度 8/16/32 位均可；具有多个 32 位定时器、8 路 10 位 PWM 输出、多个串行接口（包括 2 个 16C550 工业标准 UART、高速 I2C 接口和 2 个 SPI 接口）以及 9 个外部中断、多达 76 个可承受 5 V 电压的通用 I/O 口，同时内嵌实时时钟和看门狗，片内外设功能丰富强大；片内晶振频率范围 1~30 MHz，通过片内 PLL 可实现最大为 60 MHz 的 CPU 工作频率，具有 2 种低功耗模式——空闲和掉电，通过外部中断将处理器从掉电模式中唤醒，并可通过个别使能/禁止外部功能来优化功耗。以上特性，使其特别适用于工业控制、医疗系统、访问控制和 POS 机，同时也非常适合于通信网关协议转换器、嵌入式软 Modem，以及其他各种类型的应用。

LPC2210 的主要特性：

- 16/32 位 ARM7TDMI-S 核，LQFP144 封装。
- 16 kB 片内静态 RAM。
- 串行 BOOT 装载程序通过 UART0 来实现在系统下载和编程。
- Embedded ICE-RT 和嵌入式跟踪接口使用片内 RealMonitor 软件对任务进行实时调试并支持对执行代码进行无干扰的高速实时跟踪。
- 8 路 10 位 A/D 转换器，转换时间低至 2.44 μ s。
- 2 个 32 位定时器（带 4 路捕获和 4 路比较通道）、PWM 单元（6 路输出）、实时时钟和看门狗。
- 多个串行接口，包括 2 个 16C550 工业标准 UART、高速 I2C 接口（400 kbit/s）和 2 个 SPI 接口。

- 向量中断控制器。可配置优先级和向量地址。

- 通过外部存储器接口可将存储器配置成 4 组，每组的容量高达 16Mb，数据宽度为 8/16/32 位。

- 多达 76 个通用 I/O 口（可承受 5V 电压），9 个边沿或电平触发的外部中断引脚。

- 通过片内锁相环（PLL）可实现最大为 60MHz 的 CPU 操作频率。

- 片内晶振频率范围：1~30 MHz。

- 2 个低功耗模式：空闲和掉电。

- 通过外部中断将处理器从掉电模式中唤醒。

- 可通过个别使能/禁止外部功能来优化功耗。

- 双电源

—CPU 操作电压范围：1.65~1.95 V(1.8 V \pm 0.15 V)

—I/O 操作电压范围：3.0~3.6 V(3.3 V \pm 10%)，可承受 5V 电压。

4. 4. 3 LPC2210 控制显示接口

本文所使用的 LPC2210 平台是由广州周立功提供的基于 PHILIPS 公司的 LPC2210 的 ARM 单片机开发板—SmartARM2200。

板上资源很多，主要有：

- 2M 字节 NOR FLASH（型号为 SST39VF160）。

- 16M 字节 NAND FLASH（型号为 K9F2808U0C）。

- 标准 20 针 JTAG 调试接口/ETM 跟踪调试接口。

- 1 个 2.2 英寸 240 \times 320 262K 色彩色液晶屏 TFT6758。

- 1 个蜂鸣器。

- 4 \times 4 矩阵键盘等。

在本系统中主要使用 2.2 英寸的 TFT6758 液晶显示屏，提供显示用，同时矩阵键盘可以作为扩展人机接口使用。

TFT 6758 液晶模块工作电压为 3.3V，屏幕大小为 2.2in，点像素为 240 \times 320，内部的控制器为 HD66781 和 HD66783，采用 16 位数据连接方式，连接电路图见图 4-6，说明见表 4-2。

由于液晶的 RS 引脚接 A1，该引脚为高电平时数据操作，为低时命令操作，则液晶的操作地址为：指令口（命令口）：0x83200000，数据口：0x83200002。

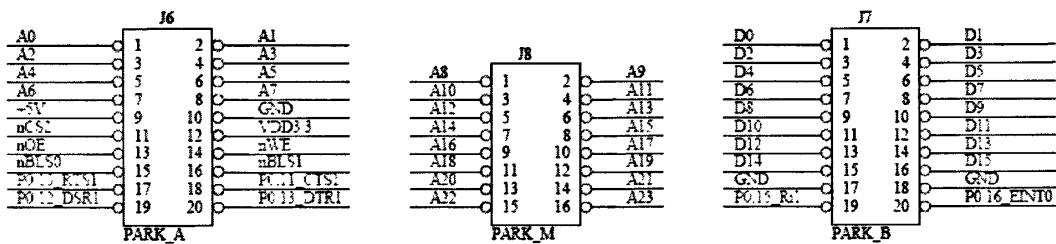
表 4—2 TFT 液晶接口电路与主板的信号连接

TFT 液晶接口	SmartARM2200 主板	功能说明
DB0~DB15	D0~D15	16 位数据口
RS	A1	地址线
nCS	LCM_nCS0	片选
new	nWE	写信号
nOE	nOE	读信号
RESET	P0. 22	用低电平复位
LEDC	P0. 21_PWM5	液晶背光开关

彩色液晶的背光是白色的 LED，必须设计相应的升压电路，CAT32TDI 芯片是专用的白光 LED 升压驱动器，可以驱动 4 个串连的白光 LED。由 LPC2210 的 P0. 21 脚来控制，可以使用 PWM 方式来控制背光亮度。

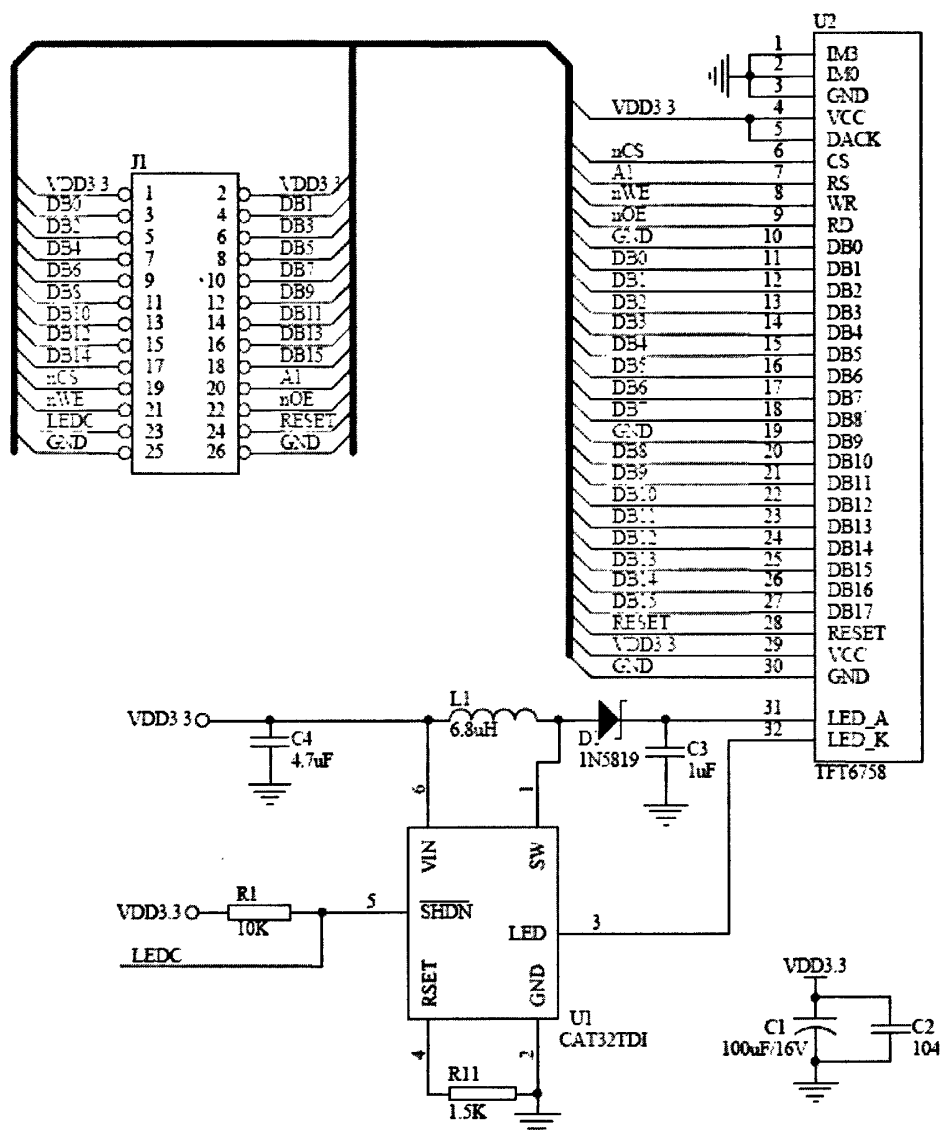
4. 4. 4 LPC2210 和 FPGA 的连接

整个图像采集系统受 LPC2210 的控制，FPGA 在 ARM 处理器的控制下完成 CMOS 传感器的启动和数据的采集以及中值滤波等操作，最后将得到的数据送到 ARM 处理器中，供后续显示的使用。ARM 板上已经提供了外接总线控制口，电路如图 4—11 所示，具有 24 根地址总线 A0~A23，16 根数据总线 D0~D15，读/写信号为 nOE、WR、BLS0 和 BLS1，片选信号为 CS2，地址范围为：0x82000000—0x82FFFFFF。



Address: 0x82000000 --- 0x82FFFFFF
图 4—11 ARM 板外设 PACK 接口电路

J7 和 FPGA 的数据输出口相连，同时将 J6 口的读控制信号 nOE 和 FPGA 相连，作为 FIFO 的输出写允许信号（见图 4—11），每读一次就启动一次数据输出；同时将一个 PIO 口作为输出，用来启动 FPGA，同时设定一个 PIO 为输入，和 FIFO 的 almost_all 信号连在一起，作为图像采集完成信号。



5. 系统软件设计及工作流程^[41-47]

5. 1 ARM 系统软件设计

5. 1. 1 ADS1.2 介绍

开发工具使用 ARM 公司提供的 ADS 1. 2 (ARM Developer Suite1. 2), ADS 集成开发环境是 ARM 公司推出的 ARM 核微控制器集成开发工具。ADS1.2 支持 ARM10 之前的所有 ARM 系列微控制器, 支持软件调试及 JTAG 硬件仿真调试, 支持汇编、C、C++源程序, 具有编译效率高、系统库功能强等特点, 可以在 Windows98、Windows XP、Windows2000 以及 RedHat Linux 上运行。ADS1.2 由 6 个部分组成, 见表 4-3 所示。

表 4-3 ADS1.2 组成部分

名称	描述	使用方式
代码生成工具	ARM 汇编器, ARM 的 C、C++编译器, Thumb 的 C、C++编译器, ARM 连接器	由 CodeWarrior IDE 调用
集成开发环境	CodeWarrior IDE	工程管理, 编译连接
调试器	AXD, ADW/ADU, armsd	仿真调试
指令模拟器	ARMulator	由 AXD 调用
ARM 开发包	一些底层的例程, 实用程序(如 fromELF)	一些实用程序由 CodeWarrior IDE 调用
ARM 应用库	C、C++函数库等	用户程序使用

5. 1. 2 BootLoader 代码的编写

ARM 应用系统中, 软件大多数采用 C 语言进行编程, 并且以嵌入式操作系统为开发平台, 这样就大大提高了开发效率及软件性能。为了使系统在上电复位后能可靠的初始化, 采用一个汇编文件作为启动代码使最常用的方法, 它可以实现向量表定义、堆栈初始化、系统变量初始化、中断系统初始化、I/O 接口初始化、外围初始化、地址重映射等操作, 这个用汇编编写的引导程序也称为 Bootloader 代码。

ARM 芯片上电或复位后, 系统进入管理模式、ARM 状态、PC (R15) 指向 0x00000000 地址处, 所以必须保证向量表代码定位在 0x00000000 处, 或者映射到

0x00000000 处, LPC2210 系统上电或复位时, 存储器映射控制寄存器等于 0, BootBlock 被重映射到地址 0x00000000 处, 选择从 BootBlock 读取中断向量。中断向量为每一个中断设置 1 个字 (4 个字节) 的存储空间, 存放一条跳转指令, 通过这条指令使 PC 指针指向相应的中断服务程序入口, 继而执行相应的中断处理程序。LPC2210 的中断向量表和其它基于 ARM 核的芯片中断向量表较类似, 只要注意 LPC2210 要使向量表所有数据 32 位累加和为零。

初始化存储器系统是初始化代码中的一个重要部分, 因为许多操作系统在开始运行之前, 希望了解存储器的组织情况。存储器系统初始化是通过软件设定 Flash、RAM 存储器的地址范围, 数据总线宽度。由于 LPC2210 是总线开放型芯片, 具有 4 个 BANK 的存储器组, 总线宽度可设置为 8 位、16 位或 32 位。

初始化堆栈、改变处理器模式, ARM 处理器有 7 种处理器模式, 分别为: 用户模式、快速中断模式、普通中断模式、管理模式、数据访问中止模式、未定义指令中止模式、系统模式。除用户模式之外的其他 6 种模式称为特权模式, 在这些模式下, 程序可以访问所有的系统资源, 也可以任意进行处理器模式的切换。每一种模式的堆栈指针寄存器 (SP) 都是独立的, 因此对程序中需要用到每一种模式都要给 SP 寄存器定义一个堆栈地址。方法是改变状态寄存器 CPSR 中的状态位, 使处理器切换到相应的模式, 然后再给 SP 赋值。这里需要注意的有: 一是要尽量给堆栈分配快速和高带宽的存储器, 因为堆栈性能的好坏对系统整体性能有非常重要的影响。二是在处理器模式切换时, 不要过早地进入用户模式, 因为进入用户模式就不能对 CPSR 进行修改切换到别的模式, 这样会对接下去地程序造成影响, 一般在系统初始化的最后阶段才将运行模式切换到用户模式。

为了 LPC2210 基本能够工作, 在进入 main() 函数前, 须对目标系统进行一些基本的初始化工作。LPC2210 有不同的存储器映射方式, 须根据硬件来设置存储器映射方式; 为了避免混乱, 最好在进入 main() 函数前设置系统各部分时钟等, 这个一般在 target.c 函数中设置。

PHILIPS 公司提供了可以供 ADS 软件使用的 6 个工程模板, 在这些模板一般包含的设置信息有 FLASH 起始地址 0x00000000、片内 RAM 起始地址 0x40000000、片外 RAM 起始地址为 0x80000000、编译连接选项及编译优化级别等等; 模板中包含了 LPC2200 系列 ARM7 微控制器的启动文件, 包括 Startup.S、Target.C; 模板还包含了 LPC2200 系列 ARM7 微控制器的头文件, 分散加载描述文件(如: mem_a.scf、mem_b.scf

mem_c.scf)等等。

主要修改如下：在 Startup. S 文件中修改，始化外部总线控制器，根据目标板决定配置，设置总线宽度。

...

ResetInit;

```

    LDR    R0, =PINSEL2
    IF :DEF: EN_CRP
        LDR    R1, =0x0f814910
    ELSE
        LDR    R1, =0x0f814914
    ENDIF
    STR    R1, [R0];
    ENDIF

```

另外要修改 InitStack 函数，用来设置堆栈和管理模式：

InitStack

```

        MOV    R0, LR
;Build the SVC stack
;设置管理模式堆栈
        MSR    CPSR_c, #0xd3
        LDR    SP, StackSvc
;Build the IRQ stack
;设置中断模式堆栈
        MSR    CPSR_c, #0xd2
        LDR    SP, StackIrq
;Build the FIQ stack
;设置快速中断模式堆栈
        MSR    CPSR_c, #0xd1
        LDR    SP, StackFiq
;Build the DATAABORT stack
;设置中止模式堆栈

```



```

        MSR      CPSR_c, #0xd7
        LDR      SP, StackAbt
;Build the UDF stack
;设置未定义模式堆栈
        MSR      CPSR_c, #0xdb
        LDR      SP, StackUnd
;Build the SYS stack
;设置系统模式堆栈
        MSR      CPSR_c, #0x5f
        LDR      SP, =StackUsr
        MOV      PC, R0

```

mem_b.scf 中需要修改的代码:

...

ERAM 0x80040000

```

{
    * (+RW,+ZI)
}

```

...

mem_a.scf 中需要修改的内容, 设定片外和片内 RAM 的起始地址:

...

ERAM 0x81000000

```

{
    * (+RW,+ZI)
}

```

...

IRAM 0x40000000

```

{      Startup.o (MyStacks)
}

```

STACKS 0x40004000 UNINIT

```

{

```

Startup.o (Stacks)

}

在 target.c 中配置存储器接口访问速度

void TargetResetInit(void)

{

#ifdef __DEBUG

MEMMAP = 0x3; /*重新配置 Bank0 的访问速度*/

#endif

#ifdef __OUT_CHIP

MEMMAP = 0x3;

#endif

#ifdef __IN_CHIP

MEMMAP = 0x1;

#endif

...

}

5. 1. 3 图像显示程序编写

为了要输出图像数据，要用 ARM 处理器 LPC2210 来控制 TFT6758，TFT6758 液晶模块驱动芯片为 HD66781，门驱动(Gate driver)芯片为 HD66783。首先根据驱动芯片的功能编写驱动程序，详细的驱动程序可见附录 lcddriver.c 文件。TFT6758 是彩色显示，在本系统中只是显示灰度图像。利用 ADS1.2 来编写，调用 LPC2200 的专用模板，ADS1.2 会将如图 5—1（a）所示的文件加入项目中，将图中的文件按 5.1.2 所分析的修改，然后将液晶驱动程序 lcddriver.c 拷贝到项目所在目录，并将其加入项目，将如图 5—1（b）中。

然后编写 main()函数，显示外部图像数据的程序可以按如下方式编写，设定在 FPGA 中缓冲的数据地址为 FIFO，使用 EasyJTAG 可以调试液晶输出图像。

void Disp_DATA(void)

{ uint32 x, y, no;

uint16 dat;

TftSetWrite(0,0);

no = 0;

for(y=0; y<128; y++)

```

{  for(x=0; x<240; x++)
    {
        dat =Read( FIFO);
        TftSendDat(dat);
        no++;
        no++;
    }
}
}

```

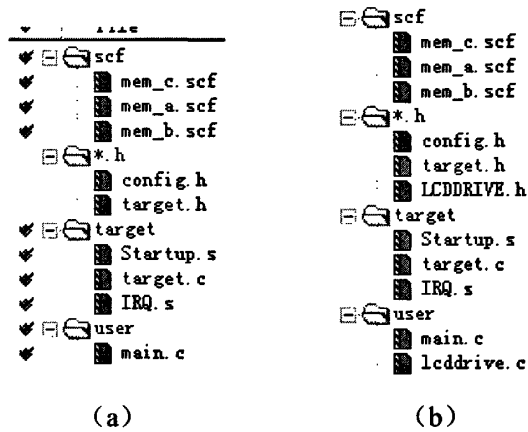


图 5—1 ADS1.2 项目图

利用液晶显示的图像可以显示出 FPGA 采集到的图像数据和中值滤波后的效果。

5.2 系统工作流程

在系统分析整个系统的软、硬件结构后，现在来分析一下系统的工作流程。系统上电后，由配置芯片 EPCS4 对 FPGA 进行上电配置。FPGA 中将装入整个图像采集系统的主体部分。ARM 由引导程序启动后，首先进行两模块的协调。由于两个模块使用的是不同的时钟，必须进行相应的协调，利用 ARM 来作为主要控制器。FPGA 上电重配置完成后，并不启动工作，在 ARM 的一个 PIO 的控制信号过来后，FPGA 启动工作，进行图像数据采集，并将采集数据进行滤波处理，然后将数据送入 FIFO 中，同时给出采样完成信号。ARM 在接收到数据后，驱动 TFT6758 显示图像。

6. 全文的总结和展望

6.1 全文总结

本论文的相关研究工作是在我所主持的孝感市 2006 年度科技计划项目《嵌入式控制器在智能病理仪器中的开发与移植研究》(孝科技发[2006]25 号)中的一个研究子课题。该图像采集系统主要由两大模块构成, FPGA 负责完成图像 CMOS 传感器的驱动和图像数据的预处理和中值滤波, 在 FPGA 中设计了一个大容量的双口 RAM 来实现缓冲图像数据的同时进行中值滤波, 并设计一个高速的 FIFO 来缓冲滤波后的图像数据。没有设计外围存储器而将图像采集和滤波系统整合在一片芯片内的设计是本文的创新点。ARM 处理器 LPC2210 负责将图像数据进行显示, 为此设计了 TFT6758 的驱动程序, 并在 ADS1.2 下仿真调试, 最后显示了采集到的图像。

经过以上的讨论, 在 FPGA 本文设计的便携式数字图像采集系统基本能正常工作, 本文所采用的大容量双口 RAM 缓冲图像数据的同时进行中值滤波, 使 FPGA 能完成整个图像采集任务, 而 ARM 处理器只作为控制器来使用。

6.2 展望

由于部分实验设备没有到位, 对于图像的采集中有许多细节问题没有作深入的研究, 这些都是在今后的领域中需要继续研究的。

首先要求图像采集系统图像的分辨率非常高时(远大于本系统的 240×128), 那么这么大的数据量的保存仅仅利用 FPGA 内部资源是绝对不够的, 为此应外接扩展 RAM, 考虑到 Cyclone II 本身带有 SDRAM 等扩展接口, 这个应是高分辨率图像采集系统的基本架构; 另外, 在图像的噪声消除方面, 仅仅采用了中值滤波, 对于图像增强处理没有涉及, 实际采到的图像比较模糊, 带有复杂图像增强、处理的系统是今后另外一个努力的方向; 第三, 本文采集的图像只是灰度图像, 对于彩色图像没有设计, 和 OV9121 外观功能几乎完全相同的 OV9620 就是彩色 CMOS 传感器。但彩色图像采集时会遇到三基色数据的处理等问题也是今后需要重点努力的方向; 第四, 本系统能否应用于动态数字图像采集也是一个值得研究的课题, 传感器 OV9121 是可以进行动态图像采集的, 但大量的图像数据的存储和处理是值得深入研究的; 第五, 本文为了显示采集到的图像数据, 是将数据读入 ARM, 利用 TFT6758 提供的驱动程序显示了一副图像数据, 几乎没有使用 MiniGUI 等其他功能, 同时, LPC2210 可以移植 uCLinux, 可以实现更加复杂的人机接口和控制界面。这些问题都有待在今后的学习中认真总结和继续研究。

参考文献

- [1] (美)冈萨雷斯(Gonzalez, R.C.)等著;阮秋琦等译. 数字图像处理(第二版)[M]. 北京: 电子工业出版社, 2003 年
- [2] 赵荣椿. 数字图像处理导论[M]. 西安: 西北工业大学出版社, 2000 年
- [3] 阮秋琦. 数字图像处理学[M]. 北京: 电子工业出版社, 2001 年
- [4] 杨遂军. 基于FPGA的高速图像采集技术研究[D]. 哈尔滨: 哈尔滨工业大学 20040601
- [5] 孙咏. 基于ARM和DSP的嵌入式实时图像处理系统设计与研究[D]. 杭州: 浙江大学 20050201
- [6] 王应军, 杨国胜, 范秋凤, 等. DSP+fpga的实时图像处理硬件系统设计[J]. 河南: 河南大学学报(自然科学版), 2006年01期: 91—93.
- [7] 陆海东, 吴明赞. 基于 DSP+FPGA 结构的小波图像处理系统设计[J]. 北京: 电子技术应用, 2006 年第 3 期: 93-95.
- [8] 徐光辉, 程东旭, 黄如, 等. 基于FPGA的嵌入式开发与应用[M]. 北京: 电子工业出版社, 2006年
- [9] 潘松, 黄继业, 王国栋. 现代DSP技术[M]. 西安: 西安电子科技大学出版社, 2003.8 第一版: 1-18
- [10] Altera CO. DSP 解决方案[EB/OL]. 2007 年. <http://www.altera.com.cn/technology/dsp/dsp-index.jsp>
- [11] 徐飞, 施晓红, 等. MATLAB应用图像处理[M]. 西安: 西安电子科技大学出版社, 2002.5第一版: 1-7
- [12] 孙兆林. Matlab 6X 图像处理[M]. 北京: 清华大学出版社, 2002 年 5 月第一版: 225-226
- [13] 张望, 常青, 喻小虎. 应用 TEXTIO 和 MATLAB 进行复杂数字系统仿真[J]. 西安: 国外电子元件, 2006 年 1 月第 1 期: 4-7.
- [14] 陈榕庭. CMOS 图像传感器封装与测试技术[M]. 北京: 电子工业出版社, 2006 年 7 月第 1 版.
- [15] 程开福 CMOS 图象传感器的原理及应用[J] 半导体情报 2001 年 10 月, 38 卷第 5 期.

- [16] 应芳琴 孟利民 陈友荣. 基于 IIC 总线技术的视频信息采集系统[J]. 上海: 电子技术, 2005 年第 3 期: 23-25.
- [17] Altera CO. "Section I. Cyclone II Device Family Data Sheet" [EB/OL] .
2005年. <http://www.altera.com.cn>
- [18] OmniVision CO. "OV9620/9121 Data Sheet" [EB/OL].
2004年. <http://www.ovt.com.cn/>
- [19] 黄睿, 方康玲. 基于 fpga 的数字滤波器的硬件实现及其应用[J] 北京: 微计算机信息, 2006 年第 22 卷第 9 期: 16-17.
- [20] 江霞. 在 matlab 中实现 fpga 硬件设计[J]. 西安: 国外电子元器件, 2005 年第 4 期: 4-6.
- [21] 李辉, 蒋秀明, 高殿斌, 等. matlab 语言在数字图像中值滤波中的应用研究[J]. 天津: 天津工业大学学报, 2003 年第 22 卷第 1 期: 87-88.
- [22] 邹英永. 基于 CPLD 技术的高速视频图像采集与显示系统[J] . 长春: 长春大学学报, 2006 年 16 卷第 4 期: 41-43.
- [23] 惠本利, 丁凤华. 基于 FPGA 的图像滤波算法实现[J]. 济南: 现代制造技术与装备, 2005 年第 4 期: 22-23.
- [24] 陈镇. 中值滤波的 FPGA 实现方案[J]. 上海: 红外, 2005 年第 10 期: 17-21.
- [25] 孙宏琦, 施维颖, 巨永锋. 利用中值滤波进行图像处理[J]. 西安: 长安大学学报, 2003 年第 23 卷第 2 期: 103-106.
- [26] 李雷鸣, 张焕春, 张波. 一种基于 FPGA 的图像中值滤波器的硬件实现[J]. 南京: 电子工程师, 2004 年 2 月第 30 卷第 2 期: 48-50.
- [33] Priyadarshan Kolte, Roger Smith, Wen Su. A Fast Median Filter using Altivec[A]. IEEE International Conference on Computer Design (ICCD'99)[C]. 1999. 384-391.
- [27] 郑鹤, 王鲁平, 李颀. 一种快速的二维中值滤波算法及其硬件实现[J]. 四川: 信息与电子工程, 2005 年 12 月第 3 卷第四期: 245-248.
- [28] 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 年 6 月第 26 卷: 61-63.
- [29] 王阿川. 基于 FPGA 高速图像采集卡的研制[J]. 上海: 自动化仪表, 2005 年 6 月第 26 卷第 6 期: 21-23.
- [30] 盛磊, 徐科军. 基于 DSP 和 FPGA 的实时视频处理平台的设计与实现[J]. 安徽:

中国科学技术大学学报, 2006 年 3 月第 36 卷第 3 期: 304-309.

[31] 刘斌兵, 刘云海, 汪燮彬. 用 FPGA 实现高速大图像采集系统[J]. 上海: 电子技术, 2006 年 01 期: 60-63.

[32] 张桂林, 张留洋. 数字图像处理算法评估系统的硬件设计[J]. 武汉: 计算机与数字工程, 2005 年 33 卷第 12 期: 88-91.

[33] 候彦宾, 李玉山, 张静. 基于 DSP 与双目 CMOS 摄像头的数字图像处理系统[J]. 北京: 电子技术应用, 2006 年第 1 期: 64-66.

[34] 刘春平, 简耀波, 刘承香, 等. 一种污渍识别算法及实现[J]. 成都: 电讯技术, 2006 年第 3 期: 74-77.

[35] 刘刚, 向健勇. 一种高速图像存储系统的设计[J]. 南京: 电子工程师, 2006 年 5 月 32 卷第 5 期: 39-41.

[36] 周剑波, 巩宪锋, 王长松, 等. 利用 FPGA 和 USB 总线的视频图像的采集与处理系统设计[J]. 北京: 北京科技大学学报, 2006 年 9 月第 28 卷第 9 期: 886-889.

[37] 王宇舟, 金声震. 基于 FPGA 的高速图像预处理系统设计[J]. 成都: 电子科技大学学报, 2005 年 2 月第 34 卷第 1 期: 12-15.

[38] PHILIPS CO. LPC2200 User Manual [EB/OL]. 2004. <http://www.philips.com.cn/>

[39] ARM CO. ARM Architecture Reference Manual[EB/OL]. 2004.

<http://www.arm.com/chinese/>

[40] ARM CO. The ARM-THUMB Procedure Call Standard [EB/OL]. 2004.

<http://www.philips.com.cn/>

[41] 周立功. ARM 嵌入式系统基础教程[M]. 北京: 北京航空航天大学出版社, 2005.

[42] 周立功. ARM 嵌入式系统实验教程(一)[M]. 北京: 北京航空航天大学出版社, 2004.

[43] 周立功. ARM 嵌入式系统实验教程(二)[M]. 北京: 北京航空航天大学出版社, 2005.

[44] 周立功. ARM 嵌入式系统实验教程(三)[M]. 北京: 北京航空航天大学出版社, 2005.

[45] 张宏财, 赵曾贻. 基于 ARM 核的 BOOTLOADER 代码的分析与设计[J]. 北京: 微计算机信息, 2006 年第 22 卷第 52 期: 137-139.

[46] 陈华, 仇潜. 基于 LPC2104 的 μ C/OS-II 引导装载系统的设计与实现[J]. 辽宁:

辽宁大学学报, 2006 第 33 卷第 1 期: 75-78.

[47] 周立功. ARM 嵌入式 MiniGUI 初步与应用开发范例[M]. 北京: 北京航空航天大学出版社, 2006

[48] Jean.Labrosse.嵌入式实时操作系统 uC/OS- II (第 2 版).邵贝贝等译[M]. 北京: 北京航空航天大学出版社, 2003.

附 录

TFT6758 驱动程序部分源程序

// 定义 LCM 像素数宏

```
#define GUI_LCM_XMAX    240                /* 定义液晶 x 轴的点数 */
```

```
#define GUI_LCM_YMAX    128                /* 定义液晶 y 轴的点数 */
```

文件名: LCDDRIVE.C

/* 功能: TFT6758 液晶模块驱动。*/

```
#include "config.h"
```

```
/******
```

* 功能: 向 TFT 液晶模块发送命令字。

```
*****/
```

```
void TftSendCom(uint16 com)
```

```
{  HD66781_IR = com;
```

```
}
```

```
/******
```

* 功能: 向 TFT 液晶模块发送数据。

```
*****/
```

```
void TftSendDat(uint16 dat)
```

```
{  HD66781_DR = dat;
```

```
}
```

```
/******
```

* 功能: 读取 TFT 液晶模块的数据。

```
*****/
```

```
uint16 TftRcvDat(void)
```

```
{  uint16  ret;
```

```
    ret = HD66781_DR;
```

```
    return(ret);
```

```
}
```

```
#define MAX_WAIT    10000    /* 等待超时控制值 */
```

```

/*****
* 功能: 向 TFT 液晶模块发送串行命令(HD66781-->HD66783)。
*****/

void TftSerialCom(uint16 indx, uint16 dat)
{
    uint32 delay;
    uint16 bak;

    TftSendCom(GATEIC_DAT);
    TftSendDat(dat);           // 发送数据
    delay = 0;
    TftSendCom(GATEIC_CON);
    while(1)
    {
        bak = TftRcvDat();     // 读 TE 位(0110H 寄存器的 d8 位)
        if((bak&0x0100) == 0) break; // 若 TE=0, 表明允许发送数据
        delay++;
        if(delay>MAX_WAIT) return; // 超时操作
    }
    indx |= 0x0100;           // 设置 TE=1
    TftSendDat(indx);         // 指定目标寄存器索引
}

/*****
* 名称: TftSetAddr
* 功能: 设置数据地址指针(坐标值 x 为 0—239, y 为 0—127)。
*****/

uint8 TftSetAddr(uint16 x, uint16 y)
{
    TftSendCom(SET_ADDR_X);   // 设置地址低 8 位
    TftSendDat(x);
    TftSendCom(SET_ADDR_Y);   // 设置地址高 9 位
    TftSendDat(y);
    if((x<240) && (y<320)) return(TRUE);
    else return(FALSE);
}

```

```
/******
```

* 功能: 设置数据地址指针(坐标值), 并发送写数据命令。

* 接着不断的写入数据即可, GRAM 地址会自动增加。

```
*****/
```

```
uint8  TftSetWrite(uint16 x, uint16 y)
```

```
{  uint8  ret;
```

```
    ret = TftSetAddr(x, y);
```

```
    TftSendCom(WR_RD_DATA);    // 写数据
```

```
    return(ret);
```

```
}
```

```
/******
```

* 功能: 设置数据填充窗口大小。

```
*****/
```

```
uint8  TftSetWindow(uint16 x0, uint16 y0, uint16 x1, uint16 y1)
```

```
{  // 参数过滤
```

```
    if(x1>239) x1 = 239;
```

```
    if(y1>319) y1 = 319;
```

```
    if((x1<=x0) || (y1<=y0)) return(FALSE);
```

```
    // 设置数据填充窗口大小
```

```
    TftSendCom(SET_WINDOW_SX);
```

```
    TftSendDat(x0);
```

```
    TftSendCom(SET_WINDOW_SY);
```

```
    TftSendDat(y0);
```

```
    TftSendCom(SET_WINDOW_EX);
```

```
    TftSendDat(x1);
```

```
    TftSendCom(SET_WINDOW_EY);
```

```
    TftSendDat(y1);
```

```
    return(TRUE);
```

```
}
```

```

/*****
* 功能：全屏填充。直接使用数据填充显示缓冲区。
*****/

*/
void Tft_FillSCR(uint16 dat)
{
    uint16 i,j;
    TftSetWindow(0,0, 239, 319);
    TftSetWrite(0, 0);
    for(i=0; i<GUI_LCM_YMAX; i++)    // 历遍所有行
    {
        for(j=0; j<GUI_LCM_XMAX; j++) // 历遍所有行
        {
            TftSendDat(dat);          // 填充数据
        }
    }
}

/*****
* 功能：控制 TFT 液晶模块复位。
*****/

void TftReset(void)
{
    IO0DIR = IO0DIR | HD66781_RST;

    IO0CLR = HD66781_RST;
    DelaymS(10);
    IO0SET = HD66781_RST;
    DelaymS(10);
}

// 定义 TFT6758 初始化数据信息
STCOM_INI  Tft6758_INI[] =
{
    // 设置电源
    {HD66783_COM, 0x0001, 0x0474, 0},// HD66783 电压设置 1
    {HD66783_COM, 0x0002, 0x02DE, 0},    // HD66783 电压设置 2
    {HD66783_COM, 0x0001, 0x0C74, 0},

```

```

{HD66781_COM, 0x0100, 0x0050, 0},// HD66781 电源控制
{HD66783_COM, 0x0000, 0x080C, 0},    // HD66783 电源控制
{HD66783_COM, 0x0002, 0x012E, 0},
{HD66783_COM, 0x0001, 0x04B2, 30},
{HD66783_COM, 0x0001, 0x04F0, 0},
{HD66783_COM, 0x0003, 0x0008, 0},// HD66783 扫描模式设置
{HD66783_COM, 0x0006, 0x09C0, 70},    // HD66783 输出设置
// 打开显示
{HD66783_COM, 0x0000, 0x0C0A, 0},
{HD66781_COM, 0x0100, 0x0110, 0},// 电源控制 1, STB=0, SLP=0, DSTB=0 (正常工作模式)
{HD66781_COM, 0x0007, 0x0001, 50},
{HD66781_COM, 0x0007, 0x0012, 0},
{HD66783_COM, 0x0000, 0x0C08, 0},
{HD66781_COM, 0x0007, 0x0003, 50},
{HD66781_COM, 0x0007, 0x0013, 0},
{HD66781_COM, 0x0007, 0x0113, 0},// 显示控制 1, D1:D0=11(使能显示)
// 其它设置
{HD66781_COM, 0x0001, 0x0100, 0},// 驱动输出控制, SS=1
{HD66781_COM, 0x0002, 0x0700, 0},// 驱动波形控制
{HD66781_COM, 0x0003, 0x1030, 0},// 启动模式设置, HWM=0, ID1:ID0 =11, AM=0
(不使用缓冲, 地址自动更新为"左上"至"右下"方式)
                                     // BRG=0, OSD=0, TRI=0 (点
像素数据为:RRRRRGGGGGBBBBB)
{HD66781_COM, 0x0004, 0x0000, 0},// 缩放控制 1
{HD66781_COM, 0x0005, 0x0000, 0},// 缩放控制 2
{HD66781_COM, 0x0008, 0x0303, 0},// 显示控制 2, FP3:FP0=3, BP2:BP0=3 (非显示
周期数)
{HD66781_COM, 0x0009, 0x142C, 0}, // 显示控制 3
{HD66781_COM, 0x000B, 0x0000, 0},    // 显示控制 4, FRCON=0, COL1:0=00
(262144 色)

```

```

{HD66781_COM, 0x000C, 0x0001, 0},    // 外部显示接口控制 1
{HD66781_COM, 0x000D, 0x0010, 0},    // 帧周期控制
{HD66781_COM, 0x000E, 0x0210, 0},    // 外部显示接口控制 2
{HD66781_COM, 0x000F, 0x0000, 0},    // 外部显示接口控制 3
{HD66781_COM, 0x0010, 0x0000, 0},    // 门驱动(Gate driver)接口控制 1
{HD66781_COM, 0x0011, 0x0000, 0},    // 门驱动(Gate driver)接口控制 2
{HD66781_COM, 0x0012, 0x0002, 0},    // 门驱动(Gate driver)接口控制 3
{HD66781_COM, 0x0013, 0x0000, 0},    // 门驱动(Gate driver)接口控制 4
{HD66781_COM, 0x0015, 0x0000, 0},    // 门驱动(Gate driver)接口控制 5
{HD66781_COM, 0x0016, 0x0000, 0},    // 门驱动(Gate driver)接口控制 6
{HD66781_COM, 0x0017, 0x0000, 0},    // 门驱动(Gate driver)接口控制 7
{HD66781_COM, 0x0018, 0x0005, 0},    // 门驱动(Gate driver)接口控制 8
{HD66781_COM, 0x0019, 0x0000, 0},    // 门驱动(Gate driver)接口控制 9
{HD66781_COM, 0x001B, 0x0000, 0},    // 门驱动(Gate driver)接口控制 10
{HD66781_COM, 0x0200, 0x0000, 0},    // GRAM 地址设置(R201H,R200H), 0x0000
{HD66781_COM, 0x0201, 0x0000, 0},
{HD66781_COM, 0x0203, 0x0000, 0},    // GRAM 写屏蔽控制(R204H,R203H), 0x0000
{HD66781_COM, 0x0204, 0x0000, 0},
{HD66781_COM, 0x0210, 0x0000, 0},    // window 地址设置(R210H,R211H):水平方向为
0--239 (x = 239+1)
{HD66781_COM, 0x0211, 0x00EF, 0},
{HD66781_COM, 0x0212, 0x0000, 0},    // window 地址设置(R212H,R213H):垂直方向为
0--319 (y = 319+1)
{HD66781_COM, 0x0213, 0x013F, 0},
{HD66781_COM, 0x0400, 0x0027, 0},    // 液晶光栅行设置 ,NL0[5:0]=39, 即
(39+1)*8=320
{HD66781_COM, 0x0401, 0x0001, 0},    // 基本显示控制, REV=1(正方向显示),
VLE=0(不滚屏)

```

```

{HD66781_COM, 0x0402, 0x0000, 0},// 显示起始行地址(y), 0
{HD66781_COM, 0x0403, 0x013F, 0},// 显示终止行地址(y), 319
{HD66781_COM, 0x0404, 0x0000, 0},// 滚屏行数, 0
{HD66781_COM, 0x0405, 0x0000, 0},    // 缩放显示起始行地址(y), 0
{HD66781_COM, 0x0406, 0x0000, 0},    // 缩放显示终止行地址(y), 0
{DATA_END,    0,    0,    0}
};

```

```

/*****

```

* 功能: 初始化 TFT 液晶模块。

```

*****/

```

```

void TftInit(void)

```

```

{ STCOM_INI  init_dat;
  int  i;
  TftReset();
  for(i=0; i<100; i++)    // 设定最多 100 条初始化命令
  {  init_dat = TFT6758_INI[i];
    if(init_dat.type==DATA_END) break;
    if(init_dat.type==HD66783_COM)
    {  TftSerialCom(init_dat.com, init_dat.dat);
    }
    else
    {  TftSendCom(init_dat.com);
      TftSendDat(init_dat.dat);
    }
    DelaymS(init_dat.dly);
  }
}

```

```

/*****

```

* 功能: 初始化 GUI, 包括初始化显示缓冲区, 初始化 LCM 并清屏。

```

*****/

```

```

*/

```

```

void GUI_Initialize(void)

```

```

{  TftInit();

```

```

    TFT_FillSCR(GUI_CCOLOR);
}

/*****

* 功能：全屏填充。直接使用数据填充显示缓冲区。

*****/

*/

void GUI_FillSCR(TCOLOR dat)
{
    TFT_FillSCR(dat);
}

/*****

* 功能：清屏。

*****/

*/

void GUI_ClearSCR(void)
{
    TFT_FillSCR(GUI_CCOLOR);
}

/*****

* 功能：在指定位置上画点。

*****/

*/

uint8 GUI_Point(uint16 x, uint16 y, TCOLOR color)
{
    /* 参数过滤 */
    if(x>=GUI_LCM_XMAX) return(0);
    if(y>=GUI_LCM_YMAX) return(0);

    /* 刷新显示 */
    TftSetWrite(x, y);
    TftSendDat(color);
    return(1);
}

/*****

* 功能：读取指定位置点的颜色数据。

*****/

```



```

*/
uint8 GUI_ReadPoint(uint16 x, uint16 y, TCOLOR *ret)
{
    uint16 dat;
    uint16 bak;
    /* 参数过滤 */
    if(x>=GUI_LCM_XMAX) return(0);
    if(y>=GUI_LCM_YMAX) return(0);

    /* 读取数据 */
    TftSetAddr(x, y);    // 设置地址
    TftSendCom(WR_RD_DATA);
    bak = TftRcvDat();
    bak = TftRcvDat();
    dat = (bak>>11);      // B 色
    dat |= (bak<<11);     // R 色
    dat |= (bak&0x07E0);  // G 色
    *ret = dat;
    return(1);
}

/*****
* 功能：画水平线。
*****/

*/
void GUI_HLine(uint16 x0, uint16 y0, uint16 x1, TCOLOR color)
{
    uint16 bak;
    if(x0>x1)                // 对 x0、x1 大小进行排列，以便画图
    {
        bak = x1;
        x1 = x0;
        x0 = bak;
    }
    GUI_Point(x0, y0, color);    // 显示第一点
    x0++;
    while(x1>=x0)

```

```

    { TftSendDat(color);          // 不断填充并显示
      x0++;
    }
}

/*****
* 功能：画垂直线。
*****/

*/

void GUI_RLine(uint16 x0, uint16 y0, uint16 y1, TCOLOR color)
{ uint16 bak;
  if(y0>y1)          // 对 y0、y1 大小进行排列，以便画图
  { bak = y1;
    y1 = y0;
    y0 = bak;
  }
  while(y1>=y0)
  { GUI_Point(x0, y0, color);    // 逐点显示，描出垂直线
    y0++;
  }
}

```

致 谢

本课题的研究与论文的撰写都是在潘永才老师的悉心指导下完成的。潘老师对科研工作的严谨态度、一丝不苟的科研作风、以及深厚的理论功底给了我极大的帮助和启发，使我受益匪浅。他是我科研工作中的一个楷模，也将会是我今后的工作学习中的重要榜样。另外还要感谢田茂老师，他给了我很多帮助和指导，在他那里我学到了很多有用的东西。

同时要感谢 3 年来给予我关心和帮助的老师，从他们那里我学到了为人师表，默默奉献的精神，以及很多书本上学不到知识。感谢我的家人，在背后默默的支持我，是我完成学业的最强大的后盾。

还有一起生活和学习的金勇、鲁启华等实验室的其他同学，也给了我很大的帮助。在毕业之际向他们表示诚挚的感谢！

在读研究生期间取得的科研成果

[1]蔡朝.基于 S3C2410 的射频无线通讯系统,湖北大学学报自然科学版,2007.增

[2]主持孝感市科技项目“嵌入式控制器在智能病理仪器中的开发与移植研究”, 2007—2008.

[3]蔡朝,叶建勇主编.电子计数基础题解精粹(数字部分),中国地址大学出版社, 2005.1

作者: [蔡朝](#)
学位授予单位: [湖北大学](#)

相似文献(10条)

1. 期刊论文 [许芬, 曾建军, Xu Fen, Zeng Jianjun](#) [基于DM642的高性能CMOS图像采集系统的设计](#) - [北方工业大学学报](#) 2007, 19(3)

本文介绍了一个基于TI DM642的高性能CMOS数字图像采集处理系统的硬件和软件设计. 系统采用美光科技的高性能CMOS图像传感器MT9T001作为敏感元件, 利用TMS320DM642和CPLD器件构造图像采集和处理单元, 设计了USB和以太网数据接口. 系统成像性能优良, 采集频率可以达到100帧/秒(VGA分辨率). 系统可用于工业流程检测、智能交通、机器人巡逻等涉及运动图像处理的领域.

2. 期刊论文 [罗钧, 吴克松, 付丽, LUO Jun, WU Ke Song, FU Li](#) [基于高分辨率CMOS传感器图像采集系统的实现](#) - [电子技术应用](#) 2008, 34(11)

为了解决高像素CMOS图像采集传感器MT9T001和一些没有视频采集控制器的DSP接口难的问题, 采用了2KB容量的线缓冲FIFO+CPLD的方案, 结合TI公司的6000系列DSP芯片TMS320C6711的EMIF接口, 在DSP中设计了采集图像的中断程序, 通过DSP将采集的图像写到SD卡中, 最终证明了本采集方案的可行.

3. 期刊论文 [梁凤梅, 赵敬](#) [基于USB2.0的高速实时图像采集系统](#) - [电脑开发与应用](#) 2009, 22(6)

采用Cypress公司的EZ-USB FX2芯片作为USB通信及主控芯片, 设计了一个基于USB2.0接口的CMOS图像采集系统. 介绍了硬件电路的设计思路及组成, 详细介绍了USB数据接口的固件设计、驱动程序开发以及上层驱动程序的设计. 实验表明, 该系统软硬件能够很好地配合工作, 完成图像的实时采样, 并且能够实时显示图像, 画质优良.

4. 期刊论文 [项小峰, 王洋昔, 李培弘, 刘济林](#) [基于TMS320VC5402的低端图像采集系统设计与应用](#) - [电视技术](#) 2003, ""(11)

分析了在实时图像处理中采用TMS320VC5402作处理器的优缺点, 设计了一种基于C5402的低端图像处理平台, 适用于分辨率较低的灰度图像处理. 还介绍了C5402利用I/O空间和I2C总线的接口实现方式, 给出了系统实例及实验结果.

5. 期刊论文 [韩毅, HAN Yi](#) [基于CMOS传感器的以太网相机的研究与实现](#) - [计算机工程与设计](#) 2008, 29(13)

给出了一种由CMOS面阵彩色图像传感器、视频解码器芯片、单片机、CPLD、SRAM阵列和以太网芯片组成的以太网相机的实现方案. 阐述了M-JPEG压缩、推挽式图像序列暂存、UDP协议的实现、上位采集软件的编制等多个重要环节的实现方法. 在此基础上搭建了以太网相机系统, 进行了实际检测, 并具体应用于车载式图像采集系统中.

6. 学位论文 [杨会伟](#) [高帧频CMOS相机图像采集系统研究](#) 2009

高帧频图像采集是研究高速瞬态现象发生机理和运动规律的一种直观的测试技术和手段. 以某型号炸点坐标测量经纬仪为应用背景, 设计了一种高帧频图像采集系统.

论文介绍了高帧频CMOS图像采集系统的发展现状, 根据功能需求把整个采集系统设计划分为成像器设计、图像采集卡设计和软件设计三个部分.

成像器设计围绕Cypress公司CMOS图像传感器LUPA-300, 设计了该图像传感器的外围电路; 基于FPGA设计了CMOS传感器的时序驱动, 实现了CMOS图像传感器的数字图像输出; 根据CMOS传感器内部寄存器各个参数性能指标, 通过FPGA与上位机串行通信完成CMOS相机自动化参数设置, 实现了CMOS图像传感器工作状态的可变配置.

图像采集卡设计主要包括Camera Link接口、FPGA控制、PCI总线接口三个功能模块. Camera Link接口主要负责接收成像器传送的高速图像数据, 并将其传输到FPGA内部; FPGA控制模块主要负责完成对高速图像数据的缓存处理, 并为整个图像采集卡提供控制命令, 协调各个功能模块间的工作; PCI总线接口采用专用的接口芯片PCI9054, 通过DMA操作实现了图像数据的高速传输.

针对图像采集卡设计了驱动程序和上层应用程序, 实现了图像数据的采集与显示. 最后运用两点校正和一点校正相结合的方法实现图像的非均匀性校正.

该高帧频图像采集系统具有10位数据位宽, 在分辨率为640×480的情况下, 图像成像单元可产生最高250帧/秒数字图像数据流, 图像采集卡可达到200帧/s的采集速率. 实验表明该高速图像采集系统具有结构简单、性能稳定可靠、实时性强、灵活性好等优点.

7. 期刊论文 [杜江, 林慧贞, 胡东亮, DU Jiang, LIN Huizhen, HU Dongliang](#) [一种基于CMOS传感器的图像采集系统设计](#) - [现代电子技术](#) 2008, 31(6)

CMOS型传感器比传统的CCD型传感器具有诸多优势, 基于CMOS型传感器的图像系统已成为研究和开发的重点. 设计一种以FPGA为核心控制芯片、基于USB接口和ADCS1121 CMOS型单色图像传感器的图像采集系统. 介绍系统的构成及其硬件设计方法, 给出USB固件程序、设备驱动程序及用户应用程序的实现方法.

8. 期刊论文 [单宝堂, 沈庭芝, 赵鹏, 郝兵, Shan Baotang, Shen Tingzhi, Zhao Peng, Hao Bing](#) [CMOS图像数据转换方法的研究与实现](#) - [仪器仪表学报](#) 2006, 27(z3)

介绍了1/2英寸300万像素CMOS数字图像传感器MT9T001. 用FPGA完成了图像传感器的参数配置, 给出了测试结果. 提出了一种硬件上简明可行的Bayer数据到RGB数据的格式转换设计方案, 用FPGA实现, 最后应用到某多路图像采集系统中.

9. 学位论文 [李凡生](#) [光学实验中图像处理应用的研究](#) 2009

图像测量技术是以现代光学为基础, 融光电子学、计算机图像学、信息处理、计算机视觉等科学技术为一体的现代测量技术. 其应用领域正在不断扩大. 近年来, 将现代科学新技术与基础实验相结合, 运用图像处理技术对光学图像进行处理是光学实验主要发展趋势. 本文在对前人研究工作进行分析讨论的基础上, 针对普通物理的光学实验中传统观察和测量手段存在的问题, 利用CMOS视频摄像头和计算机结合传统光学实验系统, 构建廉价的数字图像采集系统, 解决了实验观察及演示难的问题, 进一步提高了测量精度. 在运用图像处理技术处理双棱镜干涉条纹图像时, 提出先用数学形态滤波方法消除干涉图像的噪声; 再用区域滤波方法结合灰度平均值的方法来确定图像二值化的阈值t, 对图像进行二值化, 较好地解决了实验中直边衍射带来的影响; 最后用数学形态学的膨胀、腐蚀、开闭运算和细化方法, 得到单像素的干涉条纹. 实现干涉条纹中心的精确定位和干涉条纹间距的精确测量.

本研究将现代图像处理技术合理地融入普通物理实验中的光学实验中, 在实验中运用现代观察和测量手段, 实现数字成像技术、图像处理技术和传统技术的结合. 这种做法既体现现代技术的作用, 使学生学到并掌握先进的实验技术, 又不影响实验基本技能的训练, 充分体现传统技术与现代科技相结合的优越性.

10. 会议论文 [单宝堂, 沈庭芝, 赵鹏, 郝兵](#) [CMOS图像数据转换方法的研究与实现](#) 2006

介绍了1/2英寸300万像素CMOS数字图像传感器MT9T001. 用FPGA完成了图像传感器的参数配置, 给出了测试结果. 提出了一种硬件上简明可行的Bayer数据到RGB数据的格式转换设计方案, 用FPGA实现, 最后应用到某多路图像采集系统中.

本文链接: http://d.g.wanfangdata.com.cn/Thesis_Y1379219.aspx

授权使用: 陕西理工学院(sxlgxy), 授权号: b4e554a7-bc87-43db-8beb-9df2010e71af

下载时间: 2010年9月15日