

# 面向视觉伺服的芯片图像 Zynq 预处理系统

景鹏阁, 龚时华, 周丹宇, 于思怡

(华中科技大学 机械科学与工程学院, 武汉 430070)

**摘要:** 晶圆芯片定位精度直接影响到芯片制造的效率和质量, 视觉伺服作为一种实时控制方法, 引入了视觉反馈, 不仅能够保证芯片定位精度还具有良好的鲁棒性; 同时图像处理的实时性对视觉伺服至关重要; 传统的使用 PC 机和网线的图像处理方式实时性和准确性较差, 为了提高这一过程中视觉反馈的精度和实时性, 提出并设计了一套面向视觉伺服的芯片图像 Zynq 预处理系统; 系统通过在 Zynq PL 端的片内设计和 PS 端的控制设计, 对采集到的芯片图像进行去噪、Sobel 处理, 并给芯片粗定位, 为芯片的伺服控制提供有效数据; 经实验表明, 该系统实时性强, 位置识别稳定, 达到设计要求。

**关键词:** 芯片制造; Zynq; FPGA; 视觉伺服; 图像处理

## Chip Image Preprocessing System with Zynq for Visual Servo

JING Pengge, GONG Shihua, ZHOU Danyu, YU Sixun

(College of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan 430070, China)

**Abstract:** The positioning accuracy of wafer chip directly affects the efficiency and quality of chip manufacturing. As a real-time control method, visual servo introduces visual feedback, which not only ensures the positioning accuracy of chip but also has a good robustness. At the same time, the real-time performance of image processing is crucial for visual servo. Traditional image processing methods for using PC and network cables have the characteristics of poor in real-time and accuracy. In order to improve the accuracy and real-time performance of visual feedback in this process, a Zynq preprocessing system for visual servo-oriented chip images is proposed and designed. Through the on-chip design of Zynq PL terminal and the control design of PS terminal, the system performs the denoising, Sobel processing on the acquired chip images, and rough positioning of the chip, which provides effective data for the servo control of the chip. Experimental results show that this system has a strong real-time performance and stable location recognition, meeting the design requirements.

**Keywords:** chip manufacturing; Zynq; FPGA; visual servo; image processing

## 0 引言

在高端电子制造中, 晶圆芯片的定位精度会直接影响到芯片制造的效率和产品质量。而运动平台的多源时变扰动会使芯片位置出现不确定性, 这种不确定性带来的芯片定位误差和对控制精度的影响已不能忽视。视觉伺服控制引入了视觉反馈, 不仅能够提升对于多元时变扰动的鲁棒性, 还能提升定位精度。视觉图像处理作为视觉伺服中的重要环节, 其实时性直接影响到视觉伺服的效果。同时, 高质量的图像既可以提高观测者的视觉感知质量, 例如更容易发现晶圆芯片角度、位置等细节信息, 同时也有助于通过对图像数据研究分析进行进一步处理, 进而为后续对于芯片位置的伺服控制提供更有效准确的数据。

随着计算机技术、半导体工艺的发展, 图像处理系统的

设计结构、设计工具也在发生着变化, 传统的图像处理和数据传输方式是使用上位机+网线<sup>[1]</sup>, 这种系统的优势在于开发平台简单, 适用范围广, 但是由于上位机主频较低、网线传输速率差, 较难实现高速处理, 会大大降低视觉反馈的实时性。从 20 世纪 90 年代初开始, 除了使用 PC (Personal Computer) 机, 在 Windows 平台上编制图像处理软件, 并使用 PCI (Personal Computer Interconnect) 总线作为数据传输方式的图像处理外, 出现了脱机图像处理系统, 用集成电路或专用芯片作为处理器, 取代 PC 机<sup>[2]</sup>。进入 21 世纪以来, 脱机图像处理系统得到了迅速发展, 越来越多的设计者将嵌入式芯片作为图像处理系统的处理核心, 其中芯片的选择多以 DSP, FPGA, CPLD 等为主<sup>[3]</sup>。

使用 PC 机进行图像处理时, 在视频流数据的运算中不

收稿日期: 2024-06-20; 修回日期: 2024-08-16。

基金项目: 国家自然科学基金(52175462)。

作者简介: 景鹏阁(2000-), 男, 硕士研究生。

引用格式: 景鹏阁, 龚时华, 周丹宇, 等. 面向视觉伺服的芯片图像 Zynq 预处理系统[J]. 计算机测量与控制, 2024, 32(11): 287-293.

占优势, 而当前一般使用的以太网传输速率在 1 Gbps 以下, 图像处理周期在毫秒 (ms) 级, 因此以 PC 机+网线的视觉反馈速率达不到高端电子制造的要求。在高端电子制造中, 对于晶圆芯片位置控制的实时性要求较高, 利用 FPGA (Field Programmable Gate Array) 内部集成的丰富的逻辑资源, 发挥其流水线计算结构、高速接口、接收与处理同时进行、并行计算能力等处理优势, 可大大提高图像处理的实时性, 王建庄等<sup>[4]</sup>在研究中将流水线优化并在 FPGA 上实现, 发现同步流水线在 FPGA 中占用资源最少, 可实现性能最好, 充分发挥了 FPGA 的运算优势, 将 FPGA 运算与总线传输结合在一起, 可以将图像处理周期减小到微秒 ( $\mu\text{s}$ ) 级; 其次, FPGA 具有可重构性, 可以在不改变系统整体框架的情况下灵活实现算法更新, 方便用户的维护。因此, FPGA 已逐渐成为现阶段实时图像处理系统计算核心的首选。而一套完整的图像处理系统需要有稳定高效的控制核心, 以 ARM (Advanced RISC Machine) 处理器作为控制核心能够对所需的功能模块进行控制, 同时对 FPGA 模块进行动态配置。选择 ARM+FPGA 为基本架构, 可以满足图像处理所需的控制和计算功能, Zynq 芯片在单个芯片上包含了 ARM 与 FPGA, 可以设计出高性能、低功耗的复杂嵌入式系统的, 也可以满足多核处理需求。

Zynq-7000 系列芯片由 Xilinx 公司推出, 是以 PS (processing system) + PL (programmable logic) 为基础结构的一种组合芯片, 其 PS 端本质上是一片 ARM, 以 PL 端的 FPGA 作为 ARM 的外设之一来进行高速计算。李庆春、李祺等<sup>[5]</sup>在研究中利用 Zynq 的结构, 对 FPGA 资源合理部署并利用双核 ARM 高性能处理器, 解决了传统图像采集系统远程图像传输延时长和数据易丢失的问题。一般的板上系统在设计时会使用几块不同的物理芯片来实现各部分的功能, 而 Zynq 将接口、存储、处理、逻辑等不同功能模块集成在一个芯片内, 单个芯片就可以实现不同功能, 形成片上系统 (SoC)。在 Zynq-7000 系列芯片的开发中, 将设计的电路逻辑映射到 PL 端来实现系统的逻辑部分, 而在 PS 端作为处理核心, 可以实现设计的软件部分。PL 端本质上是一片基于 Xilinx 7 系列的 FPGA, 允许用户修改配置内部的集成电路, 具有可编程输入/输出单元、嵌入式块 RAM、底层嵌入式功能单元等丰富资源, 用户可以根据需要对系统应用进行动态修改; PS 端还拥有时钟、串口、高速缓存等诸多外设资源, PL 端的 FPGA 也是其外设之一。

Zynq 是一个的可扩展处理平台, 它的出现为大量高端嵌入式的应用场景提供了所需的运算能力, 被广泛用于汽车、通信系统和工业控制等行业。本文选择以 Zynq-7000 作为图像预处理系统的开发核心。

## 1 芯片图像预处理算法

本文设计的目的是通过图像处理获得芯片中心点坐标, 作为晶圆芯片的粗定位信息, 为芯片制造后续的位置反馈和伺服控制提供有效信息。同时对于视野内的晶圆芯片图

像给出直观的处理结果, 以便观测者对设计中的参数进行调整。在预处理部分需要对采集到的晶圆芯片视频图像进行快速、大量、重复的处理, 而 Zynq PL 端 (FPGA) 的并行运算特点恰好适合进行有此类需求的运算, 因此本文根据晶圆芯片图像的处理需求, 设计了中值滤波-Sobel 边缘检测-中心点定位的处理过程, 以实现先去噪、再提取边缘、最后进行芯片粗定位的预处理需求。

### 1.1 中值滤波

在图像传输过程中, 各种噪声的干扰不可避免, 导致图像质量退化。而在晶圆芯片图像中存在的噪声会很大程度上影响芯片定位的准确度, 从而影响后续的处理。

根据图像噪声性质, 可以把噪声分为椒盐噪声和高斯噪声两类。在本文采集到的晶圆芯片图像中, 对处理结果有显著影响的是椒盐噪声, 而中值滤波能很好的去除晶圆芯片图像中椒盐噪声的影响, 还可以较好得保留晶圆芯片的边缘特征, 而且中值滤波运算简单, 速度快<sup>[6]</sup>, 因此本文选择中值滤波算法作为晶圆芯片图像预处理过程中的第一个环节。

中值滤波的原理是: 对于图像中待处理像素点, 以该像素点为中心选取实心邻域, 将这个领域作为滤波窗口, 对滤波窗口中所有点的灰度值排列并取中位数, 以此代替目标像素点的灰度值。为了保证中位数选取的便捷, 通常选取像素点个数为正奇数的实心邻域, 如  $3 \times 3$ 、 $5 \times 5$ 。 $3 \times 3$  的滤波窗口如图 1 所示。

$f(x-1, y-1)$	$f(x-1, y)$	$f(x-1, y+1)$
$f(x, y-1)$	$f(x, y)$	$f(x, y+1)$
$f(x+1, y-1)$	$f(x+1, y)$	$f(x+1, y+1)$

图 1  $3 \times 3$  滤波窗口

### 1.2 Sobel 边缘检测

边缘检测算法实质上也是一种图像滤波。在一幅图像中, 每个像素点都有各自的像素值, 而在像素值变化剧烈的点, 往往是不同物体、或物体与背景的交界点, 通过对像素值的变化观测可以将这些点检测出来, 则可以认为提取了物体的边缘点。在进行边缘检测时, 通常的方法是对观测待处理图像中每个像素点的灰度值, 根据像素点灰度值的变化, 确定某些灰度值变化明显的点作为物体或背景的边缘点, 将这些点标识出来得到检测后的边缘信息, 这个过程中通常要利用边缘的一阶或二阶导数的规律来确定边缘位置。常用的边缘检测算法有 Sobel 算子、Canny 算子、Laplace 算子等。

本文在晶圆芯片的图像处理中, 需要对于视野中晶圆芯片的边框做快速识别, 以通过对边框的处理来找到芯片的具体位置, 经过运算完成晶圆芯片的粗定位。为了保证

晶圆芯片边框提取的准确、高效, 本文选用了 Sobel 算子作为检测晶圆芯片边框的算法, 以快速实现芯片轮廓提取<sup>[12]</sup>。Sobel 算子的优点在于简单高效、适用性强, 是一种方向算子, 适用于图像中单个目标物体有灰度渐变或噪声较多的情况。而且这种算法边缘检测效率很高, 可以很好地保证芯片图像处理的实时性<sup>[13]</sup>, 很适合作为晶圆芯片图像的预处理环节。Sobel 算子包括 X、Y 轴方向的两组 3×3 的矩阵, 如图 2 所示。

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

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

图 2 Sobel 卷积因子 Y 方向 (左) 和 X 方向 (右)

将之与待处理图像做平面卷积, 可以分别得出 X 轴、Y 轴方向的亮度差近似值, 计算方式如式 (1):

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * A, G_y = \begin{bmatrix} 1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * A \quad (1)$$

其中: A 代表目标像素点的 3×3 实心邻域, 将中心点灰度值设为  $P_0$ , 则该实心邻域像素块可表示为图 3。

$P_1$	$P_2$	$P_3$
$P_4$	$P_0$	$P_5$
$P_6$	$P_7$	$P_8$

图 3 目标像素块

横向得到灰度值为:

$$G_x = -P_1 - 2P_4 - P_6 + P_3 + 2P_5 + P_8 \quad (2)$$

同理, 纵向灰度值为:

$$G_y = -P_1 - 2P_2 - P_3 + P_6 + 2P_7 + P_8 \quad (3)$$

最终得到处理后的目标像素点灰度值:

$$G = \sqrt{G_x^2 + G_y^2} \quad (4)$$

设定检测阈值  $T$ , 若  $G > T$ , 则将目标像素点灰度值置为 0, 在图像中呈现黑色; 否则置为 255, 呈现白色。可以看出, Sobel 算子本质上计算出目标像素点邻域内像素灰度值加权差, 这一值在边缘处达到极大值, 以此能够达到边缘检测的目的。随着数据量的减少, Sobel 算子在边缘检测中可以呈现出来的效果将会更佳, 本文已经在预处理第一环节使用中值滤波的方式对图像进行去噪处理, 因此 Sobel 边缘检测算法适用于本文芯片图像预处理的第二环节。

### 1.3 芯片框选

本文在进行芯片图像预处理时, 希望得到图像内芯片的坐标, 提供位置信息。根据晶圆芯片的特点, 单个芯片的形状多为矩形, 只要确定其边缘矩形的位置, 即可得到芯片中心点。在求取中心点时, 一种方法是可以通过对边缘检测后的晶圆芯片边框做直线拟合, 再经过数学运算得

到晶圆芯片几何中心; 或者对于经过边缘检测的芯片图像, 可以采用求取最小外接矩形的方法将芯片框选出来, 再通过对最小外接矩形的运算, 得到矩形中心坐标, 则可以作为芯片中心位置的粗定位坐标。两种方法中直线拟合得到的芯片中心点坐标更精确, 但最小外接矩形法运算速度快, 符合预处理要求, 且适合在 FPGA 中设计实现, 因此本文选择通过寻找最小外接矩形的方法来确定芯片中心点。

在寻找外接矩形时, 需要将图像中的形状作为像素点集, 根据其灰度值找到其最小外接矩形。这个过程的基本方法是: 首先寻找图形点集的最小  $x$  坐标和最大  $x$  坐标, 这两个坐标确定了矩形的左右边界; 同样找到点集的最小  $y$  坐标和最大  $y$  坐标, 这两个坐标确定了矩形的上下边界; 然后以这些坐标来运算得到矩形的宽度和高度。

## 2 图像预处理的 Zynq 实现方案

本文设计的芯片图像预处理系统主要由图像的采集、处理、输出 3 个模块组成, 整体系统运行框图如图 4 所示。图像采集设备选用 YiLianrui 公司 S6 型号的 HDMI 输出信号相机, 分辨率为 1 080 P, 系统核心部分使用嵌入式开发板 Zynq-7045, 核心板的核心芯片型号 XC7Z045, 设计时分 PS 端部分和 PL 端部分, 芯片 PS 端为双核 ARM Cortex-A9, PL 端为 Kintex-7 系列 FPGA, 图像输出部分使用 1 080 P 的带 HDMI 接口的显示器, 开发环境使用 Vivado 2018.3。

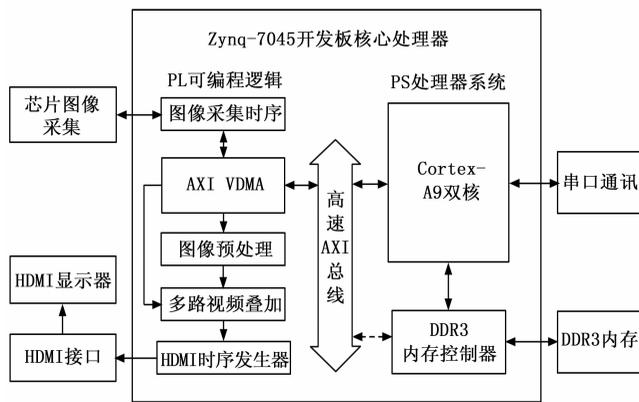


图 4 系统结构框图

根据视频的处理过程和数据流向, 可以将系统主要结构划分为三部分。

1) 图像输入部分: 通过带 HDMI 接口的摄像头采集得到视频信号, 在 Zynq 芯片的 PS 端通过扩展复用输入/输出接口 (EMIO) 引出引脚, 接收采集到的晶圆芯片视频图像数据, 并转换为视频流数据信号进行处理;

2) 图像处理部分: 将采集到的视频流数据信号转换为可以得到灰度值信息的 YcbCr 格式, 通过本文设计的处理过程, 得到中值滤波去噪、Sobel 边缘检测、芯片矩形框选后的处理结果;

3) 图像输出部分: 对于处理前后的晶圆芯片图像, 设计者应当可以直观地观察到处理结果, 图像输出部分将处理前后的视频流信号转换为视频信号, 由 Zynq 的 PS 的 EMIO 引出输出引脚, 接到带有 HDMI 接口的显示屏上。

### 2.1 图像输入设计

本文选用相机的水平/垂直分辨率为  $1\ 920 \times 1\ 080$  (pixels), 可以保证获取到  $640 \times 480$  (pixels) 的芯片图像, 相机最大帧率为 60 fps。安装连续变倍同轴微透镜及镜管来对采集过程进行调焦, 以保证获得清晰的芯片图像, 此外使用可控点光源和可控环形光源来对芯片光照环境进行增强, 在使用时可通过具体实验环境调整光源强弱, 避免低照度环境下采集图像不清晰。

在系统设计中, 采集部分要根据相机的输出信号和接口来设计。本设计采用 HDMI 接口的相机, 采集到 HDMI 引脚的四对差分信号, 将每一对差分信号根据 HDMI 接口的信号协议由串行信号转换并扩写为并行信号, 就可以将输入信号转换为 RGB888 的数据格式。在 Zynq 7000 系列器件中, PS 端与 PL 端之间主要采用 AXI (Advanced extensible Interface) 协议来实现高速通信, 这种总线是一种基于猝发式传输的片内传输协议, 内部有 5 个独立的数据通道, 分别是: (1) 读地址; (2) 读数据; (3) 写地址; (4) 写数据; (5) 写响应, 可以实现高性能、高带宽、低延迟的数据传输。

本文使用 Video In to AXI4-Stream IP 核, 将采集到的视频数据转换为视频流数据, 通过 AXI 总线的通讯协议来实现 RGB 视频信号的采集功能。

采集模块的 IP 核如图 5 所示, 输入信号包括时钟信号、图像信号、控制信号, 图像信号为 24 位 RGB888 信号, 由行同步信号 (Hsync) 和场同步信号 (Vsync) 描述像素和帧的控制信息, 由使能信号 (de) 和时钟信号 (pclk) 来确定采集的使能控制和时序信息; 输出信号 (video\_out) 为 AXI 视频流信号。

### 2.2 图像处理部分设计

#### 2.2.1 图像格式转换

在对晶圆芯片图像进行处理时, 尤其是对于要提取边缘的图片, 由于主要处理目的在于确定芯片边框和关键点

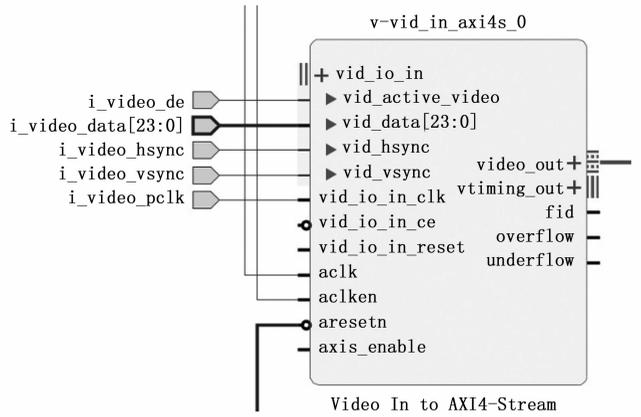


图 5 采集模块的实现

的位置信息, 所以根据算法特点, 要将晶圆芯片图像转换为灰度图的格式, 可以更直观地提取出芯片边框和关键点。YCbCr 的图像格式, 描述了一幅图像中的亮度值 (Y) 和色度值 (Cb、Cr), 其中 Cb 代表图像的蓝色色度分量, Cr 代表图像的红色色度分量。通过 YcbCr 的色彩空间描述, 可以在保留图像色彩信息的前提下, 得到图像各像素点的亮度值 Y, 这个分量可以作为灰度图中的灰度值进行处理。

在晶圆芯片图像采集部分的设计中, 相机输入信息的色彩空间为 RGB888 的数据格式, 因此要得到图像灰度值, 就要经过图像格式转换, 这个过程就是要将 RGB888 数据格式的图像信息转换为 YCbCr 的数据格式。对于每个像素点的色彩亮度信息, 可以根据色彩空间转换公式, 将图像 R、G、B 3 个分量根据权重进行乘加运算, 即可得出 Y、Cb、Cr 的 3 个分量值, 完成色彩空间描述方式的转换。若使用通用的色彩转换公式, Y、Cb、Cr 3 个分量的值在求取加权平均量时需要除以 256, 极有可能出现浮点数, 使用 Verilog HDL 语言进行浮点运算会使得计算使用 IP 过多, 占用更多运算资源, 造成运算资源的浪费和计算效率的降低。而在 Zynq 中移位运算占用资源少, 因此在色彩信息转化公式中可以不除以 256, 而是将乘加后的结果右移 8 位。最后得到转换公式可以表示为公式 (5):

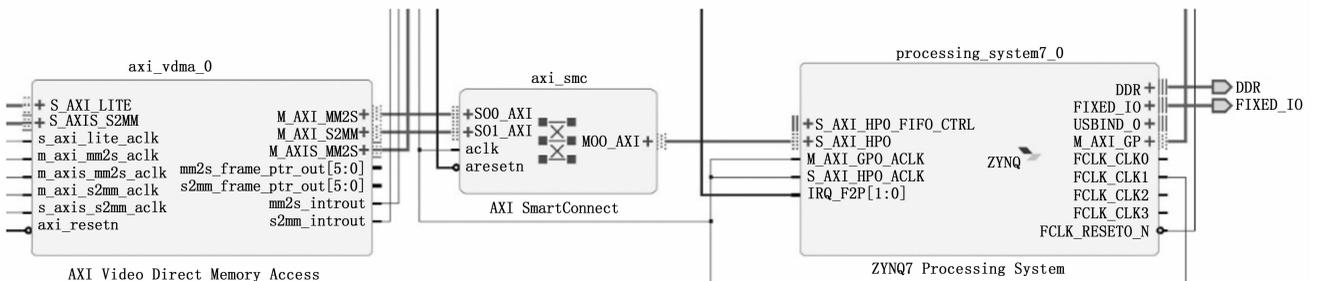


图 6 存储设计模块图

$$\begin{cases} Y = (77 * 5 + 150 * G + 29 * B) \gg 8 \\ Cb = (-43 * R - 85 * G + 128 * B + 32768) \gg 8 \\ Cr = (128 * R - 107 * G + 21 * B + 32768) \gg 8 \end{cases} \quad (5)$$

经过数据格式转换后的图像数据用于后续的图像处理, 而处理后的图像数据缓存起来, 在需要时由 ARM 端控制调用。

在色彩空间转换过程中要注意的是, 由于 Verilog HDL 语言阻塞赋值的特点, 色彩信息转换会消耗 3 个时钟周期, 应在设计中延时 3 个时钟保证数据信号的同步。

### 2.2.2 图像缓存设计

在 Zynq 设计过程中, 视频图像信息需要进行缓存, 本文工程共有两路视频信号需要缓存, 分别是来自相机的芯片原始图像 (RGB 格式) 和经过处理后的图像 (YcbCr 格式)。在缓存时通常会使用 AXI VDMA IP 核, 该 IP 核中可以通过 S2MM 接口将视频数据发送至 DDR 中, 通过 MM2S 接口从 DDR 中接收数据, 以此来实现帧缓存功能。视频流数据在 IP 核之间传输时用到 AXI 总线, 而 AXI4-Stream Video 协议提供了视频流数据的传输方式, 这种协议可以实现高速流的数据传输, 设计简单、性能高, 为 VDMA IP 核提供输入输出接口。存储设计模块如图 6 所示, VDMA 输入信号中的 S\_AXIS\_S2MM 接口接受上文中提到的输入模块处理过的视频流信号, 经过智能连接模块, 以 AXI 总线与 PS 端配置的 DDR 完成存储。

### 2.2.3 图像处理实现

在本文的 1.1 与 1.2 中已经提出了芯片边缘检测的实现方案, 对于数据转换后的 YCbCr 格式图像, 先采用中值滤波对芯片图像进行去噪, 再使用 Sobel (边缘检测) 算法进行处理, 则可以通过设定阈值, 得到芯片的边框和定位点的图像信息, 以供位姿调整。

本文在设计实现图像处理算法时, 中值滤波和 Sobel 边缘检测通过封装 IP 核的方式连接于系统模块中, 首先在 Vivado HLS 中实现算法的编写<sup>[11]</sup>, 再封装并导出 IP 核到 Vivado 中, 将该 IP 核应用于图像处理部分。如图 7 是通过 Vivado HLS 封装的 IP 核在 Block Design 中的模块连接。

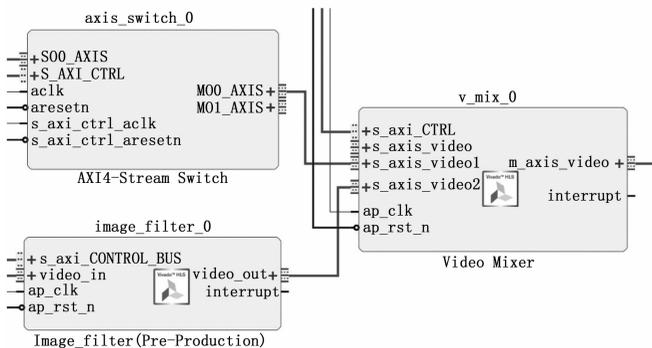


图 7 图像处理 IP 核的模块实现

在晶圆芯片的分选过程中, 芯片需要实时调整位姿,

因此要求视觉系统能够找出晶圆芯片的位置, 以进行伺服控制。本文通过求取视野中晶圆芯片的最小外接矩形来给出晶圆芯片中心点坐标, 实现晶圆芯片的粗定位。在 Vivado 中启动 sdk, sdk 中可以导入 OpenCV 库, 对经过 Sobel 处理的图像查找轮廓, 在轮廓处使用函数求取其最小外接矩形, 并画出矩形框, 再将矩形框画入芯片原始采集图像, 就可以直观地看到晶圆芯片被框选出来。通过对矩形框顶点和长宽的计算, 则可以得到芯片中心点的粗定位坐标。

### 2.3 图像输出设计

采集到的晶圆芯片原图像, 和经过处理后的图像要直观展示给设计者, 以便设计者根据处理效果调整处理参数或算法。本文设计使用 HDMI 信号作为输出信号, 与显示器连接。

在经过图像处理, 使用 AXI4-Stream to Video Out IP 核输出视频信息。VDMA IP 核交互的数据为视频流信号 AXI4-Stream, 经过该输出 IP, 可以将数据格式转换为并行的视频信号, 此并行视频信号可以作为输出信号使用。视频的输出的像素时钟使用 Clocking Wizard IP 核来产生, 时钟与分辨率和帧率对应 (1080P60), 设定为 148.5 MHz。此外 AXI4-Stream to Video Out IP 核与 VTC IP 核形成反馈控制关系, 以此来控制输入视频流与 VTC 间的相位差, VTC IP 核用于产生对应 1080P60 的时序。本文使用 Video Mixer IP 对多路视频进行叠加输出, 保证使用者既可以观察到原始图像信息, 也可以观察到处理后信息。

## 3 实验结果分析

系统设计完成后将引脚与外设完成约束, 连接外设后给评估板上电。随后将 Vivado 软件生成的比特流文件下载到 Zynq-7045 开发板, 再在 Vivado 中 Launch sdk, 在检测到硬件串连接后即可在 sdk 中发送命令进行测试。实验平台硬件结构实物图如图 8 所示。

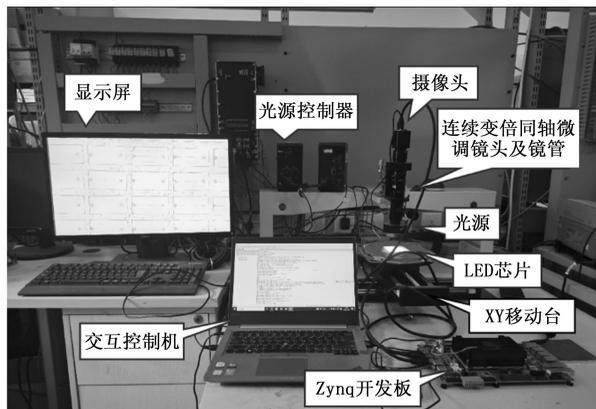


图 8 Zynq 图像处理实验平台

首先对芯片图像的原图进行采集, 打开光源并调试光源亮度与焦距到合适值, 则可以如图 9 可以看到芯片的边缘轮廓清晰, 分辨率高, 满足预定的采集要求。

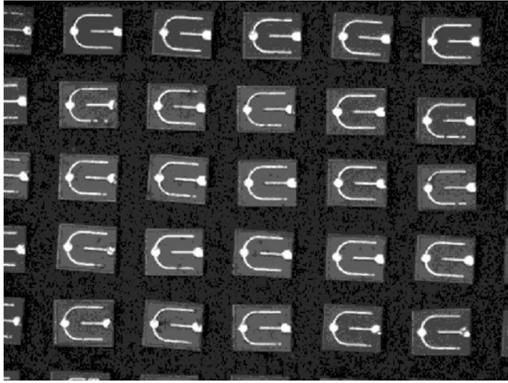


图 9 芯片图像实时采集

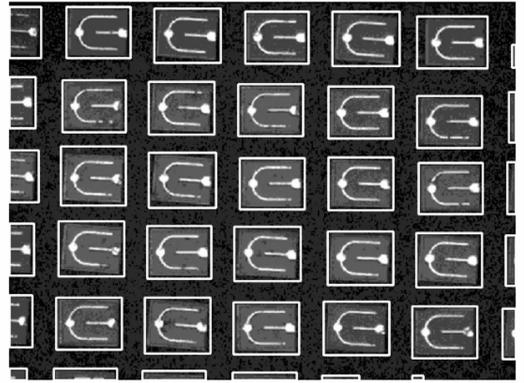


图 11 芯片位置采集与框选

在 sdk 的硬件控制命令行中输入选择边缘检测通道的控制指令, 即可在输出端看到 Sobel 边缘检测后的视频流数据结果, 如图 10 所示, 系统可以清晰地识别出芯片的边缘轮廓和左右两个定位点, 在芯片的定位和控制中, 可以以此为依据提供有效数据。

经过上述处理已经得到了芯片边框, 则可以根据该边框对芯片进行定位。对芯片进行框选, 则可以得到图 11。

图像边缘处的芯片不完整, 应在识别中被排除, 通过对矩形框尺寸的限制, 可以去除不完整的芯片框, 得到图 12。

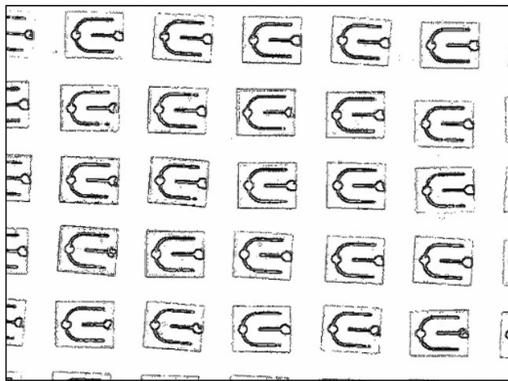


图 10 芯片图像实时处理结果

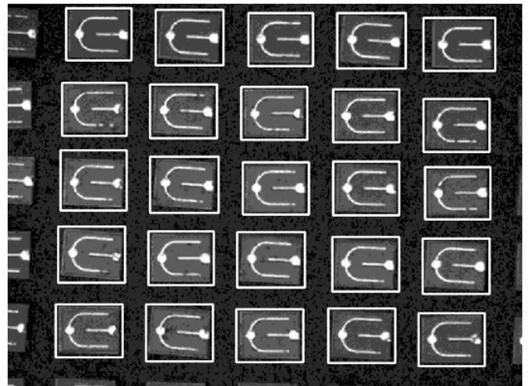


图 12 框选出完整芯片

如表 1 所示, 芯片序号以左上角为 1, 逐行递增, 其中坐标单位为 (pixels)。

本论文对芯片在不同移动速度和不同角度下采集到的芯片图像进行搜索, 对芯片的识别率做了计算, 结果如表 2 所示, 可以看出系统识别晶圆芯片的准确率较高, 对于视野中完整的晶圆芯片均能完成识别, 识别准确率达到目标效果。

在 Vivado 中运行执行 (Implementation), 在程序执行完成后可生成资源利用报告 (Report Utilization), 如图 13 所示, 通过对 Zynq PL 端的占用资源分析, 可以发现本系统的资源消耗较低, 且系统在检测速度、精准度方面也达到了目标效果, 说明系统达到了预期要求。

通过对矩形框尺寸和坐标的运算, 可以得到被框出芯片的中心点坐标, 以图 12 中的芯片为例, 统计出芯片坐标

表 1 芯片坐标统计

序号	坐标	序号	坐标	序号	坐标	序号	坐标
1	(112.5,38.5)	8	(330.0,137.0)	15	(557.5,236.5)	22	(212.5,410.5)
2	(225.0,42.5)	9	(444.0,140.0)	16	(103.5,313.0)	23	(324.5,414.0)
3	(338.0,45.0)	10	(556.0,152.0)	17	(214.5,317.5)	24	(438.5,414.0)
4	(450.5,46.5)	11	(106.0,220.5)	18	(326.0,319.5)	25	(550.0,419.0)
5	(559.5,51.0)	12	(218.5,226.0)	19	(443.5,325.0)		
6	(107.0,132.0)	13	(332.0,230.5)	20	(555.5,327.5)		
7	(217.5,134.5)	14	(444.0,230.5)	21	(100.0,408.0)		

表 2 识别统计表

实验号	速度/(mm/s)	角度/(°)	识别数	识别率/%
1	2	0	42	100.0
2	2	60	37	100.0
3	2	120	36	97.30
4	3	90	35	100.0
5	4	90	35	100.0
6	5	90	35	100.0
7	10	90	34	97.14

Summary

Resource	Utilization	Available	Utilization/%
LUT	11794	218600	5.40
LUTRAM	782	70400	1.11
FF	16287	437200	3.73
BRAM	11.50	545	2.11
DSP	14	900	1.56
IO	55	362	15.19
MMCM	1	8	12.50

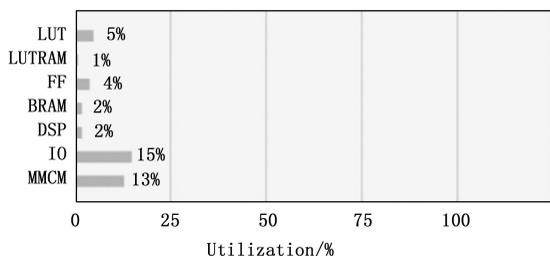


图 13 程序资源占用率分析

4 结束语

本文以 Zynq 为平台, 使用中值滤波与 Sobel (边缘检测) 算法来进行芯片图像的预处理, 并在 sdk 控制端框选出芯片位置并给出芯片坐标, 旨在为芯片定位与控制提供清晰图像和有效位置信息。采用软硬件协同的方式, 以 PS 端为控制核心, 完成部分控制操作及软件设计, 将 PL 端作为外设来布置硬件逻辑, 提供更快运算速度, 与传统的上位机控制和网线传输方式相比, 充分发挥了 Zynq 平台的优势, 达到了预期效果。本系统可以在原设计基础上搭配不同型号、不同采集通道的相机, 或增加多目数的采集通道, 也可以在处理部分搭配不同的图像处理算法, 设计更丰富的采集和处理系统, 可拓展性强, 具有很大的开发潜力。

参考文献:

[1] 黄书先. 基于 MATLAB 平台的可视化图像处理系统设计方法 [J]. 长江大学学报 (自科版), 2005 (4): 158-160.  
 [2] 林振钰, 张志杰, 刘佳琪. 基于 ZYNQ 的高清图像显示及检测系统设计 [J]. 计算机测量与控制, 2021, 29 (2): 30-34.

[3] QIANG W, QING G, JIA K. Hardware design of image information processor based on ADSP-TS201 DSPs [C] //2009 IEEE International Workshop on Imaging Systems and Techniques. IEEE, 2009: 155-158.  
 [4] 王建庄. 基于 FPGA 的高速图像处理算法研究及系统实现 [D]. 武汉: 华中科技大学, 2011.  
 [5] 李庆春, 李 祺, 刘彦君, 等. 基于 ZYNQ 的远程图像采集系统设计 [J]. 计算机测量与控制, 2022, 30 (10): 174-180.  
 [6] 巩玉奇, 陶晋宜, 杨 刚. 基于改进中值滤波的手机玻璃瑕疵图像增强方法 [J]. 电子技术应用, 2022, 48 (7): 91-95.  
 [7] O'MALLEY, RONAN, EDWARD JONES, MARTIN GLAVIN. Detection of pedestrians in far-infrared automotive night vision using region-growing and clothing distortion compensation. Infrared Physics & Technology 53.6 (2010): 439-449.  
 [8] 彭宽宽. 面向 LED 芯片分选的视觉定位技术研究 [D]. 武汉: 华中科技大学, 2020.  
 [9] 陈冠成, 吴贵燕. 基于 Zynq 芯片的实时视频处理系统设计 [J]. 现代电子技术, 2019, 42 (6): 76-78.  
 [10] 赵高长, 张 磊, 武风波. 改进的中值滤波算法在图像去噪中的应用 [J]. 应用光学, 2011, 32 (4): 678-682.  
 [11] TEWARY, SUMAN, et al. Hybrid multi-resolution detection of moving targets in infrared imagery. Infrared Physics & Technology 67 (2014): 173-183.  
 [12] 冯天桥, 毛征宇, 彭向前. 基于 FPGA 的运动目标实时检测系统设计 [J]. 电子技术应用, 2023, 49 (6): 99-103.  
 [13] 曾 俊. 图像边缘检测技术及其应用研究 [D]. 武汉: 华中科技大学, 2011.  
 [14] 袁春兰, 熊宗龙, 周雪花, 等. 基于 Sobel 算子的图像边缘检测研究 [J]. 激光与红外, 2009, 39 (1): 85-87.  
 [15] 尹雪松. 基于 Sobel 算子的飞机舱门边缘检测 [J]. 自动化应用, 2023, 64 (24): 210-212.  
 [16] SHIDDIQI A M, PRATAMA H, CIPTANINGTYAS H T. A video streaming application using mobile media application programming Interface [J]. TELKOMNIKA (Telecommunication Computing Electronics and Control), 2010, 8 (3): 293-300.  
 [17] 张文娟, 康家银. 一种用于图像降噪的自适应均值滤波算法 [J]. 小型微型计算机系统, 2011, 32 (12): 2495-2498.  
 [18] 路小波, 张光华. 基于二值图像的车牌精确定位方法 [J]. 东南大学学报 (自然科学版), 2005 (6): 972-974.  
 [19] 宁效龙, 何子力, 张昕昱, 等. 基于 Zynq 与 Qt 的视频采集与图像边缘检测系统 [J]. 信息技术与网络安全, 2019, 38 (2): 71-74.  
 [20] 逮沛龙, 符意德. 图像预处理算法在 Zynq 芯片上的设计与实现 [J]. 计算机与数字工程, 2015, 43 (6): 1118-1120.  
 [21] 谭检成, 吴定祥, 李明鑫. 基于 Zynq 的图像边缘检测系统设计 [J]. 电子测试, 2018 (9): 17-20.  
 [22] 严传高, 张乘浩, 刘马良, 等. HDMI 高速显示数据接口技术 [J]. 微纳电子与智能制造, 2020, 2 (2): 105-111.