****

**本 科 毕 业 论 文（设计）**

|  |  |
| --- | --- |
| 课题名称 | 低复杂度二维线性插值算法的FPGA实现 |
| 学 院 | 物理与电子工程学院 |
| 专 业 | 物联网工程 |
| 班级名称 | 物联网212 |
| 学生姓名 | 王力 |
| 学 号 | 32119500117 |
| 指导教师 | 浣沙 |
| 完成日期 |  |

教 务 处 制

低复杂度二维线性插值算法的FPGA实现

物联网工程212班 王力

指导教师:浣沙

摘 要如今，数字图像的使用不断增加，加工过程在商业和工业应用中扮演着重要角色。双三次插值因其高质量被广泛应用于实时图像处理系统，但真正实现双三次插值需要大量的硬件资源，这限制了其在嵌入式领域的应用。因此，研究近似双三次插值的算法，减少硬件资源消耗具有重要的实际意义。

本文在近似双三次插值算法的深入研究基础上，采用ZYNQ硬件平台和Xilinx Vivado软件开发平台，设计并实现了低复杂度二维线性插值的图像处理系统。

本文所开发的系统采用摄像头采集视频图像信息，将采集到的视频图像通过ZYNQ平台进行实时处理后，通过HDMI接口输出到显示屏上，以达到低延迟的实时图像缩放效果。

关键词 图像实时处理；双三次插值；ZYNQ

**ABSTRACT** Intravenous injections, The use of digital images has been steadily increasing, playing a vital role in commercial and industrial applications. Bicubic interpolation is widely utilized in real-time image processing systems due to its high quality. However, implementing bicubic interpolation requires substantial hardware resources, which limits its application in embedded systems. Therefore, researching approximate bicubic interpolation algorithms to reduce hardware resource consumption holds significant practical value.

Based on an in-depth study of approximate bicubic interpolation algorithms, this paper designs and implements a low-complexity 2D linear interpolation image processing system using the ZYNQ hardware platform and Xilinx Vivado software development platform.

The developed system captures video image data through a camera, processes the captured video images in real-time using the ZYNQ platform, and outputs the processed images to a display screen via an HDMI interface, achieving low-latency real-time image scaling.

**KEY WORDS** Real-time image processing; Bicubic interpolation; ZYNQ

目录

[1. 前 言 6](#_Toc28848)

[1.1 研究的背景及意义 6](#_Toc23465)

[1.2 国内外的研究情况 6](#_Toc26562)

[1.3 论文主要研究内容 8](#_Toc13079)

[2. 基于ZYNQ的非接触式静脉显像系统的整体架构 9](#_Toc15918)

[2.1 近红外光对静脉图像的增强作用 9](#_Toc25049)

[2.2 手背静脉图像处理算法 9](#_Toc17270)

[2.2.1直方图均衡算法 10](#_Toc9839)

[2.2.2高频加强滤波 11](#_Toc23810)

[2.3 开发平台选择 12](#_Toc30211)

[3. 系统硬件选型 15](#_Toc9030)

[3.1 ZYNQ开发平台 15](#_Toc26459)

[3.2 图像信息采集模块 16](#_Toc22893)

[3.3 近红外增强模块 16](#_Toc8820)

[3.4 图像显示模块 18](#_Toc6934)

[4. 系统整体实现 19](#_Toc25020)

[4.1开发平台介绍 20](#_Toc14617)

[4.1.1 Vivado开发平台介绍 20](#_Toc14913)

[4.1.2 HLS高层次综合工具 21](#_Toc17789)

[4.2 IP核介绍 22](#_Toc19035)

[4.2.1视频流-AXIS转换 IP核 22](#_Toc4093)

[4.2.2 VDMA IP核 23](#_Toc10901)

[4.2.3 ZYNQ Processing System 24](#_Toc8139)

[4.2.4 HLS视频处理IP核 25](#_Toc6875)

[4.2.5 Video Timing Controller IP核 29](#_Toc23680)

[4.2.6 RGB to DVI Video Encoder IP核 29](#_Toc30277)

[4.2.7 AXI Interconnect IP核 30](#_Toc26443)

[4.2.8 GPIO IP核 31](#_Toc2540)

[4.2.9 Processer System Reset IP核 32](#_Toc28385)

[5. 实验结果 33](#_Toc4010)

[6.结语 36](#_Toc23493)

[参考文献 37](#_Toc21441)

[致谢 39](#_Toc24534)

# 前 言

## 研究的背景及意义

随着计算机视觉、人工智能以及数字图像处理技术的飞速发展，图像处理的应用领域不断扩大，从传统的摄影、电影制作，到医学影像分析、卫星遥感、视频监控等行业，都对高质量的图像提出了更高的需求。在这些领域中，图像的分辨率往往是决定其实用性的重要因素。然而，受限于图像采集设备的硬件性能以及存储和传输带宽的限制，实际获取的图像分辨率往往难以满足特定应用场景的需求。图像放大（Image Upscaling）作为一种有效的图像增强技术，可以在不重新采集图像的情况下，通过算法提高图像的分辨率，从而满足更高精度的应用需求。

传统的图像放大方法，如双线性插值（Bilinear Interpolation）、双三次插值（Bicubic Interpolation）等，虽然计算简单，但容易导致图像边缘模糊、细节丢失等问题。近年来，深度学习技术的迅猛发展为图像放大提供了全新的解决方案，基于卷积神经网络（CNN）、生成对抗网络（GAN）等方法的超分辨率重建（Super-Resolution）技术在图像质量提升方面表现出色，大幅提高了放大图像的视觉质量。因此，研究低复杂度二维线性插值算法具有现实意义如下[3-4]：

（1）图像放大技术的研究推动了计算机视觉和数字图像处理领域的发展。通过深入研究图像放大的原理和算法，可以加深对图像数据特性的理解，并为其他图像处理任务（如去噪、去模糊、图像修复等）提供理论支持。

（2）随着数字化转型的深入，高质量图像需求的激增使得图像放大技术具有广泛的社会意义。它不仅可以提高资源的利用率，还能降低图像采集设备的成本，推动科技成果的普及化。

## 国内外的研究情况

#### 1.2.1 图像插值算法相关方法

图像插值算法是一种通过现有像素值计算图像任意位置（包括亚像素位置）像素值的技术，用于图像放大、缩小、几何变换和分辨率增强等场景。常见方法包括以下几类：最近邻插值，通过选择与目标位置最接近的像素值实现，计算简单但易产生锯齿效应；双线性插值，通过线性加权目标点邻近的四个像素值计算，提升了平滑性但仍存在模糊问题；双三次插值，利用三次多项式核函数加权计算目标点附近的16个像素值，具有较高插值精度，是平衡质量和复杂度的主流方法；Lanczos插值，基于窗函数的重采样技术，通过调整窗宽提高图像质量，但硬件实现复杂度较高；卷积神经网络（CNN）插值，利用深度学习模型实现高质量图像插值，适用于超分辨率应用，但因计算复杂度高而难以实时实现。此外，为平衡硬件资源和插值质量，许多研究提出了低复杂度近似算法，如分段线性近似双三次插值。这些方法根据应用需求和硬件平台的限制，各自展现出不同的优劣势。

在国内，关于低复杂度插值算法的研究主要集中于具体应用场景的优化与硬件化实现。部分学者将 Boukhtache 等人的分段线性插值算法与其他图像处理方法相结合，提出了多种创新性的算法优化策略。例如，在实时视频监控和遥感图像处理领域，研究人员利用图像的先验信息，对插值算法进行了简化和优化，显著提升了算法的执行效率。此外，国内在嵌入式硬件系统的工程化应用方面也取得了显著成果。一些学者将低复杂度插值算法成功应用于 ZYNQ 平台，并通过流水线优化和资源调度进一步提高了系统性能。

综上所述，国内外关于低复杂度插值算法的研究在理论优化与硬件实现方面取得了显著进展。其中，Boukhtache 等人提出的分段线性近似双三次插值算法，为低复杂度插值算法的研究奠定了重要基础。国外学者如 J. Smith 和 L. Wang、A. Kumar，以及 M. Zhang 和 Y. Li 等，通过不同的优化策略，成功地将该算法应用于各种嵌入式硬件平台，进一步降低了硬件资源消耗，同时提升了实时处理能力。国内学者则聚焦于具体应用场景的需求，结合实际工程应用优化插值算法的性能。

在未来的研究中，可以进一步探索分段线性近似算法在多领域、多平台中的适用性，结合深度学习方法或其他智能优化技术，开发出兼具高质量、低复杂度和高效率的插值算法硬件实现方案。

#### 1.2.2 图像插值算法的研究现状

传统的双三次插值算法虽然能提供较高的图像质量，但其高计算复杂度使得硬件实现成本居高不下，尤其是在 FPGA 等资源有限的嵌入式平台上。在硬件资源有限的嵌入式平台上，如何实现高效的双三次插值算法成为学术界和工程界共同关注的难题。但在本系统中，低复杂度二维线性插值算法的目的是在多数实际应用中，输出质量足以满足实现需求，并且显著降低硬件实现成本。图像插值算法相关的学术研究已经有十余年的历史，相关方法不断推陈出新。本文简要介绍几种如下：

Boukhtache 等人 于2021年提出了一种基于分段线性近似的双三次插值方法[1]，其核心思想在于以分段线性函数替代传统双三次插值核函数，从而大幅降低硬件实现中的乘法器和加法器使用量。具体来说，Boukhtache将传统双三次插值的三次多项式核函数替换为分段线性函数，通过线性插值的方式近似三次多项式函数，显著降低了计算复杂度，并且设计了一种模块化的硬件架构，将近似插值算法的各部分功能模块化，并利用流水线技术提升了数据吞吐量。

J. Smith 和 L. Wang 于2022年提出了一种高效的基于 FPGA 的图像插值硬件架构[2]。通过减少乘法器数量和优化数据路径设计，显著降低了硬件资源消耗。他们还结合流水线设计技术，提高了系统的实时处理能力。实验表明，该方法不仅能够实现近似双三次插值算法的性能，还在硬件资源利用率方面优于传统双三次插值实现。J. Smith 和 L. Wang 的研究进一步扩展了低复杂度插值算法在嵌入式系统中的实际应用，为高效图像处理硬件设计提供了新的思路。

A. Kumar 等人 受分段线性近似算法启发于2022年提出了一种自适应分段线性函数，进一步优化了硬件资源的使用效率[3]。与传统双三次插值算法相比，该方法在保证图像质量的前提下，将乘法器的使用减少了 60% 以上，同时通过模块化设计显著提升了处理效率。该研究的硬件实验结果表明，该方法不仅适用于低资源嵌入式平台，同时在实时性和稳定性方面也表现出色。A. Kumar 等人的研究进一步验证了基于近似插值核的优化方法的可行性和高效性。

M. Zhang 和 Y. Li 的研究同样基于分段线性插值核思想，提出了一种针对高分辨率显示器的低复杂度插值技术[4]。他们的方法同样基于 Boukhtache 等人的分段线性插值核思想，但通过重新设计插值核函数，进一步降低了计算复杂度。在硬件实现中，他们引入了基于 FPGA 的多模块并行优化技术，提高了数据吞吐量，确保了实时性要求。此外，该方法的图像质量与传统双三次插值相当，而硬件资源使用效率则显著提高。M. Zhang 和 Y. Li 的研究为高分辨率图像处理的硬件实现提供了新的理论依据。

## 论文主要研究内容

本文在近似双三次插值算法的深入研究基础上，采用ZYNQ硬件平台和Xilinx Vivado软件开发平台，设计并实现了低复杂度二维线性插值的图像处理系统。该系统具有低复杂度、实时性好和高效性等特点，能够满足实时图像处理的需求。本文的主要研究内容包括：

提出一种低复杂度的二维线性插值算法，并通过Matlab仿真对其性能进行验证，分析与双三次插值算法的对比，重点考察计算复杂度和图像质量的差异。仿真过程中采用多个具有代表性的图像数据集进行测试，通过均方误差（MSE）、峰值信噪比（PSNR）等指标评估算法的性能。

利用Vivado软件开发工具实现算法硬件化，包括数据流设计、模块分解、流水线优化等步骤，以满足实时处理的要求。在硬件实现过程中，特别针对模块间的数据依赖性和资源冲突问题进行了优化设计，通过时序分析和资源平衡，提升了系统的运行效率。

将处理后的图像数据通过HDMI接口输出到显示屏，实现低延迟的实时图像缩放效果。针对HDMI接口传输过程中可能的延迟问题，采用了数据缓冲和同步控制机制，确保输出图像的平滑性和一致性。同时，通过硬件测试验证了系统在不同分辨率下的稳定性和可靠性。

本文章节安排如下：

第一章为前言部分，主要介绍了非接触式静脉成像增强的意义，市场现状和静脉图像增强相关算法的研究现状，并简要介绍了本文研究的主要内容。

第二章为低复杂度二维线性插值算法的设计与分析，详细阐述了插值算法的设计过程、复杂度优化策略以及与双三次插值的性能比较。

第三章为基于ZYNQ的系统架构设计，主要介绍了硬件平台的选型、DDR4存储器的使用以及HDMI接口的图像输出流程。

第四章为系统整体实现，主要介绍了主要介绍本系统的实现流程和各模块作用，开发所使用到的软件平台Vivado和HLS开发工具，以及使用到的相关IP核。

第五章为实验结果，主要内容为搭建的实验平台的展示以及手背静脉增强的效果的分析，还提到了可见光对于实验的影响。

第七章为结语，总结了本系统所完成的工作，并对系统研发中的不足和未来工作的展望展开讨论。

# 低复杂度二维线性插值算法的设计与分析

本文提出了一种基于“一次线性插值与一次三次插值组合”（One Cubic and Four Linear Interpolation）的低复杂度二维线性插值算法。本章节中将对低复杂度二维线性插值算法进行介绍。同时还将讨论双三次插值算法的设计，并展示这两种算法在matble上的效果。

## 2.1 低复杂度二维线性插值算法

本文提出了一种基于“一次线性插值与一次三次插值组合”（One Cubic and Four Linear Interpolation）的低复杂度二维线性插值。该算法的核心思想是在图像插值过程中，首先沿一个方向（例如水平方向）使用线性插值法计算出中间值，然后再在另一个方向（例如垂直方向）应用三次插值法对这些中间值进行插值，最终得到插值结果。图像1-1(a)代表四次线性插值，线性插值只需要两个像素来插值，通过线性连接得到中间位置。图像1-1(b)代表一次三次插值，应用三次插值法对这些中间值进行插值，最终得到插值结果。如下图1-1是低复杂度二维线性插值组合：

## 

图 1-1

相较于传统的双三次插值算法需要访问16个像素，本算法仅需访问8个像素点，内存需求大大减少，这对于嵌入式平台尤其重要。低复杂度二维线性插值算法需要八个邻居像素，7个乘法器，21个加法器。如图2-1是低复杂度二维线性插值的框图。

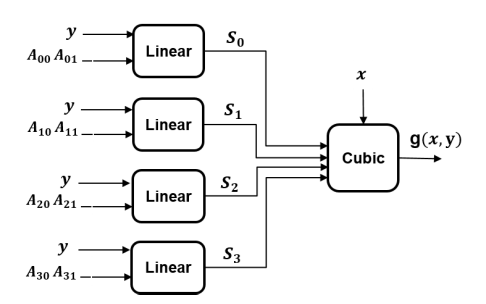


图 2-1

本文对该算法的各向同性进行了研究。该体系结构的全局插值误差

二维插值可以重写为

由于βlinear(y1) + βlinear(y2) = 1。Aij参数当其系数为正值时，将其最大化。

## 2.2 手背静脉图像处理算法

但仅仅在硬件层面，通过近红外光源和红外滤光片对手背静脉图像进行增强，由于不同人的个体差异，有时效果并不明显。因此，在通过图像通路获取了手背的近红外光图像后，还需要对图像进行一些处理，处理的目的主要是进一步增强画面中静脉纹路的锐度及与手背的对比度，以达到静脉显像增强的效果。以下介绍两种常用的手背静脉图像增强算法。

### 2.2.1直方图均衡算法

直方图均衡，英文全称Histogram Equalization, 是一种常见的图像增强算法。直方图的图表包含的信息是一幅图像灰度级的分布情况。图像的直方图反映了连续图像中的不同灰度的像素面积，或离散图像中不同灰度的像素个数在整幅图像中所占的比例。直方图均衡，就是把一幅灰度概率分布已知的图像，通过变换使之成为灰度级分布较为均匀的图像，从而增加图像的明暗对比, 实现图像的增强[15]。在近红外光照射下的静脉图像中，容易出现图片过暗或过亮的情况。此时采用直方图均衡算法，可以有效增加静脉纹理与手背间的对比度，使静脉纹理在图像上更容易被辨认。

图4-1是直方图均衡的原理示意图,即将原本灰度级分布不均匀的图像直方图（4-1左图）经过直方图均衡化操作，变成灰度级均匀分布的直方图（4-1右图）。下图中横轴G代表的是图像中单个像素的灰度值大小，纵轴P/(%)则代表了整幅图像中对应灰度值像素出现的概率。

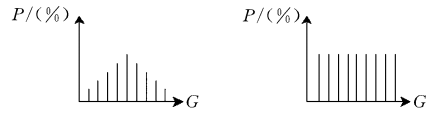


图4-1 直方图均衡化

可以发现，经过直方图均衡化后，图像各灰度级出现的概率均等(但实际操作中并不会完全相等)，图像的动态范围得到提升，图像亮度较暗的部分得到增亮,图像较亮的部分的高光则被压制，图像的对比度得到提高。

直方图均衡的原理如下：设r表示像素灰度级大小。对灰度级归一化处理，即 ,其中r = 0表示像素为纯黑色, r = 1表示像素为纯白色。对于给定的一幅图像,假设像素的灰度级随机，用概率密度函数来表示图像灰度级出现的概率。为了方便常用的离散数字图像处理,引入离散形式，,用代表离散像素的灰度级,用代表,并且下式成立:  。

其中, 。图像中出现这种灰度级的像素数量为,图像中的像素总数为n。则图像直方图均衡的数学表达式为:

 （4-1）

式中, k为灰度级数。

综上所述，直方图均衡可使图像灰度分布更加平均, 增加图像的对比度，用在手背静脉图像增强上比较合适。

### 2.2.2高频加强滤波

手背近红外光静脉图像是某一特定波段的红外光图像。相比于可见光图像丰富的颜色, 近红外光图像的灰度分布相对统一, 尤其是在红外光源强度较弱时, 静脉血管的纹理容易变得难以辨认。根据数字图像知识, 图像的灰度级信息保存在傅里叶变换的低频成分中，图像傅里叶变换的高频部分主要是图像的边缘和细节信息。即图像的傅里叶变换中的的高频部分反映的是图像中灰度级差异较大的边缘部分和图像的一些细节部分, 所以通过高通滤波器对图像进行滤波处理，可以对图像灰度急剧变化的部分起到锐化作用。而手背静脉图像中, 静脉纹理处和手背皮肤处的灰度级会有较大变化。因此,让手背静脉图像通过高通滤波器，衰减手背近红外图像中的低频成分，保留图像高频成份,可以达到增强静脉图像，使静脉更容易辨别的目的[16-17]。

二维理想高通滤波器的传递函数如式2-1所示：

 （2-1）

是从频率矩形中点向外的截止频率半径，高通滤波器以频率中点为圆心，频率为半径。将在此圆周内的低频频率分量归零，而保留圆周外的高频分量，不进行衰减。但理想的高通滤波器属于非因果系统，在物理的角度上是不可实现的。

高斯高通滤波器传递函数为：

 （2-2）

式2-2中，是像素点距傅立叶变换中心原点的距离。是截止频率。

对卷积处理而言，有空域处理和频域处理两种方法。空域处理即直接卷积，对原始像素点及其邻域直接进行对应元素相乘相加的操作得到处理后的像素值，如图4-4所示。

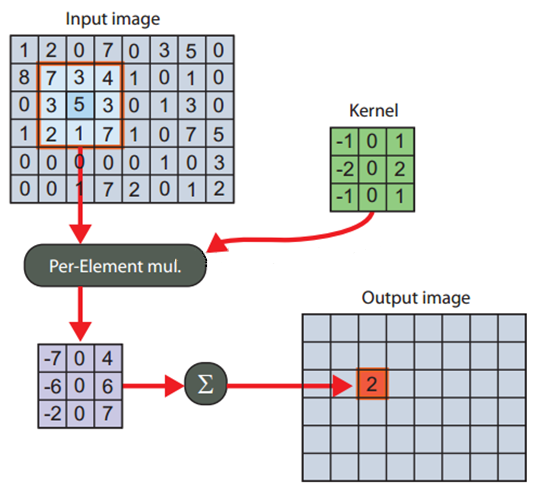


图4-4 图像的空域处理

频域处理则是先把图像进行傅里叶变换，再在频域中进行滤波等操作。即为快速FFT处理。原理即为卷积定理：时域上的卷积等于频域上的乘积。将图像和卷积核进行傅里叶变换，在频域中直接将他们相乘，然后再进行逆变换，还原成时域信息（也就是图像的空域）。频域处理的原理如式2-3，2-4所示。

 （2-3）

 （2-4）

式2-3即为卷积定理，假设为待处理的二维图像，为卷积核，则分别为，的傅里叶变换。则，时域的卷积等于它们频域的乘积。式2-4则是图像卷积的频域处理方法。将设函数I为二维图像，K为卷积核，则他们的卷积等于两者FFT快速傅里叶变换后相乘，再对相乘的结果进行IFFT快速傅里叶逆变换。o表示矩阵逐元素相乘。

## 2.3 开发平台选择

本系统实现的平台有多种选择。首先，基于PC的实现显然最为容易，丰富的器件选择和开发环境会使系统的开发难度大为降低。但PC平台实现的实用性并不强。相比于DSP或FPGA，PC平台的体积较大，在静脉穿刺广泛应用的普通门诊中，显像增强系统的便携性是不可或缺的，系统需要足够小，方便医护工作者进行移动，并且最好能够独立电源供电，因此，PC平台并不适合本系统的实现。

本系统中，对手背图像需要进行获取，存储，处理，输出等一系列操作，且考虑到效果，希望图像分辨率较高，达到720P以上，且需要保证实时性。因此，本系统对处理器的运算能力和数据的实时处理能力要求较高。当前，DSP(数字信号处理器)或者FPGA(现场可编程门阵列)是两种较受欢迎的视频图像处理平台。但是由于DSP其指令执行采用串行通信形式，处理速度相对较慢，因此无法满足本系统的要求。而FPGA平台得益于其优秀的并行运算能力，在便携性，灵活性，功耗等方面均能较好满足本系统的要求，因此经过考虑，最终采用FPGA平台进行开发。传统的FPGA，即现场可编程门阵列，是在可编程阵列逻辑PAL、通用阵列逻辑器件GAL、复杂可编程逻辑器件CPLD 等可编程逻辑器件的基础上演变而来。其内部由可编程逻辑单元CLB、可编程输入输出单元IOB和单元间的连线三部分组成的。通过硬件描述语言（VHDL和Verilog等）编程，使FPGA内部逻辑门按特定方式连接，以完成所需要的电路功能。FPGA 具有高速、高可靠性的优点[18]。

常用的FPGA平台有Altera和Xilinx两个公司的产品。2010年，Xilinx公司推出了FPGA领域第一个可扩展处理平台ZYNQ。ZYNQ平台将双核ARM Cortex-A9处理器和 FPGA 集成在了一起，这样的异构设计框架使得ZYNQ平台相比于ARM的独立开发或FPGA的单独使用，有了更为广泛的应用空间。Zynq面向许多的嵌入式应用场合，汽车电子，视频监控以及自动化控制等领域都是其大展拳脚的地方。ZYNQ平台针对视频系统开发做出了诸多优化，并契合其架构开发了一套可靠性好，数据存储能力强大的VDMA图像存储通路，因此本系统最终采用ZYNQ平台进行开发

ZYNQ芯片分为PS端（Processing System）和PL端（Programmable Logic）两大部分，整个系统整个系统以PS为中心搭建，PL部分作为扩展。在PS端，除了双核ARM Cortex-A9处理器，还有DDR接口、总线接口、DMA控制器、中断管理单元以及EMIO等。PL端和PS端通过AXI总线进行通信[19]。ZYNQ-7000系列的架构图如下图2-2所示。

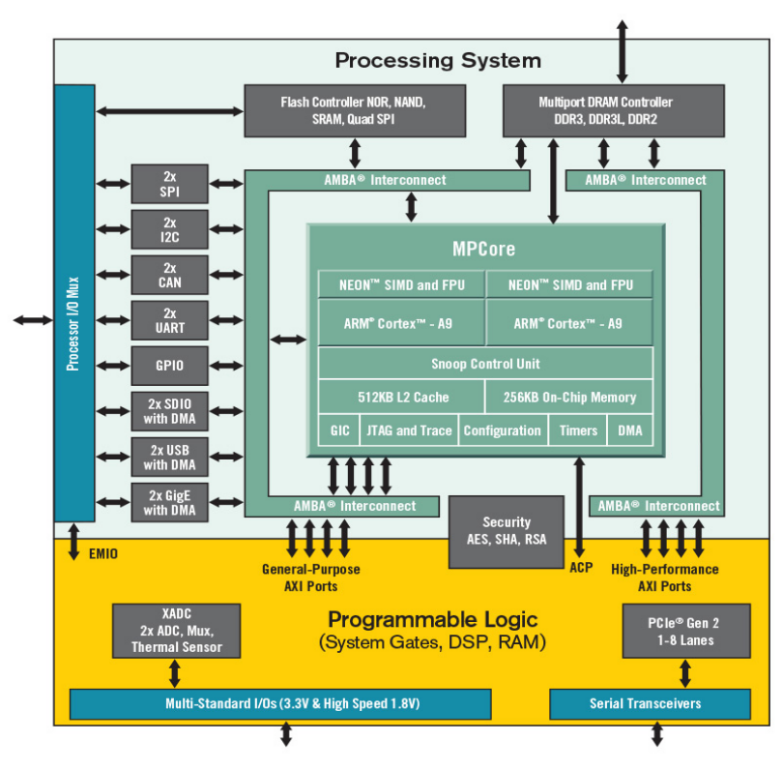


图2-2 ZYNQ-7000系列架构图

# 系统硬件选型

本设计的主要目标是在ZYNQ开发板上实现非接触式的静脉显像增强系统。本系统硬件部分主要分为以下几个模块：ZYNQ开发平台，视频图像采集模块，HDMI图像显示模块，近红外增强模块。

## 3.1 ZYNQ开发平台

在本系统中，开发平台需要外接摄像头和HDMI显示器，且图像的存储和处理需要用到存储器单元，需要ZYNQ开发平台能够提供相应的片上资源。因此，综合考虑下采用了黑金公司基于 XILINX ZYNQ7000 开发平台的AX7010 开发板。AX7010采用Xilinx公司Zynq7000系列的XC7Z010-1CLG400C作为核心处理器，该芯片具有双核ARM Cortex-A9处理器，处理器频率为667MHz，256KB的片上存储器，支持DDR3, DDR3L, DDR2, LPDDR2型号的内存，2x Quad-SPI, NAND, NOR的外部静态内存；其PL端采用Artix-7系列FPGA芯片，有28000个可编程逻辑单元，17600个查找表，35200个触发器等片上资源；而AX7010 开发板在 ARM 和 FPGA 上分别具有丰富的硬件资源和外围接口。包括2片2Gbits的高速 DDR3 SDRAM，HDMI输出接口，40针可外接摄像头的扩展接口。为图像的实时处理和输出提供可能[20] ，开发板的结构示意图如下图3-1示。

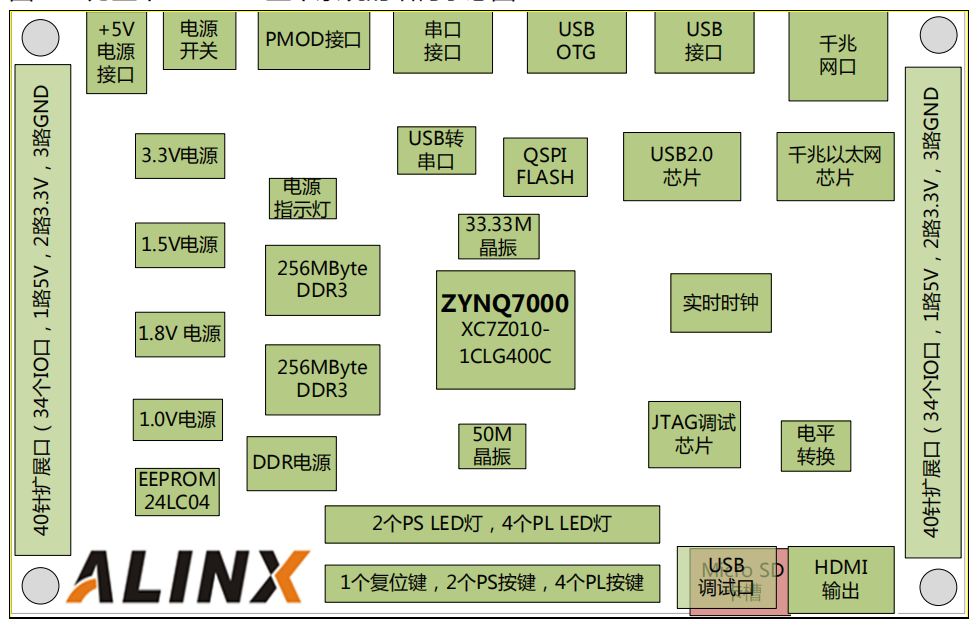


图3-1开发板结构图

## 3.2 图像信息采集模块

本系统中的视频采集模块采用的是黑金单目AN5640图像模块。它的核心是一款型号 OV5640 的彩色 CMOS图像传感器。OV5640可以输出最大分辨率为2592x1944 的 500 万像素的图像，支持 DVP（DC）、YCbCr422、RGB565、RAW 以及 JPEG 等格式的图像数据通过MIPI 接口输出。它还可以通过内部寄存器设置对拍摄图像进行白平衡、色度与饱和度等方面的预处理，但本文中并未涉及。摄像头输出视频的帧率根据寄存器的配置可以在 15-60 Fps间调节。OV5640使用 SCCB 总线配置，SCCB 和 I2C 总线兼容。在本系统中配置图像格式为RGB565输出1280×720大小的图像，由于摄像头的数据总线是8Bit，2 个时钟周期才能完整传输一个像素的数据，所以在FPGA接收端要进行处理，将两个时钟周期传递过来的数据其拼接成完整的像素数据。传感器结构图如图2-4所示。

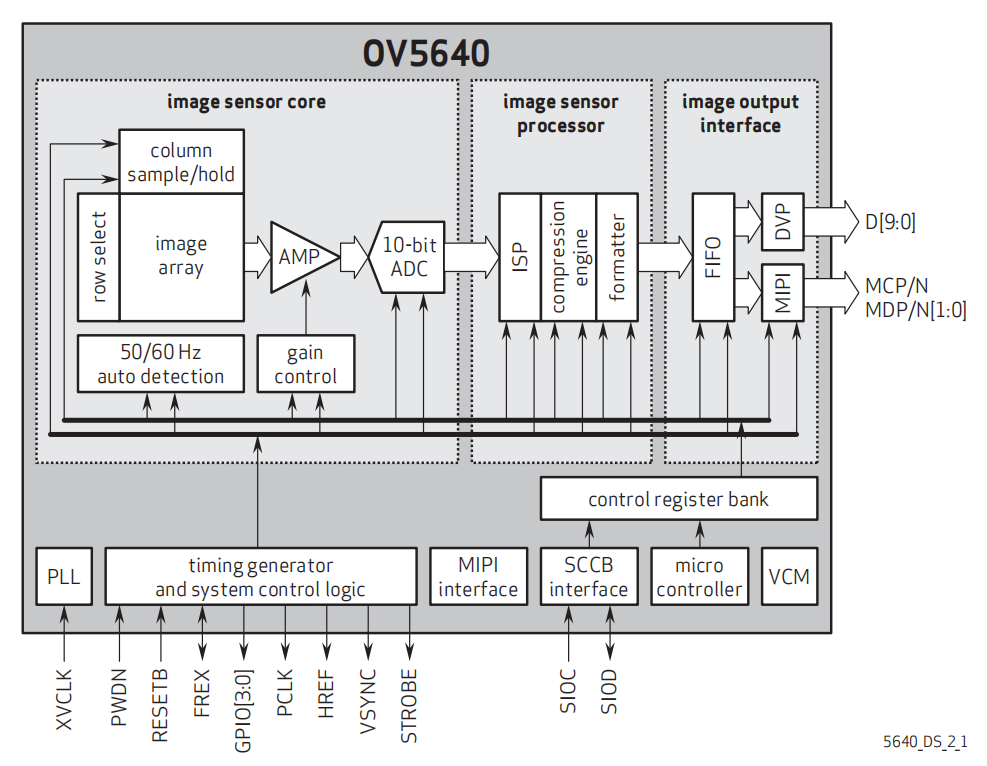


图3-2 OV5640原理图

## 3.3 图像显示模块

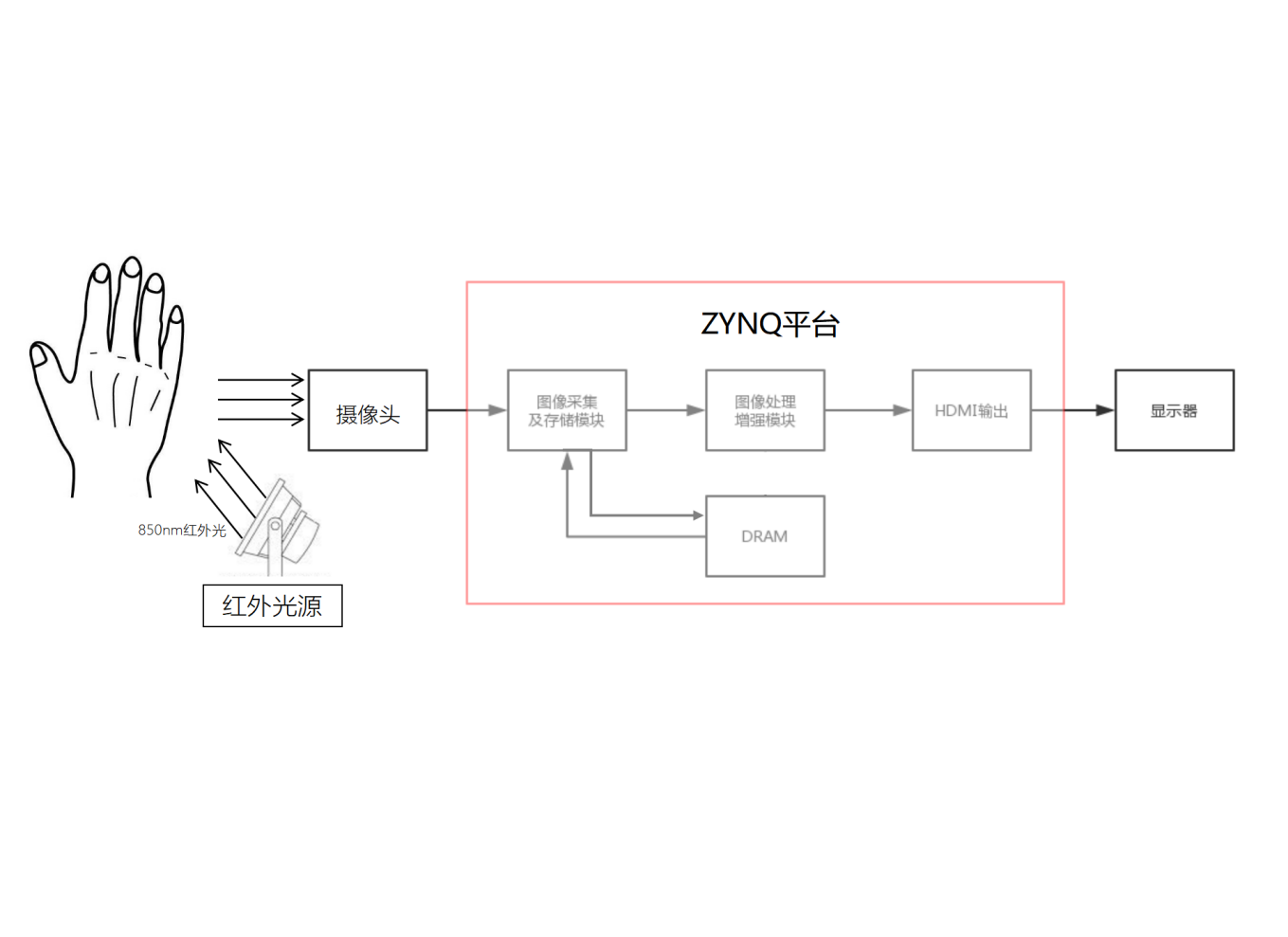
本系统中ZYNQ开发板HDMI接口输出的图像经由一台八寸便携式LED液晶显示器显示。显示器分辨率为1280\*720，恰与图像信息采集模块所采集的图像分辨率相吻合。显示器支持HDMI, VGA, AV接口，本系统使用其HDMI接口。如下图2-5所示，本系统采用的HDMI A型接口共19针，其中1到9针为数据信号。10、11、12针为时钟信号。13针为CEC针，14针为空，15针SCL针，16针为SDA针，17针为地，18针为正5V电源，19针是热插拔检测。



图3-4 HDMI引脚定义

# 系统整体实现

本系统由红外光源和滤光片组成的红外增强模块，摄像头模块，ZYNQ平台内部图像采集存储传输模块，图像处理增强模块和HDMI格式输出模块，以及外部HDMI显示器组成。系统的运行流程如下：采用850nm红外光源照射手背，然后用装有滤光片的摄像头获取人手背近红外图像，通过ZYNQ平台对图像进行存储，处理，最终通过HDMI接口输出并在显示屏上进行显示，系统的整体框图如下图4-1所示。

图4-1系统整体框图

具体到ZYNQ平台内部，其工作流程如下：将摄像头接收的近红外图像数据转换成AXI4-Stream协议格式，经过VDMA将图像数据送入DDR3.而后通过另一个VDMA ip核将图像数据取出，图像通过经由HLS工具生成的图像处理ip核进行图像处理，然后再进行格式变换，由开发板的HDMI接口输出实时视频图像。ZYNQ的推荐图像处理架构如下图4-2所示。

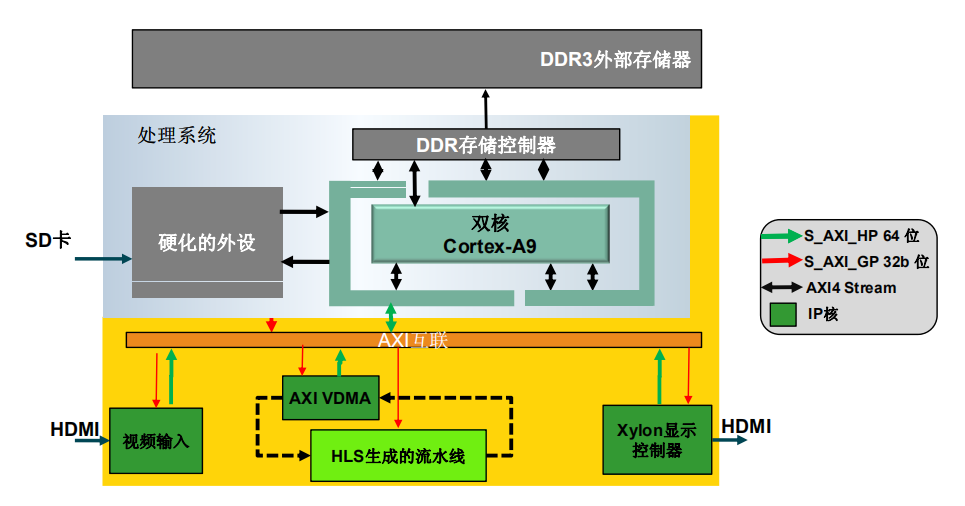


图4-2 ZYNQ图像处理架构

本系统采用Xilinx Vivado开发工具结合HLS开发工具实现。Vivado在设计方式上很大的一个特点就是增加了模块化的设计方式。只需要将单独的Verilog模块打包成IP核的形式，最后将包括ZYNQ最小系统（PS核）在内的所有IP核放在一个顶层文件中，并连接相应的端口，对IP核进行设置等，大大方便了开发。本系统正是在Vivado提供的视频相关IP核和DMA控制IP核的基础上完成的。在PL端，视频信号以AXI4-Stream格式传输，统一的传输格式方便了视频信号在各IP核间的传递。同时，AXI4-Stream格式的数据通过VDMA可以方便地存储到DDR3中或从DDR3中读出。PL端主要包含了摄像头转AXI4-Stream模块，视频存储读取模块，视频图像处理模块和AXI4-Stream转视频流输出模块。PS端作为系统的控制中心，则承担着各IP核初始化，参数配置以及摄像头初始化等功能。摄像头的正常工作需要进行初始化，PS端通过IIC接口与摄像头进行通信，对摄像头进行初始化操作；此外，PS端还需要对VDMA，图片处理IP核以及视频时序产生的IP核进行初始化和参数配置等。PS和PL端的通信通过AXI-Lite总线完成，也就是说PS端由产生的配置或控制信号通过PS端AXI-Lite接口发出，再由各IP核的AXI-Lite接口接收。因此，只有PS和PL端协同配合，才能让功能得以实现。下面的一节中，将介绍本系统使用到的软件开发工具。

### 4.1开发平台介绍

本系统的开发中主要用到了Xilinx公司的Vivado开发套件及HLS高层次开发工具，下面进行相应介绍。

### 4.1.1 Vivado开发平台介绍

Xilinx Vivado开发套件是本系统开发过程中用到的主要软件工具。它是Xilinx公司全新研发的FPGA开发环境，以IP和系统为中心，Vivado的集成度和实施速度相比传统FPGA开发环境提高了4倍。Vivado设计套件将原本独立的开发工具进行了整合集成，并且提供了一些新的开发工具，如本文中会用到的HLS高层次综合工具，以及用于PS端软件开发的SDK工具等。Xilinx Vivado工具以IP核为基础，采用图形化设计，可以帮助用户快速连接IP核，降低开发难度。Vivado同时能够创建进行基于ARM和非ARM的设计，灵活IP封装工具，还可以较为方便地实现IP和子系统的重复使用。这些特性都大大提升了开发效率，为FPGA开发提供了更简便的选择。

### 4.1.2 HLS高层次综合工具

Xilinx Vivado高级综合工具（HLS）可以将C语言程序转换为寄存器传输级（RTL）实现，然后就能够将以C语言为基础的程序应用在Vivado开发平台中，完成相应的功能。HLS工具支持C，C++以及SystemC规范，可以用这三种语言进行功能编程，代替传统的RTL创建流程，从而使IP核的构建更加灵活。 并且Vivado HLS工具支持所创建的IP核可以简单在Vivado中进行添加调用，更进一步减少了开发时间。此外，HLS还提供了一个大规模并行体系结构，所以实现的代码在性能、成本和功耗方面优于用传统处理器实现[22]。

HLS高级综合工具将软件域和硬件域联系在一起，具有以下方面的优势：

①硬件工程师可以在更高的抽象层次上开展设计工作，设计更高性能的硬件，提高系统性能。

②提高软件的系统性能。软件工程师在可以利用HLS优化FPGA设计，简化设计流程，进一步提升密集型算法的计算速度。

使用HLS高级综合设计方法，可以获得以下方面的优势：

①在利用C语言开发算法方面，可以在细节抽象出来的级别上完成工作，提高开发效率。

②在C级验证方面，相比传统的硬件描述语言，其在验证所设计的模块的功能正确性时，所花费的时间更少。

③可以通过优化指令优化代码，降低运行实验，减少所用资源，提升获得最佳方案的可能性，以求在提高硬件性能。

④通过创建可移植的可读C源代码，HLS工程能较为方便在不同的项目中重复利用，节约了开发资源[23]。

Vivado HLS工程的设计流程为：

① 编译，执行（模拟）和调试C算法。

② 将C语言合成到RTL实现，有可供选择的用户优化指令。

③ 仿真，生成仿真报告并分析资源占用，时延等情况。

④ 验证RTL实现。

⑤ 打包成一系列可供选择IP格式。

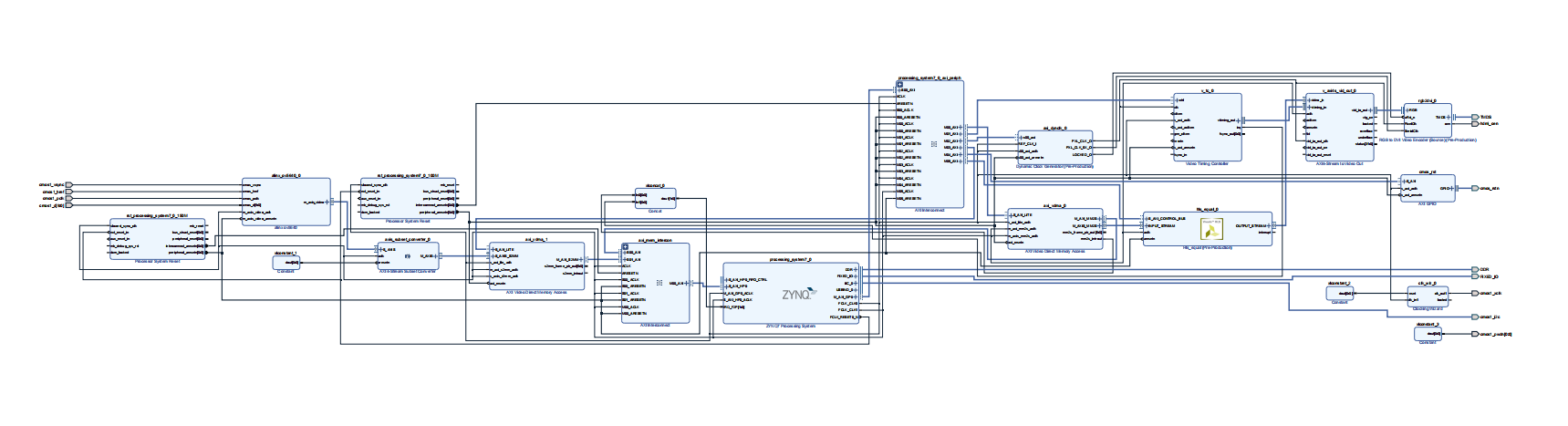
⑥ 在Vivado中调用定制IP核[24]。

此外，HLS 设计工具集成了专用的定制化的OpenCV 函数库，即hls视频库。这个函数库包含图像处理领域的许多常用算法。这些函数经过针对性优化后，在ZYNQ平台上具有相对较高的计算性能，可以在HLS工程中直接声明调用，对AXI4-Stream视频流数据进行图像处理。使用HLS 进行图像处理等应用设计，可以实现对现有资源的快速继承和移植，设计可以在FPGA上运行，图像的计算速度可以通过硬件加速以获得改善。因此，HLS 设计工具在图像处理应用上具有很大优势[25]。这也是本系统采用HLS工具的主要原因。

4.2 IP核介绍

根据上节介绍，相对于ISE，Vivado模块化的设计方式是其相比于传统FPGA开发平台的特点，开发人员只需要添加相应的IP核并进行相关设置，然后在各IP的端口间进行连线完成顶层设计，方便而且直观[26]。

本系统的实现流程如下，从数据的传输方向看，首先是摄像头数据处理。该模块作用是将摄像头输入的 RGB565格式的数据转换成 AXI4-Stream 格式，以符合数据流在ZYNQ系统中的传输和在视频处理IP核中的格式。随后数据流向VDMA IP核。VDMA在硬件部分中控制着DDR3，起到将图像数据存储到DDR3存储器中，或将图像数据从DDR3中读取出来的功能。随后另一个VDMA将数据从内存中取出，图像经过HLS生成的视频处理IP核，通过处理后，再由格式转换的Ip核，将AXI-Stream格式的图像数据转换成RGB格式的数据，再由HDMI接口输出视频图像。最终的顶层模块连接图如图4-3所示。

图4-3 工程Block Design顶层文件

下面的小节中将简要介绍本设计中使用到的IP核。

### 4.2.1视频流-AXIS转换 IP核

  Video In to AXI4-Stream IP核及AXI4-Stream to Video Out IP核是用于RGB格式视频流与AXI4-Stream协议数据流之间相互转换的IP核。Video In to AXI4-Stream IP核用于将视频源转换成AXI4-Stream接口形式，实现了接口转换。Xilinx提供的许多视频处理相关的IP核，均使用AXI4-Stream协议进行数据的发送和接收，也就是说其输入和输出端口都是AXI4-Stream接口（Slaver/Master），因此需要对视频信号的格式进行转换[27]。AXI4-Stream to Video Out IP核的功能与之恰好相反，它会将AXI4-Stream格式的视频信号转换为用户定义的输出格式，如本设计中将视频由AXI4-Stream格式转换为RGB格式，供后续的模块使用。

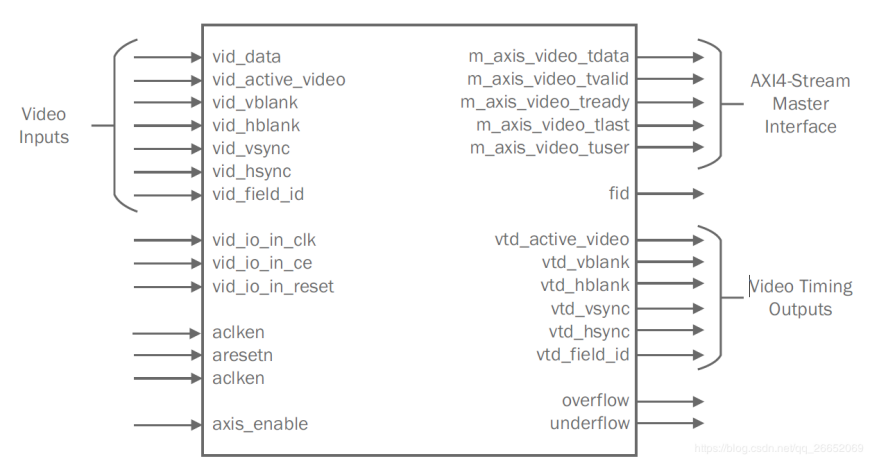


图4-4 Video In to AXI4-Stream IP核

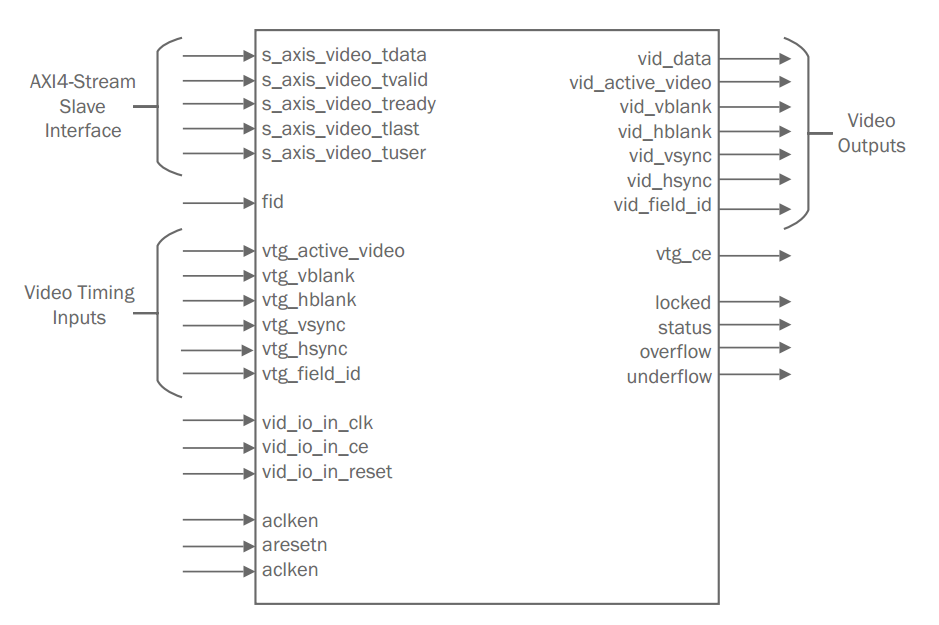


图4-5 AXI4-Stream to Video Out IP核

### 4.2.2 VDMA IP核

VDMA，全称Video Direct Memory Access，是一个专用于视频数据存储与读取的IP核。VDMA 内部有读通道和写通道，用户可以按需进行使能，即可以单独使用读通道或写通道。通过写通道可以将 AXI-Stream 格式的视频数据写入外部内存，通过读通道可以从外部内存中读取数据，并以 AXI-Stream 格式将视频数据传递输出给后续模块。因此，VDMA IP核的角色是视频数据在PL端和内存中的搬运工 ，使ZYNQ平台上视频数据在内存中的存取更加便捷。

将视频图像数据存入内存中之后，ARM可以在PS端对图像进行一些例如缩放裁剪等的处理处理，然后等待VDMA将图像读出，也可不做任何操作。VDMA 可以设置最大32个帧存，并可以由用户切换帧存读取，能够轻松地实现双缓冲和多缓冲操作。在运用VDMA进行图像通路设计的时候，通常是采取双缓冲或多缓冲的方式，以降低视频显示时延。本设计中，为了调试方便，采用了两个VDMA IP模块协同工作。一个VDMI只使能写通道，用于将前面Video In to AXI4-Stream IP核输出的AXI4-Stream格式的视频流写入DDR3；另一个VDMA只使用读通道，用于将存储在DDR3中的数据以AXI-Stream格式读取出来。

因此，在基于ZYNQ平台的视频传输处理系统中，使用 VDMA可以大大简化开发流程，加快开发速度。

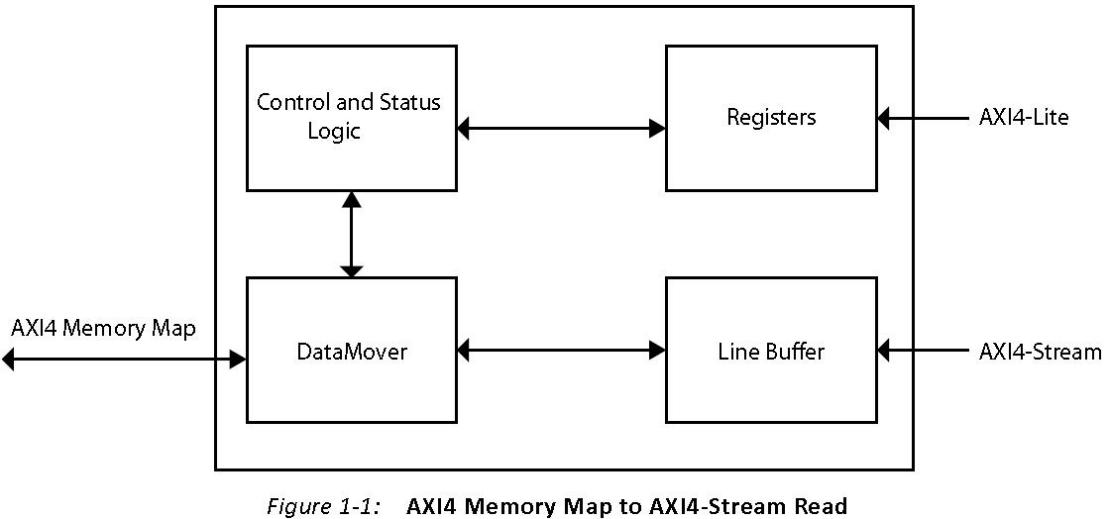


图4-6 VDMA结构框图

### 4.2.3 ZYNQ Processing System

在Vivado中，可以很方便地将处理器系统以IP核的形式添加到工程中，并对PS端进行设置，工程中添加的IP核如图4-7所示。

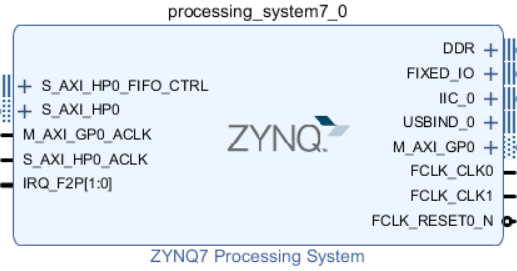


图4-7 Processing System最小系统

在下图4-8中，可以更加清晰地看到PS架构及使能的一些接口和功能。

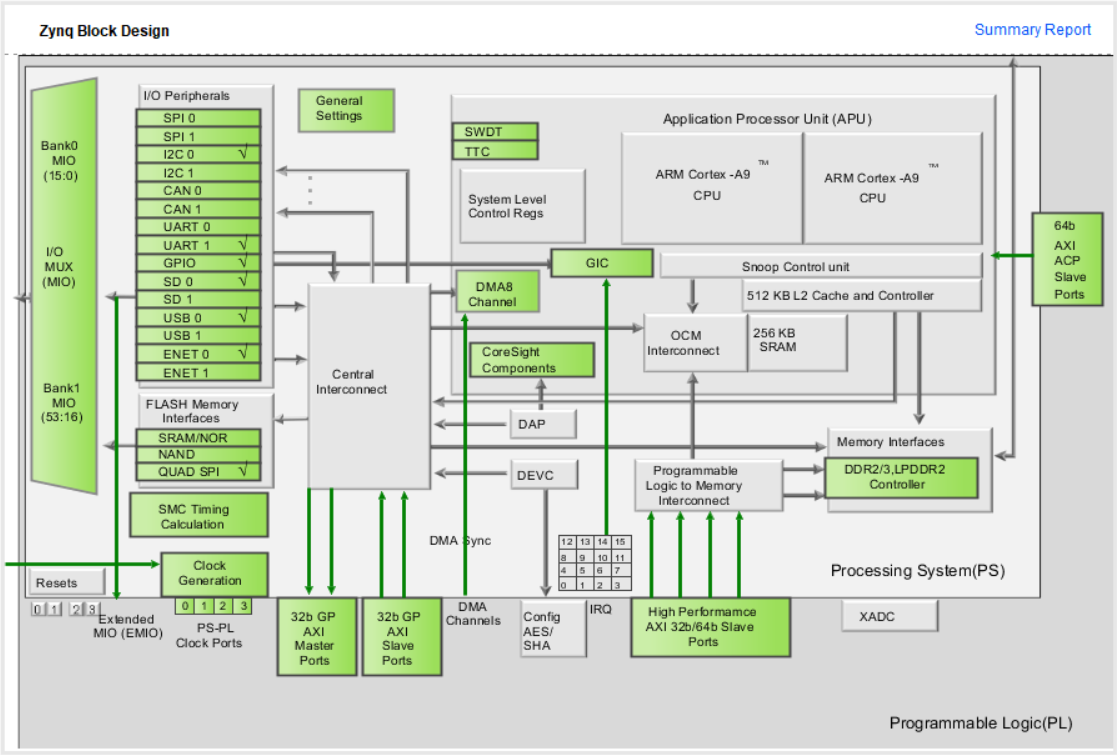


图4-8 ZYNQ片上系统硬件配置

本系统中，在PS端除了配置与VDMA通信的AXI\_HP接口，和IP核控制接口通信的AXI\_GP接口，还使能了I2C接口用于初始化摄像头，配置了两路时钟，一路100Mhz的时钟作为各IP核AXI-Lite接口的时钟信号，另一路150Mhz的时钟则用于VDMA的数据存取。

### 4.2.4 HLS视频处理IP核

在读VDMA读取了内存中的视频图像数据后，AXI4-Stream格式的视频图像就要经过HLS IP核进行算法的处理，下面详细介绍一下HLS视频处理IP核的实现。

在HLS工具中内置了hls\_video.h头文件。 此头文件中包含Vivado HLS提供 的所有图像和视频处理的功能函数。这些函数与OpenCV函数的使用方法类似，但却是为FPGA所特别优化过的，适合并行运算。本次设计中，我们采用了函数库内的hls::filter2D()函数设计了一个高通图像滤波器，以达到对静脉图像进行锐化的效果。在视频库自带的直方图均衡函数不起作用的情况下，作者尝试自行用C语言编写直方图均衡函数，但由于代码优化能力限制，所综合的代码所占用的资源过大，超出了ZYNQ开发板上的资源限制，所以最终实现中并没有采用直方图均衡算法。

Hls高级综合工具中自带的视频库函数hls::Filter2D 允许用户自定义卷积核，并应用卷积核对图像进行线性滤波。线性滤波首先需要定义一个二维滤波器矩阵（卷积核），所定义的二维滤波器矩阵不同，图像处理的效果也随之不同。定义了卷积核之后，对原来图像的每一个像素点，将卷积核的中心对准该像素，然后将卷积核中的值乘上图像中位置相对应的像素的值，再把所有结果相加作为该像素滤波后的值，这样就完成了滤波过程。其原理如图4-9所示

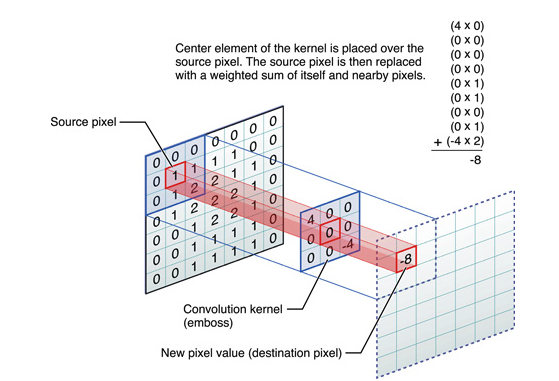


图4-9 线性滤波原理

下图4-9所示为常用的两个半径为1的高通滤波卷积核，对图像应用以下的卷积核可以实现高通滤波的效果。

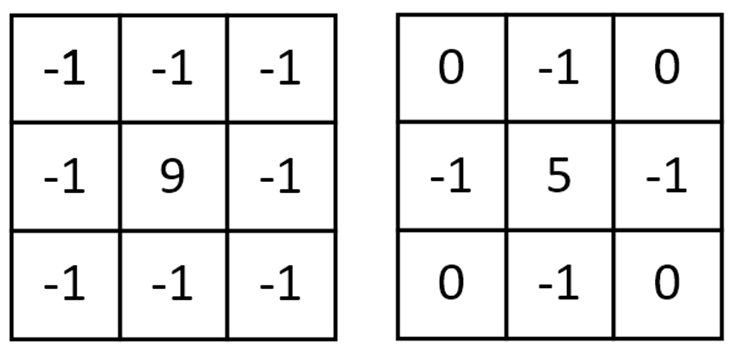


图4-10 高通滤波常用卷积核

HLS视频图像处理IP核的设计流程为：读取AXI-Stream流数据，将AXI-Stream流数据转换为hls::Mat格式，图像处理，将hls::Mat格式数据转换AXI-Stream流数据，输出图像。

下面对HLS视频处理IP核的代码做一些简要介绍：

下图对IP核的输入输出端口做了约束，将输入和输出指定为为以“INPUT\_STREAM”与“OUTPUT\_STREAM”命名的 AXI4-Stream，将控制接口分配到 AXI4 Lite 接口，指定“rows”，“cols”这两个量，可通过 AXI4-Lite 接口来配置，即图像的分辨率由AXI4-Lite 接口控制。此外还并且声明了函数中的中的 “rows”，“cols”为常量，在程序运行过程中不变，开辟寄存器单独控制[28-29]。

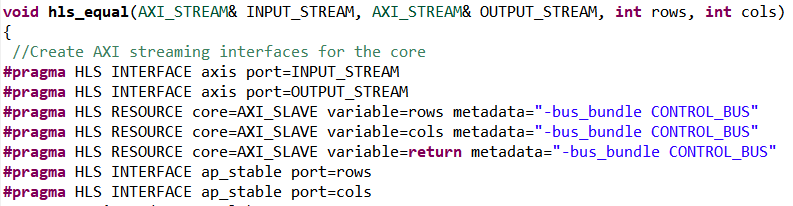


图4-11 IP核端口约束

下图中hls::Window定义了一个三阶高通卷积核，hls::Pointdin定义内核的锚点，指示内核中过滤点的相对位置（-1，-1）指示锚位于内核中心。

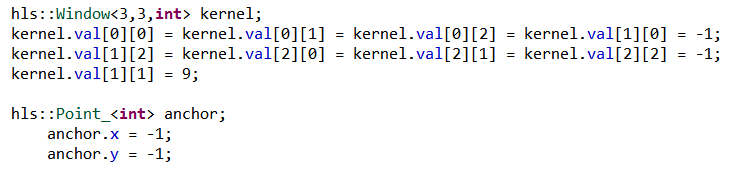


图4-12 卷积核定义

下图是视频处理核心函数，需要注意的是 #pragma HLS dataflow指令，启用数据流优化，这条指令使以下函数以流水线的方式执行。

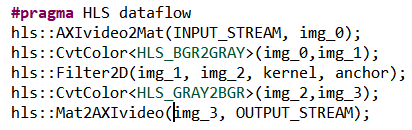


图4-13 IP核核心函数

之后对代码进行综合，可以在图4-14的仿真结果看到延时和对片上资源的占用情况，本HLS IP核共使用了3个BRAM\_18K(存储资源),3个DSP48E（计算资源），1201个触发器和3211个查找表，均远低于片上可用资源，对片上资源的占用分别为2%,3%,3%,18%。

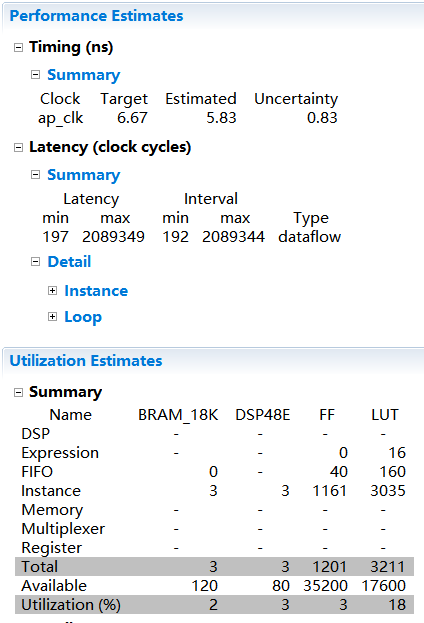


图4-14 资源占用情况

随后编写代码对其进行功能仿真，最后就可以进行生成IP核的操作了。生成的IP核框图可以在Vivado中调用，Vivado中的HLS IP核如下图4-15所示，IP核中央会有Vivado HLS的标志，表明该IP核由HLS生成。

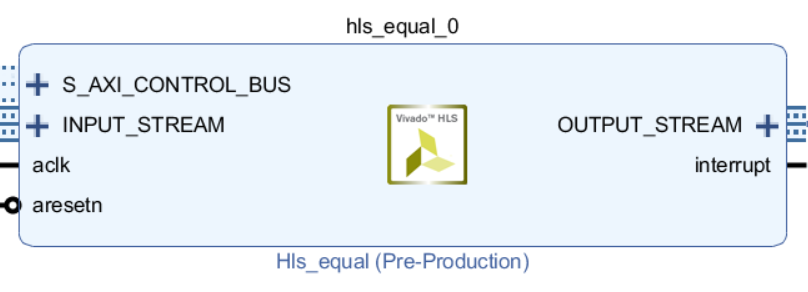


图4-15 生成的IP核

将hls生成的IP核加入到顶层原理图文件中，并连接端口，在SDK中编写驱动，就可以使用了。需要注意，hls IP核中的interrupt端口让用户可以通过中断的方式使能IP核。但本系统中并未用到中断方式，因此将本端口留空即可。

### 4.2.5 Video Timing Controller IP核

Video Timing Controller IP核可以根据用户需要，产生不同分辨率大小的视频时序。本系统采用720P分辨率的视频图像，因此设置其产生720P图像分辨率的视频时序，产生的视频时序将由vtiming\_out端口输出给AXI4-Stream to Video Out IP核，配合HLS IP核处理过的图像数据，转换成为RGB格式的视频流数据。Video Timing Controller IP核的框图如下图4-15所示

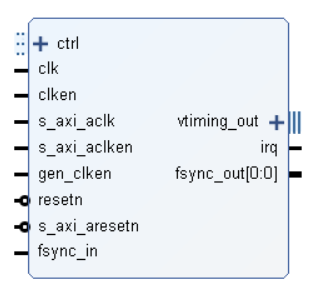


图4-16 Video Timing Controller IP核

### 4.2.6 RGB to DVI Video Encoder IP核

在AXI-Stream视频流信号经AXI4-Stream to Video Out IP核转换为RGB格式后，仍不能由HDMI接口之间输出，需要通过XIlinx提供的编码器，将RGB格式的数据以及像素时钟信号和视频的同步时钟信号，编码成为TDMS格式供HDMI接口输出。RGB to DVI Video Encoder IP核的框图如图4-17所示。

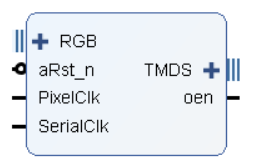


图4-17 RGB to DVI Video Encoder IP核

### 4.2.7 AXI Interconnect IP核

AXI Interconnect模块实现寄存器映射的主设备和从设备之间的连接。AXI互联模块只用于内存映射的数据传输。即AXI Interconnect IP核并不传输AXI-Stream流的数据，而是传输AXI4、AXI3或者AXI4-Lite协议的数据。具体于本系统中，PS核的的AXI\_HP接口应该与写VDMA的S2MM端口以及读VDMA的MM2S相连；而各IP核中负责传递配置信息的AXI4-Lite接口应该与PS端的AXI\_GP接口相连，用于IP核的初始化和参数配置等功能。但PS核中AXI接口数量并不足够与PL中各IP核的相应接口一一连接，因此需要用到AXI Interconnect模块充当路由功能，在接口间进行一对多或多对一的接口互联。

AXI Interconnect IP核有基本连接模式如下：

• N-to-1 Interconnect

• 1-to-N Interconnect

• N-to-M Interconnect (Crossbar Mode)

• N-to-M Interconnect (Shared Access Mode)

本系统中，采用的是N-to-1和1-to-N连接模式。其中，VDMA IP核和PS核的接口采用N-to-1模式，即两个Master口，一个Slave口，用以向内存中写入数据或从内存中读出数据[30]。N-to-1模式的示意框图如图4-18所示。

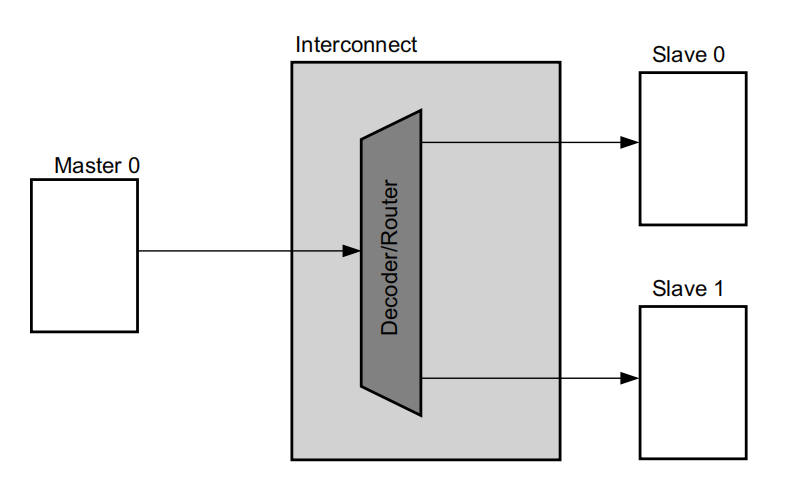


图4-18 N-to-1模式

而在PS端AXI\_GP接口和各IP核之间的AXI Interconnect则采用了1-to-N连接模式，即一个Master口，多个Slave口，1-to-N连接模式的示意框图如图4-19所示。

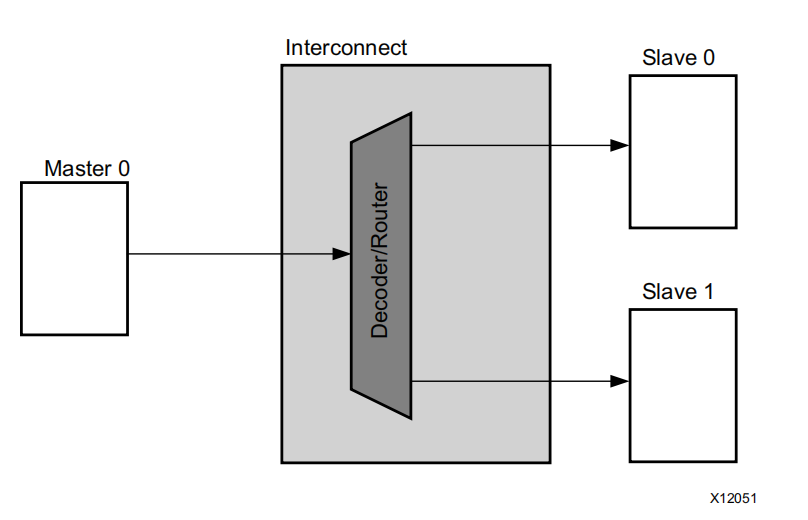


图4-19 1-to-N模式

### 4.2.8 GPIO IP核

GPIO 全称为 General Purpose Input Output，用于PS端对IO口的读写，GPIO IP核的框图如下图4-20所示。

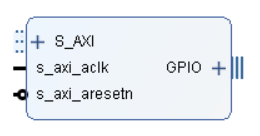


图4-20 GPIO IP核

本系统中，GPIO核的S\_AXI端连接控制接口AXI Interconnect IP核的Slave接口，GPIO端则连接摄像头I2C接口，通过PS端产生的控制信号，对摄像头进行寄存器的相关配置，以完成初始化摄像头的功能。

### 4.2.9 Processer System Reset IP核

Processer System Reset 模块可以为包括处理器，互连和外围设备产生所需的复位信号。它支持与时钟同步的异步外部复位输入，外部和辅助复位输入都可以选择为高电平有效或低电平有效。该模块的结构框图如下图4-21所示。

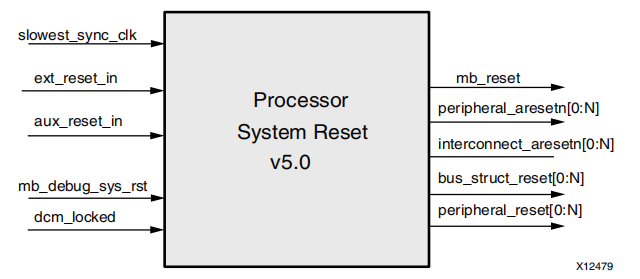


图4-21 Processer System Reset IP核

在本设计中主要用到如下几个端口：输入端口中，slowest\_sync\_clk：连接到系统中时钟，若有多个时钟，则连接最慢的一个；ext\_reset\_in：连接外部输入的复位信号，本系统中连接到PS核产生的复位信号端口。输出端口中，Peripherals\_aresetn:低电平有效的外设复位信号，连接各功能模块，如VDMA IP核，Video Timing Controller IP核，GPIO IP核等；Interconnect\_aresetn:低电平有效的Interconnect复位信号，产生AXI Interconnect IP核的复位信号，需要连接到AXI Interconnect IP核的复位端口；需要注意的是，由于本系统中有100Mhz和150Mhz两个时钟域，所以需要分别用两个Processer System Reset IP核分布产生复位信号。

# 实验结果

本实验中用到的黑金AX7010开发板如下图6-1所示，实验中共用到了其HDMI接口，JTAG接口，在图中均有标注。摄像头通过其40针拓展IO口与开发板相连接。

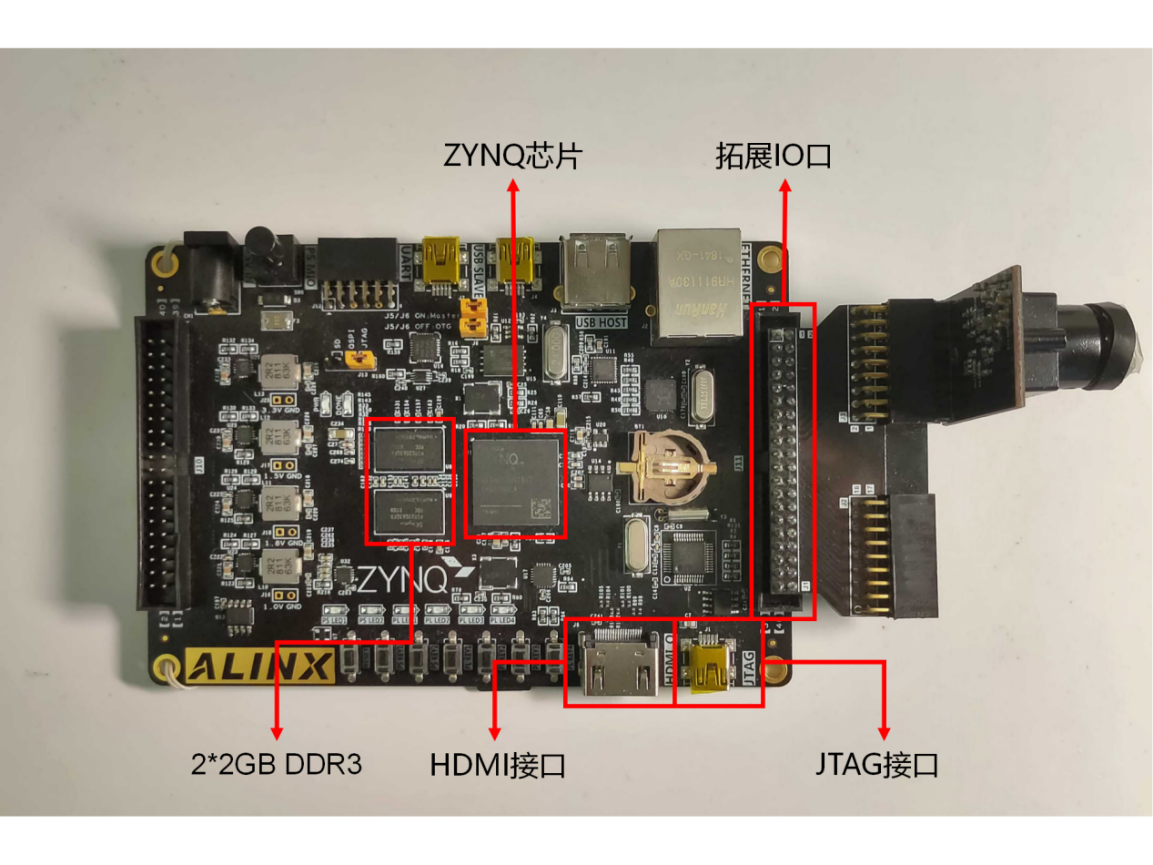
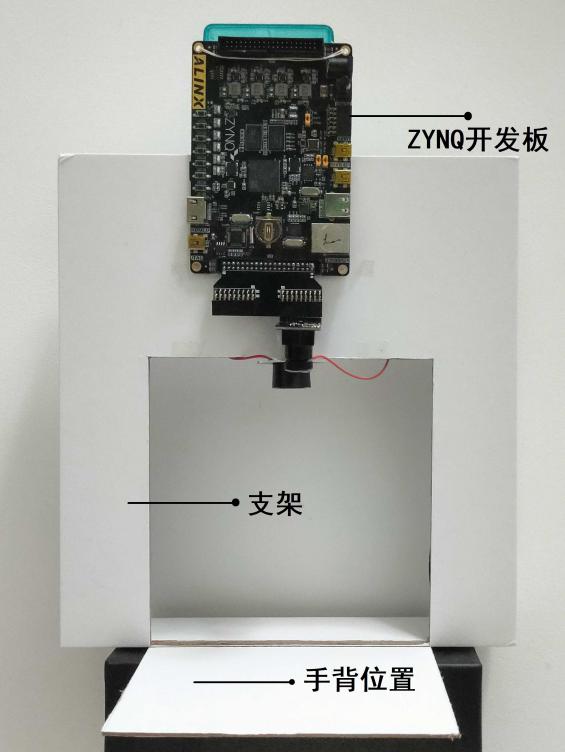
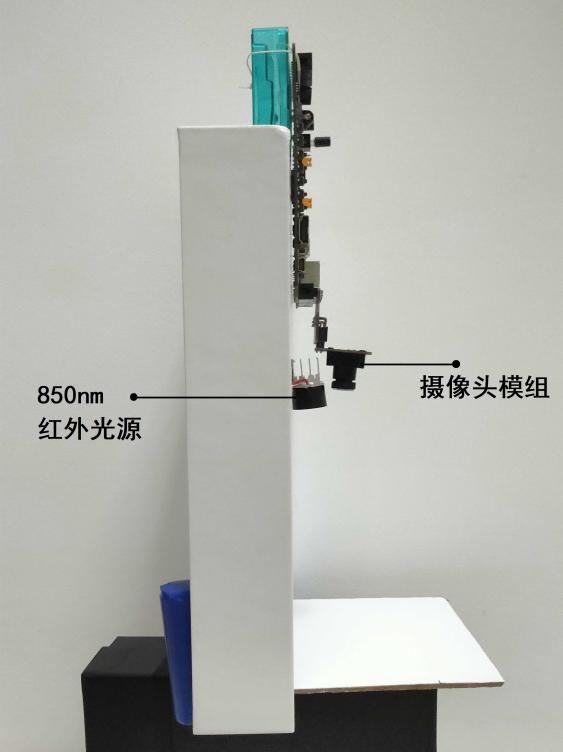


图6-1 AX7010 ZYNQ开发板

根据实验需要，搭建简易实验环境如下图图6-2所示，图(a)为实验平台主视图，图(b)为实验平台侧视图。

(a)主视图 (b)侧视图

图6-2实验平台

最终的实验环境如下图6-3所示。

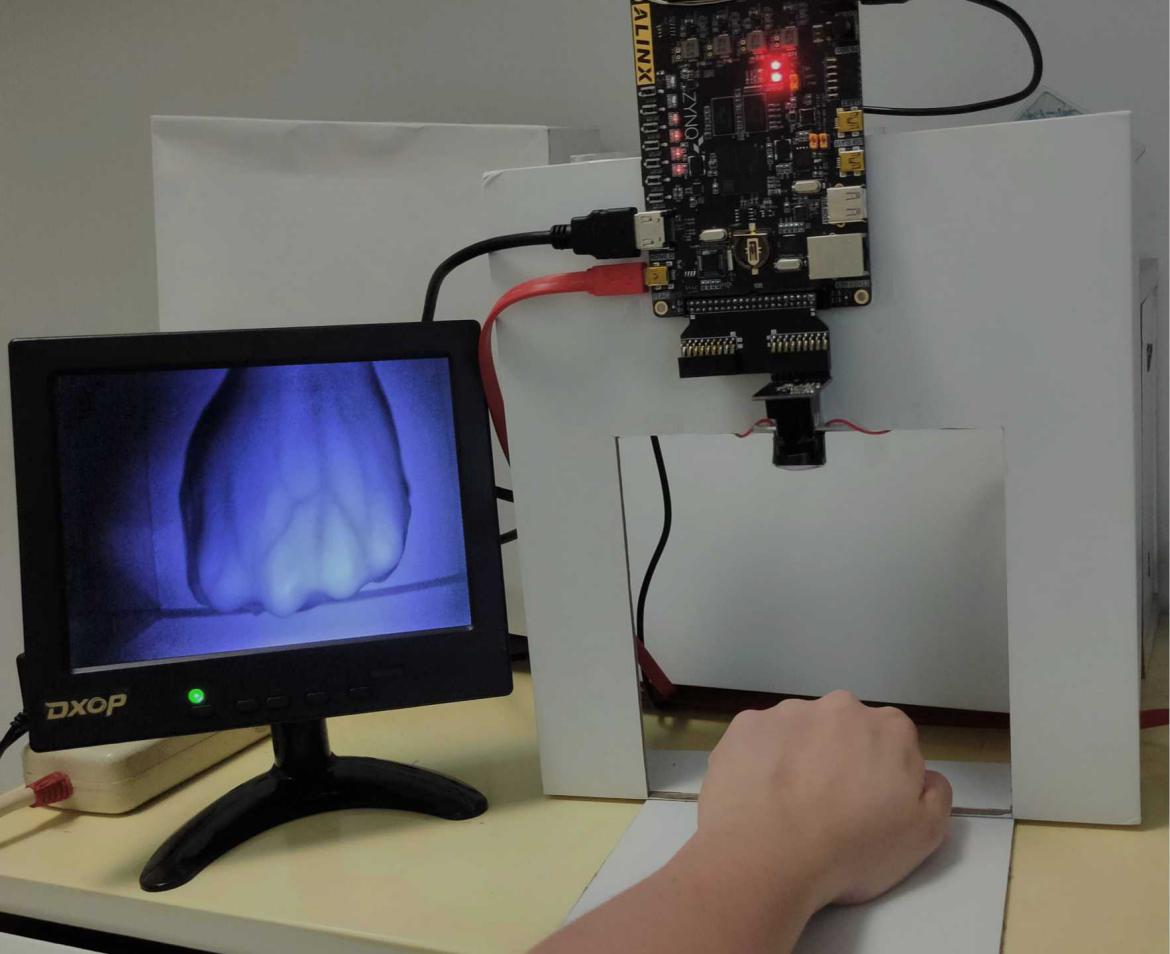
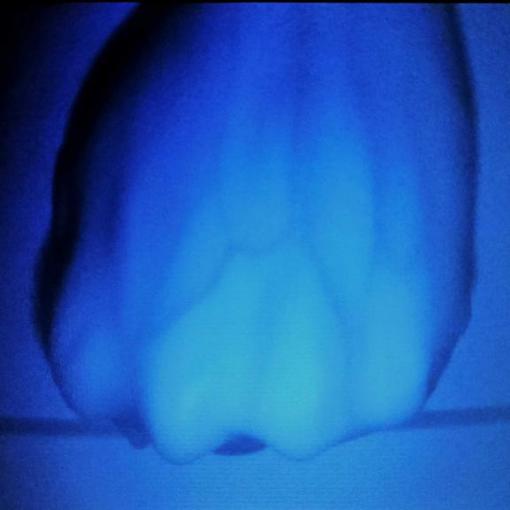


图6-3 实验环境

编译运行后，摄像头可以正常工作并在HDMI显示屏上显示实时视频画面。本系统中，由于采用了红外窄带滤光片，因此可见光对实验的影响并不大，下图6-4中，左(a)图为可见光光源较强时监视器内显示的手背近红外图像，右(b)图为几乎没有可见光照射时的图像，可以发现两者清晰度区别并不明显。

1. 有可见光源 (b)无可见光源

图6-4 可见光对成像影响

(a)未经处理 (b)仅红外无处理 (c)红外+算法增强

图6-5 实验结果

图6-5为实验结果。(a)图为未经处理的，可见光源照射下的手背静脉图像在显示屏上的显示效果，从图像中可以发现，手背的静脉纹路模糊不明显，肉眼很难找到静脉的准确位置；(b)图为经过850nm近红外光源照射和滤光片滤光后采集的手背静脉图像，可以发现颜色较深的静脉纹路相比与图(a)在手背上更加清晰，肉眼较好识别；图(c)为经过850nm近红外光源照射和滤光片滤光后，再通过高通滤波器滤波处理后的手背静脉图像，容易看出，图(c)中手背上颜色较深的静脉纹理相比于图(b)，在边缘上更加锐利，于手背皮肤的区分度得到进一步提高。从实验结果中可以得出，经过本系统处理获取的手背静脉图像，静脉的清晰度得到一定程度的提高，且视频的帧率较高，实时性较好，达到了设计目的。

# 6.结语

本设计采用ZYNQ 开发平台，在Xilinx Vivado开发套件上搭建了一套基于VDMA的图像接收处理显示通路，并针对图像采用HLS工具，设计了高通滤波的算法增强，编写了相应的驱动程序，实现了静脉显像增强功能，基本完成了设计目标。但本次设计中，灰度图像增强的直方图均衡算法未能顺利实现，而直方图均衡算法对提升图像对比度有比较大的帮助，这是本次设计的一点不足，是下一步对系统进行优化的重点。此外，处理过的手背静脉图像与原图像在静脉清晰程度上的量化比较工作由于时间所限未能开展，无法更加精确地分析系统对静脉图像的增强效果，是本次设计中的另外一点不足之处。展望未来，在实现了静脉图像加强后，可以进一步实现静脉图像在手背上的投影，代替目前的在显示屏上显示的方式，这样可以更直观地将加强后的静脉图像展示，进一步降低静脉医疗操作的难度。

# 参考文献

1. S. Boukhtache, B. Blaysat, M. Grédiac, and F. Berry, “Alternatives to bicubic interpolation considering FPGA hardware resource consumption,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 29, no. 2, pp. 349–360, Feb. 2021.
2. J. Smith, L. Wang, “Efficient Hardware Architectures for Image Interpolation in Embedded Systems,” Journal of Embedded Systems, vol. 15, no. 3, pp. 123–134, 2022.
3. Kumar, et al., “Real-Time Image Scaling Using Approximate Bicubic Interpolation on FPGA,” Proc. IEEE Int. Conf. on Image Processing, pp. 45–50, 2022.[1][2][3]
4. M. Zhang, Y. Li, “Low-Complexity Interpolation Techniques for High-Resolution Displays,” Advanced Display Technologies, vol. 10, no. 4, pp. 89–97, 2023.
5. L. Roszkowiak, A. Korzynska, J. Zak, D. Pijanowska, Z. Swiderska-Chadaj, and T. Markiewicz, “Survey: Interpolation methods for whole slide image processing,” J. Microsc., vol. 265, no. 2, pp. 148–158, Feb. 2017.
6. M.-S. Pan, X.-L. Yang, and J.-T. Tang, “Research on interpolation methods in medical image processing,” J. Med. Syst., vol. 36, no. 2, pp. 777–807, Apr. 2012.
7. C.-H. Kim, S.-M. Seong, J.-A. Lee, and L.-S. Kim, “Winscale: An image-scaling algorithm using an area pixel model,” IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 6, pp. 549–553, Jun. 2003.
8. E. Aho, J. Vanne, K. Kuusilinna, and T. D. Hamalainen, “Comments on ‘winscale: An image-scaling algorithm using an area pixel model,”’ IEEE Trans. Circuits Syst. Video Technol., vol. 15, no. 3, pp. 454–455, Mar. 2005.
9. R. Keys, “Cubic convolution interpolation for digital image processing,” IEEE Trans. Acoust., Speech, Signal Process., vol. ASSP-29, no. 6, pp. 1153–1160, Dec. 1981.
10. E. Duchon, “Lanczos filtering in one and two dimensions,” J. Appl. Meteorol., vol. 18, no. 8, pp. 1016–1022, Aug. 1979.
11. Dong, C. C. Loy, K. He, and X. Tang, “Image super-resolution using deep convolutional networks,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 38, no. 2, pp. 295–307, Feb. 2016
12. Dong, C. C. Loy, K. He, and X. Tang, “Learning a deep convolutional network for image super-resolution,” in Computer Vision. Cham, Switzerland: Springer, 2014, pp. 184–199.
13. Xilinx Inc. LogiCORE IP AXI Interconnect v2.0 Product Guide[EB/OL]2020.05.http://www.xilinx.com/support/documentation/ip\_documentation/axi\_interconnect/v2\_0/pg059-axi-interconnect.pdf

# 致谢

通过本次毕业设计，本人在专业水平上有了一定程度上的提高，尤其是Vivado开发套件的应用与HLS开发工具的使用，也对基于VDMA的视频图像处理有了一定了解。这些都要得益于此次设计任务，这次毕业设计考验了我的自主学习能力，让我收益颇丰。

我的大学生活在即将划上句号，回顾毕业设计的完成过程，遇到过许多困难。从疫情影响无法返校导致进度延后，到后面重新购置设备在家开展实验，浣沙老师对我进行远程指导，期间身体还一度抱恙被迫暂停了一段时间。但正所谓阳光总在风雨后，磕磕绊绊之后毕业设计最终还是得以完成。这里要特别感谢指导老师浣沙老师，是她的悉心指导让我得以按时递上这份答卷，浣沙老师的指导与帮助我将铭记于心。此外还要感谢四年以来所有任课老师们的谆谆教导为我打下的基础；感谢父母的鼓励和支持，给了我直面困难的勇气；最后感谢陪伴我四年青春的四位可爱的舍友们，让我这四年的生活中无时无刻不充满了欢声笑语，让我感受到家人般的温暖。