

版本信息:

版本 REV2022

时间 10/28/2022

米联客2022版FEP子卡篇-MIPI模块使用教程

Milianke 2022 FEP Card Course MIPI

电子版自学资料

常州一二三电子科技有限公司
常州米联客电子科技有限公司
版权所有

米联客 FPGA|SOC 生态 QQ 群 05(XILINX)(新群):554101168
米联客 FPGA|SOC 生态 QQ 群 04(XILINX)(已满):516869816
米联客 FPGA|SOC 生态 QQ 群 03(XILINX)(已满):543731097
米联客 FPGA|SOC 生态 QQ 群 02(XILINX)(已满):86730608
米联客 FPGA|SOC 生态 QQ 群 01(XILINX)(已满):34215299

如果 QQ 群全满无法加入, 请登录网址: <https://www.uisrc.com/f-380.html> 查看最新可以加入的 QQ 群

微信扫码注册米联客技术论坛www.uisrc.com免费享受更多资源



扫码关注微信公众平台“米联客(MILIANKE)”掌握更多信息动态



版本	时间	描述
Rev2022	2022-10-28	第一版

序 1:

FPGA 芯片是硬件技术而 FPGA 编程又称为硬件编程语言和流行的各类软件编程语言 C/C++、JAVA、python 等相比,掌握基础的硬件编程语言不是难事,难点是 FPGA 在每个专业领域的应用,只有充分理解了 FPGA,并且具有对自己所处行业专业背景认知,才能真正理解 FPGA 应该用在什么场合更加合适。

从业多年来,亲身经历了 FPGA 的发展历程,也深刻体会到未来 FPGA 应用领域可能发生的深刻变革,FPGA 从简单的逻辑门,发展到现在具备很多高速通信接口,而且最新 SOC 中也集成了 FPGA 单元,实现了 ARM 和 FPGA 单芯片。目前 XILINX 代表了业内领先的 FPGA 技术,已经可以把 FPGA\ARM\GPU\RFID 等集成到单芯片。FPGA 的逻辑资源也是达到了前所未有的密度以及超大容量。

很多人问我学习 FPGA 是否有前途,这个问题着实难以回答。我们可以一起来探讨以下几种情况,会是 FPGA 发挥作用的场合:

1)、数字 IC 设计工作

数字 IC 设计还是主要以硬件编程语言去设计数字 IC 芯片,通过硬件编程语言,软件可以把语言翻译成电路,自动布线工具可以完成布局布线,之后流片。由于流片费用非常贵,前期也可以用 FPGA 芯片模拟设计的数字 IC 芯片的功能。

2)、高速模数字信号采集分析

高速的 ADC, DAC 的数模信号处理的领域也是必然需要用到 FPGA,在无线通信、雷达信号处理等领域也都会用到。

3)、数字信号高速通信

FPGA 具备的高速接口也非常适合用于高速通信,比如 PCIE 通信、光通信、以太网通信

4)、视频图像

包括图像的拼接、缩放、高效的实时传输等, 4K 视频、8K 视频领域

5)、硬件加速算法

FPGA 的硬件加速领域也是目前的热门研究,也是 FPGA 未来最有前景的一个应用领域,已经有很多公司利用 FPGA 的硬件加速实现了很好的经济效应,但是目前 FPGA 的加速还没有做到普及,和传统的 GPU 相比,主要难度还是在开发难度上,一般小公司很难有实力取得突破。

6)、通用 CPU GPU 无法完成的工作

如果通用的 CPU 和 GPU 无法完成的工作任务,可以考虑下 FPGA 或者带 FPGA 的 SOC.

FPGA 到目前为止依然是一个小众的领域,如果专门为了学习 FPGA 而学习 FPGA 而不知道如何应用 FPGA,那么这是非常悲哀的一件事情,学习 FPGA 只是学习一门技能,而结合自己专业背景选择最合适的解决方案,解决问题才是最终的目的。

序 2:

米联客团队励志在 FPGA 领域可以贡献一份自己的力量, 让 FPGA 从业者入门槛降低、FPGA 应用难度减少, 为中小型 FPGA 团队提供必要有价值的技术资料 and 硬件支持。我们希望能和广大客户形成紧密的合作伙伴关系, 一起创造共赢, 各自实现自己的价值目标。

从 2019 年下半年开始我们米联客团队计划研发全新的 2020 版本教程。经过一年半时间的更新, 2020 版本教程研发进度进入了关键攻坚时期。

2020 版本教程是适应当前 FPGA 技术发展, SOC 技术发展, 新形势下, 米联客做出的战略决策。2020 版本教程需要解决以下几个问题:

- 1)、FPGA 基础课程, 需要解决长期以来没有认真解决好的课程内容, 包括: FPGA 的构架、FPGA 常用 IP 使用、硬件编程经验、通信接口应用、代码规范、时序分析
- 2)、新版本 vitis 软件的使用技巧
- 3)、更多讲解高速通信的基础知识和应用解决方案
- 4)、ZYNQ-SOC 到 ZYNQ-MPSOC 教程统一部署, 达到 90%以上 demo 使用方法一样。不管是学习 ZYNQ-SOC 还是 ZYNQ-MPSOC, 已经学习的 demo 具有 2 个平台之间的互通性。
- 5)、ZYNQ 和 MPSOCLinux 课程做到适合初级入门, 并提供丰富的应用 demo
- 6)、适应未来发展趋势, 增加 FPGA 高层次加速算法编程领域的课程内容
- 7)、教程的构架能够支持 7 系列 FPGA、UltraScale 系列 FPGA、UltraScale+系列 FPGA 和 MPSOC

不管有多麻烦, 不管困难多大, 面对挑战我们坚信胜利!

米联客团队
2021 年 4 月 26 日

序 3:

随着超高清产业的发展,视频,图像的分辨率在不断提高,图像传感器作为成像系统的核心设备,其成像质量至关重要。过去图像传感器主要采用并行 DVP 接口传输图像数据,因其信号完整性上受到限制,难以满足超高清视频在传输速率上的需求。因此,米联客推出了 MIPI 接口摄像头,该接口具有抗干扰性能强,传输速率快等特点,在图像传输领域被广泛使用。

米联客团队

2022 年 10 月 28 日

目录

01 MIPI 简介	2
1.1 MIPI 协议概述	2
1.2 MIPI CSI-2 总体架构	2
1.3 MIPI CSI-2 数据包格式	3
1.4 MIPI CSI-2 接口模型	5
02 实验任务	7
2.1 概述	7
2.2 系统框图	7
03 硬件设计	8
3.1 硬件概述	8
3.2 PL 图形化编程	9
3.3 mipi_csi2_rx_subsystem IP	10
3.4 Demosaic	11
3.5 添加 PIN 约束	12
3.6 编译并导出平台文件	12
04 软件设计	13
4.1 创建 SDK Platform 工程	13
4.2 创建 mipi_5640 APP 测试工程	14
05 方案演示	16
5.1 硬件准备	16
5.2 实验结果	错误！未定义书签。
附录 1：常见问题	17
1 联系方式	错误！未定义书签。
2 售后	错误！未定义书签。
3 销售	错误！未定义书签。
4 在线视频	错误！未定义书签。
5 软件下载	错误！未定义书签。
6 经验分享	错误！未定义书签。
7 官方博文	错误！未定义书签。

01 MIPI 简介

软件版本: vitis2021.1(vivado2021.1)

操作系统: WIN10 64bit

硬件平台: 适用 XILINX A7/K7/Z7/ZU/KU 系列 FPGA

登录“米联客”FPGA 社区-www.uisrc.com 视频课程、答疑解惑!

1.1 MIPI 协议概述

2003 年, MIPI 接口的出现为众多消费者厂商提供了便利, 它将显示屏接口和摄像头等接口标准化, 减少不同手机设计厂商设计过程中出现的接口规格或者数据信号不对应的问题, 从而提高各厂商的设计效率。

其中最常见的就是 MIPI DSI (Display Serial Interface) 接口和 MIPI CSI-2 接口, MIPI DSI 接口是显示屏接口, 它是将 FPGA 作为发送端把数据通过接口传输到显示屏上进行图像显示。MIPI CSI-2 接口对应摄像头接口, 此时相机系统作为发送端, FPGA 作为接收端。目前常见的摄像头接口有 DVP, USB3.0, LVDS 等等, 但是在大部分情况下, MIPI CSI-2 接口才是最合适的选择。下表列出了目前常见的几种摄像头接口参数对比。

接口类型	CSI-2	USB3.0	DVP
传输模式	串行传输	串行传输	并行传输
纠错能力	ECC 与 CRC	无	无
传输带宽	1.5Gbps/Lane	5Gbps	96Mbps
抗干扰能力	200mV 差分信号	400mV 差分信号	较弱
适用摄像头	800W 以上	500W	500W

从上表可以看出, 相比 DVP 接口和 USB3.0 接口, CSI-2 接口在图像数据的传输性能上拥有明显的优势, 其具体表现在:

相比并行传输和 DVP 接口, 串行传输使用的数据线相对较少, 而且一般采用双线差分传输, 外部噪声同时加载到并行传输的两条差分线可以相互抵消, 因此差分传输对外部噪声的抗干扰能力强。同时差分线上的信号总是往相反的方向进行跳变, 可以抵消各自跳变的噪声, 差分传输对内部噪声同样具有很强的抗干扰能力。而当使用并行传输时, 其中的任一信号线跳变时, 都会给临近的信号线带来噪声, 传输的频率越高噪声将会越发明显, 所以串行传输能够达到更高的传输速率。

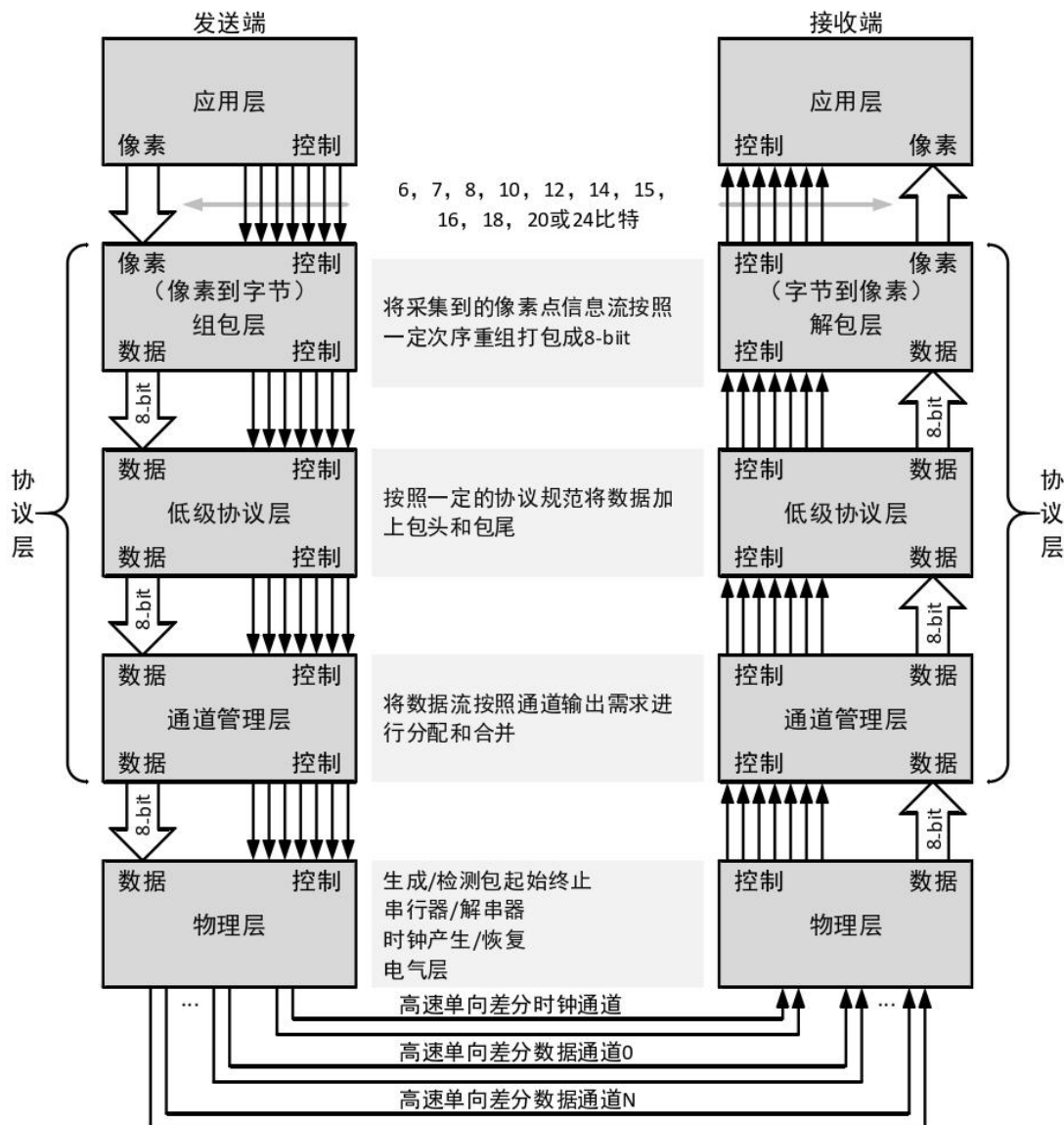
相比串行传输的 USB3.0 接口, CSI-2 接口在数据包的传输过程中加入了 ECC 和 CRC 分别作为数据包头和有效载荷的校验编码。ECC 能够监测 2 bit 以内的数据出错和纠正 1 bit 的数据出错; CRC 能够监测多 bit 的数据出错。加入校验编码后的 CSI-2 接口对数据传输将具备更高容错率。同时 CSI-2 使用 LVDS 技术, 低压幅和低电流驱动的输出使得 CSI-2 接口具备更低的噪声和更低的功耗。

在图像传输的清晰度方面, 由于 CSI-2 接口能够以更高的效率和更快的速度完成对图像数据传输, 因此 CSI-2 所支持的高清摄像头可高达千万级像素以上, 能够完成更高质量的图像传输。

在功耗与传输速率方面, CSI-2 接口支持高速 (HS) 与低功耗 (LP) 两种模式协同工作, 高速模式下使用 0.2V 低压差分电压, 用于传输图像数据, 其传输速度范围在 80Mbps 至 1000Mbps; 低功耗模式下使用 1.2V 差分电压, 用于传输控制命令, 最高传输速度为 10Mbps。接口在正常工作时, 需要在一次高速模式传输后插入低功耗模式。高速模式下低电压摆幅与低功耗下高电压摆幅的相互切换使得 CSI-2 接口既具备了较强的数据传输能力的同时也兼顾了低功耗的需求。

1.2 MIPI CSI-2 总体架构

MIPI CSI-2 协议的整体架构如下图所示:



MIPI CSI-2 协议架构可分为应用层，协议层和物理层。其中，协议层可细分为像素到字节的组包层，字节到像素的解包层，低级协议层和通道管理层。各层之间的定义如下：

应用层：该层主要面向用户，可在应用层中实现对原始图像数据进行高级编码以及处理各种算法。

协议层：由像素与字节的组包/解包层，低级协议层和通道管理层构成。

像素与字节的组包/解包层：在发送端中，组包层负责将来自应用层中包含图片像素信息的数据流打包成为字节，然后输送至低级协议层中；在接收端中，解包层将来自低级协议层中的经发送端打包过后的字节解压并还原出包含图片像素信息的数据流，然后传输至接收端应用层。

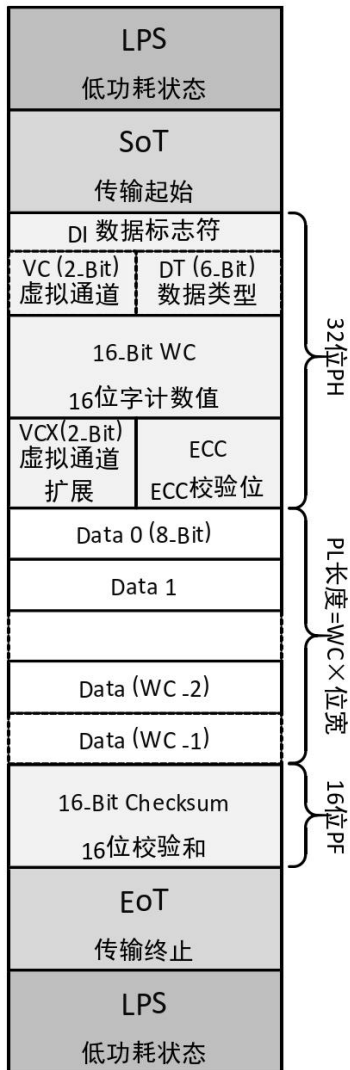
低级协议层：对位于传输起始指令和传输终止指令之间串行传输的数据建立 bit 级和 byte 级同步，并将数据传输至下一层。低级协议层的最小数据粒度为一个字节。此外，低级协议层还包含对大小端字节序分配的控制。

通道管理层：数据通道的数量可以根据应用场景的带宽需求进行选择。在发送端，接口将字节分发到一个或多个通道；在接收端，接口从通道中收集字节并将其合并重组，恢复出原始的数据流。

物理层：规范了传输媒介，电气特征，I/O 电路以及 bit 级和 byte 级的同步机制。

1.3 MIPI CSI-2 数据包格式

用于 CSI-2 中 D-PHY 物理层的数据包可分为长数据包和短数据包，其中长数据包的构成如下图所示：



它由以下三个部分组成：

一、32bit 数据包头（PH）：数据包头由以下部分构成：

8 bit 数据标识符 DI（Data Identifier）：由 2 bit 的虚拟通道号（第 7 位与第 6 位）和 6 bit 数据类型（第 5 位与第 0 位）构成，CSI-2 可以通过不同的虚拟通道号和数据类型来标志不同的数据流）。

16 bit 字计数值 WC（Word Count）：统计长数据包数据域的字节数；在短数据包里，该值可以默认是 0，同时，在有需要的情况下表示是第几帧或者是第几行。

8 bit 的 VCX+ECC 校验位：由 2 bit 的虚拟通道扩展位（第 7 位与第 6 位）和 6 bit 的 ECC 校验位（第 5 位与第 0 位）构成。通过加入 ECC 校验位，使得数据包头中允许前 24 bit（8 bit 数据标志符+16 bit 字计数值）在传输过程中纠正 1 bit 数据出错以及监测 2 bit 数据出错。

二、有效载荷：具有可变数量的 8 bit 数据字的专用数据。

三、16 bit 数据包页脚（PF）：数据包页脚具有一个元素，即 16 bit CRC 校验和，用于指示数据包是否在传输到接收过程中出现数据错误。

短数据包结构如下图所示：



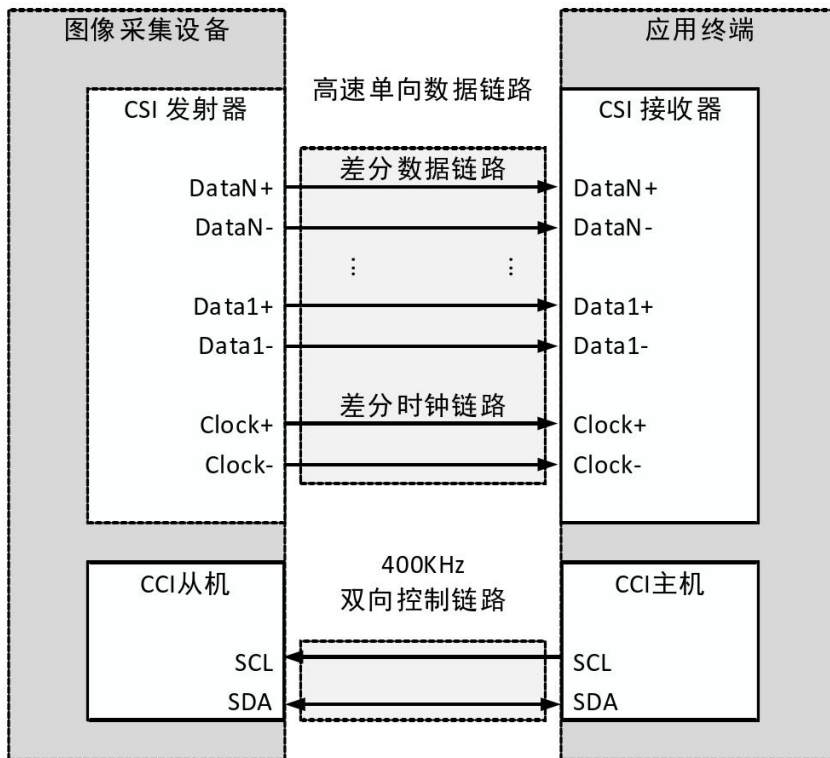
与长数据包头的结构相似，不同之处在于，长数据包头的 16 bit 字计数值 (WC) 字由短数据包数据域字段所替代，短数据包可通过 0x00 到 0x0F 来标识。同时，短数据包只包含有一个数据包头，数据包填充字段和数据包尾均不存在。对于帧同步的数据类型 (DT)，短数据包的数据域字段应为帧号。对于行同步的数据类型，短数据包的数据域字段应为行号。此外，对于短数据包通用的数据类型，其内容可由用户自定义。

1.4 MIPI CSI-2 接口模型

CSI-2 规范定义了发送端和接收端的标准数据传输和控制接口，可选择 D-PHY 或 C-PHY 物理层作为高速串行数据的传输接口选项。

C-PHY 和 D-PHY 在物理连接上存在多处不同，必须保证主机和从机同时使用 C-PHY 和 D-PHY 之间的一种作为物理层才能进行有效地通信。使用 C-PHY 作为数据传输的物理层可以获得更高的传输速率，但是 C-PHY 只支持 CSI-2，而 DPHY 同时支持 CSI-2 和 DSI。

D-PHY 物理层使用一组 2 线的单向差分时钟通道和一组或多组（一组至四组）2 线的单向差分数据通道作为连接图像采集器模块发送端和目标接收端之间的接口。其连接关系如下图所示。其中，摄像机控制接口 (CCI) 是与 I2C 标准兼容的双向控制接口。



发送端在完成对图像的各种处理之后，将按照协议对数据进行打包，通过差分信号线向接收端传输信号，差分信号线一般有一对时钟差分线和多对数据差分线，数据差分信号线的数量与需要传输的数据量的要求有关，对于数据量大的传输场景，使用多对数据线能更容易满足链路的传输要求。

一般情况下，对低于五百万像素的摄像头只需使用两对差分数据信号线，即两条数据通道。而当摄像头像素像素进一步提高到千万级别时则开启四条数据通道同时对数据进行传输，即四对差分数据信号线。

02 实验任务

软件版本: vitis2021.1(vivado2021.1)

操作系统: WIN10 64bit

硬件平台: 适用 XILINX A7/K7/Z7/ZU/KU 系列 FPGA

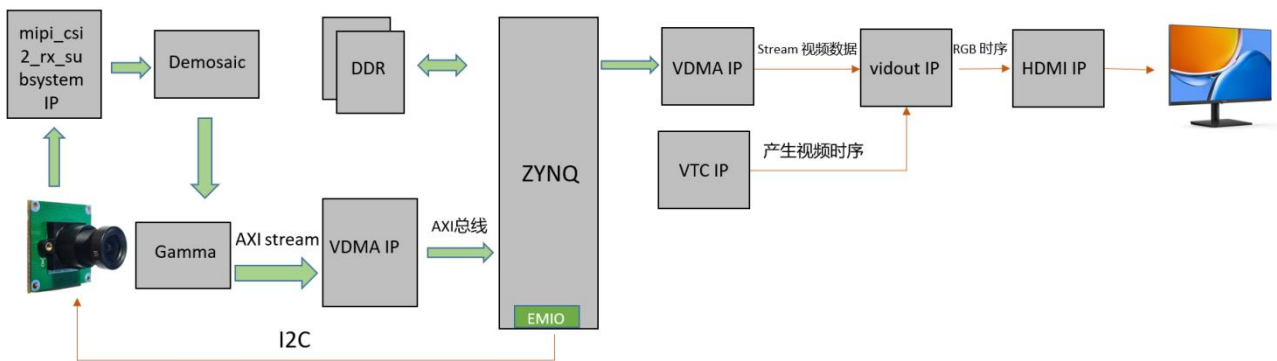
登录“米联客”FPGA 社区-www.uisrc.com 视频课程、答疑解惑!

2.1 概述

本次的实验任务是通过米联客开发板及米联客推出的 MIPI 0V5640 摄像头实现图像采集, 并通过 HDMI 屏实时显示。

2.2 系统框图

本次实验的系统框图如下:



本次实验采用两个 lane 的 MIPI 输入, MIPI 摄像头配置为 RAW10 输出。通过 **mipi_csi2_rx_subsystem** 模块进行协议解析并转换成 AXI-Stream 流数据, 然后通过 **Demosaic** 模块实现去马赛克算法, 将 RAW 格式的数据转换成 RGB 格式的数据, 之后经过 **Gamma** 校正后进入 **VDMA**, 最终在 HDMI 屏幕上显示。

03 硬件设计

软件版本: vitis2021.1(vivado2021.1)

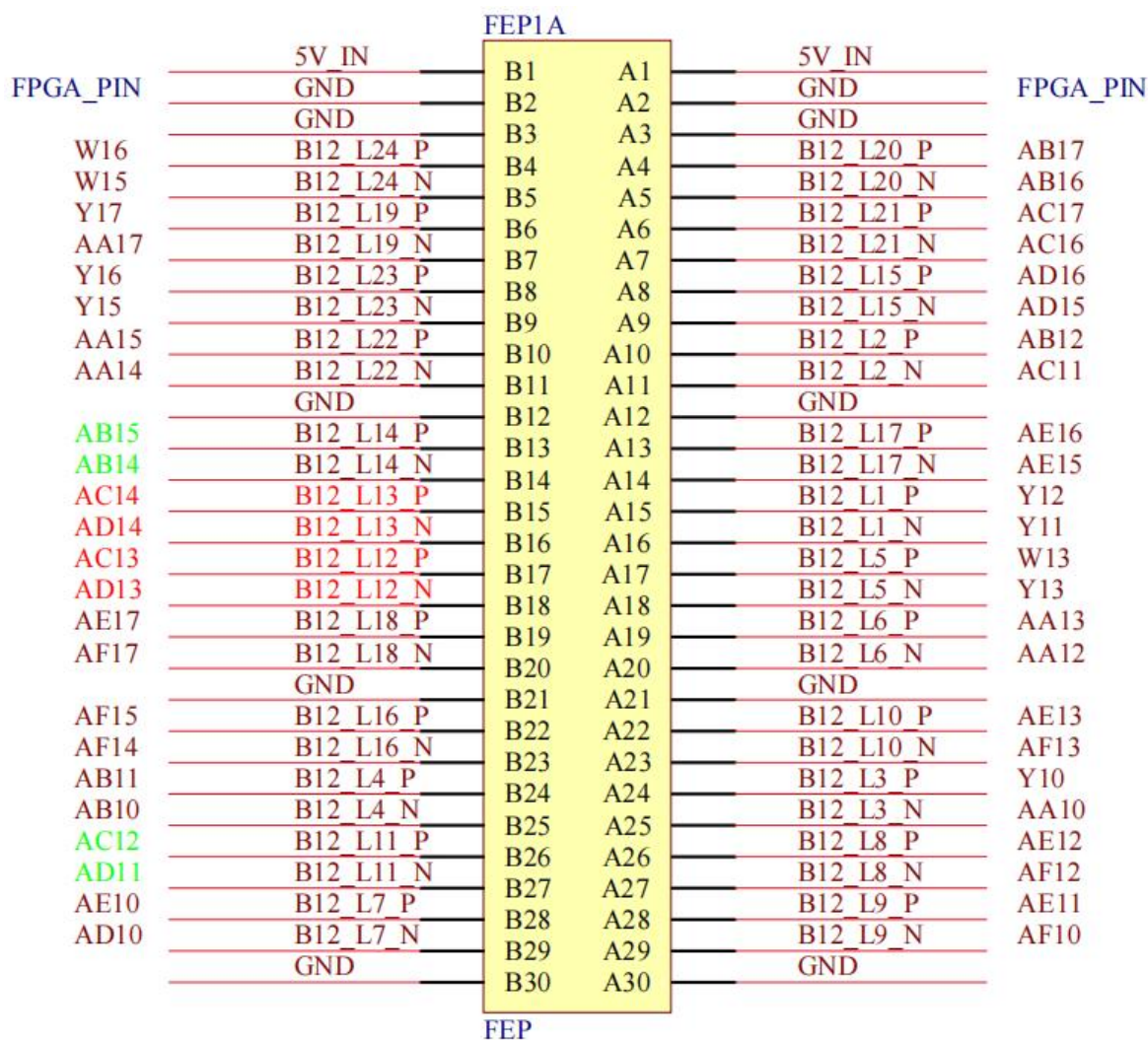
操作系统: WIN10 64bit

硬件平台: 适用 XILINX A7/K7/Z7/ZU/KU 系列 FPGA

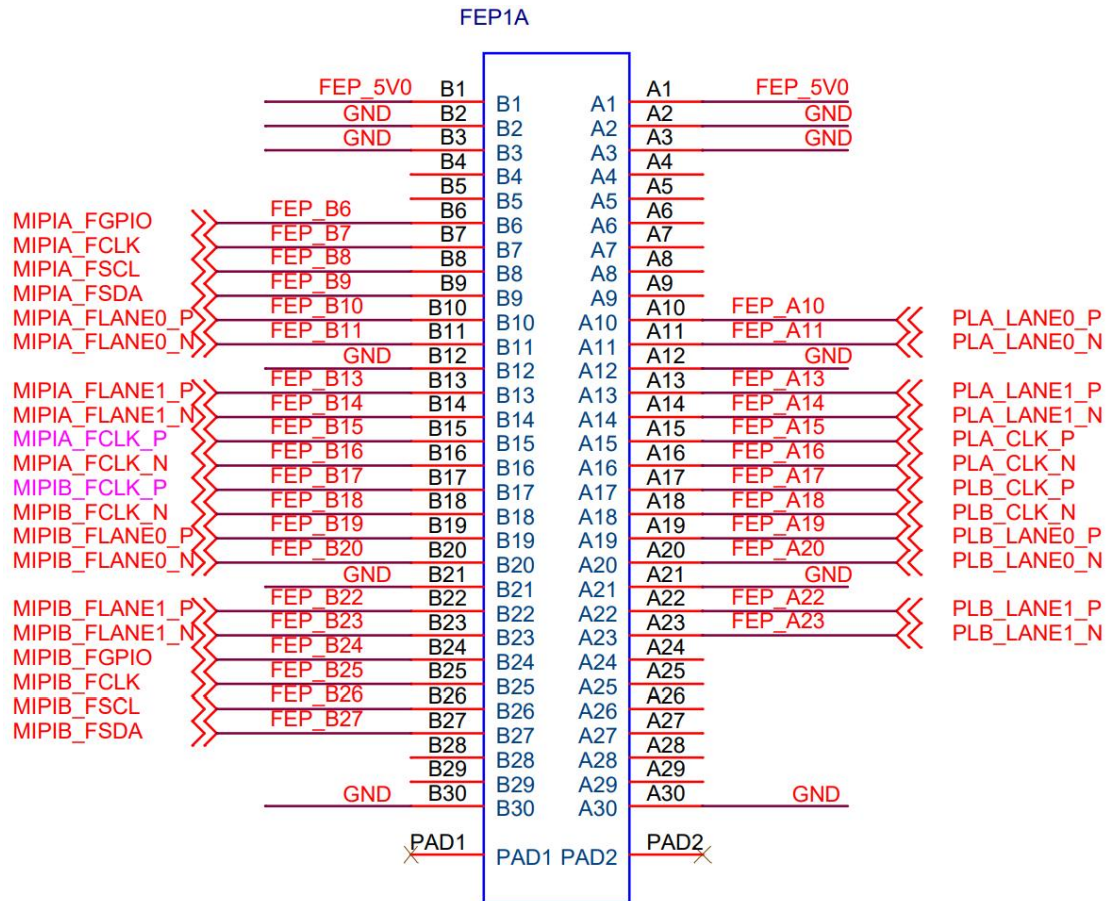
登录“米联客”FPGA 社区-www.uisrc.com 视频课程、答疑解惑!

3.1 硬件概述

H3-7100 开发板上有一个 FEP 高速扩展接口, 该接口可以用来连接 mipi 摄像头等模块。它有 96 个 IO/48 对差分, 管脚定义如下:



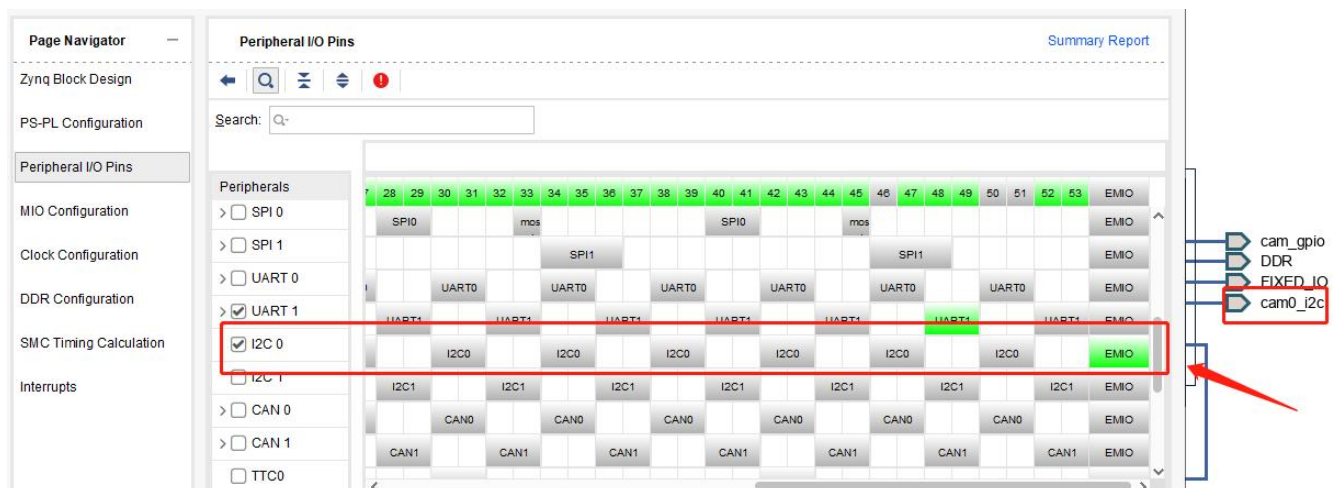
这个 FEP 高速扩展接口可以用来连接我们推出的 MIPI 摄像头子卡, MIPI 摄像头子卡的管脚定义如下图所示:



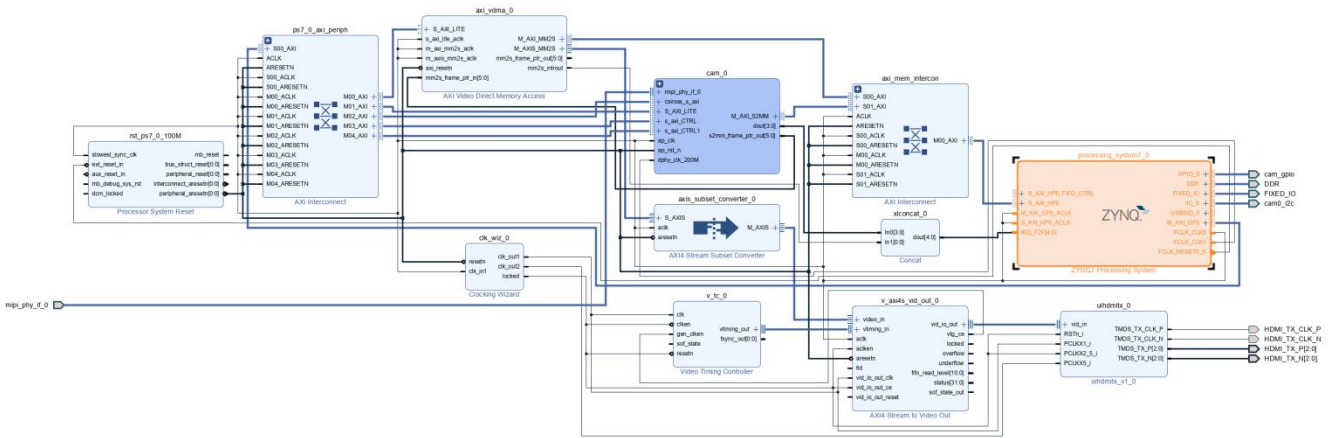
下面我们就使用 vivado 来搭建 SOC 系统工程。本工程搭建过程中使用到了 HLS IP 模块，部分用户使用时会产生 HLS IP 无法编译的问题，关于解决方案，请参考米联客官方论坛帖子《关于 HLS IP 无法编译解决方案》一文，具体网址为 <https://www.uisrc.com/t-3409.html>。

3.2 PL 图形化编程

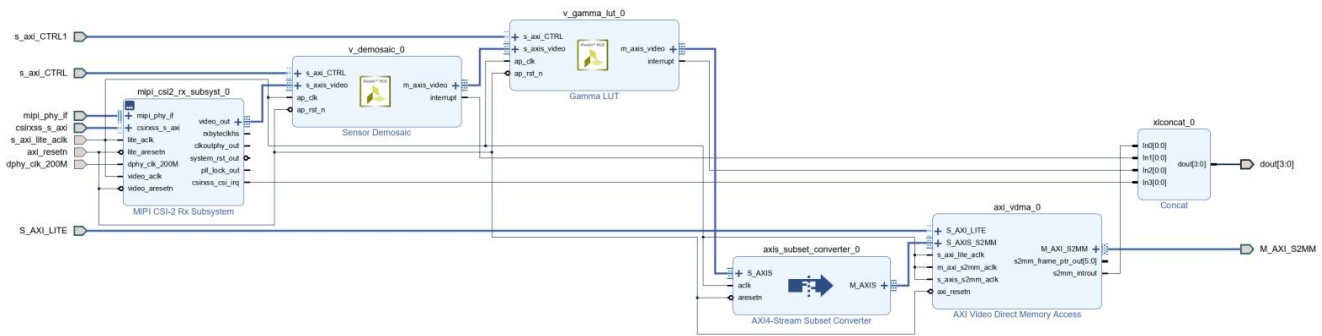
我们采用 EMIO I2C 接口方案初始化摄像头，双击 ZYNQ IP 核，勾选 I2C0 通道，并且把 I2C 的 IO 映射到 EMIO 上，如下图所示：



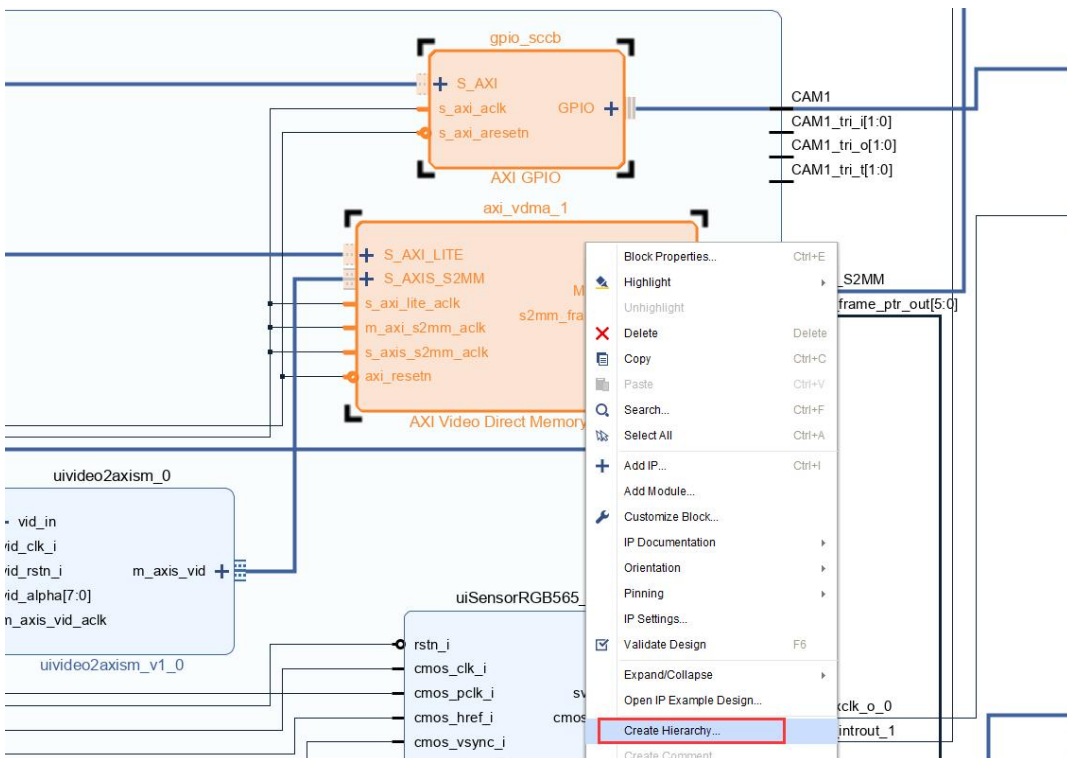
以下为完成后以 EMIO IIC 初始化摄像头的工程



深色的部分是我们对摄像头驱动部分进行的一个层级封装，这样可以让复杂的 BD 图像设计，看起来更加简洁。我们后续设置多摄像头方案时，直接复制这个 cam 封装即可。Cam 封装内 IP 如下



这个技巧如下，选中需要层级封装的 IP，右击选择 Create Hierarchy



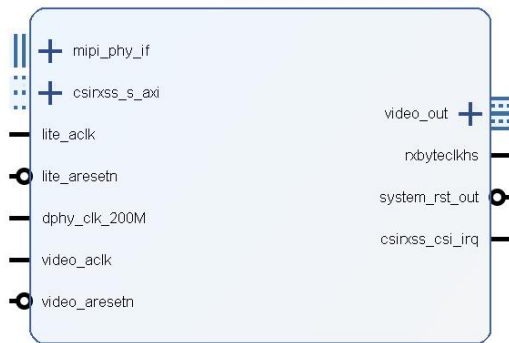
3.3 mipi_csi2_rx_subsystem IP

该模块用于 MIPI 摄像头数据的接收与解析，将 MIPI 摄像头的数据以 axi-stream 协议格式的数据输出。配置如下，数据格式选择 RAW10，选择 2 Lane，Line Rate 配置为 1000Mbps，指的是最大支持的速率，也可以根据自己的需求填写，范围为 80-2500；Pixels Per Clock 默认配置为 1，表示 1 个周期为 1 个像素；

Re-customize IP

MIPI CSI-2 Rx Subsystem (5.1)

Documentation IP Location

☐ Show disabled ports

Component Name cam_0/mipi_csi2_rx_subsys_0

Configuration Application Example Design

Subsystem Options

Pixel Format RAW10 Serial Data Lanes 2

☒ Include Video Format Bridge (VFB)

☐ Support CSI Spec V2_0

DPHY Options

IODELAY_GROUP Name ipl_csi2rx_idly_group

Line Rate (Mbps) 1000 [80 - 1500]

☒ Linerate supported by Device Datasheet

☒ D-PHY Register Interface

☐ Enable HS and ESC Timeout Counters/Registers

Calibration Mode

☒ NONE ☐ FIXED ☐ AUTO

IDELAY Tap Value 1 [1 - 31]

CSI-2 Options

☒ CSI2 Controller Register Interface

☐ Embedded non-image Interface

☐ Filter User Defined data types

Line Buffer Depth 2048

VFB Options

Allowed VC All Pixels Per Clock 1

TUSER Width 1

Resource improvement options

☒ Enable CRC ☐ Enable Active Lanes

3.4 Demosaic

该模块通过去马赛克算法，将 RAW10 格式的数据转换成 RGB 格式的数据，模块的配置如下图所示：

Re-customize IP

Sensor Demosaic (1.1)

Documentation IP Location

☐ Show disabled ports

Component Name cam_0/v_demosaic_0

Samples per Clock 1

Maximum Data Width 10

Maximum Number of Columns 1920 [64 - 8192]

Maximum Number of Rows 1080 [64 - 4320]

Interpolation Method High Resolution Interpolation

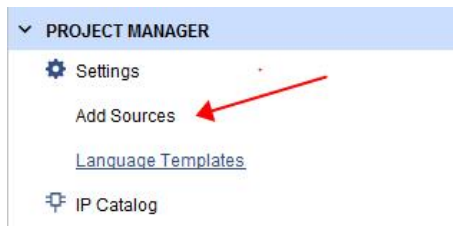
☒ Horizontal Zipper Artifact Removal

OK

Cancel

3.5 添加 PIN 约束

1: 选中 PROJECT MANAGER→ Add Sources→Add or create constraints, 添加 XDC 约束文件。



2: 打开提供例程，复制约束文件中的管脚约束到 XDC 文件，或者查看原理图，自行添加管脚约束，并保存。
配套工程路径下已经提供的 pin 脚文件。

3.6 编译并导出平台文件

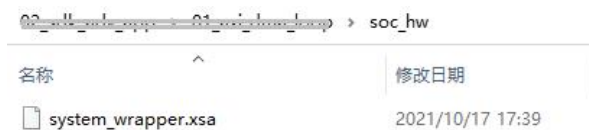
1:单击 Block 文件→右键→Generate the Output Products→Global→Generate。

2:单击 Block 文件→右键→ Create a HDL wrapper(生成 HDL 顶层文件)→Let vivado manager wrapper and auto-update(自动更新)。

3:生成 Bit 文件。

4:导出到硬件: File→Export Hardware→Include bitstream

5:导出完成后，对应工程路径的 soc_hw 路径下有硬件平台文件: system_wrapper.xsa 的文件。根据硬件平台文件 system_wrapper.xsa 来创建需要 Platform 平台。



04 软件设计

软件版本: vitis2021.1(vivado2021.1)

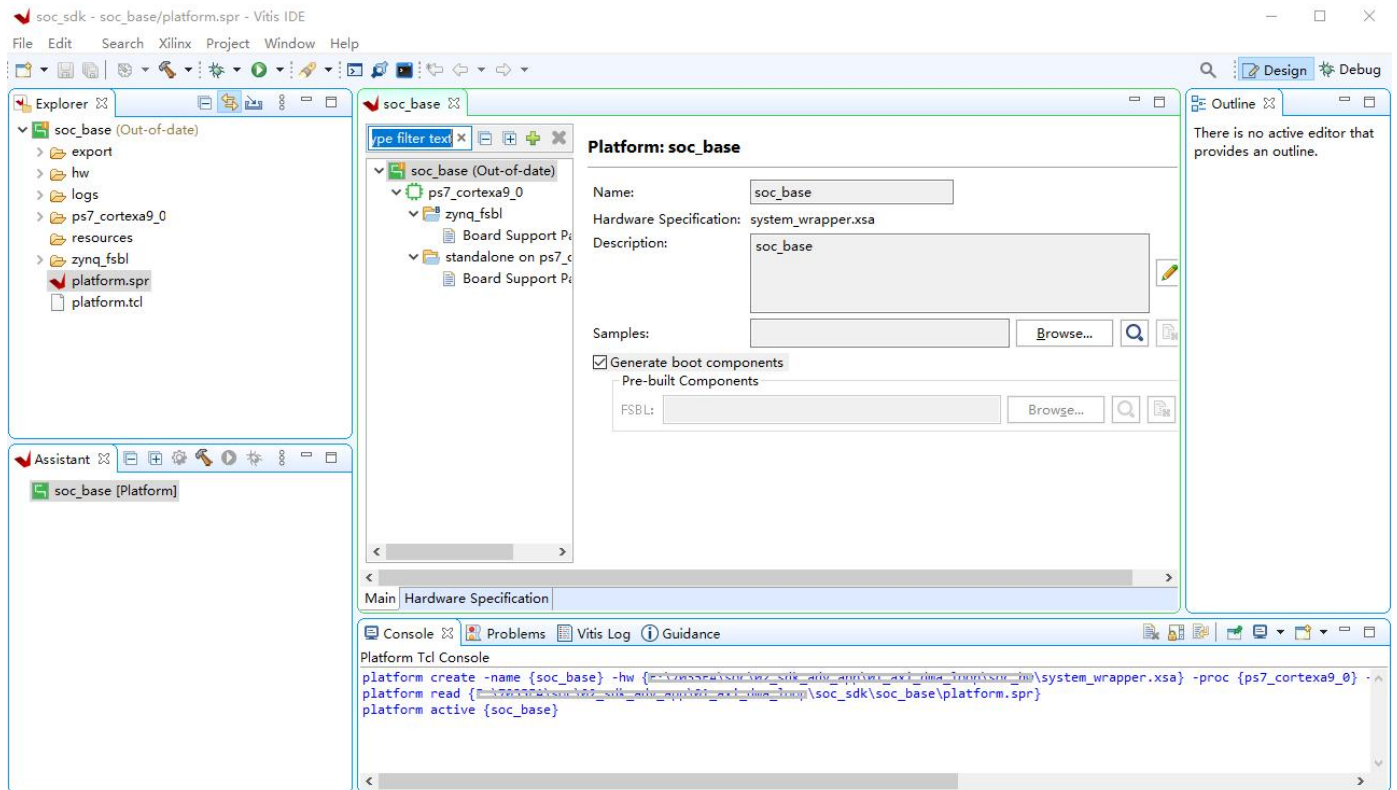
操作系统: WIN10 64bit

硬件平台: 适用 XILINX A7/K7/Z7/ZU/KU 系列 FPGA

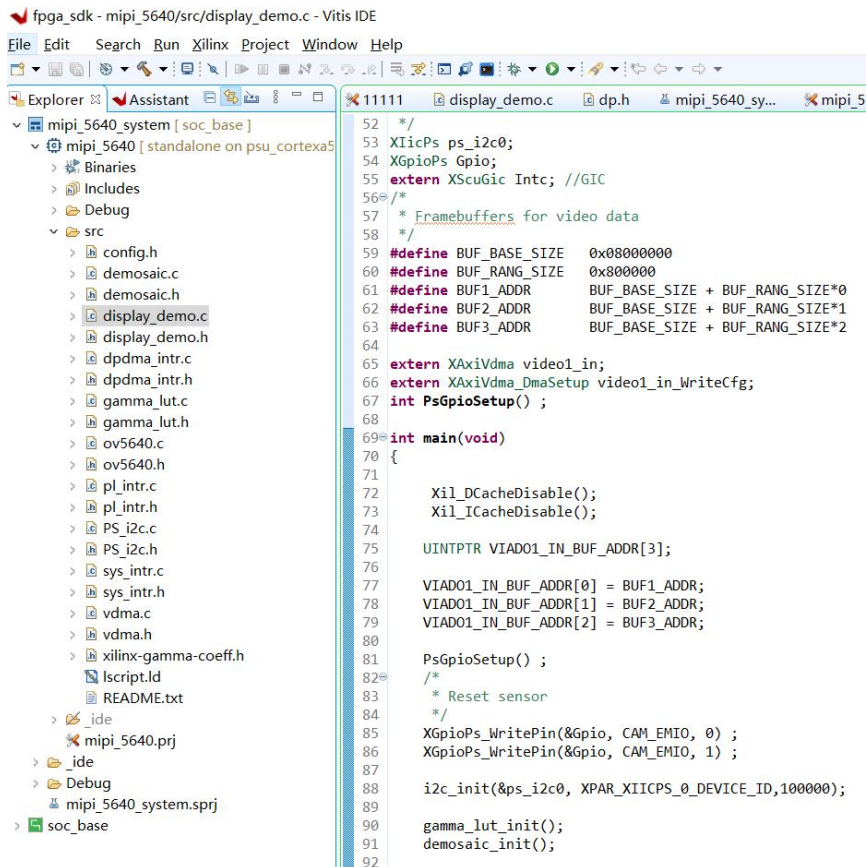
登录“米联客”FPGA 社区-www.uisrc.com 视频课程、答疑解惑!

4.1 创建 SDK Platform 工程

创建 soc_base sdk platform 和 APP 工程的过程不再重复, 如果不清楚请参考《3-2-01 米联客 2022 版 Zynq MPSOC SDK 入门篇》第一个 demo。



4.2 创建 mipi_5640 APP 测试工程



关于 Vitis 程序也较为简单，在 VDMA 的基础上，添加摄像头的初始化，VDMA 的配置，前面要进行摄像头的复位，以及 I2C 的初始化。

配置 MIPI 摄像头，以及启动摄像头的 VDMA

```
/* Start Sensor Vdma */
video1_in_WriteCfg.FrameStoreStartAddr[0] = VIADO1_IN_BUF_ADDR[0];
video1_in_WriteCfg.FrameStoreStartAddr[1] = VIADO1_IN_BUF_ADDR[1];
video1_in_WriteCfg.FrameStoreStartAddr[2] = VIADO1_IN_BUF_ADDR[2];

Video1_S2MMSetup(CAM1_AXI_VDMA_ID, &video1_in, video1_in_WriteCfg, 720, 1280*4, 1280*4);

init_intr_sys();

XAXiVdma_DmaStart(&video1_in, XAXIVDMA_WRITE);
/*
 * Initialize Sensor
 */
sensor_init(&ps_i2c0);
```

我们需要在写入 BUFF 的数据同步读出到相应的 VDMA 中，代码如下

```
3  * Initialize Sensor
1  */
2  sensor_init(&ps_i2c0);
3  /* Start Sensor Vdma */
4  video1_in_WriteCfg.FrameStoreStartAddr[0] = VIADO1_IN_BUF_ADDR[0];
5  video1_in_WriteCfg.FrameStoreStartAddr[1] = VIADO1_IN_BUF_ADDR[1];
6  video1_in_WriteCfg.FrameStoreStartAddr[2] = VIADO1_IN_BUF_ADDR[2];
7
8  video_out_ReadCfg.FrameStoreStartAddr[0] = VIADO_OUT_BUF_ADDR[2];
9  video_out_ReadCfg.FrameStoreStartAddr[1] = VIADO_OUT_BUF_ADDR[0];
10 video_out_ReadCfg.FrameStoreStartAddr[2] = VIADO_OUT_BUF_ADDR[1];
11
12 Video1_S2MMSetup(CAM1_AXI_VDMA_ID, &video1_in, video1_in_WriteCfg, 720, 1280*4, 1280*4);
13 Video_Out_MM2SSetup(VIDEO_OUT_VDMA_ID, &video_out, video_out_ReadCfg, 720, 1280*4, 1280*4);
14 XAXiVdma_DmaStart(&video1_in, XAXIVDMA_WRITE);
15 XAXiVdma_DmaStart(&video_out, XAXIVDMA_READ);
16
17 return XST_SUCCESS;
18 }
```

我们对输入的摄像头数据进行了三帧缓存，并且对输出 HDMI 数据进行了三帧图像缓存。很好的解决了单帧缓存图像撕裂的问题。

VDMA 模式我们采用的是 Circular 模式,VDMA 可以自动完成帧循环切换,同时 VertSizeInput ,HoriSizeInput 参数设置了垂直与水平像素的大小,如下图所示。

```
int vdma_read_init(short DeviceID,short HoriSizeInput,short VertSizeInput,short Stride,unsigned int FrameStoreStartAddr)
{
    XAxiVdma Vdma;
    XAxiVdma_Config *Config;
    XAxiVdma_DmaSetup ReadCfg;
    int Status;

    Config = XAxiVdma_LookupConfig(DeviceID);
    if (NULL == Config) {
        xil_printf("XAxiVdma_LookupConfig failure\r\n");
        return XST_FAILURE;
    }

    Status = XAxiVdma_CfgInitialize(&Vdma, Config, Config->BaseAddress);
    if (Status != XST_SUCCESS) {
        xil_printf("XAxiVdma_CfgInitialize failure\r\n");
        return XST_FAILURE;
    }

    ReadCfg.EnableCircularBuf = 1;
    ReadCfg.EnableFrameCounter = 0;
    ReadCfg.FixedFrameStoreAddr = 0;

    ReadCfg.EnableSync = 1;
    ReadCfg.PointNum = 1;

    ReadCfg.FrameDelay = 0;

    ReadCfg.VertSizeInput = VertSizeInput;
    ReadCfg.HoriSizeInput = HoriSizeInput;
    ReadCfg.Stride = Stride;

    Status = XAxiVdma_DmaConfig(&Vdma, XAXIVDMA_READ, &ReadCfg);
    if (Status != XST_SUCCESS) {
        xdbg_printf(XDBG_DEBUG_ERROR,
            "Read channel config failed %d\r\n", Status);

        return XST_FAILURE;
    }
}
```

由于本教程重点是针对 MIPI 子卡的运用,所以关于 VDMA IP 的仅仅是简单介绍一下,如果感兴趣的用户,可以自行《米联客 2022 版 ZynqSocSDK 高级篇》“附录 2”中对各个 IP 的用法有详细的描述。

05 方案演示

软件版本: vitis2021.1(vivado2021.1)

操作系统: WIN10 64bit

硬件平台: 适用 XILINX A7/K7/Z7/ZU/KU 系列 FPGA

登录“米联客”FPGA 社区-www.uisrc.com 视频课程、答疑解惑!

5.1 硬件准备

本实验需要用到 JTAG 下载器、USB 转串口外设, 另外需要把核心板上的 2P 模式开关设置到 JTAG 模式, 即 ON ON(注意新版本的米联客 ZYNQ 系列板卡支持 JTAG 模式, 对于老版本的核心板, JTAG 调试的时候一定要拔掉 TF 卡, 并且设置模式开关为 OFF OFF)

注: 使用 MIPI 子卡的时候, 开发板 FEP 电压需要调整为 1.8V。如果未调整, 请勿随意使用。不清楚请联系客服咨询。



5.2 实验结果



附录 1：常见问题

1 联系方式

技术交流群网址：<https://www.uisrc.com/f-380.html> 查看最新可以加入的 QQ 群

技术微信：18951232035

技术电话：18951232035

官方微信公众号（新微信公众号）：



2 售后

1、7 天无理由退货 (人为原因除外)

2、质保期限：本公司产品自快递签收之日起，提供一年质保服务 (主芯片，比如 FPGA 或者 CPU 等除外)。

3、维修换货，需提供淘宝订单编号或合同编号，联系销售/技术支持安排退回事宜。

售后维修请登录工单系统：https://www.uisrc.com/plugin.php?id=x7ree_service

4、以下情形不属于质保范畴。

A: 由于用户使用不当造成板子的损坏：比如电压过高造成的开发板短路，自行焊接造成的焊盘脱落、铜线起皮 等

B: 用户日常维护不当造成板子的损坏：比如放置不当导致线路板腐蚀、基板出现裂纹等

5、质保范畴外（上方第 4 条）及质保期限以外的产品，本公司提供有偿维修服务。维修仅收取器件材料成本，往返运费全部由客户承担。

6、寄回地址，登录网页获取最新的售后地址：<https://www.uisrc.com/t-1982.html>

3 销售

天猫米联客旗舰店：<https://milianke.tmall.com>

京东米联客旗舰店：<https://milianke.jd.com/>

FPGA|SOC 生态店：<https://milianke.taobao.com>

销售电话：18921033576

常州溧阳总部：常州溧阳市中关村吴潭渡路雅创高科制造谷 10-1 幢楼

4 在线视频

<https://www.uisrc.com/video.html>

5 软件下载

<https://www.uisrc.com/f-download.html>