Smart Glass Design Document

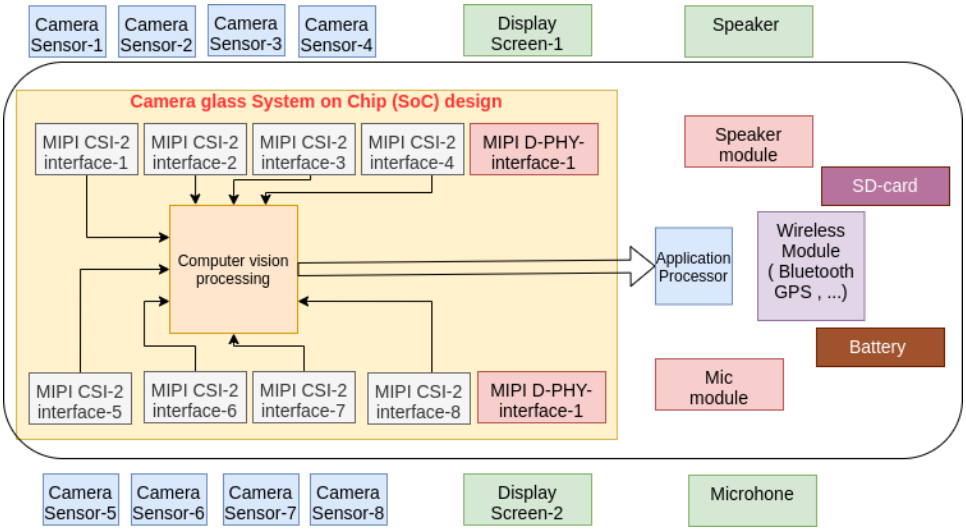
December 08, 2021

revision	Author	Data	Note
1.0	Mohammed Eladawy	08-12-2021	Initial Design Document

Purpose

The purpose of this document is to demonstrate the functionality, interfacing and implementation details of the smart glass project. The smart glass consisting of 8 cameras are used for stereo vision applications.

The Block diagram for the design is shown on figures as follows: -



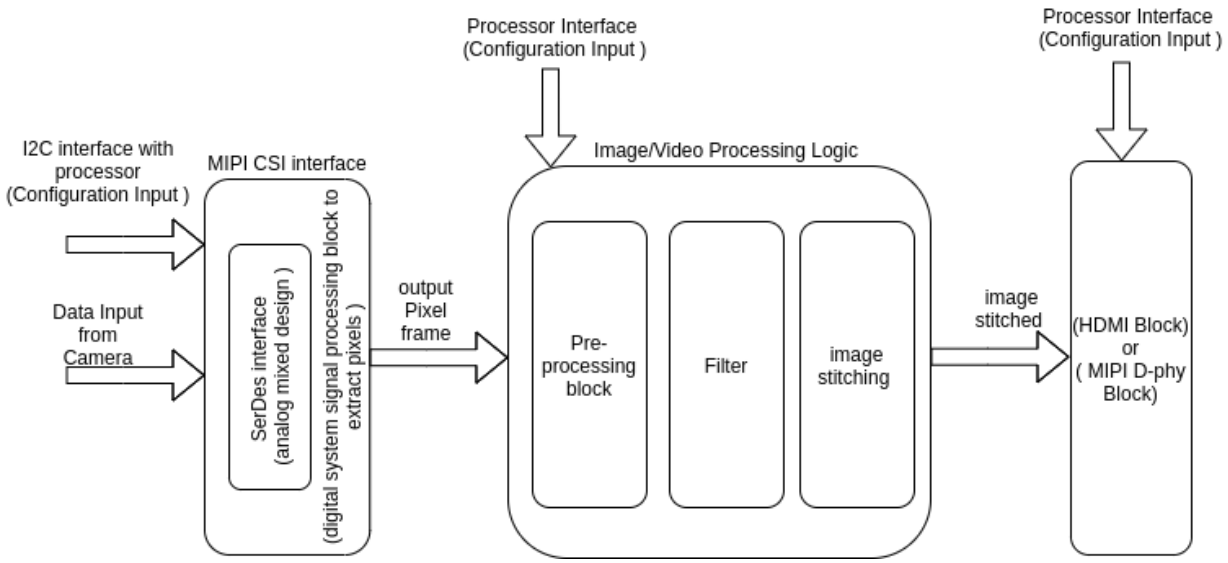
As shown in figure-1, the smart glass design components are divided into two categories: -

1- The outside components interface with smart glass camera

2- The internal hardware/software design to support the required functionality.

This document illustrates the internal hardware/software design requirements

Camera glass System on Chip (SoC) design



The main goal of project is to develop a SoC using RTL hardware language. The architecture of SoC divided into three main blocks.

Block-1: - MIPI CSI interface block, it is responsibility is receiving video/image stream and producing the output pixel frame.

Block 2: - Image / Video Processing logic, it is responsibility is decoding the received frame with required image/video processing.

Block 3: - HDMI or MIPI D-Phy block, it is responsibility is displaying the processed video/image on screen.

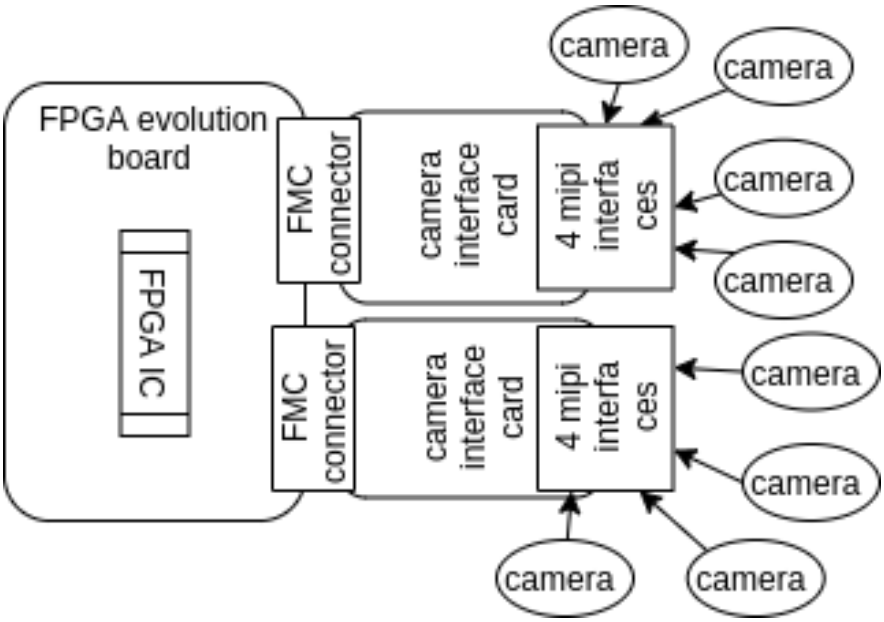
The final target of smart glass project is developing Camera glass SoC using ASIC design flow.

Design plans

The project is divided into two design plans. The first plan is developing FPGA demo oand the second plan is developing the RTL code blocks to support different image signal processing.

FPGA Demo

The FPGA demo is hardware integration of camera, Camra interface kit and Xilinx FPGA evolution board as shown in figure 2.



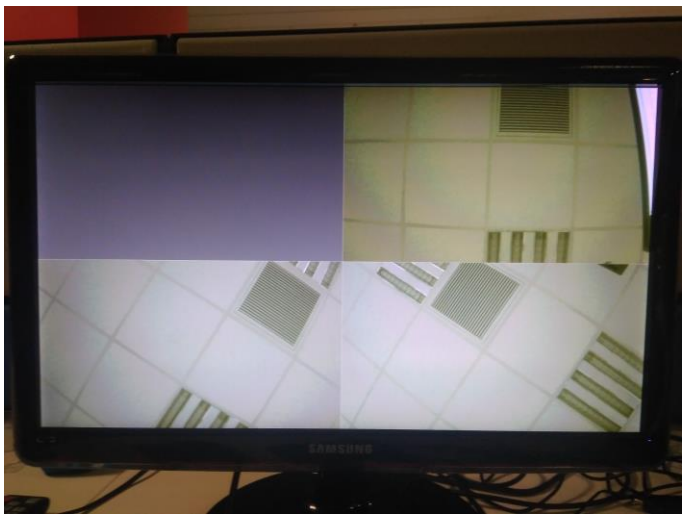
The suggested hardware components are show in the following figure.

FPGA evolution board	Zynq UltraScale+ MPSoC ZCU102 Evaluation Kit	https://www.xilinx.com/products/boards-and-kits/ek-u1-zcu102-g.html#hardware
	Zynq UltraScale+ MPSoC ZCU106 Evaluation Kit	https://www.xilinx.com/products/boards-and-kits/zcu106.html#hardware
	Xilinx Zynq-7000 SoC ZC706 Evaluation Kit	https://www.xilinx.com/products/boards-and-kits/ek-z7-zc706-g.html#hardware
Camra interface card	FMC Pcam Adapter	https://digilent.com/shop/fmc-pcam-adapter/
Camera	Sony-IMX219	https://www.arducam.com/product/arducam-raspberry-pi-camera-v2-8mp-ixm219-b0103/

Camera glass System on Chip (SoC) design on FPGA

The SoC design on FPGA is divided into three plan phases.

Phase-1: - implement SoC to display the output of 8 camera on the screen as shown in figure.



In phase 1 , Digilent used ZED evolution board which is contain FPGA based on ZYNQ 7000.

The Hardware connection is shown in following link.

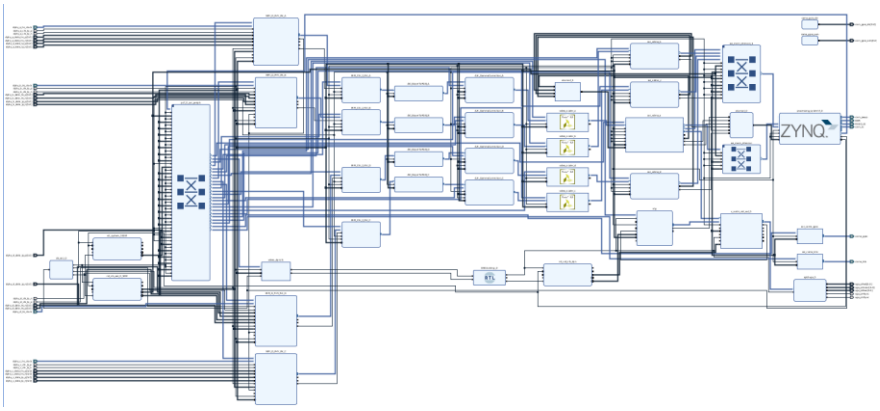
https://digilent.com/reference/learn/programmable-logic/tutorials/zedboard-fmc-pcam-adapter-demo/start

I developed the SoC on xilinx FPGA using FPGA IP with IP from Digilanet.

I started developing SoC on FPGA starting from SoC implemented on the following link

https://github.com/Digilent/ZedBoard-FMC-Pcam-Adapter-DEMO?_ga=2.83766494.918358314.1638359103-622627578.1633875439

The SoC on Xilinx FPGA for ZYNQ 7000 is shown on the following figure



As the suggested FPGA board is new FPGA series (Zynq UltraScale+ MPSoC ZCU102 Evaluation Kit), I started modification on SoC design to be Implemed on new FPGA series.

The following modifications are done: -

Modification	Status
1- replace ZYNQ with MPsoC	Replaced successfully
2- replace “mipi c-phy" IP from Digilanet with mipi c-phy xilinx IP	Replaced successfully
3- update “AXI_BayerToRGB” to be implanted on UltraScale	Updated successfully
4- update “AXI_GammaCorrection” to be implanted on UltraScale	Updated successfully
5- update video_scaler to be implanted on UltraScale	Updated successfully
6- Replace VGA out interface with HDMI	Replaced successfully
7- Integrate modifications from 1 to 6 together	Integrated successfully

The above modifications are run on Xilinx FPGA design flow as following

Run folw	Status
1-compilation	finished successfully
2- synthesis	Finished successfully
3- Implementation	Finished successfully
4- bit stream generation	Failed
5- software implantation using SDK	Suspended
6- test ion FPGA	Suspended

I spent time to figure out why bit stream generation is failed, I found answer on the following link

https://forum.digilentinc.com/topic/20170-fmc-pcam-adapter-features/

There are two reasons for failing of bit stream generation: -

1- The current version of “FMC Pcam Adapter” adaptor from Digilanet don’t support Zynq UltraScale+ board, because circuit implementation of FMC connector with Zynq UltraScale+ MPSoC ZCU102. There is conflict on pin connection between MIPI c-phy xilinx IP and FMC connection.

(Digilanet mention that the new vesriob will support UltraScale+ board)

2- The current version of “FMC Pcam Adapter” adaptor is limited support

IP developing (RTL coding for image processing library)

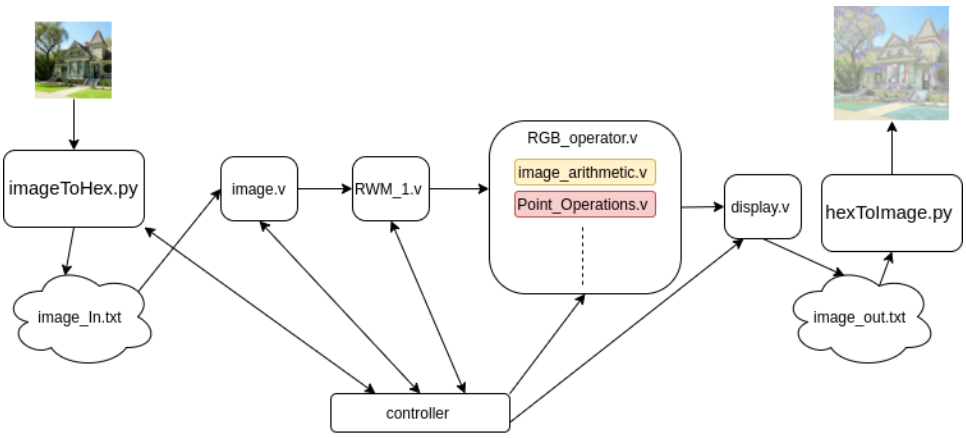
The goal of this design step is developing RTL code for image processing library. This library should support different image processing operations as shown in the following link.

https://homepages.inf.ed.ac.uk/rbf/HIPR2/wksheets.htm

The following table show the required IP.

IP name	Description	Status
imageToHex.py	Python code to read image and convert it to pixel	successful
hexToImage.py	Python code to convert pixel to image	successful
image.v	Verilog code to read pixel and generate required signal for controlling. and produce the 3 components of image (Red, green, blue)	successful
RWM_1.v	store the RGB pixel bytes coming from the camera to internal register.	successful
RGB_operator.v	Perform the required operation of each color pixel (R,G,B respectively)	successful
image_arithmetic.v	Perform the arithmetic operation on each pixel (the supported arithmetic operation are addition, subtraction, multiplication, division by 2 ^n, and , or , not)	successful
display.v	Verilog code to store pixel in file to be converted to image	successful
Controller.v	Control the operation for all moules	successful

The following figure show how different module connected together



Suggested design and design-planning.

Image processing IP modules.

The next design phase for image processing library should include design the following IPs

IP name	Description
Point Operations	functions applied to individual pixels
Geometric Operations	image rotation, translation and scaling
Image Analysis	labeling image pixels
Morphology	pixel shape-based analysis
Digital Filters	noise reduction and other enhancement filters
Feature Detectors	edges and others feature
Image Transforms	Fourier, Hough and other transforms
Image Synthesis	noise image data

Also, in other phase of design the following

IP name	Description
mipi c-phy	Module interface with camera/video and generate the required pixels
Mipi D-phy	Module interface with screen to diplay image/video

Demo on FPGA

The suggested design plan for demo as follows

IP name	HW Comp.	Description
Phase 1: - display 8 camera image/video on TV screen.	FPGA kit+ pcam card + camera + TV screen	In this design phase, we will use interface PCAM to show the output of 8 camera on TV screen through HDMI port. In this phase we will use IP devloped by Xilinx.
Phase-2: - display **stitched** camera image/video on TV screen.	FPGA kit+ + pam card + camera + TV screen	In this design phase, we will use interface PCAM to show the stitched output of 8 camera on TV screen through HDMI port (output image/video after performing image processing on it)
Phase-3: - display **stitched** camera image/video on smart glass display	FPGA kit+ pam card + camera + smart glass	In this design phase, we will use interface PCAM to show the stitched output of 8 camera on smart glass screen through (output image/video after performing image processing on it)

Demo on smart glass

IP name	HW Comp.	Description
Phase 1: - design circuit for smart glass using FPGA IC	Circuit developed using FPGA	Design circuit for receiving image/video from 8 camera and display output to on screen
Phase-2: - display **stitched** camera image/video on camera glass screen	Circuit developed using FPGA + smart glass screen	In this design phase, we