logiADAK-VDF-ZU-SDSoC

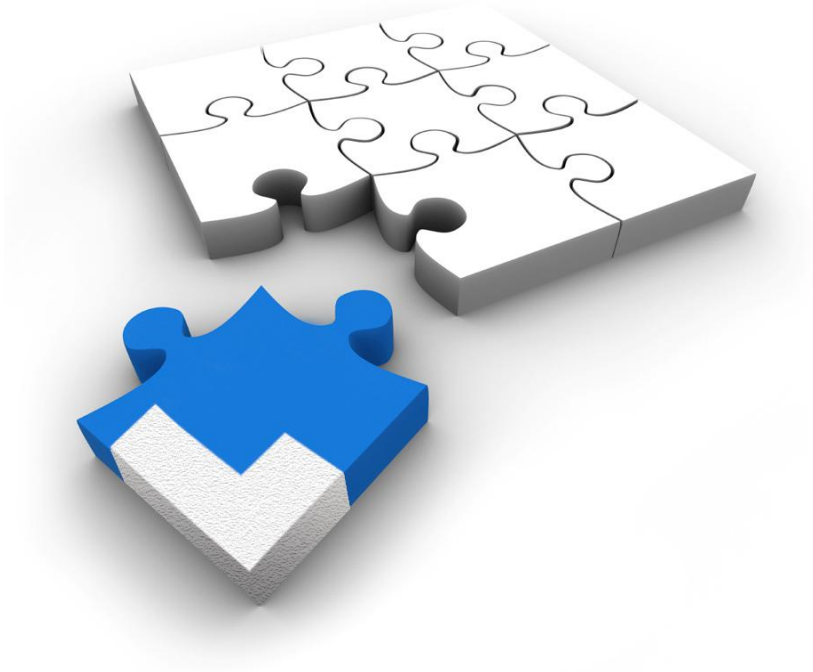
Video Design Framework - Reference Designs for Xylon logiVID-ZU Vision Development Kit

Prepared for the Xilinx® SDSoC™ Development Environment

User‘s Manual

Version: 2.00.a

logiADAK_VDF_ZU_SDSoC_v2.00.a.docx



1 ABOUT THE FRAMEWORK

The logiADAK-VDF-ZU-SDSoC Video Design Framework enables Xylon logiVID-ZU Vision Development Kit users to quickly utilize the provided hardware platform for their own development of the Xilinx® All Programmable Zynq® UltraScale+™ MPSoC based embedded multi-camera vision systems.

The framework includes pre-verified logicBRICKS reference designs for video capture from Xylon video cameras with the TI® FPD-Link III high-speed digital video interface and the HDMI video input, and the display output under the Linux operating system. Reference designs are prepared for both, hardware-centric Vivado® Design Suite and software-centric SDSoC™ Development Environment. This document describes the designs prepared for the Xilinx SDSoC Development Environment, and the Vivado compatible designs are described in the logiADAK_VDF_ZU document.

The complete camera-to-display MPSoC designs, which are compact and use just a fraction of available programmable logic, significantly save the design time. Instead of starting from scratch and having to spend months designing and building a new design framework, logiADAK-VDF-ZU-SDSoC design framework users can immediately focus on specific vision-based parts of their next MPSoC design. The logiVID-ZU hardware platform can be installed on test vehicles (cars, robots...) and used in exhaustive tests, i.e. for testing of the new ADAS developments in the test vehicle and under different road conditions.

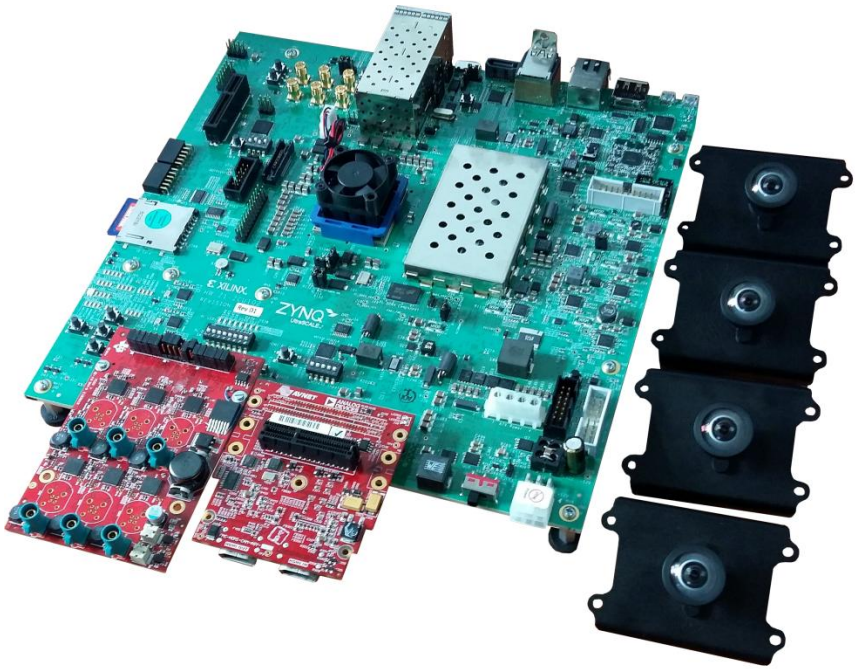


Figure 1: Xylon logiVID-ZU Vision Development Kit

logiADAK-VDF-ZU-SDSoC reference designs include Xylon logicBRICKS IP cores and hardware design files prepared for Xilinx SDSoC Development Environment. To provide the SDSoC users the complete embedded C/C++ development experience, the supplied SDSoC reference designs include

the Sobel video filter example. This example shows kit users how to integrate their own vision processing logic between video input and video output IP cores, and how to implement it in software or in programmable logic.

The Linux OS and software drivers for logicBRICKS IP cores enable software developers to efficiently work with the framework, without knowing the hardware implementation details.

The logiADAK-VDF-ZU-SDSoC Video Design Framework is functionally equal to Xylon's logiADAK-VDF-ZU Video Design Framework, which is fully implemented in the Xilinx Vivado Design Suite.

1.1 Programmable Logic Utilization

The logiADAK-VDF-ZU-SDSoC reference designs utilize just small fractions (Table 3) of available programmable logic resources in the Xilinx Zynq UltraScale+ MPSoC XCZU9EG device. Free resources can be utilized by users who can also alter the pre-defined logicBRICKS configurations and change the programmable logic utilization.

Table 1: CAM-HDMI Reference Design Programmable Logic Utilization

Family (Device)	F (MHz)			LUT[1]	FF[1]	IOB[2]	BRAM	MULT/ DSP48/E	DCM / CMT	BUFG	GTx	Design Tools
	mclk[4]	vclk[4]	rclk									
Zynq US+[3] (XCZU9EG-2)	(150/100)	(150/100)	100	5400	6129	54	11	13	1	7	0	Vivado 2017.1

Table 2: FOUR-CAM Reference Design Programmable Logic Utilization

Family (Device)	F (MHz)			LUT[1]	FF[1]	IOB[2]	BRAM	MULT/ DSP48/E	DCM / CMT	BUFG	GTx	Design Tools
	mclk[4]	vclk[4]	rclk									
Zynq US+[3] (XCZU9EG-2)	(150/100)	(150/100)	100	13835	13725	70	65	40	1	10	0	Vivado 2017.1

Notes:

1) Assuming the following configuration: ITU656,YUV output, 32-bit AXI4-Lite register interface, 64-bit AXI4 memory interface with max. burst size of 64 words, scaling in both directions with multipliers (DSP48s), output stride set to 2048 pixels
2) Assuming only video inputs are routed off-chip, register and memory interfaces are connected internally
3) Only burst size of 16 words is supported on HP ports in the Xilinx Zynq US+ MPSoC
4) logiCVC/logiWIN clock frequencies

Table 3: Free Programmable Logic Resources

	Available in XCZU9EG	Used Resources	
		CAM-HDMI	FOUR-CAM
Look-Up Tables (LUTs)	274,080	~ 2%	~ 5%
Flip Flops (FFs)	548,160	~ 1%	~ 3%
Block RAM (36 kB BRAM)	912	~ 1%	~ 7%
DSP slices (MULT/DSP)	2,520	~ 1%	~ 2%

1.2 About Xilinx SDSoC Development Environment

The Xilinx SDSoC development environment is a member of the Xilinx SDx™ family that provides a greatly simplified ASSP-like C/C++ programming experience including an easy to use Eclipse IDE and a comprehensive design environment for heterogeneous Zynq® All Programmable SoC and MPSoC deployment.

Complete with the industry's first C/C++ full-system optimizing compiler, SDSoC delivers system level profiling, automated software acceleration in programmable logic, automated system connectivity generation, and libraries to speed programming.

To access the capabilities of the SDSoC, please visit www.xilinx.com/sdsoc

Xylon is an SDSoC development environment-qualified Xilinx Alliance Member and offers logicBRICKS IP cores, complete Xilinx All Programmable based solutions and design services.

1.3 Hardware Requirements

The logiVID-ZU Vision Development Kit (Figure 1) includes the following hardware, which is utilized by reference designs provided in the logiADAK-VDF-ZU-SDSoC design framework:

- 1x Xilinx Zynq UltraScale+ MPSoC ZCU102 Evaluation Kit[1] (ver 1.0) with the XCZU9EG-FFVB1156-2 device
- 1x Avnet HDMI IN/OUT FMC Module (Part Number: AES-FMC-HDMI-CAM-G)
- 1x Spectrum Digital FPD-Link III Deserializer FMC Module (Part Number: 703769-0001)
- 4x 1-Mpix Xylon Cameras (FPD-Link III version)
- 1x SD card
- 4x FAKRA cable assemblies
- Power supply

[1] – OEM kit version without the Xilinx Vivado Design Suite seat

1.3.1 FPD-LINK III Deserializer FMC Module

The FOUR-CAM reference design delivered with the logiADAK-VDF-ZU requires the Spectrum Digital FPD-Link III Deserializer FMC module (Figure 2), which comes as a part of the hardware kit deliverables.

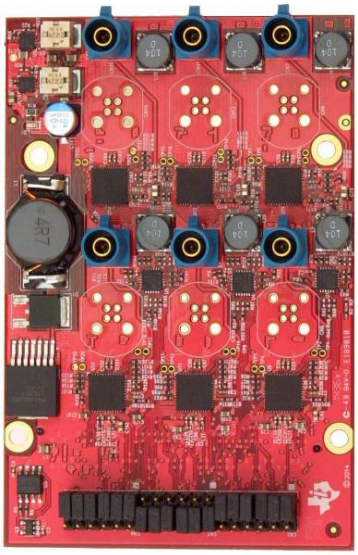


Figure 2: Spectrum Digital FPD-Link III Deserializer FMC Module

This FMC module comes slightly modified to enable operation with the 1.8 V VADJ, which is available on the ZCU102 development board. PCB modifications are described in the `data` folder within the reference designs installation.

1.3.2 HDMI Input/Output FMC Module

The CAM-HDMI reference design delivered with the logiADAK-VDF-ZU requires the HDMI Input/Output FMC module, as a part of the hardware kit deliverables.

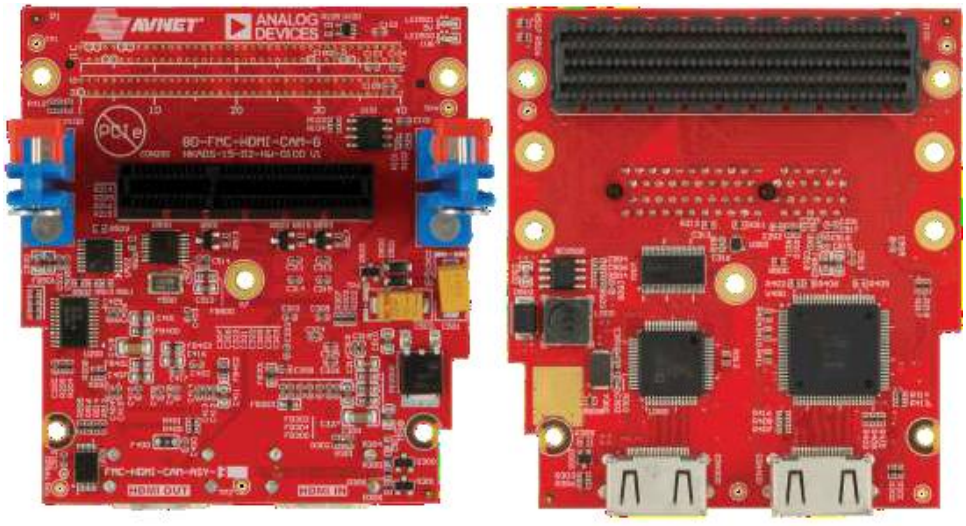


Figure 3: Avnet HDMI Input/Output FMC Module

1.3.3 Xylon Camera

For transmissions of high-definition uncompressed video and camera control data the logiVID-ZU development kit includes Xylon cameras compatible with the TI FPD-Link III high-speed digital video interface. Each camera includes the OmniVision OV10635 1-megapixel camera sensor that combines high-definition 1280x800p30 WXGA (HD) video with the color high dynamic range (HDR) functionality, FPD-Link III serializer (transmitter) board, the Sunex DSL219 miniature fish-eye Wide FOV lens and a short cable lead with a connector. The same camera cable is used for power, control data and HD video transmissions. All camera parts are enclosed in the aluminium housing designed by Xylon. The Its rugged metal construction provides excellent lens and imager module protection and enables safe and easy test vehicle installations.

1.4 Software Requirements

The logiADAK-VDF-ZU-SDSoC reference designs and Xylon logicBRICKS IP cores are fully compatible with the Xilinx SDSoC Development Environment 2017.1. Future design releases shall be synchronized with the newest Xilinx development tools.

1.5 Design Deliverables

1.5.1 SDSoC platform

- Two reference designs prepared for Xilinx SDSoC Development Environment
- Contains the pre-built hardware files for faster completion of software build process
- Supports Linux applications
- Includes software drivers for included logicBRICKS IP cores
- Xylon evaluation logicBRICKS IP cores:
    - logiCVC-ML Compact Multilayer Video Controller
    - logiWIN Versatile Video Input
    - logiI2C I2C Bus Master Controller

1.5.2 Software

- Linux user space drivers with driver examples
- Demo application sources
- Bare-metal software drivers for logicBRICKS IP cores
- logiVIOF VideoIn-VideoOut Library

1.5.3 Binaries

- Precompiled SD Card image for the fastest demo startup
    - Camera/HDMI demo
    - Four Camera demo

1.6 Reference Designs

The logiADAK-VDF-ZU-SDSoC video design framework includes two reference designs:

- FOUR-CAM reference design implements four parallel video inputs from Xylon cameras, and the display output with the RGB graphic overlay. All video inputs are stored in the video memory, and by mean of the on-board push buttons, the user can select each of them for the single camera or all cameras full screen display output. The display output demonstrates the Sobel filtering, either on the full screen, or on one camera input in the tiled four-camera view.

- CAM-HDMI reference design implements a single video input, and the display output with the RGB graphic overlay. Video can be sourced by the Xylon camera, or through the HDMI video input, which is particularly suited for use with the PC and for the playback of prepared test videos, i.e. road videos that make test cases for ADAS algorithms implemented in the Xilinx MPSoC. The design displays a single video source, filtered with Sobel filter, and automatically switches to the HDMI video input upon detection of the plugged-in HDMI cable.

Both reference designs contain the Sobel filter as the example of the C-code based hardware accelerator implemented by the SDSoC Development tool. The Sobel example shows kit users how to integrate their own vision processing logic between video input and video output IP cores, and how to implement it in software or in programmable logic.

2 LOGICBRICKS IP CORES

2.1 About logicBRICKS IP Library

Xylon's logicBRICKS IP core library provides IP cores optimized for Xilinx All Programmable FPGA and SoC devices. The logicBRICKS IP cores shorten development time and enable fast design of complex embedded systems based on Xilinx All Programmable devices.

The key features of the logicBRICKS IP cores are:

- logicBRICKS can be used in the same ways as Xilinx IP cores within the Xilinx Vivado Design Suite, and require no skills beyond general tools knowledge. IP users setup feature sets and programmable logic utilization through implementation tools' Graphical User Interface (GUI).
- Each logicBRICKS IP core comes with the extensive documentation, reference design examples and can be evaluated on reference hardware platforms. Xylon provides evaluation logicBRICKS IP cores to enable risk-free evaluation prior to purchase.
- Broad software support – from bare-metal software drivers to standard software drivers for different operating systems (OS). Standard software support allows graphics designers and software developers to use logicBRICKS in a familiar and comfortable way.
- Xylon assures skilled technical support.

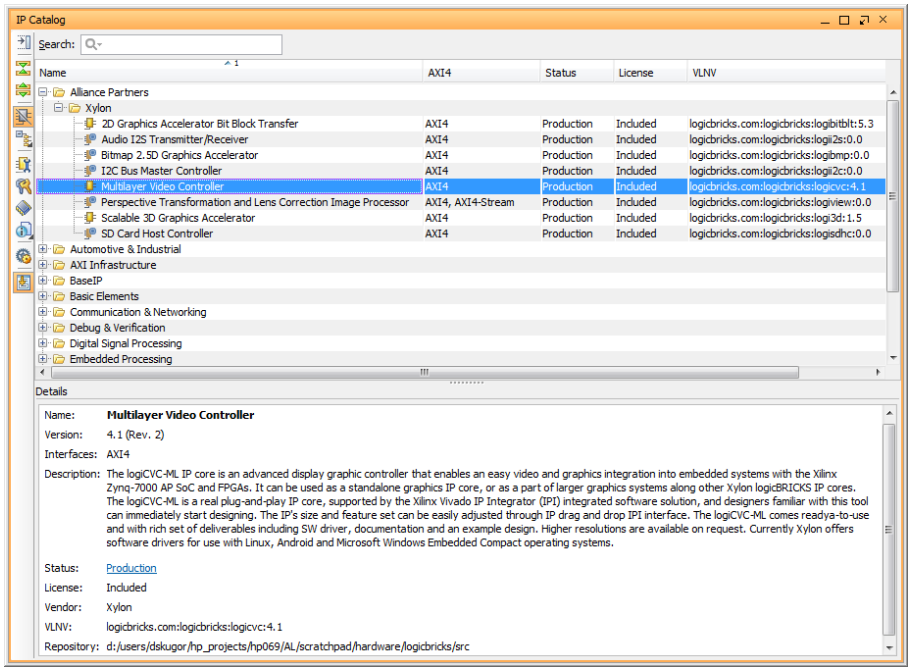


Figure 4: logicBRICKS IP Cores Imported into the Vivado IP Catalog

The Figure 4 shows imported logicBRICKS IP cores into Vivado Design Suite, while the Figure 5 shows a typical logicBRICKS IP core's configuration GUI.

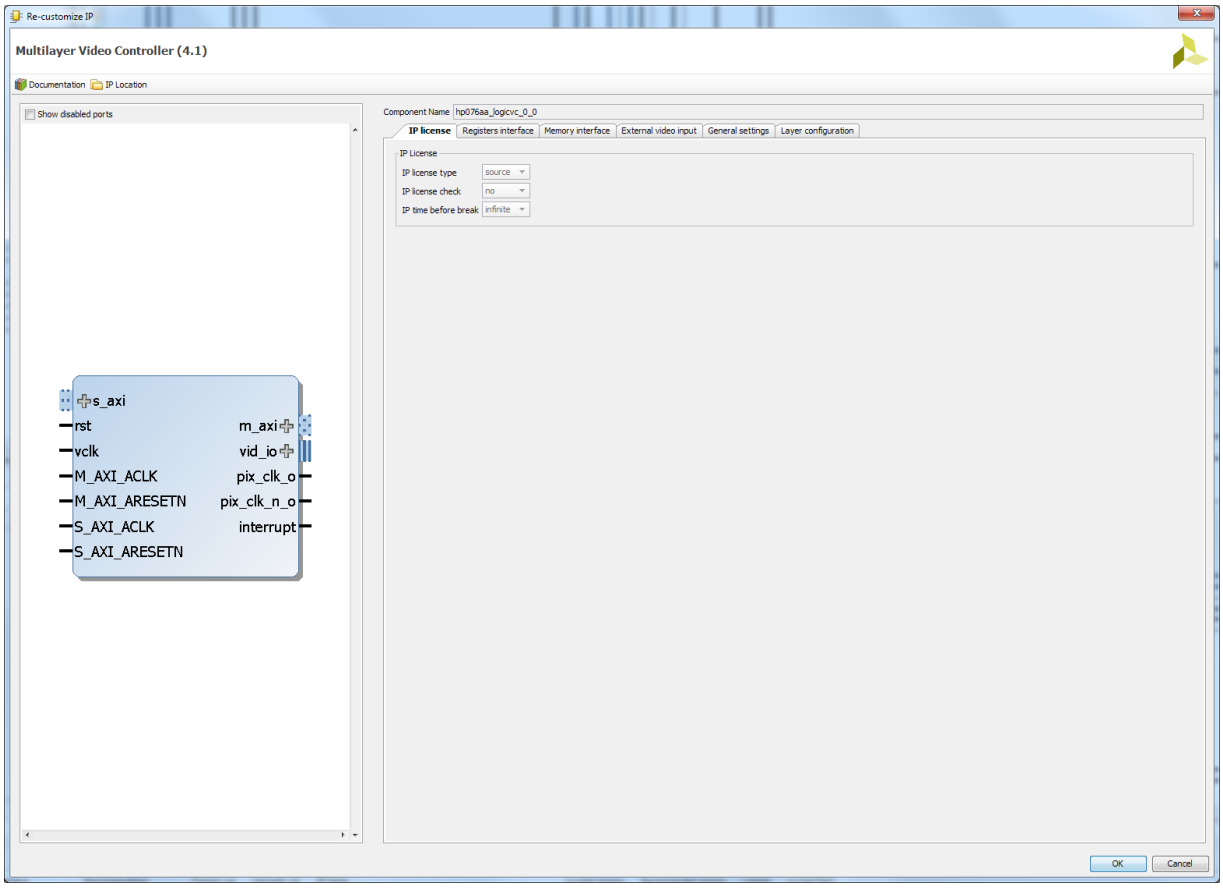


Figure 5: Example of logicBRICKS IP Configuration GUI

To access logicBRICKS IP cores' User's Manuals, double-click on the specific IP core's icon, and then the Documentation icon in the opened IP configuration GUI. Choose either the Product guide to open the manual, or the Change Log to open IP core's change log.

logicBRICKS User's Manuals contain all necessary information about the IP cores' features, architecture, registers, modes of operation, etc.

2.2 Evaluation logicBRICKS IP Cores

Xylon offers free evaluation logicBRICKS IP cores which enable full hardware evaluation:

- Import into the Xilinx Vivado tools (IP Integration)
- IP parameterization through the GUI interface
- Simulation (if Xilinx tools support it)
- Bitstream generation

The logicBRICKS evaluation IP cores are run-time limited and cease to function after some time. Proper operation can be restored by reloading the bitstream. Besides this run-time limitation, there are no other functional differences between the evaluation and fully licensed logicBRICKS IP cores.

Evaluation logicBRICKS IP cores are distributed as parts of the Xylon reference designs: http://www.logicbricks.com/logicBRICKS/Reference-logicBRICKS-Design.aspx.

Specific IP cores can be downloaded from Xylon's web shop: http://www.logicbricks.com/Products/IP-Cores.aspx.

2.3 logicBRICKS IP Cores Used in This Design

2.3.1 logiCVC-ML Compact Multilayer Video Controller

The logiCVC-ML IP core is an advanced display graphics controller for LCD and CRT displays, which enables an easy video and graphics integration into embedded systems with Xilinx Zynq-7000 All Programmable SoC and FPGAs.

This IP core is the cornerstone of all 2D and 3D GPUs. Though its main function is to provide flexible display control, it also includes hardware acceleration functions: three types of alpha blending, panning, buffering of multiple frames, etc.

- Supports all Xilinx FPGA families
- Supports LCD and CRT displays (easily tailored for special display types)
- Display resolutions up to 8192x8129 (including 4K2K resolution)
- Available SW drivers for: Linux and Microsoft Windows Embedded Compact OS
- Support for higher display resolutions available on request
- Supports up to 5 layers; the last one configurable as a background layer
- Configurable layers' size, position and offset
- Alpha blending and Color keyed transparency
- Pixel, layer, or Color Lookup Table (CLUT) alpha blending mode can be independently set for each layer
- Packed pixel layer memory organization:
  - RGB – 8bpp, 8bpp using CLUT, 16bpp 5-6-5, 24bpp 8-8-8 and 30bpp 10-10-10
  - YCbCr – 16bpp (4:2:2), 20bpp (4:2:2), 24bpp (4:4:4), 30bpp (4:4:4)
- Configurable ARM® AMBA® AXI4 memory interface data width (32, 64, 128 or 256 bits)
- Programmable layer memory base address and stride
- Simple programming due to small number of control registers
- Support for multiple output formats:
  - Parallel display data bus (RGB or YCrCb) with 1, 2 or 4 pix per clock: 12x2-bit, 15, 16, 18, 20, 24, 30 bit
  - Digital Video ITU-656: PAL and NTSC
  - LVDS output format: 3 or 4 data pairs plus clock
  - Camera link output format: 4 data pairs plus clock
  - DVI output format (currently not supported in US and US+ devices)
- Supports synchronization to external parallel input

- Versatile and programmable sync signals timing
- Double/triple buffering enables flicker-free reproduction
- Display power-on sequencing control signals
- Parametrical VHDL design that allows tuning of slice consumption and features set
- Prepared for Xilinx Vivado tools

More info: http://www.logicbricks.com/Products/logiCVC-ML.aspx
Datasheet: http://www.logicbricks.com/Documentation/Datasheets/IP/logiCVC-ML_hds.pdf

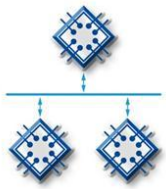
2.3.2 logiWIN Versatile Video Input

The logiWIN IP core enables easy implementation of video frame grabbers. Input video can be decoded, real-time scaled, de-interlaced, cropped, anti-aliased, positioned on the screen... Multiple logiWIN instances enable processing of multiple video inputs by a single Xilinx device.

- Supports versatile digital video input formats:
  - ITU656 and ITU1120 (PAL and NTSC)
  - RGB
  - YUV 4:2:2
- Maximum input and output resolutions are 2048 x 2048 pixels
- Built-in YcrCb to RGB converter, YUV to RGB converter and RGB to YcrCb converter
- Embedded image color enhancements: contrast, saturation, brightness and hue for ITU and YUV separately
- Real-time video scale-up (zoom in) up to 64x
- Real-time video scale-down (zoom out) down to 16 times
  - Lossless scaling down to 2x, or 4x in cascade scaling mode
- Supports video input cropping and smooth image positioning
- Configurable register interface; ARM® AMBA® AXI4-Lite
- ARM® AMBA® AXI4 and AXI4-Lite bus compliant
- Compressed stencil buffer in BRAM (mask over output buffer)
- Supports pixel alpha blending
- Provides "Bob" and "Weave" de-interlacing algorithms
- Supported big and little Endianness memory layout
- Double or triple buffering for flicker-free video
- Prepared for Xilinx Vivado tools

More info: https://www.logicbricks.com/Products/logiWIN.aspx
Datasheet: https://www.logicbricks.com/Documentation/Datasheets/IP/logiWIN_hds.pdf

2.3.3 logiI2C I2C Bus Master



The logiI2C is Xylon logicBRICKS IP core compatible with the I2C serial bus interface standard. The IP core supports single master I2C communications and enables bug-free data transfers.

- Master I2C serial bus controller
- Supports single master operation
- ARM® AMBA® AXI4-Lite bus compliant
- 16 locations deep TX and RX data FIFO
- Supported transmission speeds:
  - Normal – 100 Kbps
  - Fast - 400 Kbps
  - High speed – 3.5 Mbps
- Prepared for Xilinx Vivado tools

More info: https://www.logicbricks.com/Products/logiWIN.aspx
Datasheet: https://www.logicbricks.com/Documentation/Datasheets/IP/logiWIN_hds.pdf

2.4 logicBRICKS IP Cores for Video Processing

Xylon offers several logicBRICKS IP cores for video processing on Xilinx All Programmable FPGA, SoC and MPSoC devices:

logiVIEW Perspective Transformation and Lens Correction Image Processor

Removes fish-eye lens distortions and executes programmable transformations on multiple video inputs in real time. Programmable homographic transformation enables: cropping, resizing, rotating, transiting and arbitrary combinations. Arbitrary non-homographic transformations are supported by programmable Memory Look-Up Tables (MLUT).

More info: http://www.logicbricks.com/Products/logiVIEW.aspx
Datasheet: http://www.logicbricks.com/Documentation/Datasheets/IP/logiVIEW_hds.pdf

logiISP Image Signal Processing (ISP) Pipeline

The logiISP Image Signal Processing Pipeline IP core is a full high-definition ISP pipeline designed for digital processing and image quality enhancements of an input video stream in Smarter Vision embedded designs based on Xilinx All Programmable devices.

More info: http://www.logicbricks.com/Products/logiISP.aspx
Datasheet: http://www.logicbricks.com/Documentation/Datasheets/IP/logiISP_hds.pdf

logiHDR High Dynamic Range (HDR) Pipeline

Ultra High Definition (UHD, including 4K2Kp60) HDR pipeline for camera image quality enhancements. Enables extraction of the maximum detail from high-contrast scenes, i.e. scenes with objects highlighted by a direct sunlight and objects placed in extreme shades.

More info: http://www.logicbricks.com/Products/logiHDR.aspx
Datasheet: http://www.logicbricks.com/Documentation/Datasheets/IP/logiHDR_hds.pdf

3 GET AND INSTALL THE DESIGN FRAMEWORK

Customers entitled for the logiADAK-VDF-ZU-SDSoC installation package delivery get the unique FTP or web download account from Xylon. To purchase the design framework, please visit our webshop: http://www.logicbricks.com/Products/logiADAK-VDF-ZU.aspx

3.1 Installation Process

Installation process is quick and easy. The logiADAK-VDF-ZU-SDSoC framework can be downloaded as a cross-platform Java JAR self-extracting installer. Please make sure that you have a copy of the JRE (Java Runtime Environment) version 6 or higher on your system to run Java applications and applets. Double-click on the installer's icon to run the installation.

At the beginning, you will be requested to accept the design framework license – Figure 6. For installation in Linux OS, please follow instructions:

http://www.logicbricks.com/logicBRICKS/Reference-logicBRICKS-Design/Xylon-Reference-Designs-Linux-Installation.aspx.

If you agree with the conditions from the Xylon license, click NEXT and select the installation path for your logicBRICKS reference design (Figure 7). The installation process takes several minutes. It generates the folder structure described in the paragraph 3.1.1 Folder Structure.

Figure 6: Installation Process – Step 1

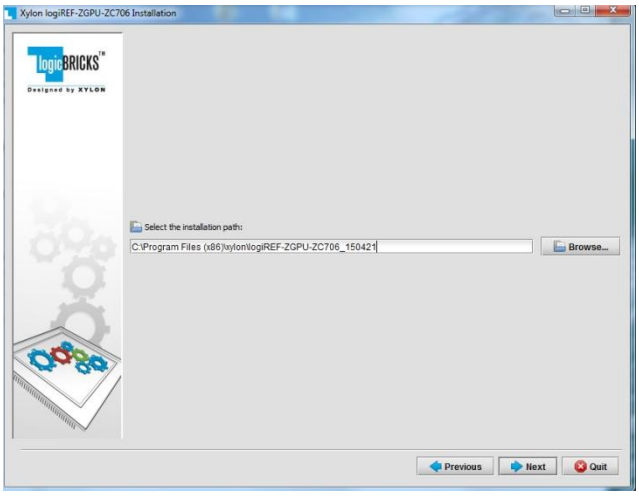


Figure 7: Installation Process – Step 2

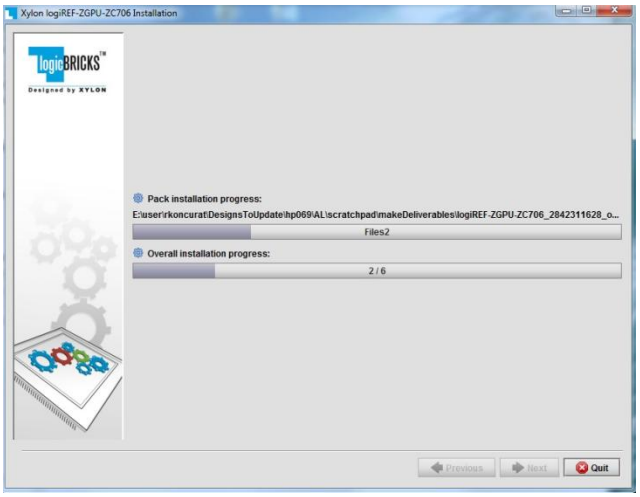


Figure 8: Installation Process – Step 3

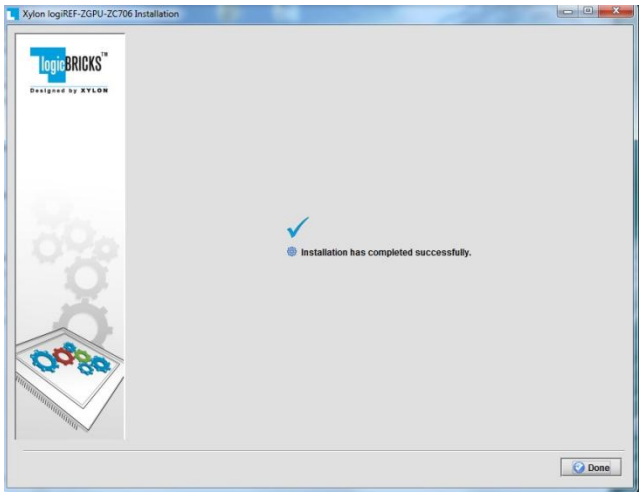


Figure 9: Installation Process – Step 4

3.1.1 Filesystem Permissions of the Installed Folder (Microsoft® Windows® OS)

The reference design installed in the default path `C:\Program Files\xylon` may inherit read-only filesystem permissions from the parent folder. This will block you in opening the hardware project file in Xilinx Vivado tools. Therefore it is necessary to change the filesystem permissions for the current user to "Full control" preferably.

To change the user permissions for `C:\Program Files\xylon` folder and all of its subdirectories, right click on the `C:\Program Files\xylon` folder and select "Properties". Under "Security" tab select "Edit". Select "Users" group in the list and check "Full control" checkbox in the "Allow" column.

3.2 Folder Structure

Figure 10 gives a top level view of the directories and files included with the logiADAK-VDF-ZU-SDSoC video design framework. Table 4 explains the purpose of directories.

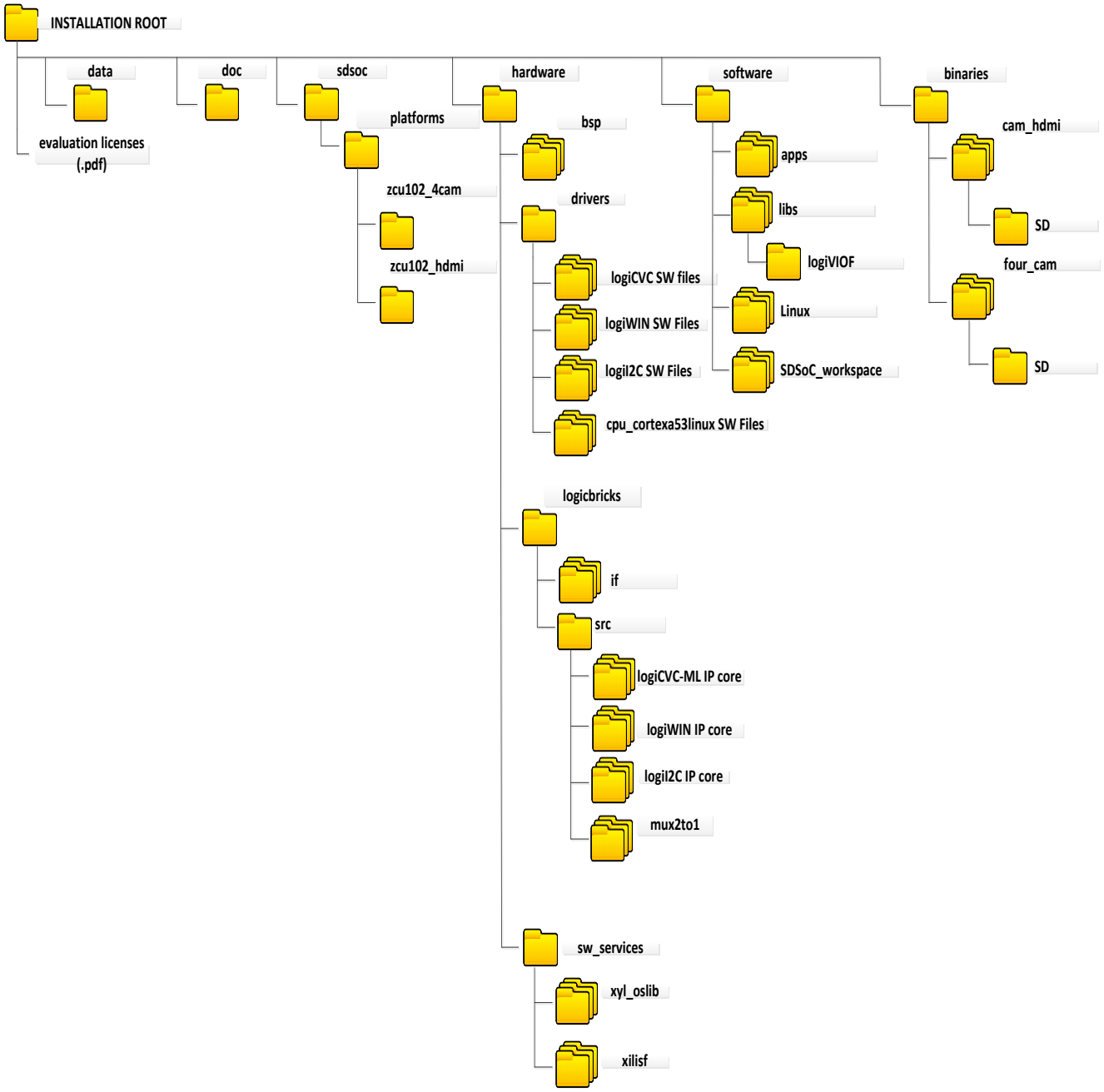


Figure 10: The Folder Structure

Table 4: Explanation of the logiADAK-VDF-ZU-SDSoC folder structure

Folder		Purpose
INSTALLATION ROOT		This folder contains the *start.html* page – the jump-start navigation page through the reference design.
data		Additional hardware documentation/datasheets/manuals.
doc		Project documentation.
sdsoc/platforms		SDSoC platform files.
	zcu102_hdmi	Cam/hdmi platform files.
	zcu102_4cam	Four cam platform files.
hardware		
	bsp	Xylon Linux user space Board Support Package (BSP); custom Xylon BSP compatible with the Xilinx SDK. It enables users to quickly build Linux User space applications within the SKD workspace.
	drivers	Standalone (bare-metal) drivers for logicBRICKS IP cores with documentation and examples.
	logicbricks/if	Xylon custom IP core interfaces (bus definitions).
	logicbricks/eval	Evaluation logicBRICKS IP cores. IP cores' User's Manuals are stored in doc subdirectories.
	sw_services	xyl_oslib – Xylon OS abstraction library for Xilinx Xilkernel embedded kernel – use in standalone (non-OS) applications.
software		
	apps	Demo applications source code files.
	libs/logiVIOF	logiVIOF source code files.
	Linux/kernel	Linux kernel and device tree configuration files.
	SDSoC_workspace	Xilinx SDK workspace folder for building bare-metal applications.
binaries		Prepared binaries ready for download to SD card.
	cam_hdmi/SD	Cam/hdmi platform with hardware implemented sobel filter.
	four_cam/SD	Four cam platform with hardware implemented sobel filter.

4 GETTING LOGICBRICKS IP LICENSES

The logiADAK-VDF-ZU-SDSoC installation comes with the evaluation versions of the logicBRICKS IP cores, and in order to be able to change the provided reference designs, you need to request the proper licenses from Xylon.

Please contact Xylon Technical Support Service support@logicbricks.com and immediately provide your Ethernet MAC ID number or Sun Host ID.

For instructions how to find your Ethernet MAC or host ID, please visit:
http://www.logicbricks.com/Documentation/Article.aspx?articleID=KBA-01186-M0JXKD..

For each logicBRICKS IP core used in the logiADAK-VDF-ZU reference designs Xylon will generate and send to you separated e-mails with the license keys (file) and full instructions for setting up the license key and downloading the logicBRICKS IP core. Please follow the provided instructions.

If you experience any troubles during the registration process, please contact Xylon Technical Support Service – support@logicbricks.com.

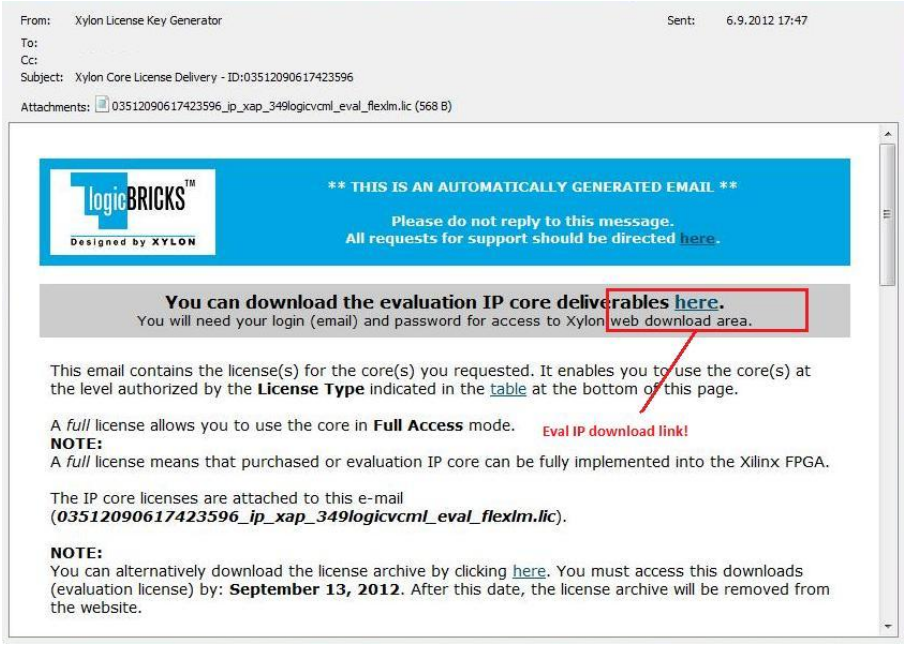


Figure 11: E-mail with logicBRICKS License and Download Instructions

5 LOGIADAK-VDF-ZU-SDSOC REFERENCE DESIGNS

The logiADAK-VDF-ZU-SDSoC framework contains two SDSoC platform designs:

- zcu102_hdmi
- zcu102_4cam

Xylon's platforms for the targeted hardware kit are designed following Xilinx's SDSoC platform design specifications [REF 2].

Platform designs are based on the logiADAK-VDF-ZU reference designs functionality with addition of the Sobel filter implementation. Sobel filter is implemented between active input and display output. Filter implementation provides a simple example how to implement C-code based accelerator in the Programmable Logic by SDSoC tool usage.

The **`zcu102_hdmi`** and **`zcu102_4cam`** platforms support Linux applications and contain precompiled Linux drivers for the included logicBRICKS IP cores. Linux and Bare-metal software drivers' source files are included within the reference design deliverables – please see the Table 4.

The provided platforms also include the pre-built MPSoC hardware folder for faster user workflow described in [REF 3]. The bitstream contained in platform hardware folder enables users to run and verify custom developed applications, entirely in software and without engaging in the lengthy process of hardware implementation. However, that process cannot be used with the provided demo application if the sobel filter function is marked for hardware acceleration. Alternatively, user can generate bitstream for the Cam/HDMI and Four-Cam applications and then uncheck the **`Generate Bitstream.`** It will enable the tools to rebuild only software portions of the design.

5.1 CAM-HDMI SDSoC Design and Memory Layout

This platform design (Figure 12) supports a single video camera or a single HDMI video input (only one active at a time), and the single display output.

The following Zynq UltraScale+ MPSoC resources are not used by the **`zcu102_hdmi`** platform and can be used for other purposes within the SDSoC development environment:

- Clocks:
  - Clock id 1 (default): 100 MHz
  - Clock id 2: 150 MHz
  - Clock id 3: 200 MHz
- PS-PL ports:
  - M_AXI_HPM0_FPD
  - M_AXI_HPM1_FPD
  - S_AXI_HP2_FPD
  - S_AXI_HP3_FPD

- Interrupts:
  - Over xlconcat IP, ports from 3 to 7

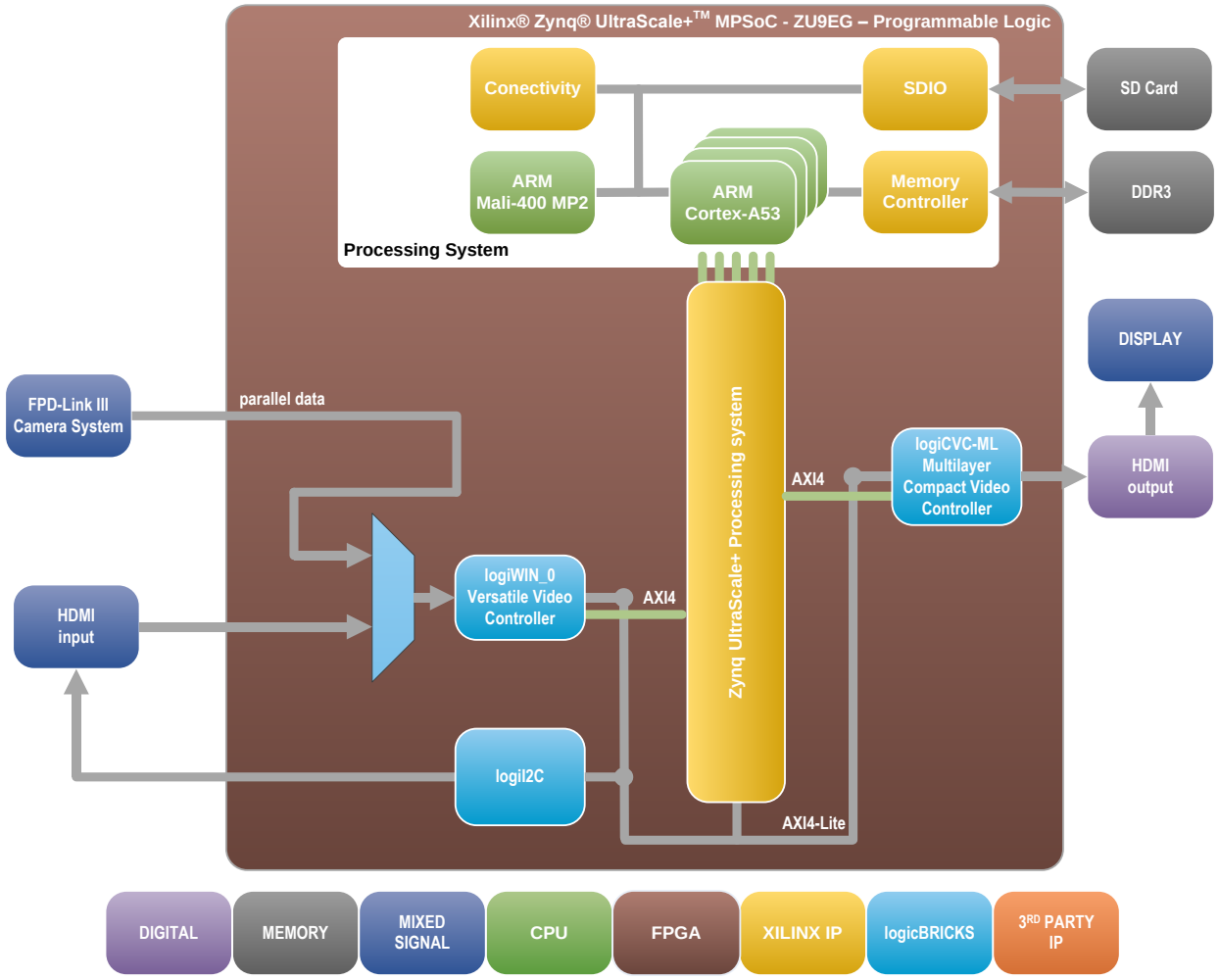


Figure 12: CAM-HDMI MPSoC Design – Block Diagram

(Clock Generator Module and other utility IP cores are not shown)

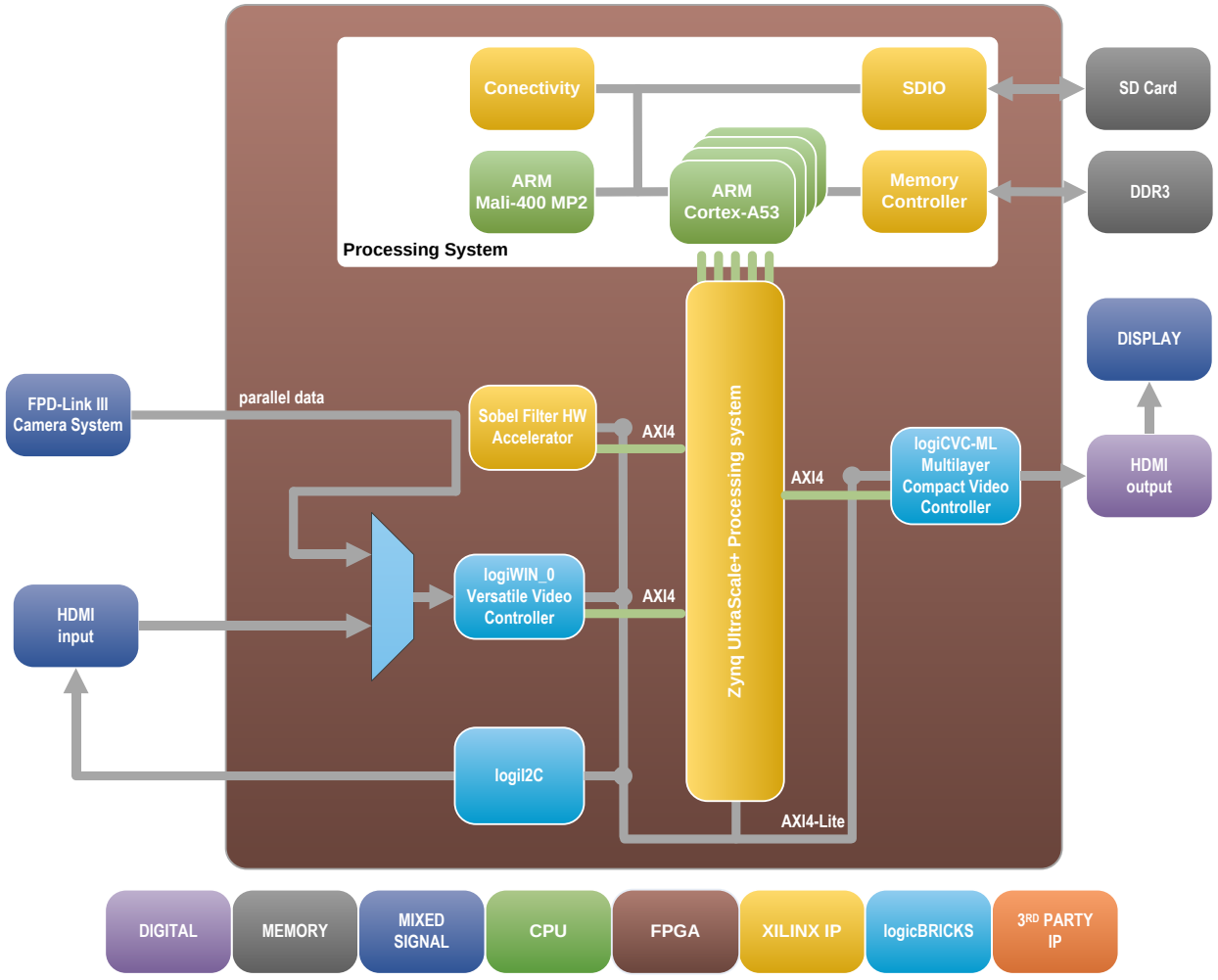


Figure 13: CAM-HDMI MPSoC Design – Block Diagram (Platform with the Sobel Filter IP)

(Clock Generator Module and other utility IP cores are not shown)

The Figure 14 shows the memory layout of video buffers. Each logiCVC-ML layer has its own reserved memory space and use multiple video buffers (not shown).

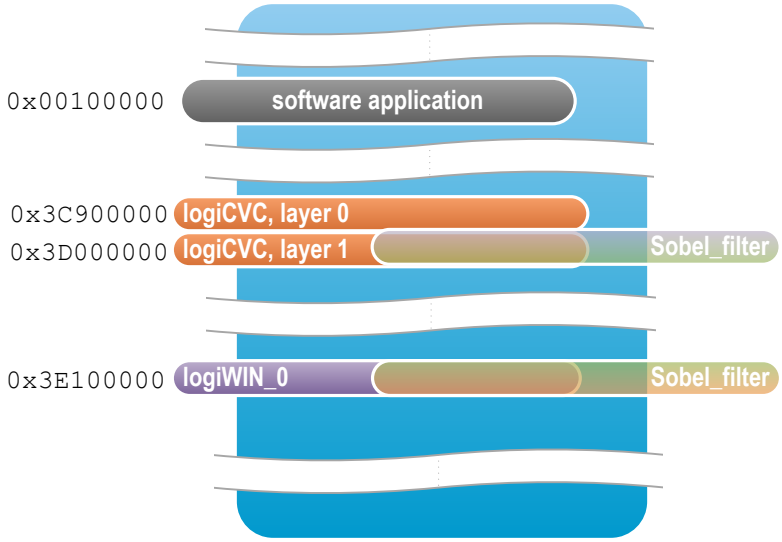


Figure 14: CAM-HDMI Design Memory Layout

5.2 FOUR-CAM SoC Design and Memory Layout

This platform design (Figure 15) supports four camera inputs, and the single display output. The following Zynq UltraScale+ MPSoC resources are not used by the **`zcu102_4cam`** platform and can be used for other purposes within the SDSoC development environment:

- Clocks:
  - Clock id 1 (default): 100 MHz
  - Clock id 2: 150 MHz
  - Clock id 3: 200 MHz
- PS-PL ports:
  - M_AXI_HPM0_FPD
  - M_AXI_HPM1_FPD
  - S_AXI_HP3_FPD
- Interrupts:
  - Over xlconcat IP, ports from 6 to 7

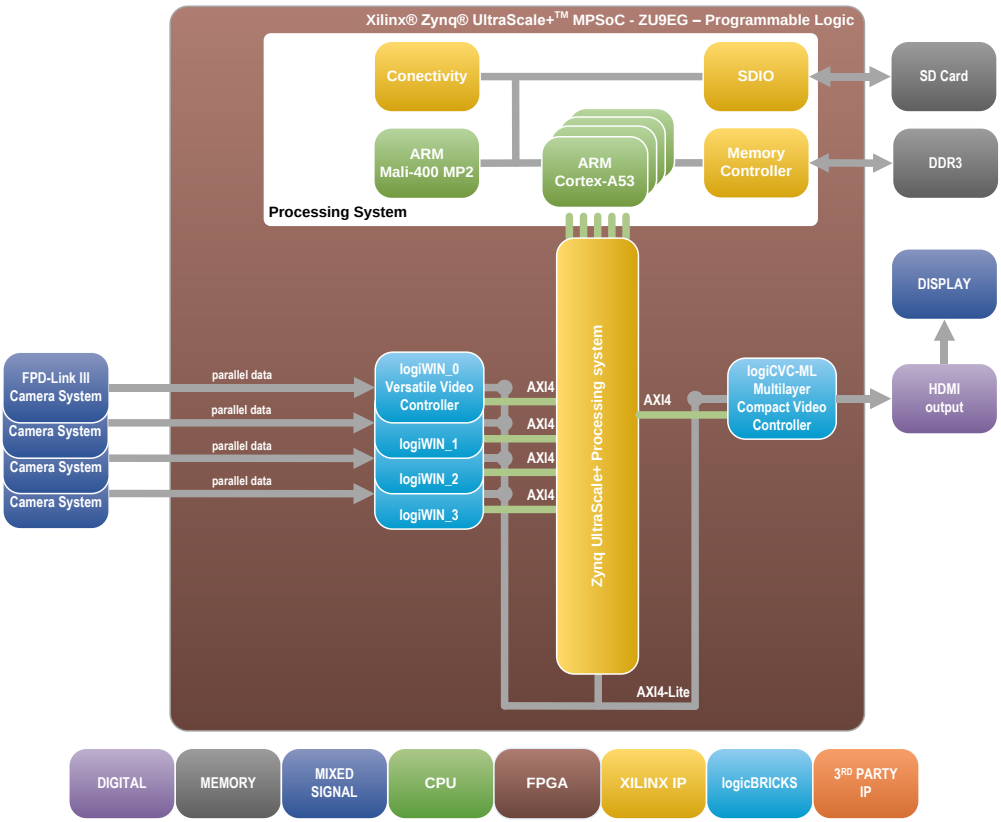


Figure 15: FOUR-CAM MPSoC Design– Block Diagram

(Clock Generator Module and other utility IP cores are not shown)

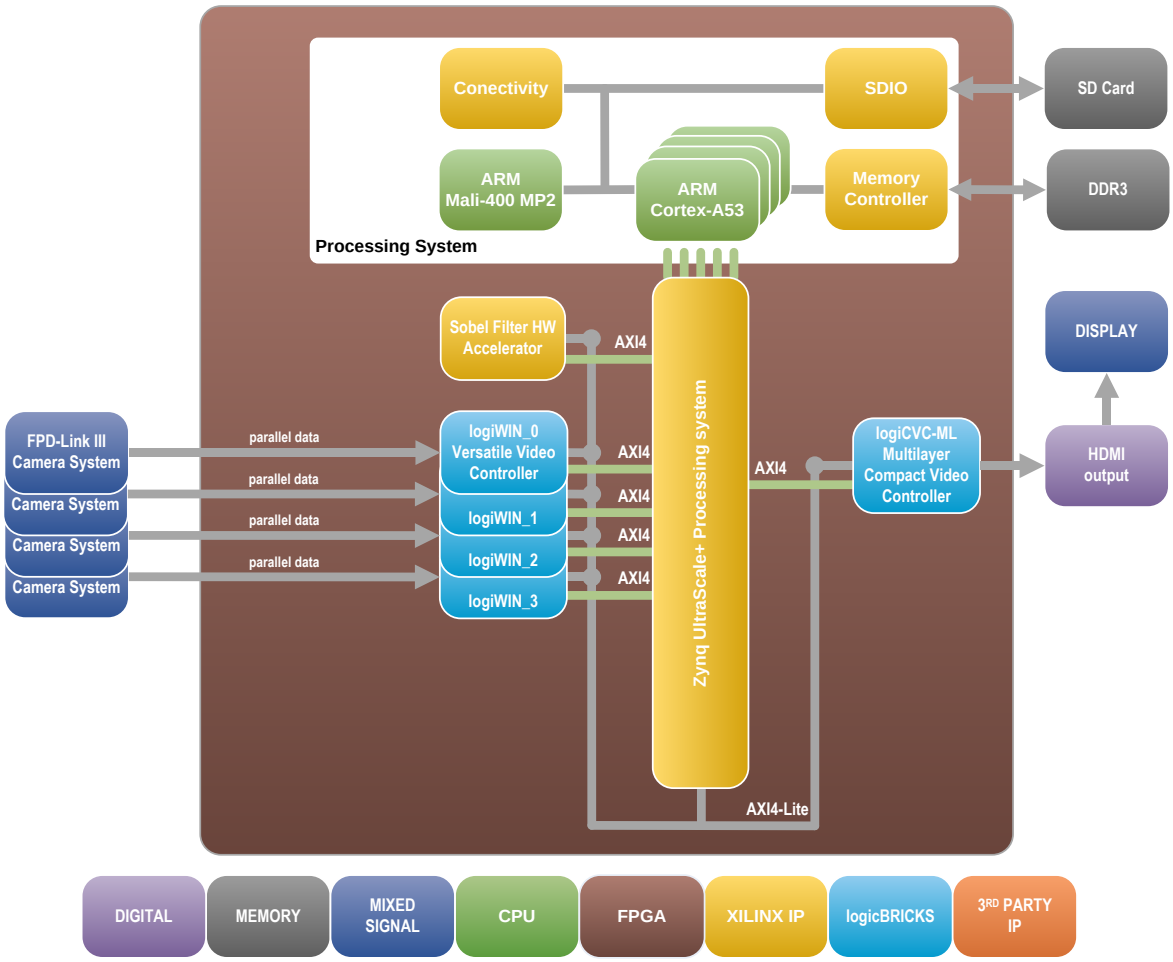


Figure 16: FOUR-CAM MPSoC Design– Block Diagram (Platform with the Sobel Filter IP)

(Clock Generator Module and other utility IP cores are not shown)

The Figure 17 shows the memory layout of video buffers. Each logiCVC layer has its own reserved memory space and multiple video buffers (not shown).

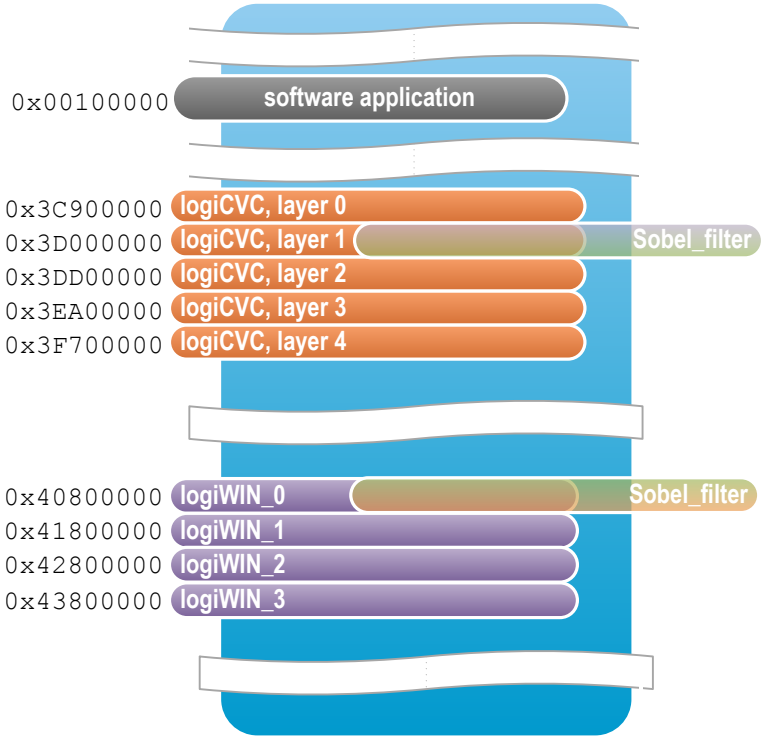


Figure 17: FOUR-CAM Design Memory Layout

5.3 Using the logiADAK-VDF-ZU-SDSoC Platforms and the Hardware Accelerator Based on the C-code Description

Please copy the *zcu102_hdmi* and *zcu102_4cam* platform folders from ***logiADAK-VDF-SDSoC_vX_Y_Z/sdsoc/platforms/*** to folder *<SDSoC_install_dir>/platforms* within your SDSoC installation folders.

5.4 Using and Configuring Hardware Accelerator as C-code Based IP core

In the provided software demo applications the sobel filter algorithm is contained in one C++ function `sds_sobel()`(*`sds_sobel.cpp`*).This function is used as a simple example of the C-code based hardware accelerator implementation generated by the SDSoC tool.

The HLS generated Sobel filter accelerator connects to an AXI-Stream accelerator adapter that drives all inputs and captures all outputs. The adapter has two FIFO interfaces for streaming video data to and from the accelerator. The adapter converts the video data streams to the AXI-Stream format to connect to the AXI DMA Rx and Tx channels via the AXI-Stream router. The instantiated

data mover is an AXI DMA in the scatter-gather mode. The width, height, and stride parameters are simple control signals (scalars) driven by the adapter.

Note that this memory-to-memory pipeline is not video-timing accurate. The DMA reads and writes video frames without inserting any horizontal blanking in between video lines. An interrupt is issued by the DMA engine separately for the read and write channels upon completion of a frame read/write operation.

The DMA engine translates AXI-Stream data to AXI-Memory-mapped interface. The output from the SDSoC generated accelerator is able to connect to the full-power domain high-performance PL master non-coherent AXI interface.

The AXI-Lite control interface is of the SDSoC generated accelerator is able to connect to the full-power domain high-performance PS master AXI interface.

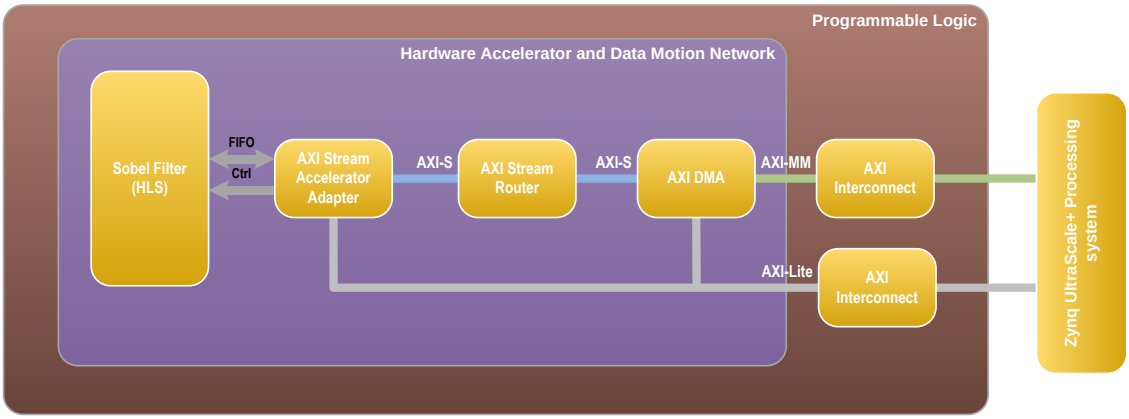


Figure 18: Hardware Accelerator and Data Motion Network entirely generated by the SDSoC tool based on a C-code description

5.5 Software Description

Software components:

- Linux 4.9 (tag "xilinx-v2017.1"), APF driver updated to 2017.2 and improved memory management.
- v4l2 driver, https://github.com/logicbricks/driver_v4l2_logiwin, updated driver v1.1 is located here: /software/Linux/kernel/linux-xlnx-xilinx-v2017.1/kernel_src_diff/drivers/media
- framebuffer driver, https://github.com/logicbricks/driver_fb_logicvc, updated driver v4.2 is located here: /software/Linux/kernel/linux-xlnx-xilinx-v2017.1/kernel_src_diff/drivers/video
- logiVIOF library, for detailed documentation see: /software/libs/logiVIOF/doc/html/index.html
- Xilinx SDSoC FPGA hardware processing wrapper

Figure 19 illustrates the software architecture used in this video framework.

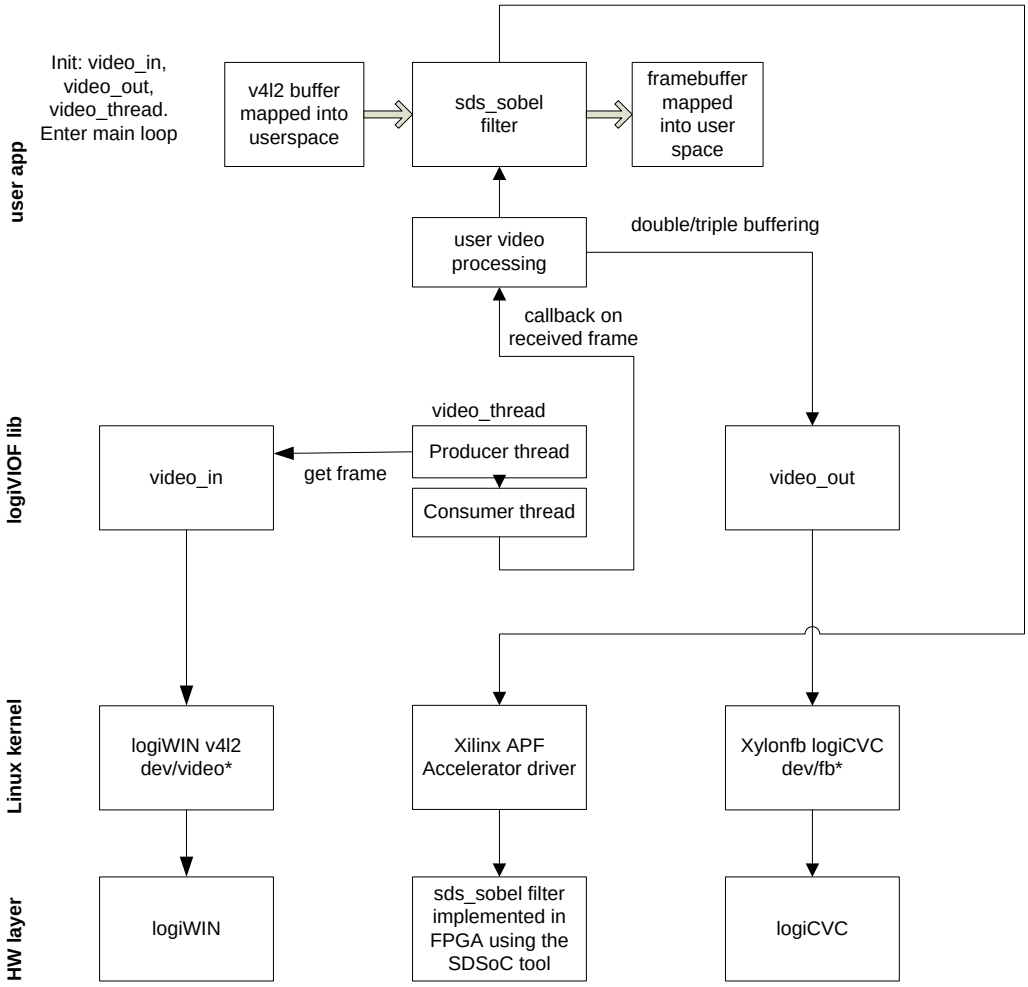


Figure 19: The Software Architecture

5.5.1 Demo application

There are two demo applications:

- FOUR-CAM for four cameras operation
- CAM-HDMI for mixed camera/HDMI operation

Both applications use the same C-code source and the only difference between them is the DVI_INPUT_PRESENT symbol defined in the CAM-HDMI project.

Both applications demonstrate how to display the flicker-free and processed video from an external video input by using the software buffering synchronization.

The applications use the logiVIOF (VideoIn-VideoOut) library (Figure 19) for VideoIn and VideoOut initialization. After the initialization, the logiVIOF VideoThread calls user video processing call-back function upon the reception of the video frame. The video processing call-back in this demo calls the sds_sobel filter processing function, which can be implemented either in software (processing system) or hardware (programmable logic) using the SDSoC tool. The processed video frame is further passed to the logiVIOF VideoOut and displayed on the screen. Video output synchronization is software controlled and uses double or triple video buffering.

For details about the logiVIOF software library, please check:
/software/libs/logiVIOF/doc/html/index.html

Main functions used in the demo are:

- demoInit
- demoStart
- demoStop
- demoDeinit

demoInit initializes video in and video out:

```
demoInit:
   ▪ vout_init            //create logiVIOF VideoOut interface to control
                          //frame buffer device (xylonfb dev/fb*)
   ▪ vout_layerHWBufferingEnable  //enables hardware buffering
   ▪ vout_layerDisable    //disable layer
   ▪ vout_layerFill       //clear layer
   ▪ vout_layerSetOffset  //reset offset
   ▪ sds_mmap             //map output buffers
   ▪ vin_init             //create logiVIOF VideoOut interface to control
                          //framegrabber (xylon v4l2 dev/video*)
```

After the **demoInit**, the **demoStart** function is executed:

```
demoStart:
   ▪ vin_set_format       //set input format if mode is changed between
                          //one camera / four cameras
   ▪ vout_layerFill       //clear layer
   ▪ vout_layerSetPositionAndSize //depending on mode set layer geometry
                                  //for one or all cameras
   ▪ vout_layerEnable     //depending on mode enables one
                          //or all four layers
   ▪ video_thread_init    //create and start logiVIOF videoThread with
                          //pointer to user callback function
```

The CAM-HDMI reference design enables work with the HDMI input and the video input from Xylon's video camera. The plugged in HDMI video input takes precedence. The FOUR-CAM reference design enables the application user to toggle between the single camera and the four camera views.

Upon the **demoStart** execution, the program enters the main loop and checks the on-board push-buttons and keyboard buttons. User can stop the application by calling the **demoStop** function:

```
demoStop:
   ▪ video_thread_deinit  //stops frame grabber, exit threads
   ▪ vout_layerDisable    //disables all layers
```

The **demoStart** function is called with the user options after the end of the **demoStop** function.

If user exits the application, the **demoDeinit** function deinitializes video inputs and the video output:

```
demoDeinit:
   ▪ sds_munmap           //unmap output buffers
   ▪ vin_deinit           //deinitializes VideoIn
   ▪ vout_deinit          //deinitializes VideoOut
```

5.5.2 Input resolution and the framerate

Both reference designs use 1280x800@30 input video resolution for both, the Xylon camera and the HDMI video input.

5.5.3 Output resolution and the frame rate

Both reference designs use 1920x1080@60 output resolution.

6 QUICK START

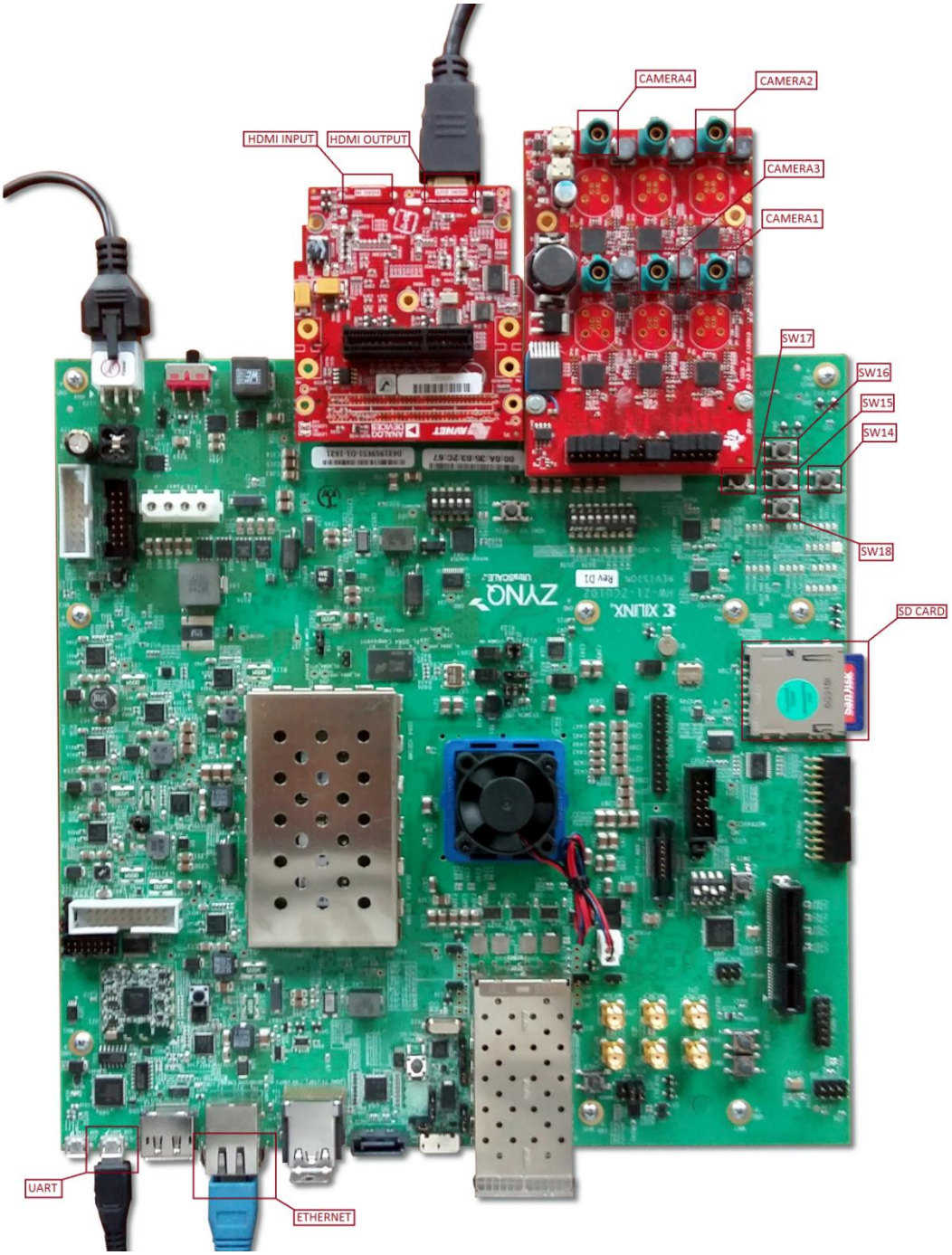


Figure 20: FOUR-CAM and CAM-HDMI HW Setup

6.1 Run the Precompiled Linux Demo Examples

To enable rapid testing of the hardware setup, Xylon provides application demo binaries in the *binaries* folder of the deliverables. There are two demo applications: the camera/HDMI and the four cameras demo.

To run the four cameras demo copy the content of the *binaries/four_cam/SD* folder to the root folder of the SD card.

To run the camera/HDMI demo, copy the content of the *binaries/cam_hdmi/SD* folder to the root folder of the SD card.

SD card should be formatted as FAT32.

Optionally, you can use a serial terminal program (baud rate 115200 8N1) and the USB UART connection to the ZCU102 board to monitor the system's operation.

For full explanation of the ZCU102's features and settings, please check the documentation Xilinx XTP426.

6.2 Demo controls

Start the FOUR-CAM demo design and the display will show the video stream from one of the four attached video cameras. Change the displayed camera input by pressing the number '1', '2', '3' or '4' on a keyboard, or by pressing SW16 or SW18 push-buttons on the ZCU102 board. Toggle demo mode between one camera, or all cameras by pressing the number '5' on a keyboard, or by pressing the SW15 push-button on the ZCU102 board. Stop the application by pressing 'q' on the keyboard.

Start the CAM-HDMI demo design. The attached PC monitor will show the HDMI video, if there is the HDMI video source connected to the HDMI video input of the FMC card. Otherwise, the display will show the video input from the attached video camera. Stop the application by pressing 'q' on the keyboard.

6.3 Change the Delivered Software

6.3.1 Xilinx Development Software

The logiADAK-VDF-ZU-SDSoC video design framework reference designs and the Xylon logicBRICKS IP cores are fully compatible with Xilinx development tools – Vivado Design Suite 2017.1. Future design releases shall be synchronized with the newest Xilinx development tools.

Licensed users of Xilinx tools can use their existing software installation for the logiADAK-VDF-ZU-SDSoC evaluation and modifications.

6.3.2 Set Up Linux System Software Development Tools

Set of ARM GNU tools are required to build the Linux software and applications. The complete tool chain for the Zynq UltraScale+ Multiprocessor SoC can be obtained from the Xilinx ARM GNU Tools wiki page: http://wiki.xilinx.com/Install+Xilinx+Tools. Access to tools requires a valid, registered Xilinx user login name and password.

6.3.3 Set Up git Tools

Git is a free Source Code Management (SCM) tool for managing distributed version control and collaborative development of software. It provides the developer a local copy of the entire development project files and the very latest changes to the software.

Visit http://wiki.xilinx.com/using-git to get instructions how to use Xilinx git.

To get the latest version of Xylon logicBRICKS software drivers for Linux operating system, please visit Xylon's git: https://github.com/logicbricks.

6.3.4 Setting up the SDK workspace

All logiADAK-VDF-ZU-SDSoC software applications are delivered in the source code to enable users to do software customizations. If the hardware platforms has been changed it is necessary to rebuild the board support packages and FSBL files. Delivered SDK workspace contains required board support packages. This paragraph explains how to setup the Xilinx SDK environment for these customizations.

Quick steps required to set up the SDK workspace:

1. If the CAM-HDMI MPSoC design is modified, open the recreated and modified ***CAM-HDMI Vivado design,*** go to: File → Export hardware (select option „*Include bitstream")*
2. If the FOUR-CAM MPSoC design is modified, open the recreated and modified ***FOUR-CAM Vivado design,*** go to: File → Export hardware (select option „*Include bitstream")*
3. Open SDK, select workspace: ***logiADAK-VDF-ZU-SDSoC_vX_Y_Z/software/SDK_workspace***
4. Open C/C++ perspective: Window → *Perspective* → *Open Perspective* → *Other* → *C/C++*
5. In the SDK go to: *Xilinx Tools* → *Repositories* → *New,* add ***logiADAK-VDF-ZU-SDSoC_vX_Y_Z/hardware*** directory
6. In the SDK go to: *Project* and uncheck *Build automatically* (optional, but recommended)
7. In the SDK go to: *File* → *Import* → *General* → *Existing projects to workspace* → *Next,* in *Select a root directory* choose ***logiADAK-VDF-ZU-SDSoC_vX_Y_Z/software/SDK_workspace***, select all projects and click *Finish*
8. *Each BSP needs to be regenerated. Right click on BSP and select Re-generate BSP Sources. Do this for all BSPs.*
9. Open *bsp_linuxuserspace_zcu102_hdmi* folder and copy *psu_cortexa53_0 include* and *lib* folders contents to *platforms/zcu102_hdmi/aarch64-linux-gnu* folder.
10. Copy *SDK_workspace/fsbl_zcu102_hdmi/Release/fsbl_zcu102_hdmi.elf* file to *platforms/zcu102_hdmi/boot* folder.

11. Open *bsp_linuxuserspace_zcu102_4cam* folder and copy *psu_cortexa53_0 include* and *lib* folders contents to *platforms/zcu102_4cam/aarch64-linux-gnu* folder.
12. Copy *SDK_workspace/fsbl_zcu102_4cam/Release/fsbl_zcu102_4cam.elf* file to *platforms/zcu102_4cam/boot* folder.

6.3.5 Setting up the SDSoC workspace

All logiADAK-VDF-ZU-SDSoC software applications are delivered in the source code to enable users to do software customizations. This paragraph explains how to setup the Xilinx SDSoC environment for software customizations.

Quick steps required to set up the SDSoC workspace:

1. Make sure platform copy (section 5.3) is done.
2. Open SDSoC SDK, select workspace: ***logiADAK-VDF-ZU-SDSoC _vX_Y_Z/software/SDSoC_workspace***
3. In the SDSoC SDK go to: *Project* and exclude *Build automatically* (optional, but recommended)
4. In the SDSoC SDK go to: *File → Import → General → Existing projects to workspace → Next,* in *Select a root directory* choose ***logiADAK-VDF-ZU-SDSoC _vX_Y_Z/software/SDSoC_workspace***, select all projects and click *Finish*
5. For optimized SW implementation it is recommended to set project build configuration to *SDRelease.* In Project explorer window -> Right click on the SDSoC project: *Build Configurations -> Set Active -> SDRelease*
6. If you want to use hardware acceleration, function *sds_sobel()* needs to be marked as [H]. This is done by default, so no change is needed.
7. If the *Build automatically* option has been disabled, build all imported applications manually.
8. If you want to run the sobel filter in SW, function *sds_sobel()* needs to be toogled to SW. In Project explorer window:
    - Browse to *src/sds_sobel.cpp* file. Expand content of the file by clicking arrow mark on left side (|>)
    - On *sds_sobel* function: right click and Toggle HW/SW (mark [H] disappears)
9. If the *Build automatically* option has been disabled, build the application manually.
10. Copy the sd_card folder content to the to the root folder of the SD card.
    - sd_card folder find in the application build generated folder. Build generated folder has the same name as the chosen build configuration and is located in the SDSoC workspace application folder (e.g. for the build configuration *SDRelease* and application *zcu102_4cam_linux* the build generated folder is *zcu102_4cam_linux/SDRelease*).

Due to dependencies between the application and the logiVIOF library, please make sure to use the same configuration options (Release/Debug) for both, the application and the logiVIOF library.

6.4 Software Instructions – Linux Software

Xylon provides Linux Framebuffer and V4L2 logiWIN driver. Zynq UltraScale+ tool chain, Linux kernel and file system used for development and demonstrations of Xylon drivers are provisions of Xylon.

- Linux kernel building instructions, and `dts` files can be found in *software\Linux\kernel* folder.
- Running Linux applications with the ZCU102 board setup for the precompiled SD card image.

6.5 Debugging SDSoC Application with TCF Agent

1. Launch the application's SDSoC workspace
2. Open **`Debug Configurations`** window **`Run -> Debug Configurations`**…
3. To create a new Debug Configuration, double click on the **`Xilinx SDSoC Application`** section on the left hand side of the Debug Configuration GUI
4. Set **`Debug Type`** to **`Linux Application Debug`**

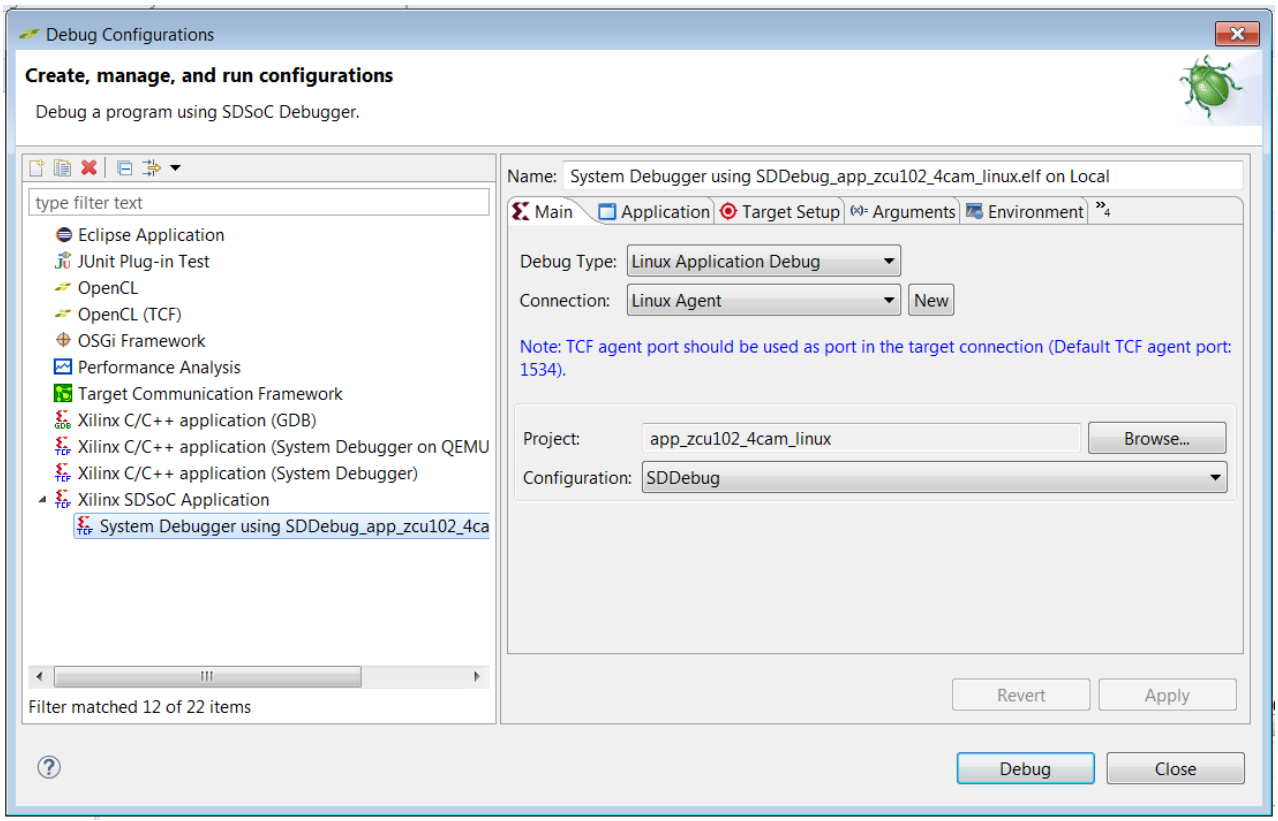


Figure 21: SDK Workspace – Debug Configurations

5. Create new connection; set **`Host`** to correct target IP and set **`Port`** to 1534

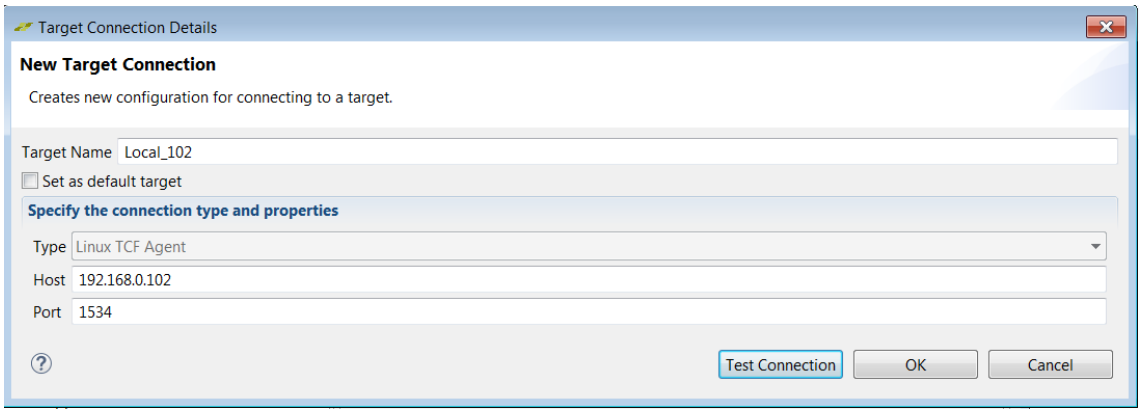


Figure 22: SDK Workspace – Create the new Connection

6. Switch to **`Application`** Tab
7. Enter **`Project Name`**, **`Local File Path`**, **`Remote File Path and Working dir`**

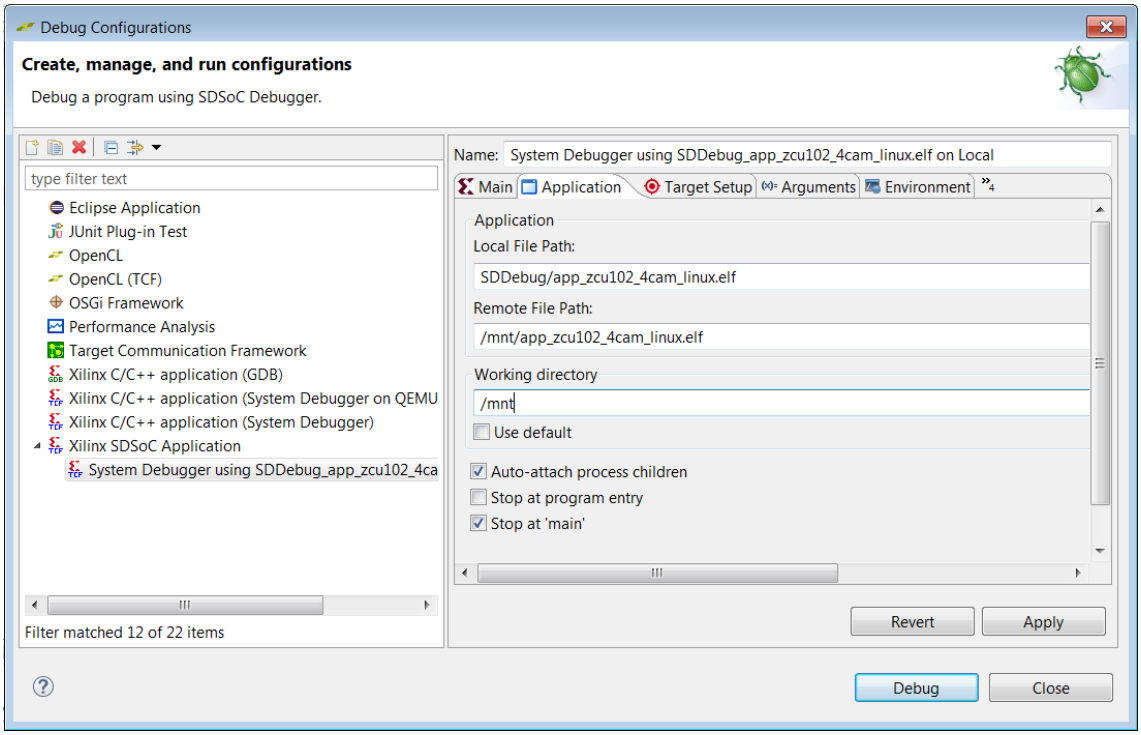


Figure 23: SDK Workspace – Application Tab

8. If shared libraries are used set paths to in the **`Environment`** tab
9. Start **`Debug`**

7 REVISION HISTORY

Version	Date	Author	Approved by	Note
1.00.a	July 28th, 2017	A.Bogdanic, D.Stimac		Initial draft
2.00.a	September 5th, 2017	A.Bogdanic, D.Stimac	G. Galić	Initial release