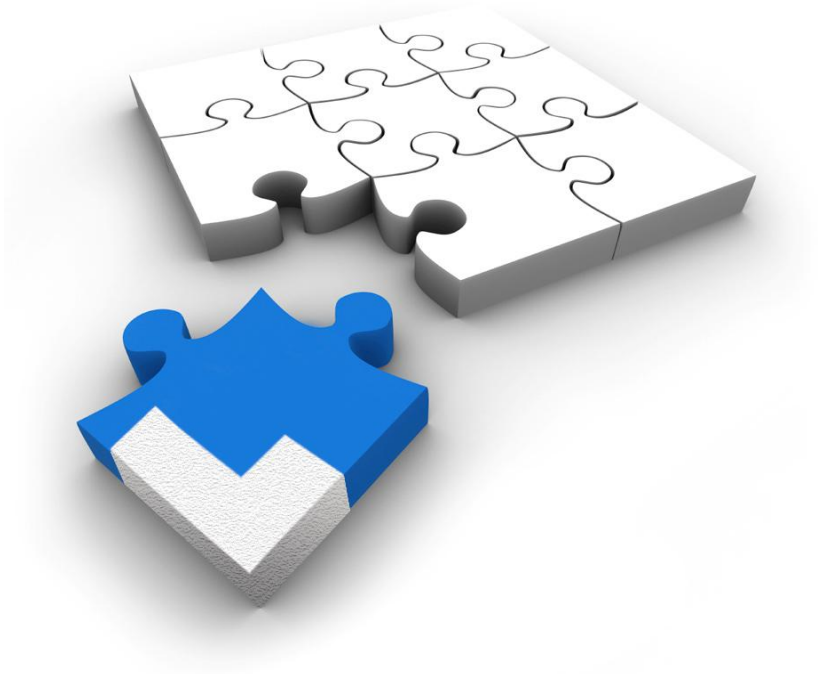
logiADAK-VDF

Video Design Framework - Reference Designs for Xylon logiADAK and logiVID-Z Development Kits

User‘s Manual

Version: 3.00.a

logiADAK_VDF_v3.00.a.docx



All rights reserved. This manual may not be reproduced or utilized without the prior written permission issued by Xylon.

Copyright © Xylon d.o.o. logicBRICKS® is a registered Xylon trademark.
All other trademarks and registered trademarks are the property of their respective owners.
This publication has been carefully checked for accuracy. However, Xylon does not assume any responsibility for the contents or use of any product described herein. Xylon reserves the right to make any changes to product without further notice. Our customers should ensure to take appropriate action so that their use of our products does not infringe upon any patents.

Copyright © Xylon d.o.o. 2017 All Rights Reserved

Copyright © Xylon d.o.o. 2017 All Rights Reserved

1 ABOUT THE FRAMEWORK

The logiADAK-VDF Video Design Framework enables Xylon logiADAK Automotive Driver Assistance Kit and logiVID-Z Vision Development Kit users to quickly utilize the provided hardware platforms for their own development of the Xilinx® Zynq®-7000 All Programmable SoC based multi-camera computer vision systems.

The framework includes pre-verified logicBRICKS reference designs for video capture from Xylon video cameras 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 reference designs prepared for the Xilinx Vivado Design Suite, and the SDSoC compatible designs are described in the logiADAK-VDF-SDSoC document.

The complete camera-to-display SoC 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 design framework users can immediately focus on specific vision-based parts of their next SoC design. Supported hardware platforms 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.

logiADAK-VDF reference designs include Xylon logicBRICKS IP cores and hardware design files prepared for Xilinx Vivado Design Suite. Hardware designers can customize designs and add their own IP cores through the Vivado IP Integrator (IPI). 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.

Figure 1: Xylon logiADAK Automotive Driver Assistance Kit

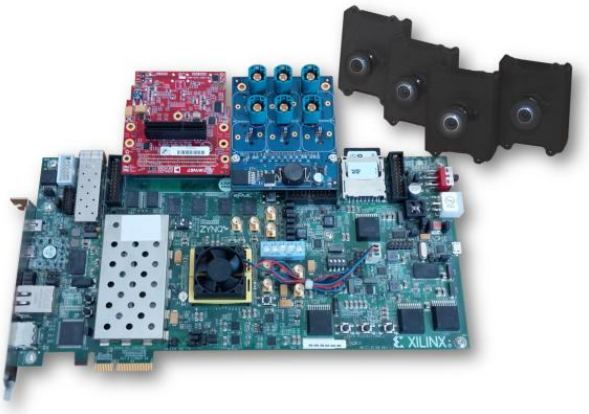


Figure 2: Xylon logiVID-Z Vision Development Kit

Copyright © Xylon d.o.o. 2017 All Rights Reserved

1.1 Programmable Logic Utilization

The logiADAK-VDF reference designs utilize just small fractions (Table 3) of available programmable logic in the Xilinx Zynq-7000 AP SoC XC7Z045 device. The 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[2]	IOB	BRAM	MULT/ DSP48/E	DCM / CMT	GTx	Design Tools
	mclk[4]	vclk[4]	rclk								
Zynq-7000[3] (XC7Z045-2)	(150/100)	(80/100)	100	5564	6343	56	17	13	2	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	GTx	Design Tools
	mclk[4]	vclk[4]	rclk								
Zynq-7000[3] (XC7Z045-2)	(150/100)	(80/100)	100	21484	25887	42	85	52	2	0	Vivado 2017.1

Notes:

1) Assuming the following configuration: ITU656, RGB565 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 1024 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-7000 SoC
4) logiCVC/logiWIN clock frequencies

Table 3: Free Programmable Logic Resources

	Available in XC7Z045	Used Resources	
		CAM-HDMI	FOUR-CAM
Flip Flops (FFs)	437,200	~ 2%	~ 6%
Look-Up Tables (LUTs)	218,600	~ 3%	~ 10%
Block RAM (36 kB BRAM)	545	~ 3%	~ 16%
DSP slices (MULT/DSP)	900	~ 1%	~ 6%

Copyright © Xylon d.o.o. 2017 All Rights Reserved

1.2 Hardware Requirements

The logiADAK Automotive Driver Assistance Kit (Figure 1) includes the following hardware, which is utilized by reference designs provided in the logiADAK-VDF design framework:

- 1x Xilinx Zynq-7000 SoC ZC706 Development Kit[1] with XC7Z045 FFG900 -2 AP SoC
- 1x Xylon FMC add-on board for up to 6 camera connections
- 5x Xylon Cameras
- 1x SD card
- 5x Rosenberger cables
- Power supply

Learn more about the logiADAK kit: http://www.logicbricks.com/Products/logiADAK.aspx

The logiVID-Z Vision Development Kit (Figure 2) includes the following hardware, which is utilized by reference designs provided in the logiADAK-VDF design framework:

- 1x Xilinx Zynq-7000 SoC ZC706 Development Kit[1] with XC7Z045 FFG900 -2 AP SoC
- 1x Xylon FMC add-on board for up to 6 camera connections
- 1x Avnet HDMI Input/Output FMC Module
- 4x Xylon Cameras
- 1x SD card
- 4x Rosenberger cables
- Power supply

Learn more about the logiVID-Z kit: http://www.logicbricks.com/Products/logiVID-Z.aspx

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

1.2.1 HDMI Input/Output FMC Module

The CAM-HDMI reference design delivered with the logiADAK-VDF requires the HDMI Input/Output FMC module (Figure 3), which is available with the logiVID-Z kit and not included with the Xylon logiADAK Kit. In order to use this reference design, the logiADAK kit users need to purchase the HDMI Input/Output FMC module from Avnet (Part Number: AES-FMC-HDMI-CAM-G).

Copyright © Xylon d.o.o. 2017 All Rights Reserved

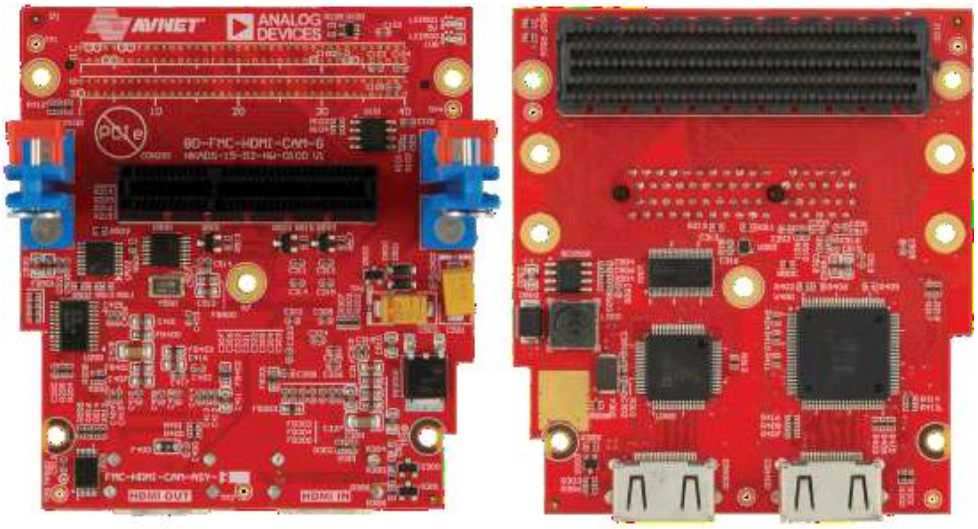


Figure 3: Avnet HDMI Input/Output FMC Module

1.2.2 Xylon Camera

For transmissions of the high-definition uncompressed video and camera control data, Xylon kits include hardware and software that is completely developed by Xylon: the LVDS-based serial interface, the Xylon video camera and an add-on LVDS FMC receiver board for up to six camera connections to the video processor implemented in the Xilinx Zynq-7000 AP SoC.

Each Xylon video 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, LVDS serializer (transmitter) board, the the Sunex DSL219 miniature fish-eye Wide FOV lens and a short cable lead with a connector. The logiADAK kit also includes the 5th camera equipped with the Sunex DSL947 Narrow Field of View (FOV) miniature lens, which is in Xylon ADAS demos used for the forward-looking collision avoidance and in-cabin driver status monitoring.

All camera parts are enclosed in the waterproof aluminium housing. The housing is sealed with rubber gaskets to ensure a weather-proof rating of IP67. Its rugged metal construction provides excellent lens and imager module protection and enables safe and easy test vehicle installations.

1.3 Software Requirements

The logiADAK-VDF reference designs and 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.

Copyright © Xylon d.o.o. 2017 All Rights Reserved

1.4 Design Deliverables

1.4.1 Hardware Design Files

- Configuration bitstream file for the programmable logic and the SDK export of the reference design that allows for instant design check-up and software changes
- Two reference designs prepared for the Xilinx Vivado Design Suite
- Xylon evaluation logicBRICKS IP cores:
    - logiCVC-ML Compact Multilayer Video Controller
    - logiWIN Versatile Video Input
    - logiI2C I2C Bus Master Controller
    - logiVLINK Vanilla LVDS Multimedia Data Link Receiver

1.4.2 Software

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

1.4.3 Binaries

- fsbl, fpga bitstream
- Linux binaries:
    - uboot, dtb, root file system
    - uImage
    - Camera/HDMI demo
    - Four Camera demo

1.5 Reference Designs

The logiADAK-VDF 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.

- 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 SoC. The design displays a single video source, and automatically switches to the HDMI video input upon detection of the plugged-in HDMI cable. This design requires the Avnet HDMI FMC add-on board, which is not a part of the logiADAK kit.

Copyright © Xylon d.o.o. 2017 All Rights Reserved

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.

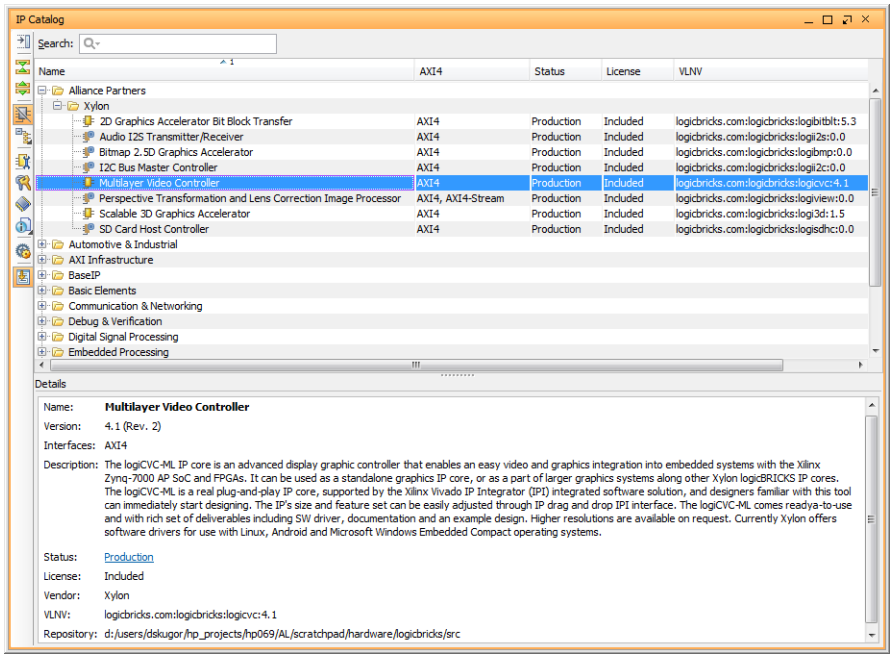


Figure 4: logicBRICKS IP Cores Imported into the Vivado IP Catalog

Copyright © Xylon d.o.o. 2017 All Rights Reserved

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

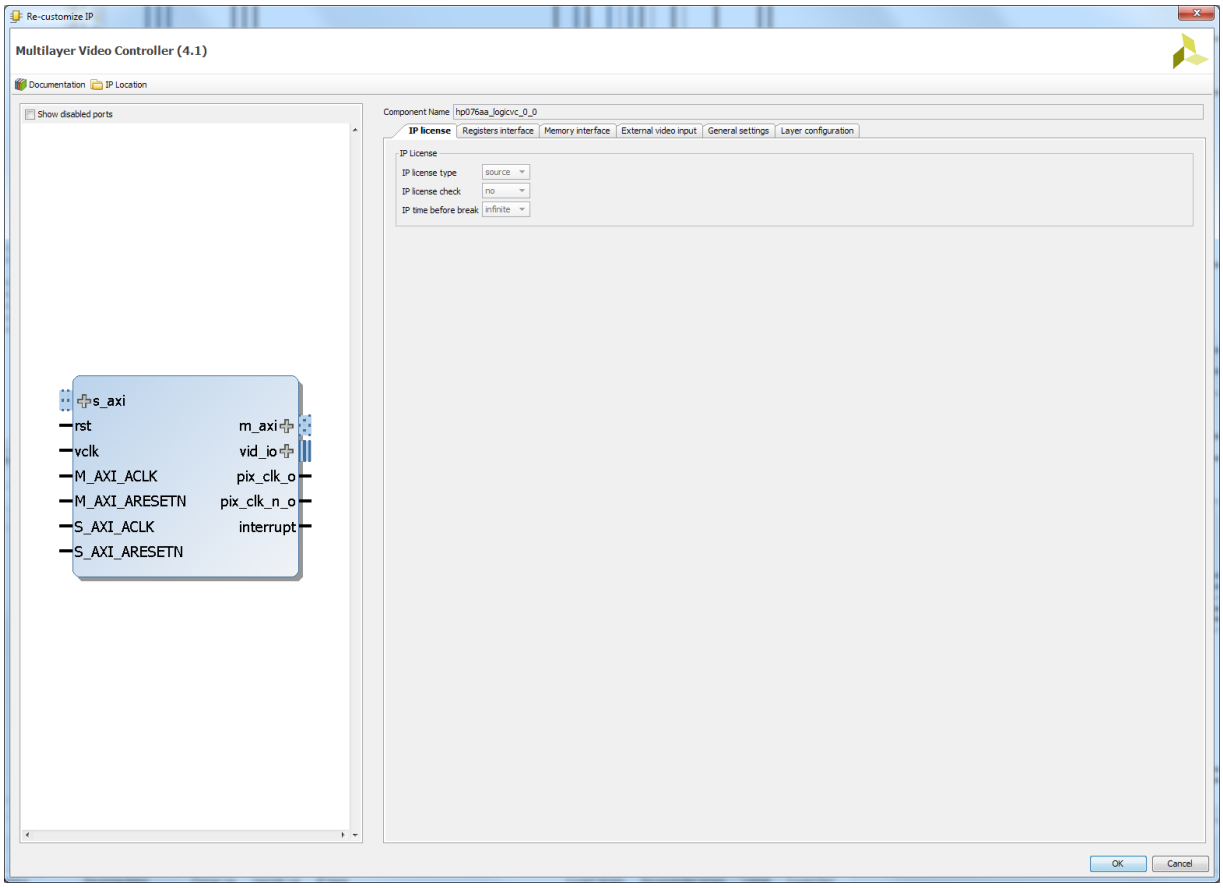


Figure 5: Example of logicBRICKS IP Configuration GUI

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

Copyright © Xylon d.o.o. 2017 All Rights Reserved

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 SoC/MPSoCs 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)
- 64x1 to 2048x2048 display resolutions
- 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 Hi-color RGB 565 and True-color 24bpp
  - YCbCr – 16bpp (4:2:2) and 24bpp (4:4:4)
- Configurable ARM® AMBA® AXI4 memory interface data width (32, 64 or 128)
- 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): 12x2-bit, 15-bit, 16-bit, 18-bit or 24-bit
  - YCbCr 4:4:4 or 4:2:2 output format
  - 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
- Supports synchronization to external parallel input

Copyright © Xylon d.o.o. 2017 All Rights Reserved

- HW cursors
- 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

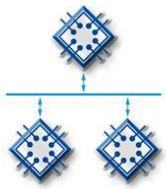
The logiWIN IP core enables easy implementation of video frame grabbers. Input video can be decoded, real-time scaled, de-interlaced, cropped, anti-aliased, positioned on the screen... Multiple logiWIN instances enable processing of multiple video inputs by a single Xilinx device.

- 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

Copyright © Xylon d.o.o. 2017 All Rights Reserved

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

2.3.4 logiVLINK Vanilla Multimedia Data Link Receiver

The logiADAK kit integrates proprietary logicBRICKS LVDS Camera Interface serial link for joint transmissions of high-speed video, audio and control data. Xylon ADAS solutions have been already successfully used in projects incorporating other video links (examples: TI FPD-Link III, Ethernet…) for video transmissions between different video cameras and the Zynq-7000 AP SoC or FPGA device. More info: info@logicbricks.com

2.4 logicBRICKS IP Cores for Video Processing

Xylon offers several logicBRICKS IP cores for video processing on Xilinx FPGA and SoC programmable devices:

logiVIEW Perspective Transformation and Lens Correction Image Processor

Removes fish-eye lens distortions and executes programmable transformations on multiple video inputs in a 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

Copyright © Xylon d.o.o. 2017 All Rights Reserved

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 Zynq-7000 All Programmable SoC and 7 Series FPGA 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

Copyright © Xylon d.o.o. 2017 All Rights Reserved

3 GET AND INSTALL THE DESIGN FRAMEWORK

To download the evaluation version and to purchase the design framework, please visit our online catalog: http://www.logicbricks.com/Products/logiADAK-VDF.aspx

3.1 Installation Process

Installation process is quick and easy. The logiADAK-VDF 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

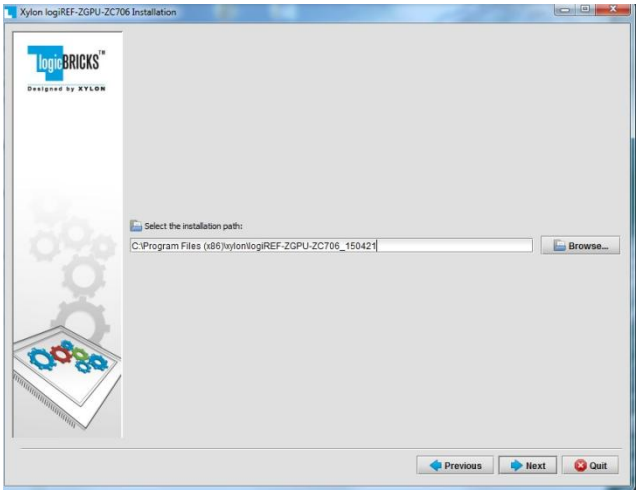


Figure 7: Installation Process – Step 2

Copyright © Xylon d.o.o. 2017 All Rights Reserved

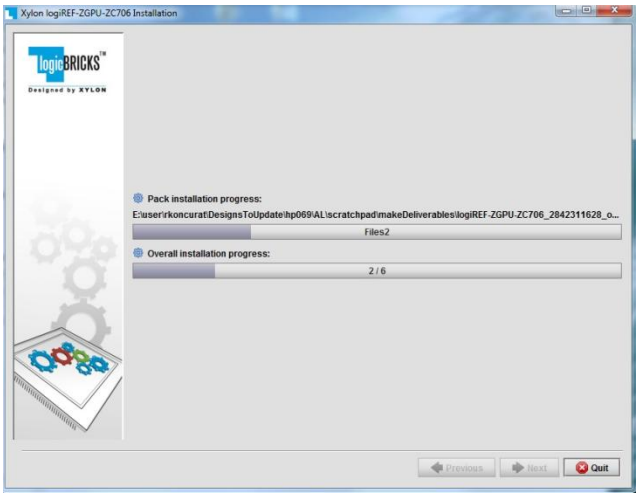


Figure 8: Installation Process – Step 3

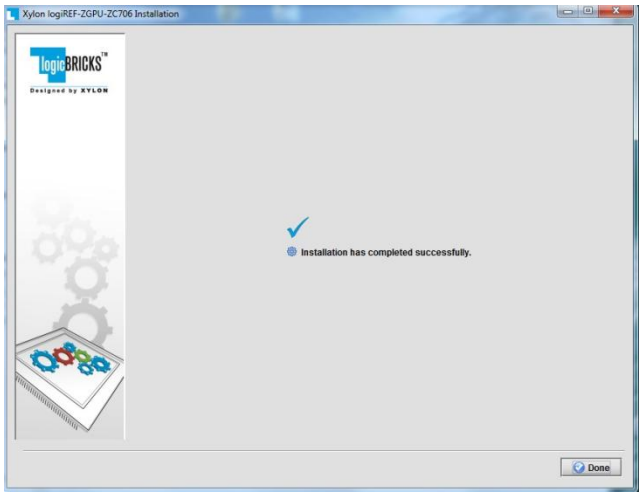


Figure 9: Installation Process – Step 4

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.

Copyright © Xylon d.o.o. 2017 All Rights Reserved

Folder Structure

Figure 10 and the **Error! Reference source not found.** explains the folder structure of the logiADAK-VDF-SDSoC video design framework.

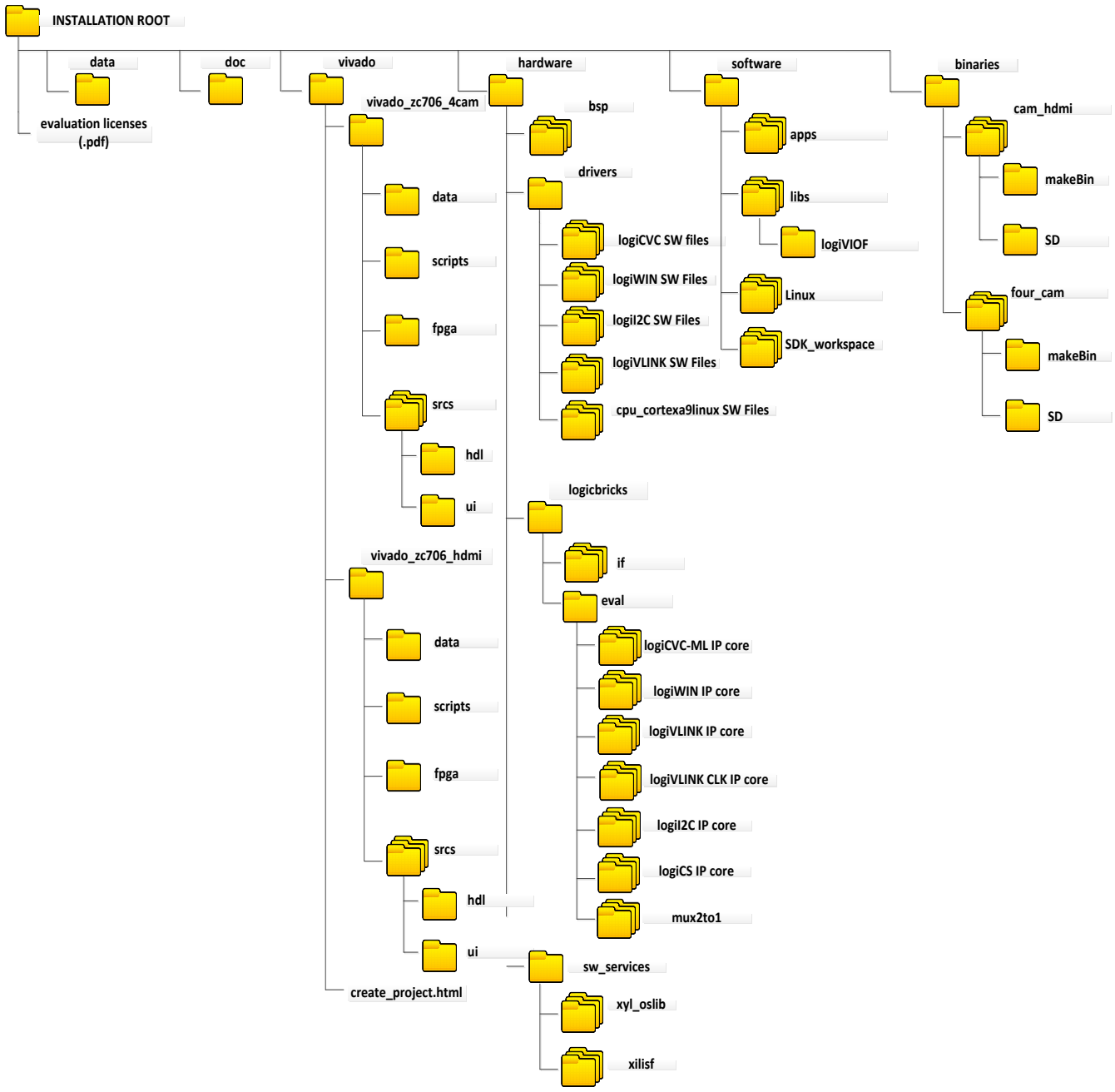


Figure 10: The Folder Structure

Copyright © Xylon d.o.o. 2017 All Rights Reserved

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.
vivado		
vivado_zc706_hdmi		This folder contains the complete Vivado project and files necessary for regenerating project from TCL scripts.
	data	Design constraints files (XDC).
	srcs	Block design GUI script and HDL wrappers.
	scripts	TCL scripts to create block design from scratch.
	fpga	ZC706 reference design bitstream.
vivado_zc706_4cam		This folder contains the complete Vivado project and files necessary for regenerating project from TCL scripts.
	data	Design constraints files (XDC).
	srcs	Block design GUI script and HDL wrappers.
	scripts	TCL scripts to create block design from scratch.
	fpga	ZC706 reference design bitstream.
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.
	SDK_workspace	Xilinx SDK workspace folder for building bare-metal applications.
binaries		
	cam_hdmi/makeBin	Utility script for creating boot.bin file.
	cam_hdmi/SD	Prepared binaries ready for download.
	four_cam/makeBin	Utility script for creating boot.bin file.
	four_cam/SD	Prepared binaries ready for download.

Table 4: Explanation of the logiADAK-VDF Folder Structure

Copyright © Xylon d.o.o. 2017 All Rights Reserved

4 GETTING LOGICBRICKS IP LICENSES

The logiADAK-VDF 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 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.

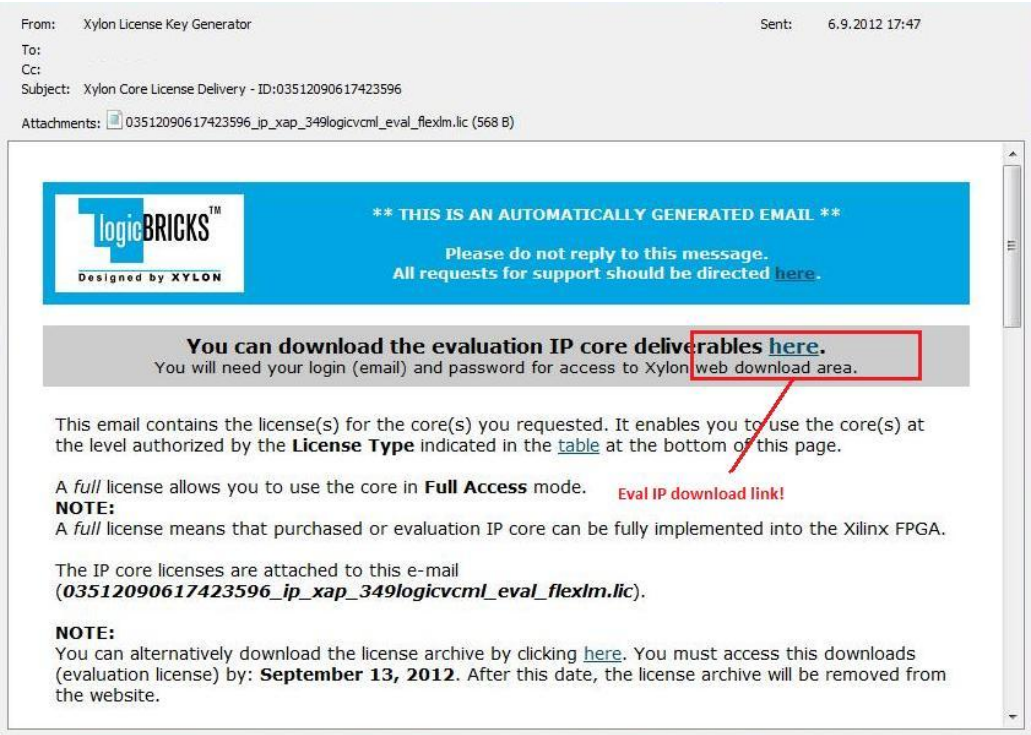


Figure 11: E-mail with logicBRICKS License and Download Instructions

Copyright © Xylon d.o.o. 2017 All Rights Reserved

5 LOGIADAK-VDF REFERENCE DESIGNS

5.1 CAM-HDMI SoC Design and Memory Layout

This SoC 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.

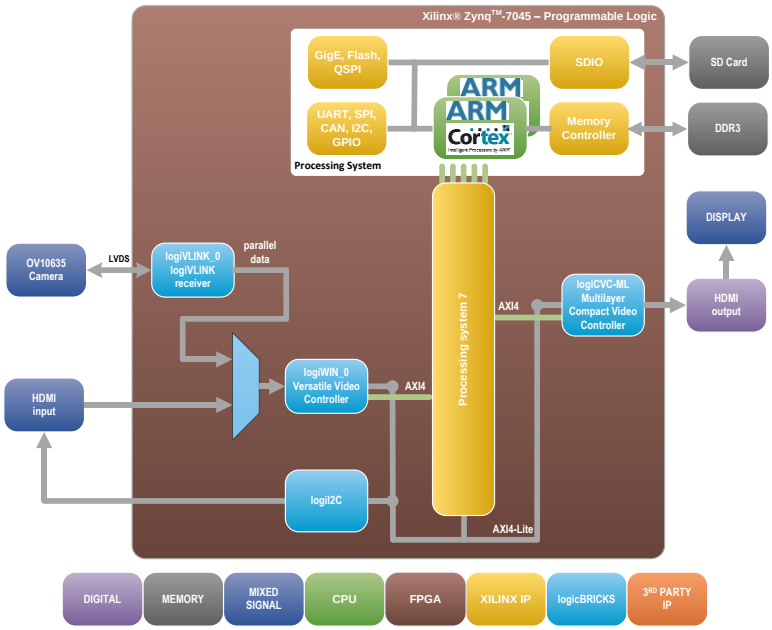


Figure 12: CAM-HDMI SoC Design – Block Diagram

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

The Figure 13 shows the memory layout of video buffers. Each logiCVC-ML layer has its own reserved memory space and use multiple video buffers (not shown).

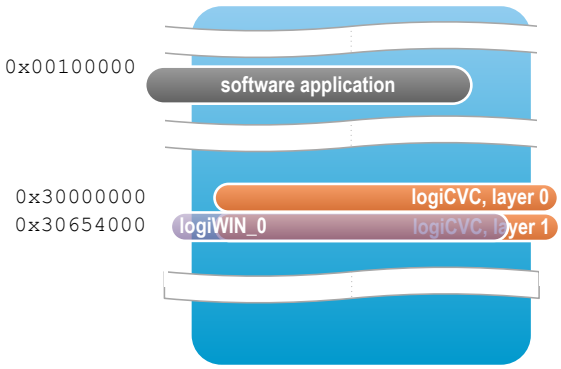


Figure 13: CAM-HDMI Design Video Memory Layout

Copyright © Xylon d.o.o. 2017 All Rights Reserved

5.2 FOUR-CAM SoC Design and Memory Layout

This SoC (Figure 14) design supports four camera inputs, and the single display output.

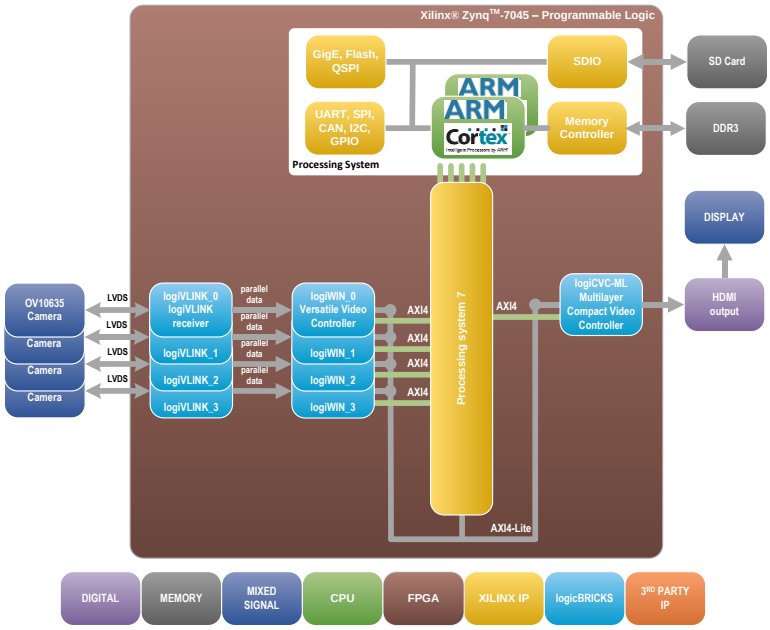


Figure 14: FOUR-CAM SoC Design– Block Diagram

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

The Figure 15 shows the memory layout of video buffers. Each logiCVC layer has its own reserved memory space and multiple video buffers (not shown).

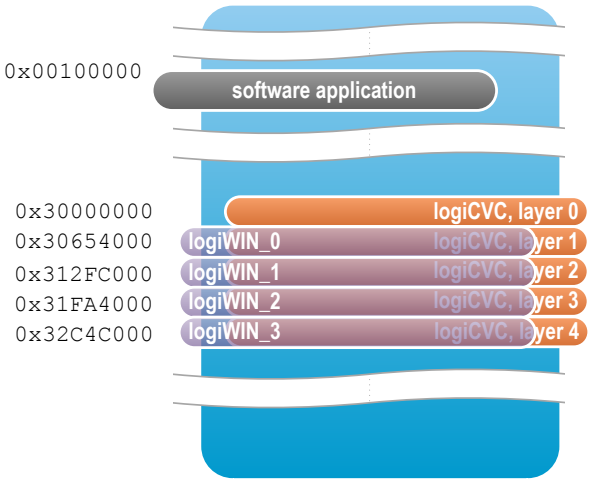


Figure 15: FOUR-CAM Design Memory Layout

Copyright © Xylon d.o.o. 2017 All Rights Reserved

5.3 Video Input/Output Synchronization

Both designs use hardware synchronization implemented between logiWIN and logiCVC-ML IP cores. SoC systems implementing video input units has video input frame rates and video output frame rates rarely equal and must cope with frame rate conversions from lower to higher frame rate, or vice versa.

5.3.1 logiWIN Hardware Buffering Implementation

Double/triple buffering state machine is placed outside the logiWIN. The logiWIN output pin *curr_vbuff[1:0]* sends to the double/triple buffer external controller information to which buffer the logiWIN currently writes. On the input pin *next_vbuff[1:0]* the double/triple buffer external controller sends to the logiWIN information to which buffer the next frame should be written. For double buffering *next_vbuff* value changes between 0 and 1, and for triple buffering between 0, 1 and 2. If double/triple buffering is not in use, the *next_vbuff* must be set to 0. Output signal *sw_vbuff_req* signals to the external controller that the current buffer *curr_vbuff[1:0]* is written and requests buffer switching. External controller grants buffer switching over *sw_vbuff_grant* together with the pointer to the next buffer *next_vbuff[1:0]*.

5.3.2 logiCVC-ML Hardware Buffering Implementation

External video synchronization requires three separate frame buffers (buffer_0, buffer_1 and buffer_2 implemented in the video memory). In SoC designs with the logiCVC-ML display controller IP core, three frame buffers must be setup for every logiCVC-ML graphic layer. The triple buffering method provides an advantage over the double buffering synchronization method, since the video input units do not have to wait on buffers swapping as there is always a spare frame buffer for new frame data writing.

To support this feature, the logiCVC-ML uses video input synchronization control port which consists of *e_curr_vbuff*[C_NUM_OF_LAYERS*2-1:0] and *e_switch_vbuff*[C_NUM_OF_LAYERS-1:0] input signals, and *e_next_vbuff* [C_NUM_OF_LAYERS*2-1:0] and *e_switch_grant*[C_NUM_OF_LAYERS-1:0] output signals.
With the input signals *e_current_vbuff*[n*2+1:n*2] and *e_switch_vbuff*[n] external video source signals to logiCVC-ML layer n the currently written buffer and when to switch buffers (typically on the end of its active frame of external video source). With the output signal *e_switch_grant*[C_NUM_OF_LAYERS-1:0] the logiCVC-ML grants the video source to start writing its next frame to *e_next_vbuff*[n*2+1:n*2] buffer.

The logiCVC-ML IP core is constantly sampling *e_current_vbuff* and *e_switch_vbuff* inputs with the memory clock. When *e_switch_vbuff* high state is detected, the logiCVC-ML samples *e_current_vbuff* and asserts *e_switch_grant* along with the associated *e_next_vbuff*. External logic should constantly sample *e_switch_grant* signal, and when it detects that *e_switch_grant* is high, it should sample *e_next_vbuff* and de-assert *e_switch_vbuff*. When logiCVC-ML detects *e_switch_vbuff* low, it de-asserts *e_switch_grant* signal on the next memory clock cycle. *e_switch_vbuff* and *e_switch_grant* signals are used as handshake signals between logiCVC-ML and external logic. This kind of implementation supports buffers switching between logiCVC-ML and external logic running on synchronous and on asynchronous clocks.

Copyright © Xylon d.o.o. 2017 All Rights Reserved

To enable external frame buffer synchronization for a specific graphic layer, user has to enable it by setting the EN_EXT_VBUFF_SW bit to 1 in the corresponding layer control register.

If external video input signals are connected to the logiCVC-ML's video input synchronization control port and synchronization are turned off (EN_EXT_VBUFF_SW=0), logiCVC-ML will always signal the external video input to write data to buffer 0, i.e. *e_next_vbuff*[n*2+1:n*2]=0. At the same time, logiCVC-ML will work in the CPU synchronization mode so it will read memory buffer, which is defined with layer address register.

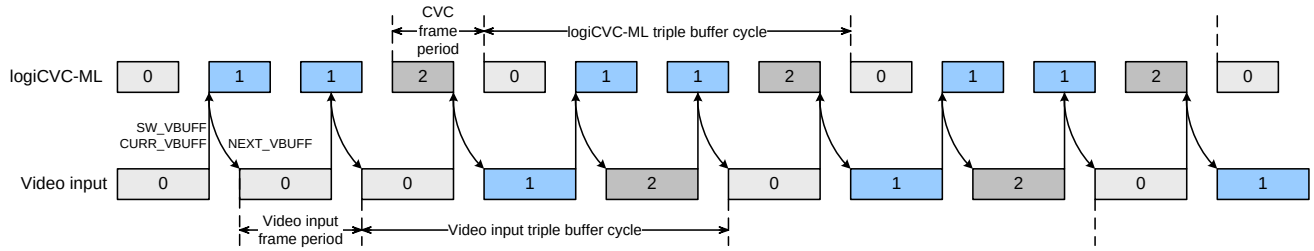


Figure 16: Triple buffering example when logiCVC-ML refresh rate is higher than video input

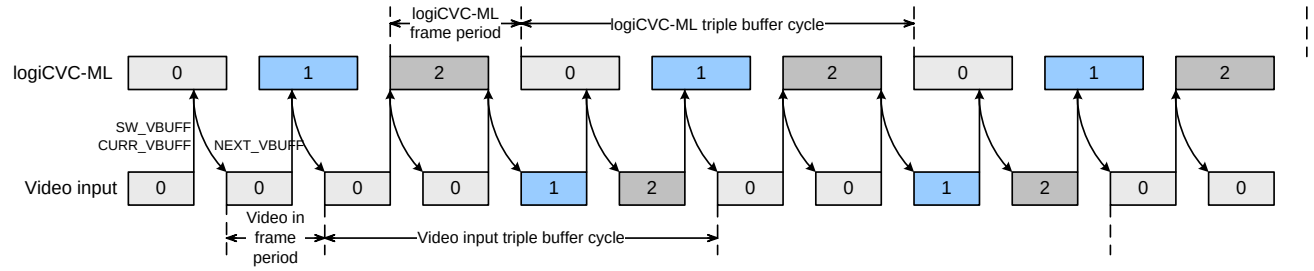


Figure 17: Triple buffering example when logiCVC-ML refresh rate is lower than video input

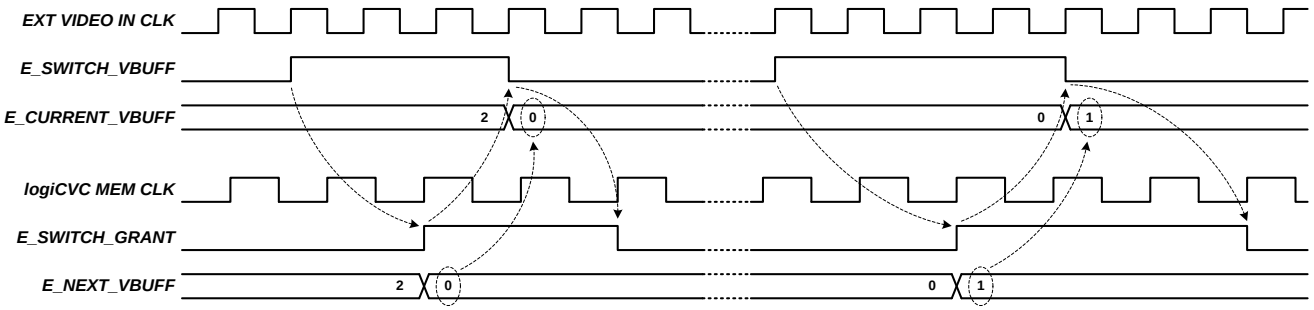


Figure 18: External buffer control signals timing diagram

Copyright © Xylon d.o.o. 2017 All Rights Reserved

5.4 Restoring Full SoC Design from Xylon Deliverables

Xylon provides all necessary design files and TCL scripts to enable full project restore in the Xilinx Vivado Design Suite 2017.1. To regenerate the Vivado designs, please go to Windows Start button and start the Vivado 2017.1 Tcl Shell.

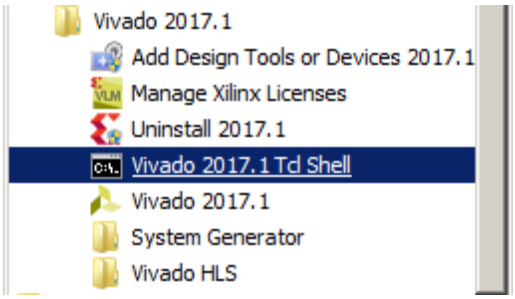


Figure 19: Vivado 2017.1 Tcl Shell

Navigate to either ***logiADAK-VDF_vX_Y_Z/vivado/vivado_zc706_hdmi/scripts*** folder for the CAM-HDMI Design, or to ***logiADAK-VDF _vX_Y_Z/vivado/vivado_zc706_4cam/scripts*** folder for the FOUR-CAM design.

To regenerate the selected design, run the `project.tcl` script (type `source project.tcl` to Vivado Tcl Shell as shown by Figure 20). Upon the design regeneration, the Vivado GUI will be started and the Block Diagram will be opened as shown in Figure 21.

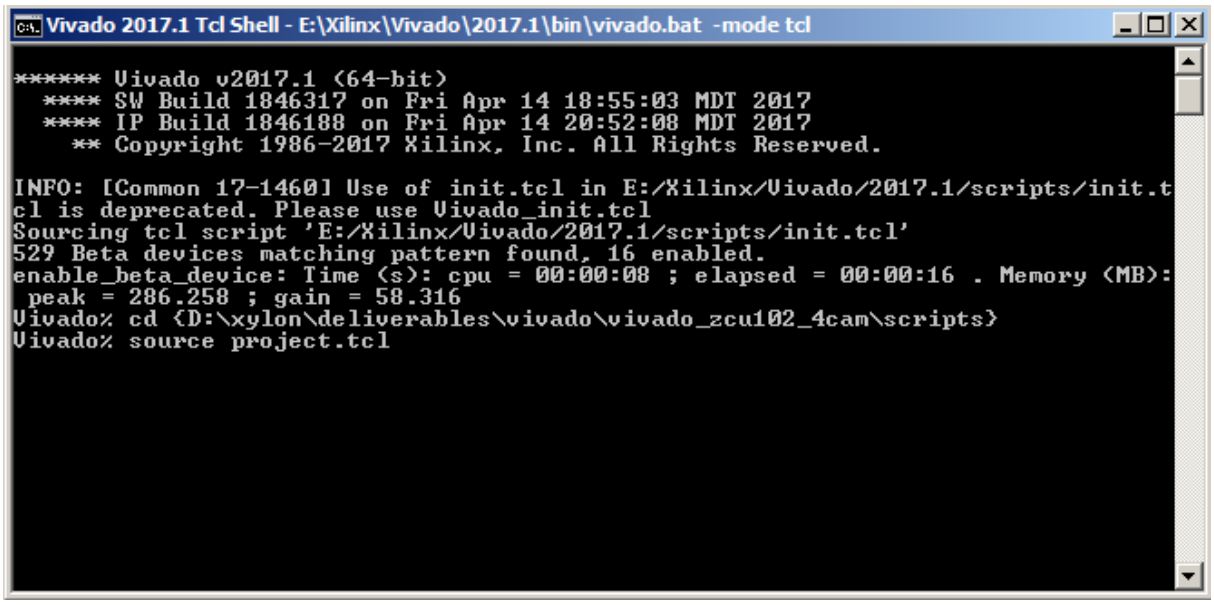


Figure 20: Vivado 2017.1 Tcl Shell – Regenerating the Project Example

Copyright © Xylon d.o.o. 2017 All Rights Reserved

Please note that design regeneration takes some time. Also, during the regeneration process, the Vivado GUI may close and re-open, so do not be alarmed by these events.

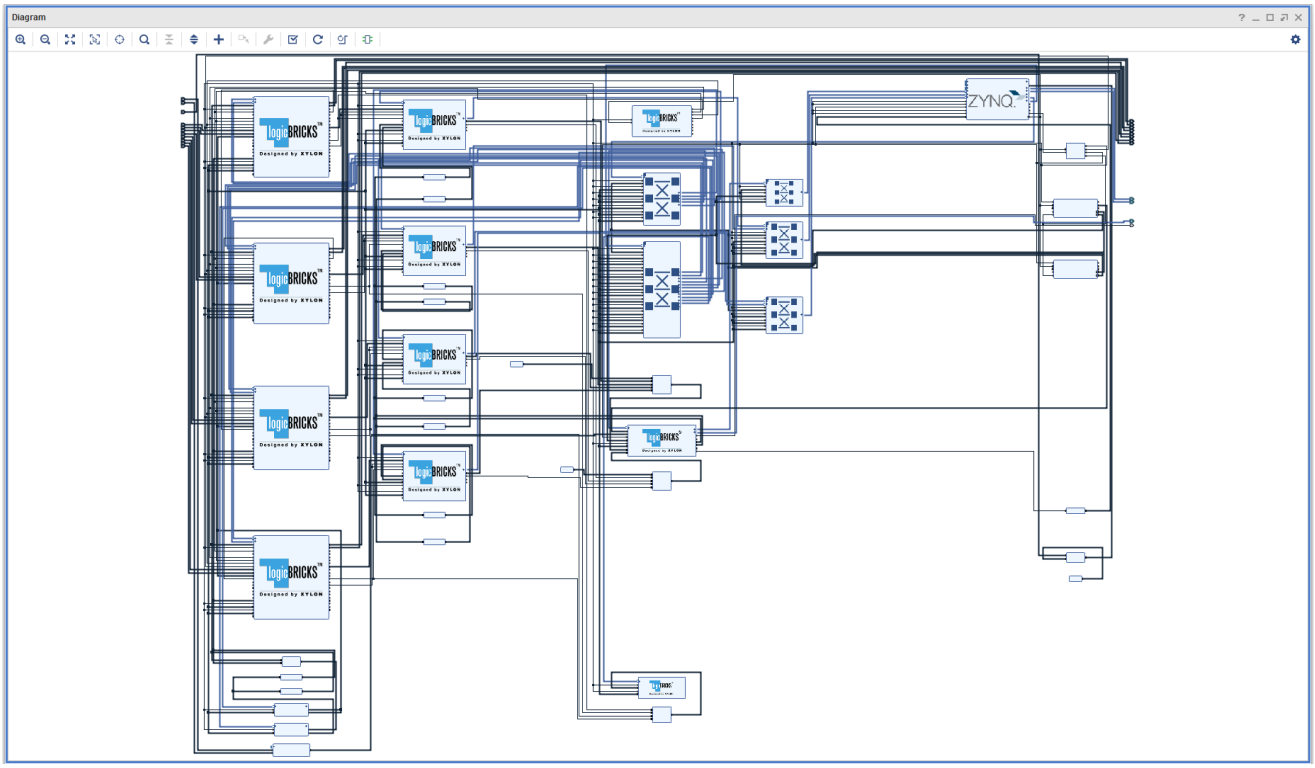


Figure 21: FOUR-CAM Vivado IP Integrator Block Diagram

Copyright © Xylon d.o.o. 2017 All Rights Reserved

5.5 SOFTWARE DESCRIPTION

Software components:

- v4l2 driver, https://github.com/logicbricks/driver_fb_logicvc, 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

Software architecture is shown in the picture below.

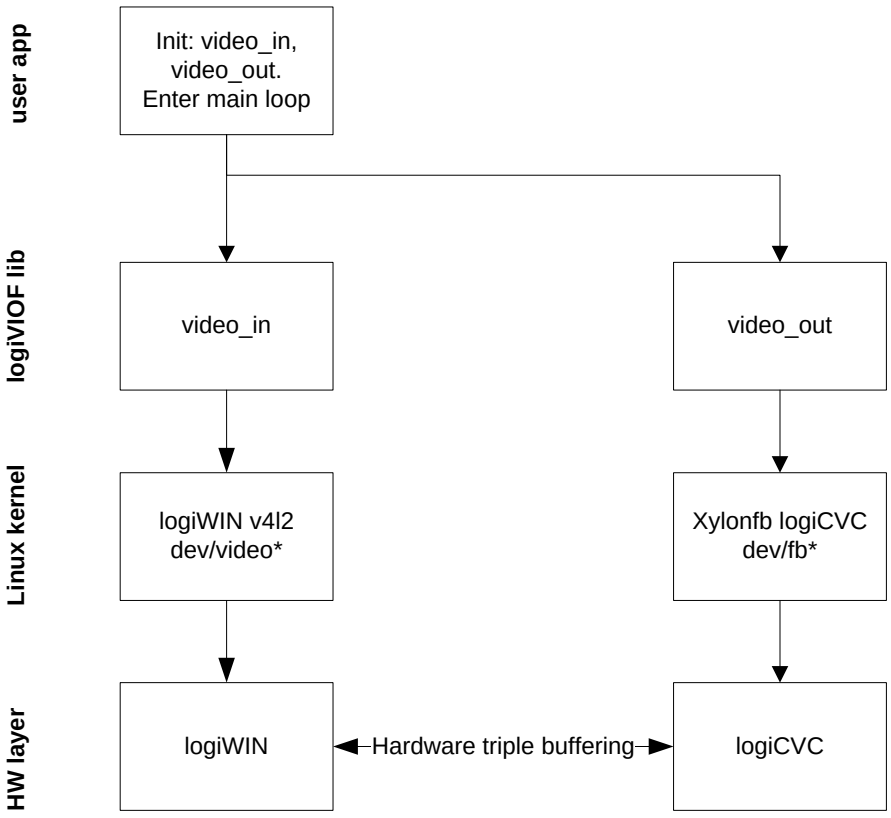


Figure 22: Software architecture

Copyright © Xylon d.o.o. 2017 All Rights Reserved

5.5.1 Demo applications

There are two demo applications:

- FOUR-CAM : demonstrates the usage with four cameras
- CAM-HDMI : demonstrates the usage with one camera/HDMI video input

Both applications use the same C source code, 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 video from an external video input by using the hardware buffering synchronization.

Video inputs and outputs are initialized by the logiVIOF library (Figure 22). For details about this software library, please check: /software/libs/logiVIOF/doc/html/index.html.

Main functions used in the demo:

- 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
   ▪ vin_init               //create logiVIOF VideoOut interface to control
                            //framegrabber (xylon v4l2 dev/video*)
   ▪ vin_enable             //framegrabber start
```

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_layerSetPositionAndSize  //depending on mode set layer geometry
                                   //for one or all cameras
   ▪ vout_layerEnable       //depending on mode enables one
                            //or all four layers
```

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.

Copyright © Xylon d.o.o. 2017 All Rights Reserved

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:
   ▪ 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:
   ▪ vin_deinit                  // deinitializes VideoIn
   ▪ vout_deinit                 // deinitializes VideoOut
```

5.5.2 Input resolution and the frame rate

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

Copyright © Xylon d.o.o. 2017 All Rights Reserved

6 QUICK START

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/bin* folder to the root folder of the SD card.

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

SD card should be formatted as FAT32.

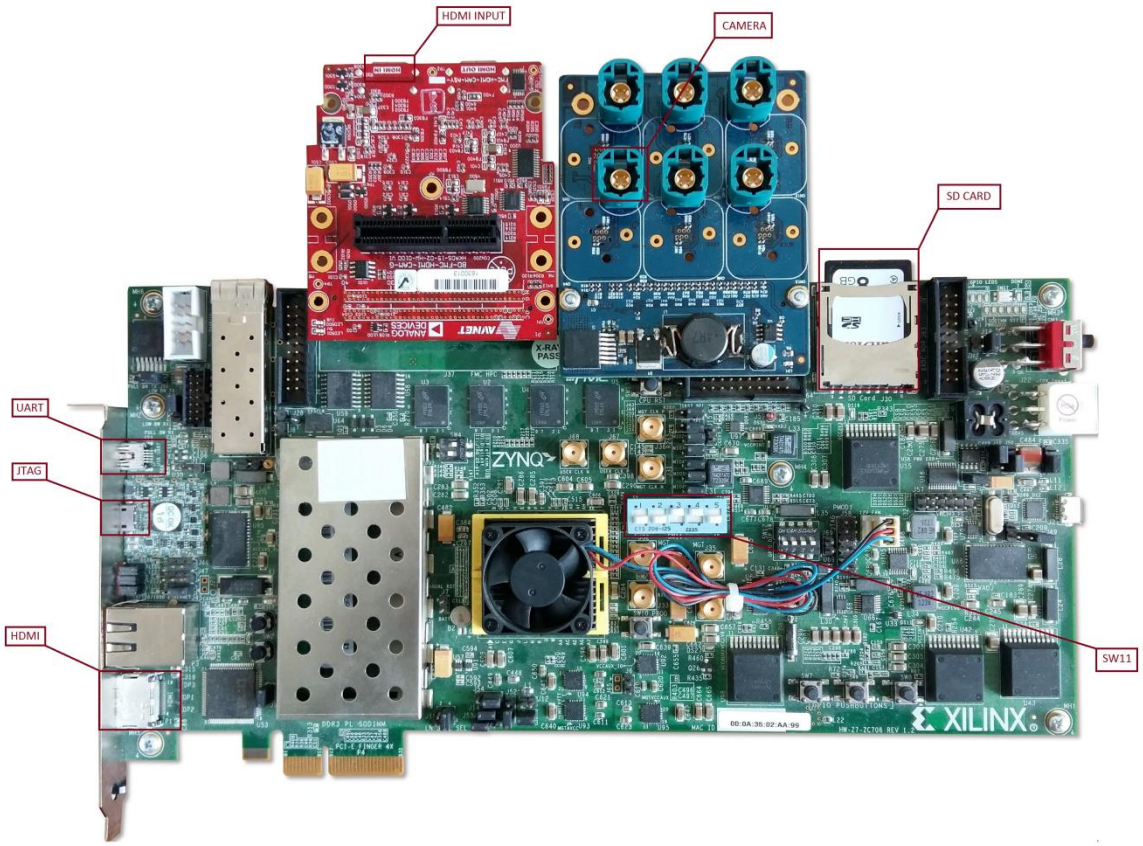


Figure 23: CAM-HDMI HW Setup

Copyright © Xylon d.o.o. 2017 All Rights Reserved

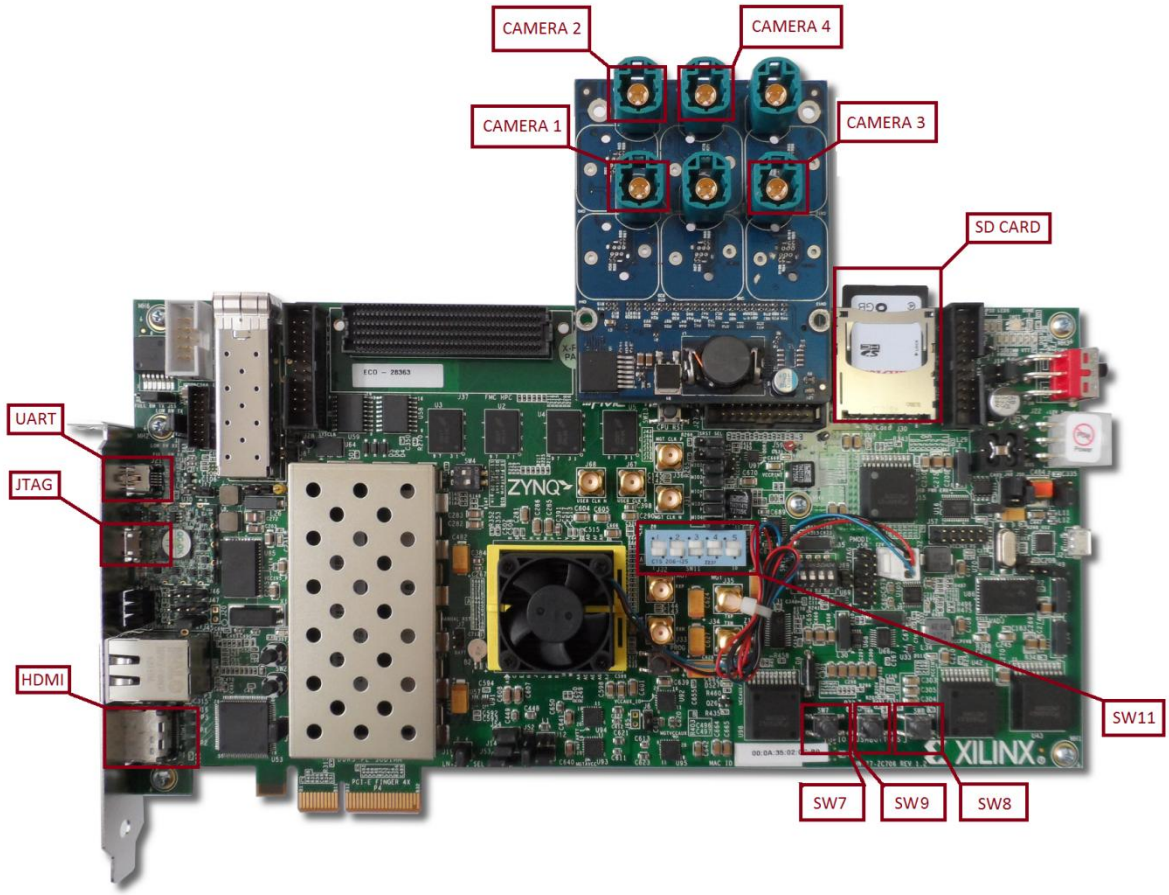


Figure 24: FOUR-CAM HW Setup

Optionally, you can use a serial terminal program (baud rate 115200 8N1) and the USB UART connection to the ZC706 board to monitor the system's operation.

For full explanation of the ZC706 board's features and settings, please check the documentation Xilinx UG954.

6.2 Demo controls

Start the FOUR-CAM demo design and the display output 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 left (SW7) or right (SW8) push-buttons on the ZC706 board. Toggle demo mode between one camera, or all cameras by pressing the number '5' on a keyboard, or by pressing central (SW9) push-button on the ZC706 board. Stop the application by pressing 'q' on the keyboard.

Copyright © Xylon d.o.o. 2017 All Rights Reserved

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 on 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 video design framework reference designs and 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 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-7000 All Programmable SoC can be obtained from the Xilinx ARM GNU Tools wiki page: http://wiki.xilinx.com/zynq-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 software applications are delivered in the source code to enable users to do software customizations. This paragraph explains how to setup the Xilinx SDK environment for software customizations.

Quick steps required to set up the SDK workspace:

1. If the CAM-HDMI SoC design is modified, open the recreated and modified ***CAM-HDMI Vivado design,*** go to: File → Export hardware (select option „*Include bitstream")*

Copyright © Xylon d.o.o. 2017 All Rights Reserved

2. If the FOUR-CAM SoC 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_vX_Y_Z/software/SDK_workspace***
4. In the SDK go to: *Xilinx Tools → Repositories → New,* add ***logiADAK-VDF_vX_Y_Z/hardware*** directory
5. In the SDK go to: *Project* and exclude *Build automatically* (optional, but recommended)
6. In the SDK go to: *File → Import → General → Existing projects to workspace → Next,* in *Select a root directory* choose ***logiADAK-VDF_vX_Y_Z/software/SDK_workspace***, select all projects and click *Finish*
7. *Each BSP needs to be regenerated. Right click on BSP and select Re-generate BSP Sources. Do this for all BSPs.*
8. If the *Build automatically* option has been disabled, build all imported applications manually

If the error "**cannot find -lrsa**" occurs when building the **fsbl_zynq_**, please re-generate the **bsp_fsbl_zynq_** and build the **fsbl_zynq_** again.

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 the Linux Framebuffer driver. Zynq 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
- Running Linux applications with the ZC706 board setup for the precompiled SD card image

6.5 Debugging Software Application with the TCF Agent

1. Launch the application SDK workspace
2. Open **`Debug Configurations`** window and **`Run -> Debug Configurations`**…
3. To create a new Debug Configuration, double click on the **`Xilinx C/C++ application (System Debugger)`** section on the left hand side of the Debug Configuration GUI
4. Set **`Debug Type`** to **`Linux Application Debug`**

Copyright © Xylon d.o.o. 2017 All Rights Reserved

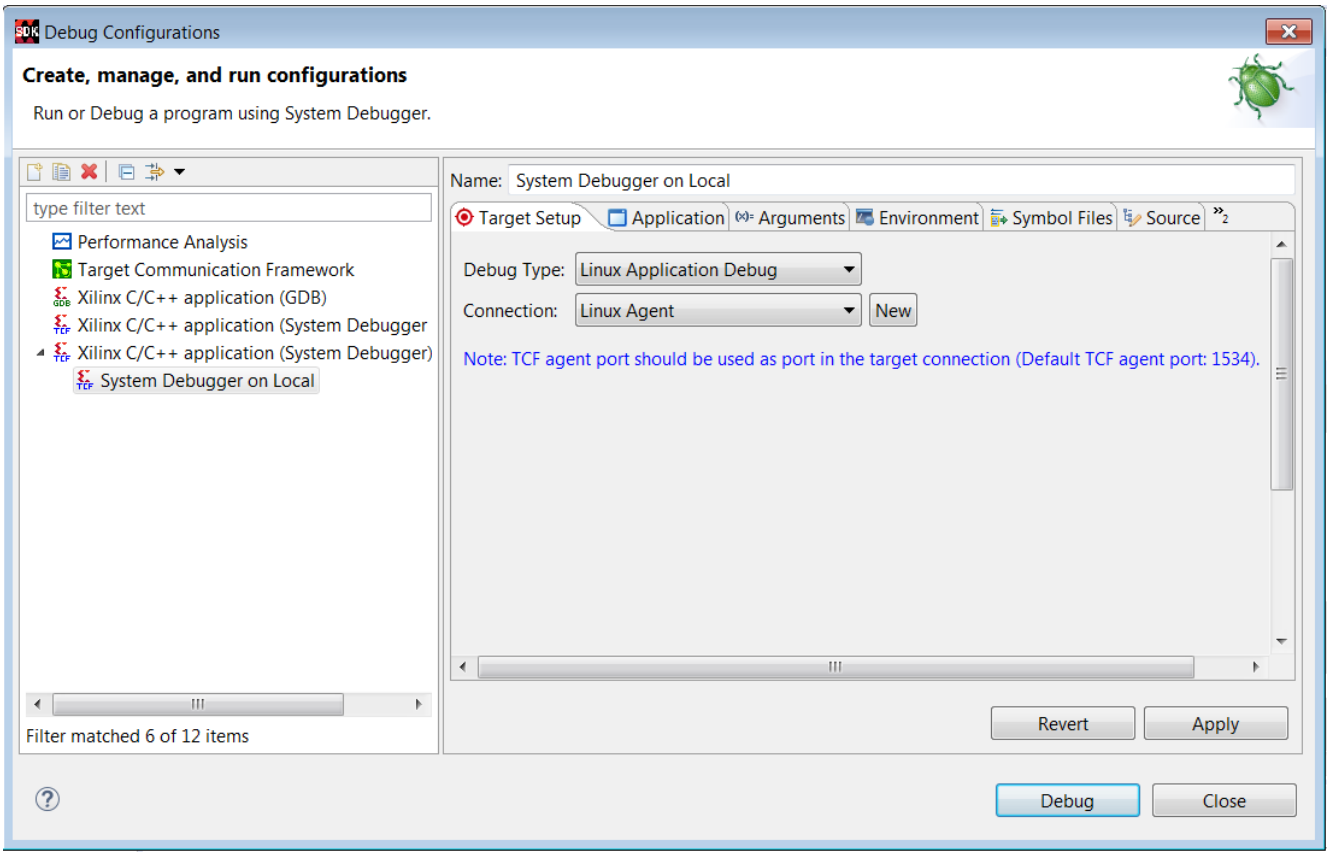


Figure 25: SDK Workspace – Debug Configurations

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

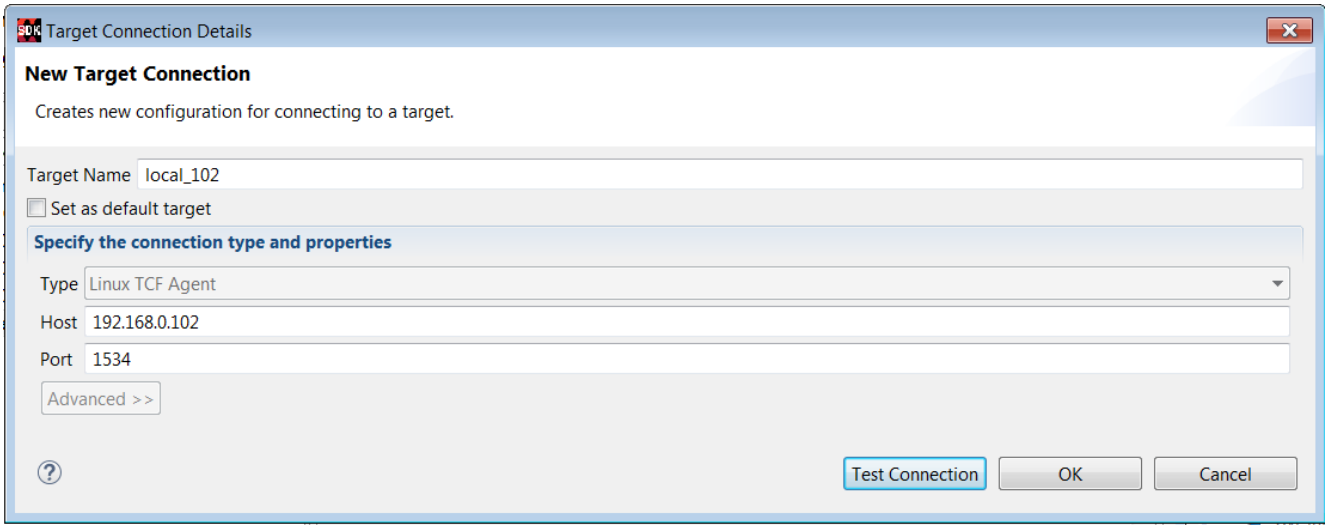


Figure 26: SDK Workspace – Create the new Connection

Copyright © Xylon d.o.o. 2017 All Rights Reserved

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

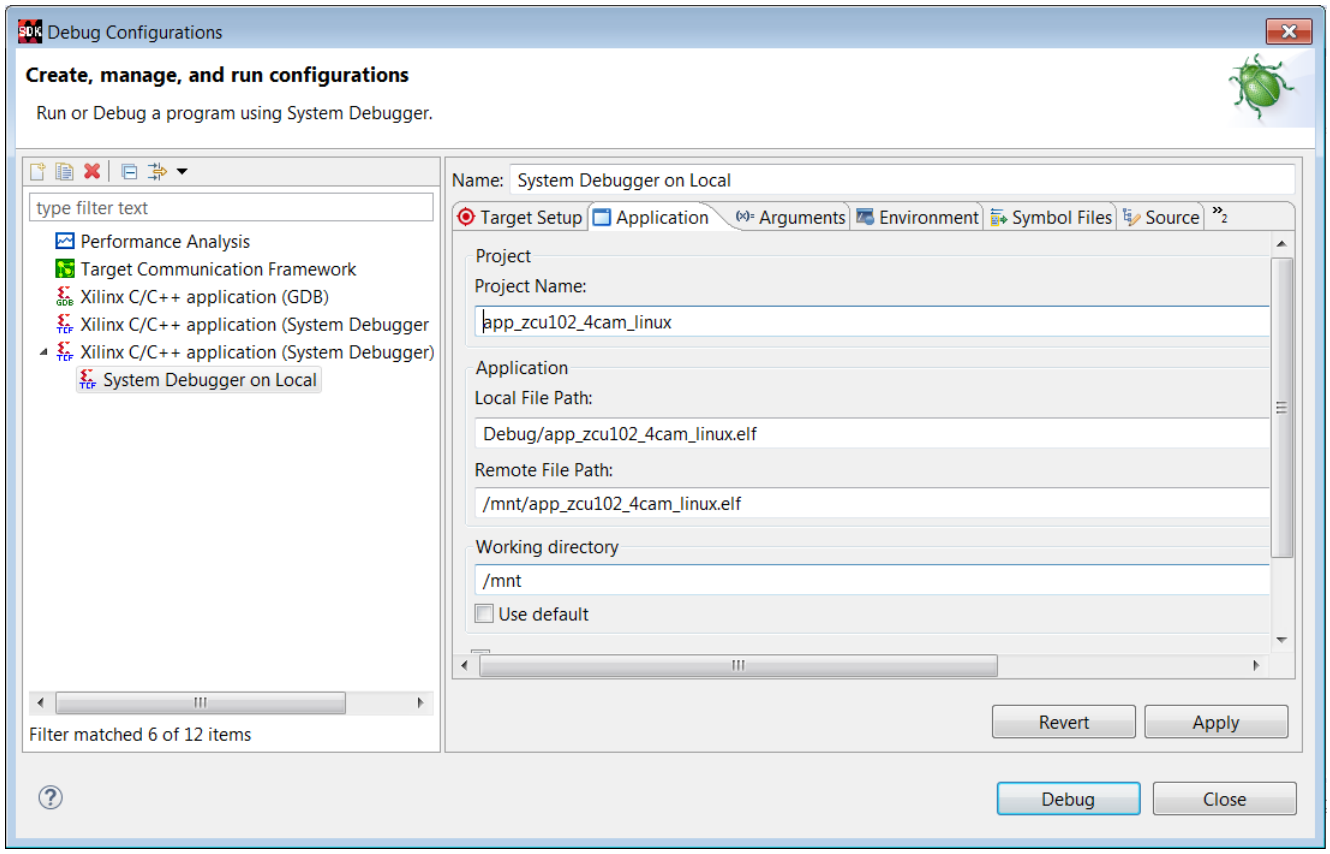


Figure 27: SDK Workspace – Application Tab

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

Copyright © Xylon d.o.o. 2017 All Rights Reserved

7 REVISION HISTORY

Version	Date	Author	Approved by	Note
1.00.a	March 30th, 2016	A.Bogdanic	G.Galic	Initial
1.00.b	May 2nd, 2016	D.Štimac	A.Bogdanic	Added chapter: "Setting up the SDK workspace" Updated folder structure block diagram and table content.
1.01.a	July 15th, 2016	A.Bogdanic		Version of Vivado Development Tool changed to 2016.2 Added External video input synchronization description.
2.00.a	November 21st, 2016	A.Bogdanic		Project version changed to 2.00.a to match the SDSoC project version.
3.00.a	July 19th, 2017	D. Štimac	G.Galic	Version of Vivado Development Tool changed to 2017. Added logiVIOF decription. Revised software description chapter. Added chapter 6.5 Debugging Software Application with the TCF Agent

Copyright © Xylon d.o.o. 2017 All Rights Reserved