Zynq-7000 SoC ZC702 Base Targeted Reference Design (Vivado Design Suite 2015.4)

User Guide

Vivado Design Suite

UG925 (v2015.4) July 1, 2018

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
06/21/2012	1.0	Initial Xilinx release.
10/08/2012	2.0	Updated for ISE® Design Suite 14.2. Replaced AXI VTC with VTC throughout. Changed Sobel engine to Sobel filter. Chapter 1, Introduction: Added R, RW, COR, SoC, and SC to Table 1-1. Expanded descriptions of the TRD. Updated the block diagram in Figure 1-1. Chapter 2, Functional Description: In Figure 2-1, replaced the continuous line with a dotted line for Monitor and between VIDEO_MUX0 and Clk_detect and revised AXI Interconnect inputs. Added slice registers to Table 2-1. In Table 2-2, changed SOBEL_ENGINE to FILTER_VDMA and FILTER_ENGINE. Added FILTER_0_interrupt row in Table 2-3, and SOBEL_VDMA changed to FILTER_VDMA. In Table 2-4, GPIO bit number 7 became N/A. In PL Clocks, page 19, (FCLKCLK[3:0]) became (FCLK_CLK). In PL Reset, page 19, (FCLKRESETN[3:0]) became FCLK_RESET[3:0]_N and FCLKRESETN[0] changed to FCLK_RESET0_N. In Clocking, page 19, PS FCLKCLK[0] changed to PS FCLK_CLK0. In Table 2-5, clock signal sources and uses were updated. In Table 2-9, added xFilter, changed frequency and connection of s_axi_CONTROL_BUS_ACLK and deleted s_axi_SOBEL_CONTROL_ACLK. Changed frequency of INTERCONNECT_ACLK bus. Changed SOBEL_ENGINE to FILTER_ENGINE and SOBEL_VDMA to FILTER_VDMA. Added information to the end of the table for DVI2AXI_0, CRESAMPLE_0, and YUV2RGB_0 components. In Processor System Reset Module, page 22, input to the proc_sys_reset core is generated by PS Proc_sys_reset_1_N. In Clock Detect, page 23, FMC changed to FMC-IMAGEON. The v_cresample, v_ycrcb2rgb, vsrc_sel, sobel_filter_top, and dvi2axi sections were added. Starting in fmc imageon hdmi in, page 25, YCbCr was changed to YCrCb. In Figure 2-4, added xFilter to the Device Drivers. In the Boot Loader section, removed "HDMI in" chip. In xFilter, page 30, updated the introductory text for clarity and reformatted the IOTCL arguments. Modified the description of Figure 2-6 for clarity. Modified the descriptions of Figure 2-8 and Figure 2-9 for clarity and added description of cases 4, 5, and 6. Modified Software Sobel Filter Processing, page 45 for clarity. Added xFilter sections to Xilinx Linux Kernel, page 33. Figure 2-4, GUI for TRD Application was replaced and notes about the GUI were added. AXI was added to the Plot Graph descriptions. DDR3 was removed from Figure 2-8 and Figure 2-9. The Software Sobel Filter Processing, page 45 description and API changed. Appendix A, Register Description: New sections were added from Sobel Filter Registers, page 46 to the end of the appendix. Removed redundant instances of TPG. Appendix C, Directory Structure: Figure C-1, Directory Structure, was updated. Appendix F, Additional Resources: PG012, *LogiCORE IP Chroma Resampler Product Guide* and PG014, *LogiCORE IP YCrCb to RGB Color-Space Converter Product Guide* were added to references. Appendix G, Regulatory and Compliance Information was added. Appendix H, Warranty was added.

Date	Version	Revision
11/12/2012	2.1	Updated for ISE® Design Suite 14.3. Chapter 1, Introduction: Document and web site references changed throughout the book. In Base TRD Key Features, page 11, *1 GB DDR3 running at 533 MHz* was removed. Chapter 2, Functional Description: Figure 2-1 line colors changed. Table 2-1 cell values changed. Table 2-2 Peripheral names changed. Under Clocking, page 19, RGB2YUV block was removed. In Table 2-6, the processor is ps7_0, and PERF_MON_HP0, PERF_MON_HP2, VTC_0, and DVI2AXI_0 sections changed. The *dvi2axi Converter* heading changed to *dvi2axi* on page 25. Figure 2-6 was replaced and a GUI explanation added. Figure 2-7 was added to show minimized GUI mode. In section Graphical User Interface, page 41, the File Browser and Command Line Shell paragraphs were removed. The directory structure changed in Figure C-1. Appendix F, Additional Resources: Appendix was reorganized. A link was added for the Master Answer Record for the ZC702 board. Appendix G, Regulatory and Compliance Information: A Declaration of Conformity link was added.
11/19/2012	2.1.1	Made typographical edits.
01/31/2013	3.0	Updated for ISE® Design Suite 14.4. Chapter 2, Functional Description: Figure 2-1 was changed in the Performance Monitor and TPG_0 areas. Table 2-1 changed Used 19,052 to 19,805, 42 to 43, 23,881 to 24,094, and % of Used 35 to 37. Table 2-2, deleted PERF_MON_HP2 row and renamed next row to PERF_MON_HP2_HP0_HP2. APU, page 17, end of Cortex-A9 Core section added "For this TRD, both ARM cores run at 667 MHz." Table 2-6, PERF_MON_HP0 changed to PERF_MON_HP0_HP2. In that same section added new row SLOT_1_AXI_ACLK 150 0 Yes clk_150mhz and removed PERF_MON_HP2 section. In TPG_0 section, changed clk to aclk, 148 to 150, video_clk_int to clk_150mhz, S_AXI_ACLK to s_axi_aclk, and DVI2AXI_0 section to VID_IN_AXI4S. *Reset Module* changed to Processor System Reset Module, page 22. *AXI VDMA Video Direct Memory Access*, page 21 changed to AXI Video Direct Memory Access. Heading VTC on page 22 changed to Video Timing Controller. In the same section, VTC IP in last paragraph changed to "Video Timing Controller IP." Added final sentence in that section "For detailed information on ...". The *TPG* section changed to Test Pattern Generator, page 24. Section logiCVC-ML, page 25 was added. In AXI Performance Monitor, page 25 added the final sentence, "For more information..." In Chroma Resampler, page 26 edited the last sentence and added "and licensing information ..." AXI Performance Monitor, page 25 changed Instance: PERF_MON_HP0_HP2 and dropped 's' at Instance. The last sentence in the same section changed from "Two AXI Performance Monitors are instantiated" to "Two slots of AXI Performance Monitors are used." Video Multiplexer, page 26 added the final sentence "For detailed information ..." The *dvi2axi* section on page 25 was removed. Figure 2-4 changed "Xilinx DMA" block to "Xilinx AXI VDMA." Table 2-9 changed "core DMA" to "Xilinx AXI VDMA." XVDMA Driver, page 35 changed "Xilinx DMA" to "Xilinx AXI VDMA." XFILTER_STOP, page 31 bullet switched "On demand mode" with "Continuous mode" in both sentences. Application, page 40 added a new sentence at end "The Linux kernel is in SMP mode and both ..." Figure 2-6 and Figure 2-7 were replaced. In Control and Decision Making, page 43 the bullet "Sobel Filter (using the xFilter driver)" was added. In User Space Device Control, page 44 deleted "Sobel filter" bullet.
01/31/2013, continued	3.0	Appendix A, Register Description: Removed the *AXI TPG Registers* section. Modified Table A-1 and added a row for Bit Position 7. Appendix C, Directory Structure: The directory structure in Figure C-1 changed. Appendix F, Additional Resources: Added references for PG043 and PG103.

Date	Version	Revision
06/07/2013	4.0	Updated ISE Design Suite version 14.3 to version 14.5 throughout document. Added chroma resampler and YCrCb to RGB color space converter IP, and clarified third bullet under The Base Targeted SoC Reference Design, page 8. Changed "Two AXI Performance Monitors" to "AXI Performance Monitor" under Base TRD Key Features, page 11. Updated Figure 2-1, page 15, Clocking, page 19, Table 2-6, page 20, Programmable Logic, page 19, to delete Clk_detect references. Updated values in Table 2-1, page 16. Table 2-4, page 18 with VID_IN_AXI4s information. Added Video Color Space Formats, page 27. and I2C Sub-system, page 27. Added Linux Boot, page 31 to replace earlier Boot loader contents. Updated Table 2-7, page 27 to add UIO, Imageon, ADV7511, and ADV7611 information. Added ADV7511 V4L2 Driver, page 36 and Xylon Frame Buffer Driver, page 37. Updated Figure 2-6, page 39 and Figure 2-7, page 40. Updated User Space Device Control, page 44 with sobel_lib information, and deleted reference to `udriver.h` file. Added Appendix B, Extended Display Identification Data. Updated Figure C-1, page 56. Deleted "Set Up Linux System Software Development tools" from Appendix E.
10/14/2013	5.0	Updated ISE Design Suite version 14.5 to Vivado Design Suite version 2013.2 throughout document. Changes quantiy of Axi Performance Monitors from two to one under Base TRD Key Features, page 11. Updated Table 2-1, page 16. Deleted AXI TPG from Clocking, page 19. Updated the names of the instances in Processor System Reset Module, page 22, AXI Interconnect, page 22, AXI Video Direct Memory Access, page 23, Video Timing Controller, page 24, Test Pattern Generator, page 24, logiCVC-ML, page 25, AXI Performance Monitor, page 25, fmc imageon hdmi in, page 25, Chroma Resampler, page 26, YCrCb to RGB Color-Space Converter, page 26, Video Multiplexer, page 26, Sobel Filter, page 26, and Video In to AXI4-Stream, page 27. Added Sobel Edge Detection, page 30. Updated Software Architecture, page 31. Updated Appendix C, Directory Structure to reflect new updated directory structure. Added Appendix D, PetaLinux Software Development Kit.
02/21/2014	6.0	Updated references to Vivado Design Suite version 2013.2 to version 2013.3 throughout document. Revised block description in Figure 2-1 from *DMAS channel* to *DMA 8 Channel*. Revised the Instance description in the first column, first row of Table 2-2. Corrected cross-reference to Figure 3-7. Added `autostart.sh` to the directory structure shown in Figure C-1. Revised the description in column two, row four in Table C-1. Added Appendix D, PetaLinux Software Development Kit. Revised Appendix F, Additional Resources to conform to the board and kit document reference and style format du jour.
08/27/2014	7.0	Added DRM and V4L2 Acronyms to Table 1-1. Modified Figure 2-1. Modified values in Table 2-1. Modified list of instances in AXI Interconnect in Chapter 2. Modified list of instances in AXI Video Direct Memory Access in Chapter 2. Modified list of instances in Video Timing Controller in Chapter 2. Modified list of instances in Test Pattern Generator in Chapter 2. Modified list of instances in logiCVC-ML in Chapter 2. Modified list of instances in AXI Performance Monitor in Chapter 2. Modified list of instances in fmc imageon hdmi in in Chapter 2. Deleted Chroma Resampler and YCrCb to RGM Color Space Converter cores. Added Clock Multiplexer in Chapter 2. Modified list of instances in Sobel Filter in Chapter 2. Modified list of instances in Video In to AXI4-Stream in Chapter 2. Modified Figure 2-7, Video Color Space Formats. Expanded Software Architecture in Chapter 2, added TRD Software Components. Modified Figure 2-5, Software Architecture: Top Level View. Replaced Table 2-9, Linux Kernal Drivers Used by the Base TRD. Deleted obsolete sections from Chapter 2. Modified API Sobel filter command in Software Sobel Filter Processing in Chapter 2. Modified Device Tree Blob in Appendix D.

Date	Version	Revision
12/15/2014	8.0	Updated content to include changes for Vivado Design Suite version 2014.4 and to reflect changes to the TRD file. Revised AXI interconnect Bulleted descriptions under The Base TRD programmable logic includes:, page 11. Revised Linux kernal and Linux application description under Base TRD software includes:, page 11. Revised Figure 2-1 by grouping fmc_hdmi_input functions and grouping processing functions in dashed boxes. Updated PL hardware utilization values in Table 2-1. Updated instance path descriptions in Table 2-2. Updated Interrupt information in Table 2-3. changed the term video multiplexer to clock multiplexer in Table 2-4. Changed the FPGA logic 75 MHz clock frequency to 50 MHz throughout document. Clock signal names including 75 MHz in the signal name were also revised to include 50 MHz. Instance names throughout document were updated or the instance path description was simplified. Updated directory structure shown in Figure C-1. Removed Getting Started with the PetaLinux SD section in Appendix D. Removed Getting Started with the PetaLinux SD section in Appendix E.
07/31/2015	9.0	Updated content to include changes for Vivado Design Suite version 2015.2 and to reflect use of AXI IIC interface. Added note to Introduction. Revised Figure 2-1 by removing PS I2C-1 (EMIO) reference and adding AXI IIC FMC interface. Updated PL hardware utilization values in Table 2-1. Added AXI IIC instance path information in Table 2-2. Updated Interrupt information in Table 2-3. Added AXI IIC clock information in Table 2-6. Added AXI IIC Bus Interface, page 26. Revised descriptions for AD7511 and AD7611 drivers in Figure 2-4. Updated Figure 2-7 and Figure 2-8 screen captures. Updated User Space Device Control, page 41.
01/04/2016	2015.4	Changed the document version number from 9.0 to 2015.4 (This is a new convention where the document version = the Vivado software version). Updated all references to the Vivado Design Suite version to version 2015.4. Revised note on page 9. Added video Input block to Figure 1-1. Replaced fmc_hdmi_input group of blocks with tpg_input and fmc_hdmi_input group of blocks in Figure 2-1. Updated PL hardware utilization values in Table 2-1. Added axi_vdma_3 instance path information in Table 2-2. Removed hdmi_int_b interrupt, added tpg_input_s2mm_introut interrupt, assigning it to ID 65, and reassigned ID values for axi_perf_mon_1_interrupt, processing_mm2s_introut, and processing_s2mm_introut in Table 2-3. Updated the GPIO bit purpose descriptions in Table 2-4. Revised fourth paragraph under Clocking, page 19. Updated source description for fmc_imageon_in_0_clk_pin in Table 2-5. Removed clock_mux_1 information, revised logicvc_1 vclk connection name, revised axi_vdma_1 s_axis_s2mm_aclk connection name, added axi_vdma_3 information, revised v_tc_1 clk connection name, revised v_tpg_1 aclk connection name, revised fmc_imageon_hdmi_in clk connection name, revised vid_in_axi4s_1 vid_in_clk connection name, and added vid_in_axi4s_2 information in Table 2-6. Added axi_vdma_3 instance under AXI Video Direct Memory Access, page 23. Removed Clock Multiplexer section on page 25. Added HDMI_IN_VDMA to Table 2-7. Removed references to IMAGEON under I2C Sub-system section. Updated Figure 2-4, Figure 2-7, Figure 2-8, and Figure 2-10.
07/01/2018	2015.4	Editorial updates only. No technical content updates.

Table of Contents

Introduction

This user guide provides a functional description the Zynq®-7000 SoC (SoC) ZC702 base targeted reference design (TRD). Register descriptions for IP/logic implemented in programmable logic (PL), Base TRD package directory structure, and pointers to enable you to further develop embedded platforms based on Zynq-7000 SoC architecture are included in this guide. The TRD is part of the ZC702 evaluation kit.

To run the demonstration application and implement hardware and software for the Base TRD, refer to the Xilinx Zynq-7000 ZC702 Base TRD webpage at wiki.xilinx.com/zc702-base-trd.

Note: Starting with Vivado Design Suite version 2015.4, this TRD uses the pin compatible FMC-HDMI-CAM module in place of the FMC IMAGEON card.

The Base Targeted SoC Reference Design

The Base TRD showcases various features and capabilities of the Zynq-7000 SoC Z-7020 device for the embedded domain in a single package using a Xilinx standard Zynq platform Linux-based video pipeline design.

The Base TRD consists of two processing elements: the Zynq-7000 SoC processing system (PS) and a video accelerator based on PL. The SoC allows you to implement a specific functionality either as a software program running on the Zynq-7000 SoC PS or as a hardware design inside the PL. The Base TRD demonstrates an optimization of how you can seamlessly switch between a software or a hardware implementation, contributing to ease of use. The TRD also demonstrates the value of offloading computation-intensive tasks onto PL, thereby freeing the CPU resources available for user-specific applications.

Software developers can leverage the Base TRD and start programming immediately using the widely known Eclipse-based integrated development environment (IDE), GNU compiler tool chain, Linux operating system (OS), and libraries. Embedded hardware designers now have immediate access to the industry standard ARM® Cortex™-A9 core processor system and video IPs running on PL out of the box. In addition, the TRD also provides customers with access to the various PL-based video components such as video direct memory access (VDMA), video timing controller (VTC), video test pattern generator (TPG), Sobel filter and the Xylon logiCVC-ML display controller (www.logicbricks.com/Products/logiCVC-ML.aspx). These IP cores are part of the video pipeline implemented in the PL.

The Base TRD consists of a PS based on the embedded ARM Cortex-A9 core processor, a video processing pipeline implemented in PL, and a Linux-based software application that includes a Qt-based graphical user interface (GUI) (doc.qt.nokia.com) to provide user control and monitoring. The Linux-based software platform and software application run on the ARM Cortex-A9 cores. The software application works in tandem with hardware and provides you with the choice of offloading computation-intensive processing to the PL-based hardware subsystem.

The Zynq-7000 SoC Base TRD reference design (Figure 1-1) consists of these components:

- A dual ARM Cortex-A9 core processor-based embedded Linux OS, board support package (BSP), and U-Boot boot loader
- PL-based hardware IPs that enable acceleration of computation-intensive video processing tasks
- Linux application to configure and control the video IPs in the PL and the PS

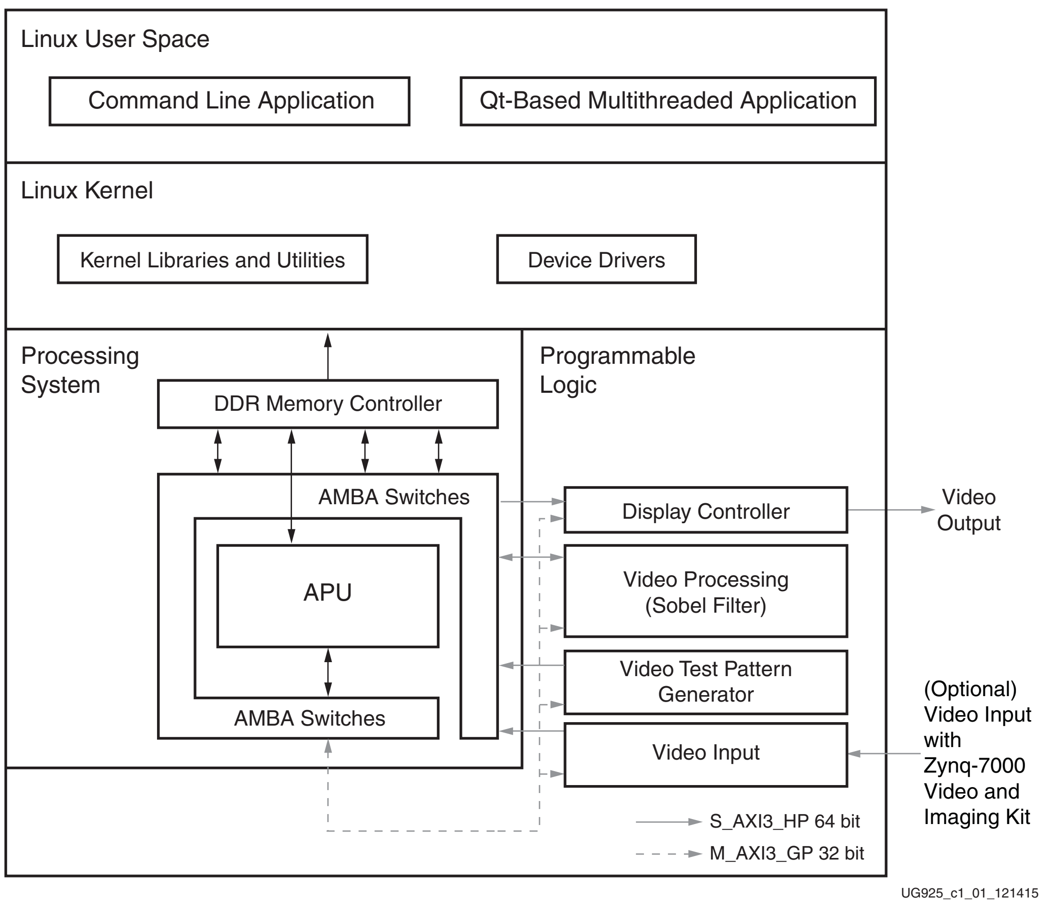


Figure 1-1: **Zynq-7000 SoC Base TRD System Block Diagram**

The TRD deliverables include source code for RTL design and software packages such as the Linux OS, device drivers, the application, and the GUI.

Note: The camera supplied with the video kit is not supported with the TRD.

Base TRD Key Features

Components in the Base TRD are further described in this section.

The Base TRD processing system (PS) includes:

- A dual ARM Cortex-A9 core
- ARM AMBA® AXI interconnect
- Multi-protocol, 32-bit DDR DRAM controller
- Standard peripheral interfaces including USB, UART, I2C, SD MMC, and GPIO

The Base TRD programmable logic includes:

- Two AXI interconnects, 64-bits wide at 150 MHz
- One AXI interconnect, 32-bits wide at 50 MHz
- AXI VDMA(s)
- A full HD video input and output interface
- A Sobel accelerator
- AXI Performance Monitor

Base TRD software includes:

- Xilinx Linux kernel
- Linux device drivers for TRD-specific IPs
- QT and command line Linux application demonstrating the video processing pipeline

List of Acronyms

Table 1-1 lists acronyms used in this document.

Table 1-1: **Acronyms**

Acronym	Definition
AFI	AXI FIFO interface
APU	Application processor unit
BSP	Board support package
COR	Clear on read
DTB	Device tree binary
DTS	Device tree source

Table 1-1: **Acronyms *(Cont'd)***

Acronym	Definition
EDK	Embedded Development Kit
SoC	System on a chip
DRM	Direct rendering manager
FMC	FPGA mezzanine card
FPS	Frames per second
FSBL	First-stage boot loader
GIC	General interrupt controller
GUI	Graphical user interface
HD	High definition
IDE	Integrated development environment
IOP	Input/output peripherals
IP	Intellectual property
JTAG	Joint Test Action Group
KFLOPS	Kilo floating-point operations per second
NVM	Nonvolatile memory
OCM	On-chip memory
OS	Operating system
PL	Programmable logic (inside the Zynq-7000 SoC)
PS	Processing system
R	Read only
RTL	Register transfer level
RW	Read/write
SC	Self clear
SD	Secure Digital
SD MMC	Secure Digital Multimedia Card
SDK	Software Development Kit
SoC	System on Chip
TDP	Targeted Design Platform
TPG	(Video) Test Pattern Generator
TRD	Targeted Reference Design
TTC	Triple-timer counter
VDMA	Video direct memory access
VTC	Video timing controller

Table 1-1: **Acronyms *(Cont'd)***

Acronym	Definition
V4L2	Video for Linux
ZC702	Platform development board based on the Zynq SoC Z-7020 device
Zynq Z-7020	An implementation of the Zynq-7000 SoC with a fixed feature set and PL capabilities

Functional Description

This chapter describes the Zynq®-7000 SoC ZC702 Base Targeted Reference Design (TRD) hardware design, software system, and video demonstration application components. It also describes how data flows through the various connected IPs and includes information about the flow of application control.

To build hardware and software for the Base TRD, refer to the Xilinx Zynq-7000 Base Targeted Reference Design wiki page wiki.xilinx.com/zc702-base-trd.

Hardware Architecture

The block diagram for the Base TRD is shown in Figure 2-1. This design has two parts:

- Processing system (PS)
- Video IPs and custom logic implemented in programmable logic (PL)

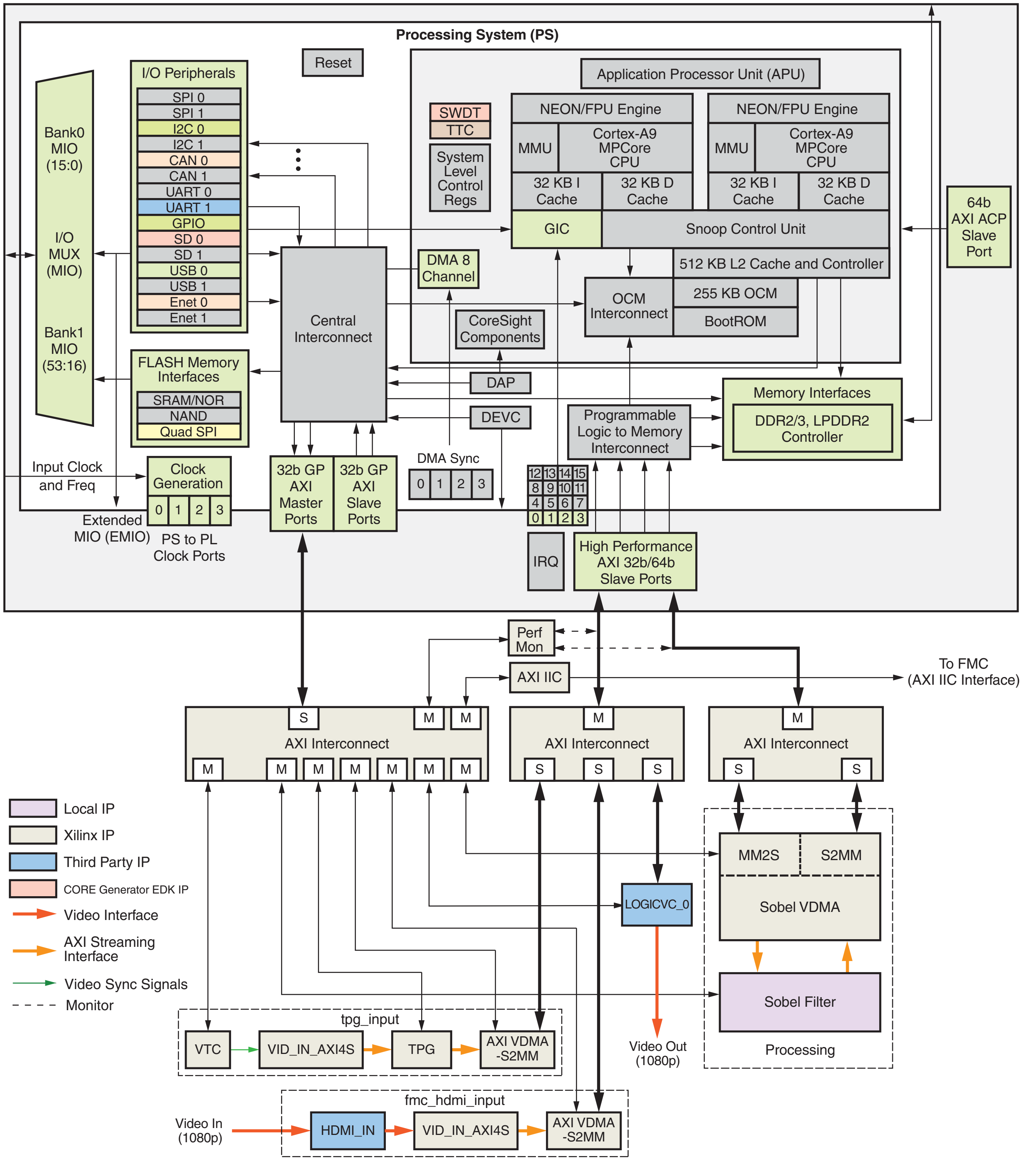


Figure 2-1: **Zynq-7000 SoC Base TRD Hardware Block Diagram**

This system is implemented in a Zynq-7000 SoC device (XC7Z020-CLG484-1) using the Vivado® Design Suite.

The PL hardware utilization for the implemented design is shown in Table 2-1.

Table 2-1: **PL Hardware Utilization for Device XC7Z020-CLG484-1[(1)]**

FPGA Components	Total Available	Used	% Used
LUTs	53,200	20,327	38
I/Os	200	43	21
Slice registers	106,400	25,772	24
FPGA Logic Memory			
RAMB36/FIFO	140	19	13
RAMB18	280	31	11

Notes:

1. The figures provided here are only indicative of nature and can vary between different tool chain versions.

The PL-implemented video IP and custom logic address map is shown in Table 2-2.

Table 2-2: **FPGA Logic Address Map for the Zynq-7000 SoC ZC702 Base TRD**

Instance	Peripheral	Base Address	High Address
v_tc_1	axi_vtc	0x40070000	0x4007FFFF
v_tpg_1	axi_tpg	0x40080000	0x4008FFFF
axi_vdma_1	axi_vdma	0x40090000	0x4009FFFF
axi_vdma_2	axi_vdma	0x400B0000	0x400BFFFF
axi_vdma_3	axi_vdma	0x40020000	0x4002FFFF
axi_iic_1	axi_iic	0x40040000	0x4004FFFF
logicvc_1	LogiCVC	0x40030000	0x4003FFFF
axi_perf_mon_1	axi_perf_mon	0x400F0000	0x400FFFFF
image_filter_1	sobel_filter_top	0x400D0000	0x400DFFFF

System Configuration

Processing System

This design makes full use of these four major components in the PS:

- Application processor unit (APU)
- Interconnect
- Input/output peripherals (IOP)
- Memory interfaces

This section describes some of the features of the PS used in this design. For detailed information about the complete feature set including a functional description, see the *Zynq-7000 SoC Technical Reference Manual* (UG585) [Ref 1].

APU

The APU includes the dual ARM Cortex-A9 core processor, snoop control unit (SCU), L2 cache controller, on-chip memory (OCM), 8-channel DMA, system watchdog timer (SWDT), and triple-timer controller (TTC) blocks.

Cortex-A9 Core - The ARM Cortex-A9 core processor implements the ARMv7 architecture and runs 32-bit ARM instructions, 16-bit and 32-bit Thumb instructions, and 8-bit Java byte codes in the Jazelle state. The media processing engine implements ARM NEON coprocessor technology, a single instruction multiple data (SIMD) architecture that adds instructions targeted at audio, video, 3D graphics, image, and speech processing. For this TRD, both ARM cores run at 667 MHz.

General Interrupt Controller - The GIC collects interrupts from various sources and distributes these interrupts to each of the ARM cores. The interrupt distributor holds the list of pending interrupts for each ARM Cortex-A9 core processor and then selects the highest priority interrupt before issuing it to the Cortex-A9 processor interface. Interrupts of equal priority are resolved by selecting the lowest ID. A total of 64 shared peripheral interrupts (PL interrupts + PS I/O peripheral interrupts) are supported, starting from ID 32. Table 2-3 lists interrupt IDs for interrupts coming from PL.

Table 2-3: **Interrupt IDs for PL-Generated Interrupts**

Interrupt line	ID	Type	Source
axi_iic_1_iic2intc_irpt	62	Level	axi_iic_1
logicvc_1_interrupt	63	Level	logicvc_1
fmc_hdmi_input_s2mm_introut	64	Level	axi_vdma_1
tpg_input_s2mm_introut	65	Level	axi_vdma_3
axi_perf_mon_1_interrupt	66	Level	axi_perf_mon_1
processing_mm2s_introut	67	Level	axi_vdma_2
processing_s2mm_introut	68	Level	axi_vdma_2

Interconnect

The interconnect unit connects all PS and PL master and slave devices. There are a total of four Advanced eXtensible Interface (AXI) slave ports dedicated for AXI masters residing in the PL, and four of these ports contain deep FIFOs to improve data throughput. Two AXI master ports provide access to AXI slaves in the PL. In this design, masters in PL are connected through two AXI slave ports with deep FIFOs. One AXI master port is used to access registers in AXI slave IPs in PL.

An advanced peripheral bus (APB) master port is provided for accessing software programmable registers of all PS modules. The top level switch is AXI3-compliant, the soft IPs provided by Xilinx are AXI4-compliant, and the soft AXI interconnect IP provides protocol bridging as needed.

S_AXI_HP - The high performance slave AXI interfaces (S_AXI_HP) connect the PL to AFI blocks in the PS. The PL has four AXI masters out of which two are connected to the S_AXI_HP0 port and two are connected to the S_AXI_HP2 port. The HP port enables a high throughput datapath between AXI masters in the programmable logic and the processing system DDR3 memory. The main aim of the AXI FIFO interface (AFI) units is to smooth out this variable latency, allowing the ability to stream data continuously from DDR to the PL masters and from the PL masters to DDR. The PL-side interface of AFI runs on the clock coming from the PL. In this design, a 150 MHz clock is connected from the PL side. The DDR-side clock is running on 2/3 of the DDR_CLK (533 MHz). The high performance AXI interface module provides several hooks to assist in bandwidth management of masters connected to different PL ports. Controlling issuance capability available from the PL port is one of the hooks exercised in this design to obtain a fair share of bandwidth between two masters, SOBEL VDMA, and the display controller.

M_AXI_GP - This AXI master port interfaces with AXI slave IPs in PL through an AXI Lite interconnect. The CPU manages initializing and controlling the video pipeline through this port.

IOP - The IOP unit includes communication peripherals. GPIO, Ethernet, USB, I2C, and SD controllers from the PS are used extensively in this design.

GPIO - The 64-bit general purpose input/outputs (GPIOs) are connected to the PL through the extendable multiplexed I/O (EMIO) interface. Sixty-four bits are divided into two banks, each of 32 bits. Because each GPIO bit can be dynamically configured as input or output, GPIO bits are used in this design for many functions. Table 2-4 lists the GPIO bit and purpose in design.

Table 2-4: **GPIO Bits Functional Description**

GPIO Bit Number	Net Name	Purpose
1	ps7_0_GPIO_O[1]	Selects the video input source, either external HDMI or test pattern
0	ps7_0_GPIO_O[0]	FMC I2C multiplexer reset.
2	ps7_0_GPIO_O[2]	TPG Reset

Memory Interfaces

The memory interfaces unit includes the DDR memory controller and nonvolatile memory (NVM) controllers. The DDR memory controller includes a 4-port arbiter. One AXI port is dedicated for ARM CPU access and two ports are dedicated for high performance AXI interface master devices in the programmable logic. The remaining port is shared by all other AXI masters. In this design, DDR3 is configured to run at 533 MHz, and the AXI interface is running at 355 MHz.

PL Clocks

The PS provides four fully programmable clocks (FCLK_CLK) to the PL. These clocks are routed directly to PL clock buffers to serve as a frequency source for the PL. The clock wizard module in PL gets a 100 MHz clock from FCLK_CLK0.

PL Reset

The PS provides four FCLK_RESET[3:0]_N fully programmable reset signals to the PL. These signals are asynchronous to PS clocks. The PL logic reset blocks in this design receive input from FCLK_RESET0_N and generate necessary reset signals for the design implemented in PL.

Programmable Logic

Clocking

The FPGA logic design has three clock domains: AXI MM (memory-mapped) interconnect, AXI register interface, and video clock. These domains run at 150 MHz, 50 MHz, and 148.5 MHz, respectively.

The clock generator module receives a 100 MHz input clock from the PS FCLK_CLK0 and generates 50 MHz and 150 MHz. The AXI Lite interconnect works on 50 MHz. Apart from the AXI Lite interconnect, the register interface of AXI VDMA, AXI TPG, logiCVC-ML, VTC, and axi_perf_mon_1 are driven by the 50 MHz clock.

Two instances of the AXI_MM interconnect connected to the HP port of the PS run on 150 MHz. The S2MM (stream to memory map) and MM2S (memory map to stream) channels of VDMAs are running at 150 MHz. The 150 MHz clock drives the logiCVC-ML memory read interface and also the AXI slave interface of the Sobel filter.

The video clock comes from the onboard clock synthesizer or the FMC card. This video clock is used by the input video modules v_tc_1, v_vid_axi4s_1, v_tpg_1 and for axi_vdma_1 s2mm clock.

Table 2-5 lists system clocks.

Table 2-5: **System Clocks**

Clock Signal	Source	Frequency	Use
FPGA_CLK	PS - FCLK_CLK0	100 MHz	Input clock to clock generator
clk_50mhz	Internal mixed-mode clock manager (MMCM)	50 MHz	Clock for AXI Lite interconnect, generated by clock generator
clk_150mhz	Internal MMCM	150 MHz	Clock for AXI MM interconnect, generated by clock generator
VIDEO_CLK_P, VIDEO_CLK_N	External differential video clock coming from clock synthesizer on board	148.5 MHz	Clock for display controller and video receiving blocks
fmc_imageon_in_0_clk_pin	External video clock coming from Imageon FMC	148.5 MHz	Clock for video receiving modules

Based on user clock configuration inputs, the clock generator determines the correct configuration of the PLLs.

Table 2-6 shows clock requirements of master and slave peripherals connected in system and their connection.

Table 2-6: **PL Clock Configuration**

Component	Frequency (MHz)	Phase	Buffered	Connection
clk_wiz_1				
• CLKIN	100		Yes	FPGA_CLK
• CLKOUT0	50		Yes	clk_50mhz
• CLKOUT1	150		Yes	clk_150mhz
Processor				
processing_system7_1				
• FCLK_CLK0	100	0	Yes	FPGA_CLK
• M_AXI_GP0_ACLK	50	0	Yes	clk_50mhz
• S_AXI_HP0_ACLK	150	0	Yes	clk_150mhz
• S_AXI_HP2_ACLK	150	0	Yes	clk_150mhz
Buses				
axi_interconnect_hp0				
• INTERCONNECT_ACLK	150	0	Yes	clk_150mhz
axi_interconnect_hp2				
• INTERCONNECT_ACLK	150	0	Yes	clk_150mhz
axi_interconnect_gp0				

Table 2-6: **PL Clock Configuration *(Cont'd)***

Component	Frequency (MHz)	Phase	Buffered	Connection
• INTERCONNECT_ACLK	50	0	Yes	clk_50mhz
Peripherals				
proc_sys_reset_1_clk50				
• Slowest_sync_clk	50	0	Yes	clk_50mhz
proc_sys_reset_clk150				
• Slowest_sync_clk	150	0	Yes	clk_150mhz
logicvc_1				
• S_AXI_ACLK	50	0	Yes	clk_50mhz
• mclk	150	0	Yes	clk_150mhz
• vclk	148.5	0	Yes	video_clk_1
image_filter_1				
• SYS_CLK	150	0	Yes	clk_150mhz
• s_axi_CONTROL_BUS_ACLK	150	0	Yes	clk_150mhz
axi_vdma_2				
• m_axi_mm2s_aclk	150	0	Yes	clk_150mhz
• m_axi_s2mm_aclk	150	0	Yes	clk_150mhz
• m_axis_mm2s_aclk	150	0	Yes	clk_150mhz
• s_axi_lite_aclk	50	0	Yes	clk_50mhz
• s_axis_s2mm_aclk	148.5	0	Yes	clk_150mhz
axi_vdma_1				
• m_axi_s2mm_aclk	150	0	Yes	clk_150mhz
• s_axi_lite_aclk	50	0	Yes	clk_50mhz
• s_axis_s2mm_aclk	150	0	Yes	vtiming_mux_0_video_clk
axi_vdma_3				
• m_axi_s2mm_aclk	150	0	Yes	clk_150mhz
• s_axi_lite_aclk	50	0	Yes	clk_50mhz
• s_axis_s2mm_aclk	150	0	Yes	vtiming_mux_0_video_clk
axi_perf_mon_1				
• SLOT_0_AXI_ACLK	150	0	Yes	clk_150mhz
• SLOT_1_AXI_ACLK	150	0	Yes	clk_150mhz
• S_AXI_ACLK	50	0	Yes	clk_50mhz
• CORE_ACLK	150	0	Yes	clk_150mhz
v_tc_1				
• clk	148.5	0	Yes	video_clk_1

Table 2-6: **PL Clock Configuration *(Cont'd)***

Component	Frequency (MHz)	Phase	Buffered	Connection
• S_AXI_ACLK	50	0	Yes	clk_50mhz
v_tpg_1				
• aclk	148.5	0	Yes	clk_150mhz
• s_axi_aclk	148.5	0	Yes	clk_50mhz
fmc_imageon_hdmi_in				
• clk	148.5	0	Yes	fmc_imageon_hdmi_clk
vid_in_axi4s_1				
• vid_in_clk	148.5	0	Yes	fmc_imageon_hdmi_clk
• aclk	150	0	Yes	clk_150mhz
vid_in_axi4s_2				
• vid_in_clk	148.5	0	Yes	video_clk_1
• aclk	150	0	Yes	clk_150mhz
axi_iic_1				
• s_axi_aclk	50	0	Yes	clk_50mhz

Processor System Reset Module

Instances: proc_sys_reset_clk50, proc_sys_reset_clk150

The proc_sys_reset module implements a reset scheme. Input to the proc_sys_reset core is generated by PS FCLK_RESET0_N. The polarity of input reset to this block is indicated by parameter C_EXT_RESET_HIGH. In this design, C_EXT_RESET_HIGH is set to 0 as reset generated by PS is active-Low. This block generates various types of resets, such as reset for interconnect, peripheral reset, and so on. All the blocks in the PL are driven by interconnect reset, which is active-Low in polarity.

For detailed information about the complete feature set and a functional description of the proc_sys_reset IP, see the *LogiCORE IP Processor System Reset Module Product Specification* (PG164) [Ref 2].

AXI Interconnect

Instances: axi_interconnect_gp0, axi_interconnect_hp0, axi_interconnect_hp2

FPGA logic design has two interconnects for AXI memory-mapped masters and one interconnect for the AXI register interface.

AXI memory-mapped interconnects are connected to masters like AXI_VDMA and logiCVC-ML. Slaves connected to these interconnects includes HP0 and HP2 ports of Zynq-7000 SoC PS. This interconnect operates at 150 MHz and the data width is 64-bit wide.

The AXI register interface is clocked at 50 MHz. The Zynq-7000 SoC PS GP0 port acts as master on this interconnect and connected slaves have register maps. AXI TPG and VTC are examples of slaves connected to this interconnect. The operations of the video pipeline are controlled by registers inside every IP. Depending upon data flow required in the video pipeline, the processor writes these registers through the AXI Lite interconnect. The AXI Lite interconnect accepts write or read transfers from the CPU, performs address decoding, selects a particular slave, and establishes a communication channel between the CPU and the slave device.

For detailed information about the complete feature set and a functional description of the AXI Interconnect IP, see the *LogiCORE IP AXI Interconnect* (PG059) [Ref 3].

AXI Video Direct Memory Access

Instances: axi_vdma_1, axi_vdma_2, and axi_vdma_3

AXI VDMA has an AXI streaming interface on one side and an AXI memory-mapped interface on the other side. The VDMA has two channels: MM2S (memory-mapped to streaming) and S2MM (streaming to memory-mapped). The MM2S channel reads the number of data beats programmed through the C_MM2S_MAX_BURST_LENGTH parameter and presents it to the slave device connected through the streaming interface. The data width of the streaming interface can be different than the memory-mapped interface and controlled through C_M_AXIS_MM2S_TDATA_WIDTH. The data width of the S2MM memory-mapped interface is controlled by the C_M_AXI_MM2S_DATA_WIDTH parameter.

The S2MM channel receives data from the master device connected through the streaming interface. The C_S_AXIS_S2MM_TDATA_WIDTH parameter decides the width of the streaming interface. Data received on the streaming interface is then written into the system memory through the memory-mapped interface. The C_M_AXI_S2MM_DATA_WIDTH parameter decides the data width of the memory-mapped interface and C_S2MM_MAX_BURST_LENGTH governs the burst length of the write transaction.

In this design, the streaming interface data width is set to 32-bit wide and the memory-mapped interface is configured as 64-bit wide. The AXI VDMA is used in simple register direct mode, which removes the area cost of the scatter/gather feature. Initialization, status, and management registers in the AXI VDMA core are accessed through an AXI4-Lite slave interface. To get the best possible throughput for AXI VDMA instances, the maximum burst length is set to 16. In addition, the master interfaces have a read and write issuance of 8 and a read and write FIFO depth of 512 to maximize throughput. The line buffers inside the AXI VDMA for the read and write sides are set to 4K deep and the store and forward feature of the AXI VDMA are enabled on both channels to improve system performance and reduce the risk of system throttling.

For detailed information on the complete feature set and a functional description of AXI VDMA IP, see the *LogiCORE IP AXI Video Direct Memory Access Product Guide* (PG020) [Ref 4].

Video Timing Controller

Instance: v_tc_1

The VTC is a general purpose video timing generator and detector. The input side of this core automatically detects horizontal and vertical synchronization pulses, polarity, blanking timing, and active video pixels. This information can be used by application software to take various decisions and for configuration of the video pipeline. In the current design, application software measures resolution of external video and then decides whether to switch to the external video source or not. The same feature can be expanded in the future to configure the video pipeline based on input resolution.

The output side of the core generates the horizontal and vertical blanking and synchronization pulses. The width and interval of these pulses are configured through the AXI Lite interface. The AXI TPG block generates a video test pattern based on video timing pulses generated by VTC. In this design, VTC is used to generate video timing signals to match Full HD (1080p60) video format.

The video timing generator/detector block and AXI Lite interface of this core work on a single clock domain (that is, the video clock).

For detailed information on the complete feature set, a functional description, and licensing information for Video Timing Controller IP, see the *LogiCORE IP Video Timing Controller Product Guide* (PG016) [Ref 5].

Test Pattern Generator

Instance: v_tpg_1

The Test Pattern Generator contains an AXI register interface to access slave control registers from a processor. This IP can generate patterns like color bars, horizontal and vertical burst patterns, and zone plates. The generation of pattern is controlled through the pattern control register. It also enables the overlay of a box on a selected pattern. The motion control register controls the speed at which the box moves over a selected pattern. In this design, color bars are used with a moving box. The box size register controls the size of the box, and the color of the box is selected by the box color register. The width and height of the pattern is equal to 1920 x 1080, selected through the line length and frame height register.

For detailed information on the complete feature set and a functional description of Test Pattern Generator, see the *LogiCORE IP Test Pattern Generator Product Guide* (PG103) [Ref 6].

logiCVC-ML

Instance: logicvc_1

The logiCVC-ML is a multi-layer video display controller from Xylon (www.logicbricks.com/Products/logiCVC-ML.aspx). The logiCVC-ML controller refreshes the display image by reading the video memory and converting the read data into a data stream acceptable for the display interface. It generates control signals for the display, and supports multiple layers with video processing functions such as alpha blending, transparency, and move around.

For detailed information about the complete feature set, a functional description, and license information for logiCVC-ML IP, refer to the Xylon data sheet (www.logicbricks.com/Products/logiCVC-ML.aspx).

AXI Performance Monitor

Instance: axi_perf_mon_1

The AXI Performance Monitor can monitor and analyze system behavior on the AXI interface. This core is used in the Base TRD to measure read and write throughput on AXI slave ports of the PS (HP0 and HP2), which are used to access DDR memory from PL. The core consists of the AXI4-Lite interface to configure and control the core.

This core is configured to measure the read and write throughput by counting the number of transactions per second. When the configured time interval expires, measured throughput in bytes is loaded into a register and read by the software application.

Two slots of AXI Performance Monitors are used to measure read and write throughput of HP0 and HP2 simultaneously.

For detailed information on the complete feature set and a functional description of AXI Performance Monitor, see the *LogiCORE IP AXI Performance Monitor Product Guide* (PG037) [Ref 7].

fmc imageon hdmi in

Instance: fmc_imageon_hdmi_in_1

This IP core receives video from FMC-IMAGEON, in YCrCb 4:2:2 format, with embedded vblank and hblank signals, and extracts blanking information.

Sobel Filter

Instance: image_filter_1

This IP has the AXI4-Lite interface through which the IP is configured and controlled.

The number of rows and columns are configured using AXI interface, and the filtering process starts when the Start register is written through the Sobel interface.

The Sobel filter detects the edge in the video frame and the processed frame is sent out on AXI stream interface.

Video In to AXI4-Stream

Instance: v_vid_in_axi4s_1

This IP interfaces a video source to the AXI4-Stream interface. It also handles video data clock boundary crossing between the video clock domain and AXI4-Stream clock domain.

For detailed information on the complete feature set and a functional description of Video in to AXI4-Stream IP, see the *LogiCORE IP Video In to AXI4-Stream v1.0* (PG043) [Ref 9].

AXI IIC Bus Interface

instance: axi_iic_1

This IP provides an AXI IIC bus interface controller for the FMC. It is used to configure the ADV7611 HDMI receiver on the FMC.

Video Color Space Formats

Table 2-7 shows the video color space formats available in the video pipeline.

Table 2-7: **Video Color Space Formats**

<table>
<tr><th>Video IP</th><th>Input Video Format</th><th>Output Video Format</th></tr>
<tr><td>HDMI_IN (ADV7611)</td><td rowspan="3">YCbCr 4:2:2</td><td rowspan="3">YCbCr 4:2:2</td></tr>
<tr><td>VID_IN_AXI4S</td></tr>
<tr><td>TPG_0 (Test Pattern Generator)</td></tr>
<tr><td>TPG_VDMA (S2MM)</td><td rowspan="4">YCbCr 4:2:2</td><td rowspan="4">YCbCr 4:2:2</td></tr>
<tr><td>FILTER_VDMA (S2MM and MM2S)</td></tr>
<tr><td>HDMI_IN_VDMA</td></tr>
<tr><td>Sobel Filter</td></tr>
<tr><td>LogiCVC – Layer 0</td><td>RGB</td><td>YCbCr 4:2:2</td></tr>
<tr><td>LogiCVC – Layer 1</td><td>YCbCr 4:2:2</td><td>YCbCr 4:2:2</td></tr>
<tr><td>HDMI_OUT (ADV7511)</td><td>YCbCr 4:2:2</td><td>Automatic</td></tr>
</table>

Because all the VDMAs operate on YCbCr 4:2:2, the DDR memory always has the YCbCr 4:2:2 video format.

I2C Sub-system

I2C is a two-wire bus for attaching low-speed peripherals. It uses two bidirectional open-drain lines, SDA (serial data) and SCL (serial clock), pulled up with resistors. In standard mode, a 7-bit address space and a 400 kHz bus speed are used. In this reference design, both PS and PL I2C controllers are used as bus masters to configure several I2C slaves. The bus hierarchy is shown in Figure 2-2.

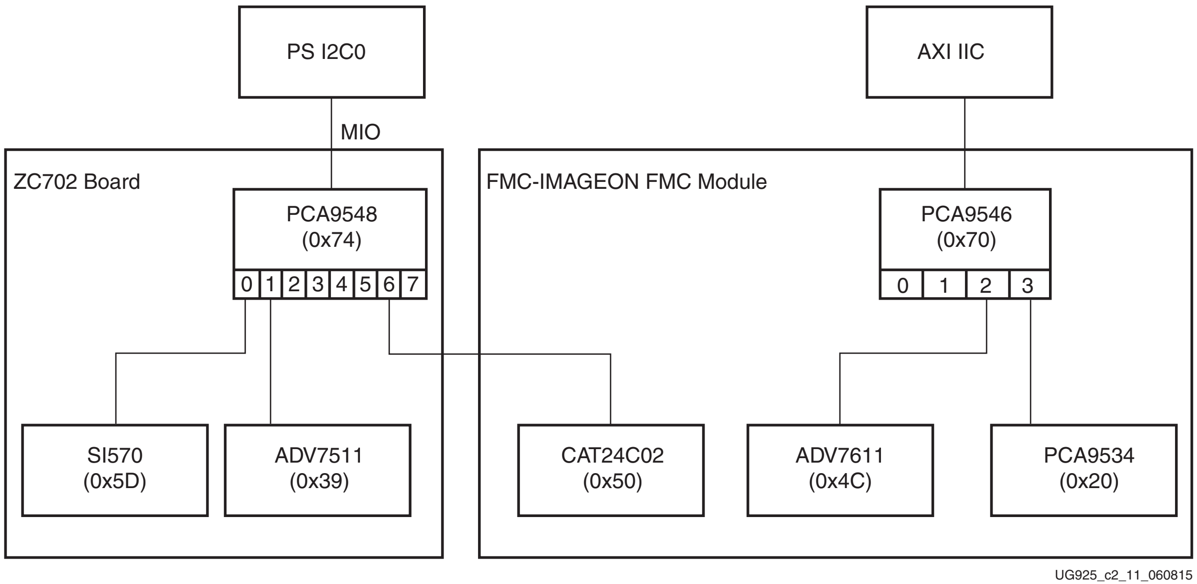


Figure 2-2: **I2C Sub-system**

The PS I2C0 controller is connected via hard-wired MIO signals (SCL on MIO[50] and SDA on MIO[51]) to an 8-channel I2C multiplexer (PCA9548) at address `0x74` on the ZC702 board. The following I2C clients are connected to the I2C multiplexer and used in this design: A programmable clock synthesizer (SI570) is connected to channel 0 at address `0x5D`; an HDMI encoder/transmitter (ADV7511) is connected to channel 1 at address `0x39`; and, optionally, a 2 Kb EEPROM (CAT24C02) is connected to channel 6 at address `0x50` on a mezzanine card plugged into the FMC2 slot.

The AXI IIC bus interface (axi_iic_1) is connected to a 4-channel I2C multiplexer (PCA9546) at address `0x70` on the FMC mezzanine card (if present). The following I2C clients are connected to the I2C multiplexer and (optionally) used in this design: An HDMI decoder/receiver (ADV7611) is connected to channel 2 at address `0x4C`; an 8-bit I/O expander (PCA9534) is connected to channel 3 at address `0x20`.

The SI570 clock synthesizer is used to generate an accurate video clock that is used to drive the display output. The logiCVC display controller configures the SI570 to generate a specific video clock frequency based on the selected video resolution (e.g., for 1920x1080 at 60 frames per second [1080p60]), the video clock needs to be set to 148.5 MHz. The generated clock is a differential clock connected the PL. The same clock is used not only to drive the display output, but it also connects to the VTC timing generator which in turn provides video timing signals to the TPG.

The ADV7511 HDMI transmitter is driven by the logiCVC display controller. Its video input interface consists of a video clock, a 16-bit video data bus, a data enable, and horizontal and vertical sync signals. The HDMI transmitter serializes the incoming video stream and sends it to the display through HDMI. The video input format is YCbCr 4:2:2. The SPDIF pin for audio is unused and therefore not connected in this design. The interrupt pin is connected to the GIC and the signal is routed through the PL. The main purpose is to detect if a display is connected to the HDMI port or not; this feature is called hot plug detect (HPD). Furthermore, the ADV7511 is used to query the Extended Display Identification Data (EDID), a standard published by the Video Electronics Standards Association (VESA). The main purpose of the EDID is to provide information on the video resolutions supported by a video sink, in this case a display monitor. This information is communicated back to the display controller which then uses the information to generate the proper timing signals to drive the HDMI transmitter which in turn drives the display.

The FMC standard defines an I2C bus interface to optionally support the Intelligent Platform Management Interface (IPMI). The Field Replaceable Unit (FRU) for IPMI is a data structure stored inside the CAT24C02 2 Kb EEPROM of the FMC mezzanine card. It holds the inventory information, such as vendor ID and manufacturer, part number, version number, etc.

The ADV7611 HDMI receiver on the FMC mezzanine card is used to connect an external video source like an HD camera or video player via HDMI. The interface to the PL consists of a video clock and a 16-bit video data bus. The video format interfacing the PL is YCbCr 4:2:2 with embedded sync signals (i.e. there are no separate signals for data enable, horizontal and vertical sync to reduce pin count). The SPDIF pin for audio is unused in this design and hence not connected. The active-low reset and HPD control pins are inputs, connected to the I/O expander.

The PCA9534 I/O expander is used to drive the reset line and HPD control pin of the ADV7611 HDMI receiver. The HPD signal is only asserted after the EDID has been programmed into the internal EEPROM of the ADV7611. On the video input side, the EDID is used to advertise the video resolutions supported by the HDMI receiver to the video source (e.g. an HD media player). The video source will receive the HPD event and then determine the video output resolution based on the EDID information read from the ADV7611 HDMI receiver. Some video sources do not attempt to read the EDID and therefore *allow* driving a potentially unsupported video resolution.

Note that more complex ICs like the ADV7511 or the ADV7611 can also have internal I2C bus structures to communicate with sub-components (e.g., the ADV7611 is comprised of 7 I2C sub-devices with the EDID EEPROM being one of them).

Sobel Edge Detection

Sobel edge detection is a classical algorithm in the field of image and video processing for the extraction of object edges. Edge detection using Sobel operators works on the premise of computing an estimate of the first derivative of an image to extract edge information. By computing the x and y direction derivatives of a specific pixel against a neighborhood of surrounding pixels, it is possible to extract the boundary between two distinct elements in an image. Due to the computational load of calculating derivatives using squaring and square root operators, fixed coefficient masks have been adopted as a suitable approximation in computing the derivative at a specific point. In the case of Sobel, the masks used are shown in Table 2-8

Table 2-8: **Sobel Operator Masks**

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

The visual effect of applying a Sobel operation on a frame of video or an image is shown in Figure 2-3.

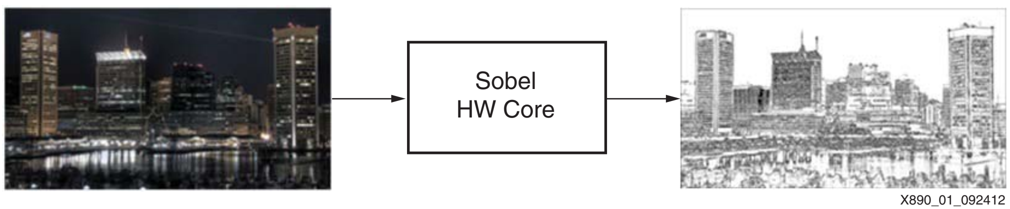


Figure 2-3: **Sobel Edge Detection**

There are different ways of implementing the C code for Sobel edge detection depending on the target execution platform and performance requirements. The sobel implementation is in `/hardware/vivado_hls/src/sobel.cpp`.

The C code shows that applying the masks of Figure 2-3 to an image creates a two-dimensional traversal in memory along a 3 x 3 window. Because this window accesses different rows in the image, the benefits of memory locality to reduce memory bandwidth requirements are limited.

To achieve the target performance of 60 frames per second at 1080p, the inner for-loops computing `x_weight` and `y_weight` must be fully unrolled.

Complete unrolling of the inner for-loop structure in the context of the HLS tool has the effect of exposing the entire datapath for the computation of the gradient at a pixel to the compiler. It also exposes all the memory accesses required to complete the computation as fast as possible. In this case, achieving maximum performance requires nine read operations to the memory storing the input image to construct the 3 x 3 window on which the gradient is computed. This creates a memory bandwidth bottleneck, which must be addressed at the algorithmic level.

As described in *Implementing Memory Structures for Video Processing in the Vivado HLS Tool* (XAPP793) [Ref 8], the algorithm targeted at the HLS tool must have a notion of proper memory architecture for achieving performance. In the context of Sobel edge detection, proper memory architecture requires only a single access to the global memory storing the input image and local buffer storage to create the 3 x 3 computation window. Line buffers and other memory structures must be part of the C code given to the HLS tool to generate the correct design. Implementing Memory Structures for Video Processing in the Vivado HLS T*ool* provides the details on how to create line buffers and memory structures for image and video processing. The code in the TRD has been modified to express tiered memory architecture to meet the performance requirements of the design.

Software Architecture

This section explains the software architecture for the Zynq-7000 SoC ZC702 Base TRD. Figure 2-5 illustrates a top-level view of the software architecture.

A multi-threaded Linux application is responsible for running the TRD demonstration. This application uses the Qt-based GUI, which is displayed through Display Monitor, to obtain user inputs. Depending on the inputs, it calls device drivers to configure the hardware and to enable a particular datapath.

Zynq Base TRD software stack implements standard Linux video framework for data capture, processing, and display.

The data capture block is controlled by V4L2 and media framework. Captured frames are then processed either in the hardware block or in the application depending on the sobel mode selection.

Finally, video buffers are submitted to DRM driver for the scan out.

The Zynq Base TRD uses the below mentioned Linux frameworks.

- V4L2 framework
- Media framework
- Linux DMA framework
- Device Rendering Manager (DRM) framework

Figure 2-4 shows the software schematic.

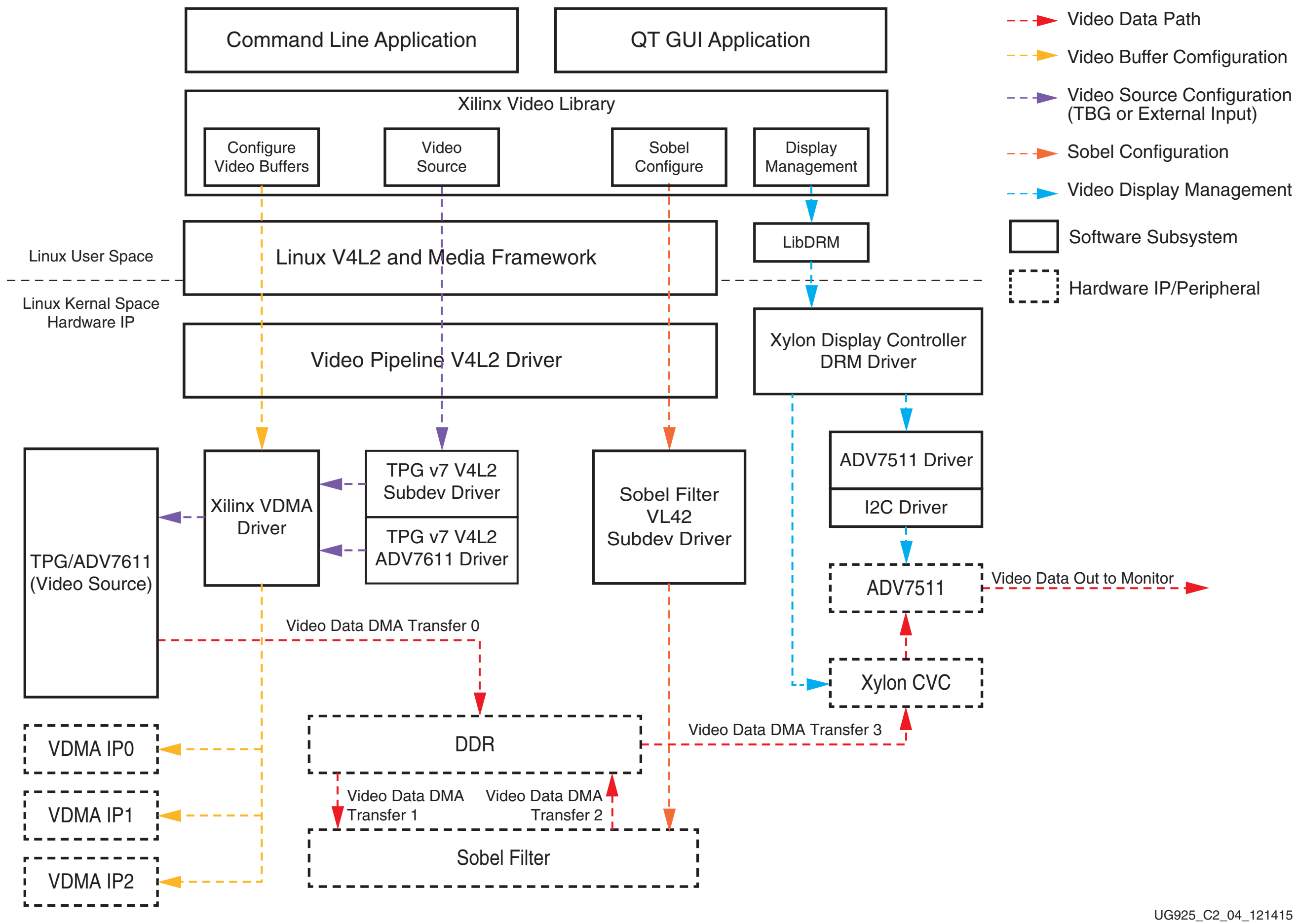


Figure 2-4: **Software Schematic**

TRD Software Components

Following are the software components (Linux User space and Kernel space) included in the Base TRD System. The components are either native to the Linux Operating System or supplied by Xilinx third party partners, or developed in-house at Xilinx.

Control Applications

This consists of a command line Linux user space application and a QT GUI based application that operates the video pipeline of the base TRD. As the logic of operating the video pipeline is common, some is captured in the Xilinx Video Library. Interfaces exported by this library are used by the command line and QT GUI applications.

- V4L2 interface is used by the application for configuring the required video device (for example, TPG, Sobel IP, etc.)

- V4L2 interface is used to manage video buffers used for streaming IN and processing of video data.
- DRM interface is used by the application to display video data.
- Media framework is used to configure device internal topology at run time.

Some of the significant logic implemented in Xilinx Video Library is:

- Video Buffer Management

  This logic is responsible for supplying video buffers (allocated in kernel space) to receive Video data from TPG/External video source. The logic invokes V4L2 pipeline driver logic, which in turn depends on the VDMA IP driver to manage the transfer of video data into DDR.

  If the Sobel filter is enabled, the Video Buffer management logic facilitates transfer of incoming video data from video buffers to Sobel IP and from Sobel IP to DRM driver buffers.

- Video Source Selection

  This logic interfaces with Media framework to configure enabled video devices.

  This logic selects the source of video data (TPG or External Input). This logic invokes the entry points of the Video pipeline driver, and subsequently the TPG/ADV7611 V4L2 Sub-device driver to select the video source based on user input.

- Sobel Configuration

  This logic enables and configures the Sobel Filter hardware IP. This logic invokes the entry points of the Sobel Filter V4L2 Sub-device driver to control the Sobel IP.

- Media Framework

  This is the standard media framework in Linux. Discovering a device internal topology, and configuring it at run time, is one of the goals of the media framework. There are three media topologies:

  - External video input –> vcap_hdmi output
  - Test Pattern Generator –> vcap_tpg output
  - Filter HLS –> vm2m_hls output

- Linux V4L2 framework

  This is the standard V4L2 framework in Linux. The Xilinx video pipeline driver registers with the V4L2 framework to provide control of the various elements of the video pipeline.

- Video Pipeline Driver

  This is a Xilinx V4L2 compliant driver that registers with the Linux V4L2 framework. The TPG/ADV7611 and Sobel filter V4L2 sub device drivers register with this pipeline driver to allow the user application to manage the base TRD video pipeline adhering to the V4L2 standard.

  The Video pipeline driver also makes use of the Xilinx VDMA driver to move video data within the base TRD video pipeline.

- Xylon Display Controller DRM Driver

  This driver (from Xylon) adheres to DRM framework in Linux and drives the Xylon CVC display controller. The DRM driver facilitates the DMA included in Xylon CVC to transfer video data (typically from TPG/ADV7611/Sobel IP) in DDR to Xylon CVC for display by ADV7511 (HDMI) to monitor. The driver also initializes the ADV7511 peripheral on ZC702 board through the Xilinx I2C driver.

- Xilinx VDMA Driver

  Xilinx VDMA driver adheres to the Linux DMA framework and controls the VDMA IP for facilitating video data transfers in the base TRD Video pipeline. The VDMA drivers APIs are invoked by the Xilinx video pipeline V4L2 driver to move video data within the video pipeline under control of the buffer management logic of application software.

- TPG Subdev Driver

  TPG driver adheres to the Linux V4L2 and Media framework, and registers with both the frameworks. The V4L2 control interface is used to configure TPG for generating different test patterns.

- ADV7611 Subdev Driver

  ADV7611 driver adheres to Linux V4L2 and Media framework and registers with both the frameworks. It also provides an interface to get/set EDID data, set/get video timing, etc.

- Sobel Filter V4L2 Subdev Driver

  This V4L2 sub-device driver is used to control the Sobel filter IP.

- ADV7511 Driver

  The ADV7511 driver is used by the Xylon CVC DRM driver to configure the display of video output over HDMI. The ADV7511 driver is stacked on the I2C driver. It reads EDID data that has been broadcasted by Monitor.

The Zynq Base TRD uses PetaLinux framework for building/customizing software artifacts. See Appendix D, PetaLinux Software Development Kit for more details on PetaLinux SDK.

Three major software components are involved in the Base TRD:

- Xilinx Boot Loader
- Xilinx Linux kernel
- Application

Figure 2-5 shows the software architecture.

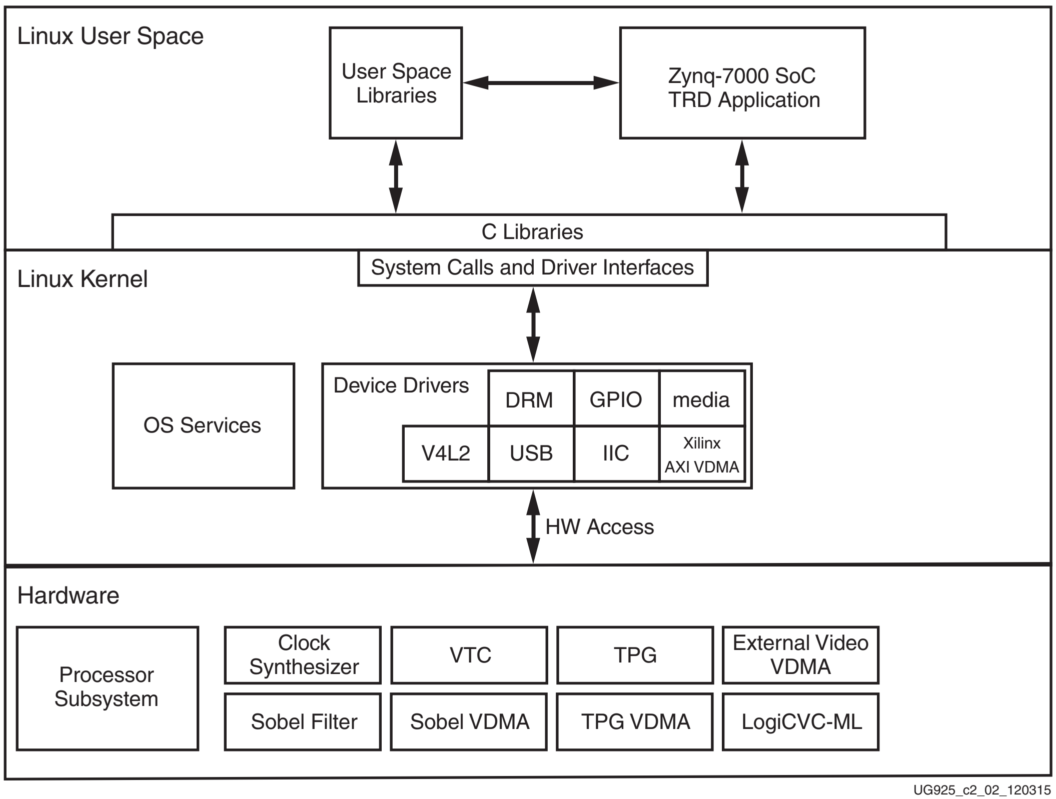


Figure 2-5: **Software Architecture: Top Level View**

Boot Loader

Responsible for the power-on boot-up process, the non-changeable boot code resides in the boot ROM. At power-on, the boot ROM reads the boot mode register to determine the boot mode. The boot mode is user-configurable.

This TRD uses SD card boot mode for booting.

In all modes except JTAG, the boot ROM reads the boot configuration header from the selected boot media, which is based on the boot mode register. In SD Card mode the boot header is located in the `BOOT.bin` file, located on the first partition of the SD card. For other boot modes, see the *Zynq-7000 SoC Technical Reference Manual* (UG585) [Ref 1].

File Elements

The file contains these elements:

Boot header - It contains information about the other contents of `BOOT.bin` along with their offsets and sizes, as well as whether this is a secure or non-secure boot.

First Stage Boot Loader (FSBL) - FSBL is responsible for initializing the minimum required hardware to program the PL bitstream, and load and execute U-Boot. For details, see Boot Loader, page 35.

Bitstream - PL hardware bitstream that gets programmed in FSBL.

U-Boot - Second stage boot loader responsible to complete initializing the hardware, and load and execute the Linux kernel. For details, see Boot Loader, page 35.

Boot Loader

A two-stage boot loader is used for the Zynq-7000 SoC Linux boot-up.

FSBL - Initializes the required hardware in the PS along with PL programming and loads the second-stage boot loader, U-Boot. The FSBL source code is generated through the Petalinux SDK, depending on the hardware design specification.

U-Boot - Loads the kernel image in the DDR memory. U-Boot is an open source universal boot loader used across various embedded platforms. The source code, customized for Zynq-7000 SoC Linux, is available on the Xilinx Open Source ARM Git Repository: https://github.com/xilinx/.

Refer to the Zynq-7000 Base Targeted Reference Design wiki page at wiki.xilinx.com/zc702-base-trd. to build the FSBL and U-Boot, boot image `BOOT.BIN`.

Xilinx Linux Kernel

The Base TRD uses the Petalinux kernel which is based on the mainline open source kernel Git tree, adding support for a variety of Xilinx IP core drivers and reference boards. The source code is available on the Xilinx Open Source ARM Git Repository: https://github.com/xilinx.

Table 2-9 lists kernel drivers used for the Base TRD.

Table 2-9: **Linux Kernel Drivers Used by the Base TRD**

Linux Driver	Function	Called By
Xylon DRM	Drives the display controller (logiCVC-ML) to display the application UI and control datapath	• TRD application uses libdrm library to interface with DRM driver. • Linux fbconsole uses frame buffer interface to emulate console.
Xilinx video test pattern generator (V4L2)	Configures TPG and provides control using standard V4L2 interface. Registers itself a media entity.	• TRD application uses v4l2 interface to configure TPG parameters. • Xilinx video pipeline driver uses tpg subdev for stop/stop operations
Xilinx video pipeline	Configures xilinx video pipeline. Registers as a media and video device. Sets up links and configures all entities in the pipeline.	• TRD application uses media control lib to set up link. • TRD application uses V4L2 interface for V4L2 buffers handling and stream on/off operations.
Xilinx VDMA engine driver	Configures Xilinx VDMA IP.	Xilinx video pipeline driver uses slave-DMA API to configure/start/stop VDMA.
Xilinx APM driver (UIO)	Provides a UIO-based framework for monitoring APM device.	TRD application uses UIO framework to map device address space, and then control device using write/read as per defined register map.
Xilinx IIC driver	Configures IIC controller. Provides I2C write/read functions.	Used by Analog device transmitter/receiver chips (ADV7511, ADV7611).
ADV7511 encoder	DRM compliant HDMI transmitter driver.	Display controller (logiCVC-ML).
ADV7611	V4L2 compliant HDMI receiver driver.	Used by Xilinx video pipeline driver.

Application

Figure 2-6 describes various components of the application.

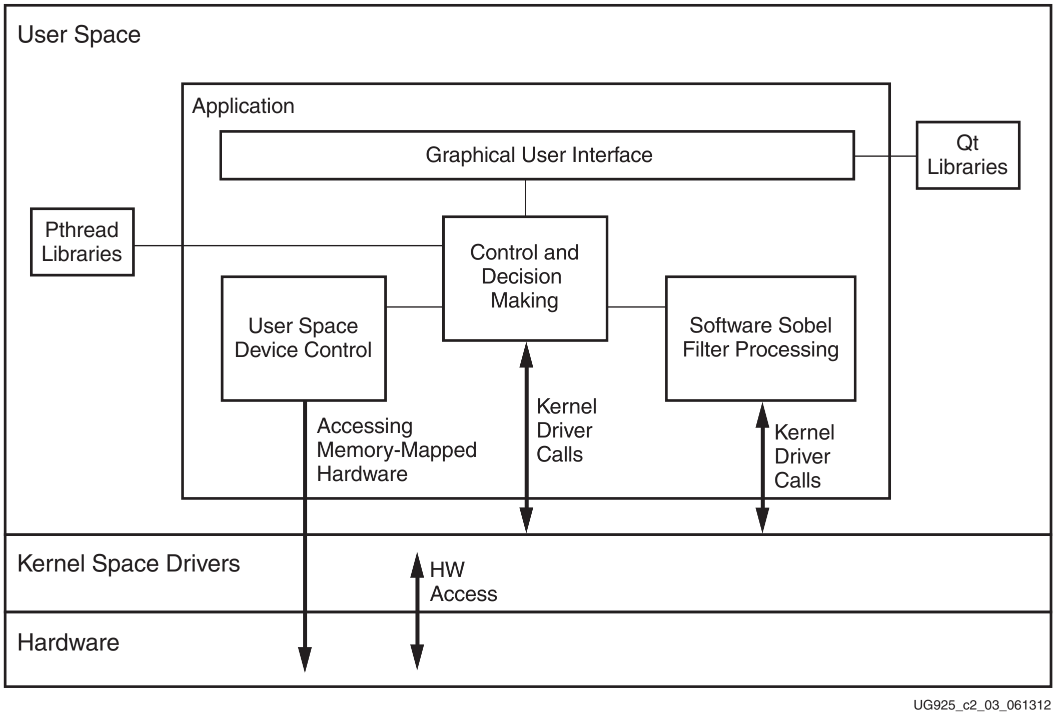


Figure 2-6: **Application Functional Blocks**

The application is divided into the following functional blocks:

- GUI
- Control and decision making
- User space device control
- Software Sobel filter processing

The first three components run in one thread of the application, while the software Sobel filter runs in a second separate thread. The Linux kernel is in SMP mode and both the cores are utilized for running different threads. At run time, the kernel assigns one of the cores for software Sobel filter processing and other core for running the rest of the application and the operating system.

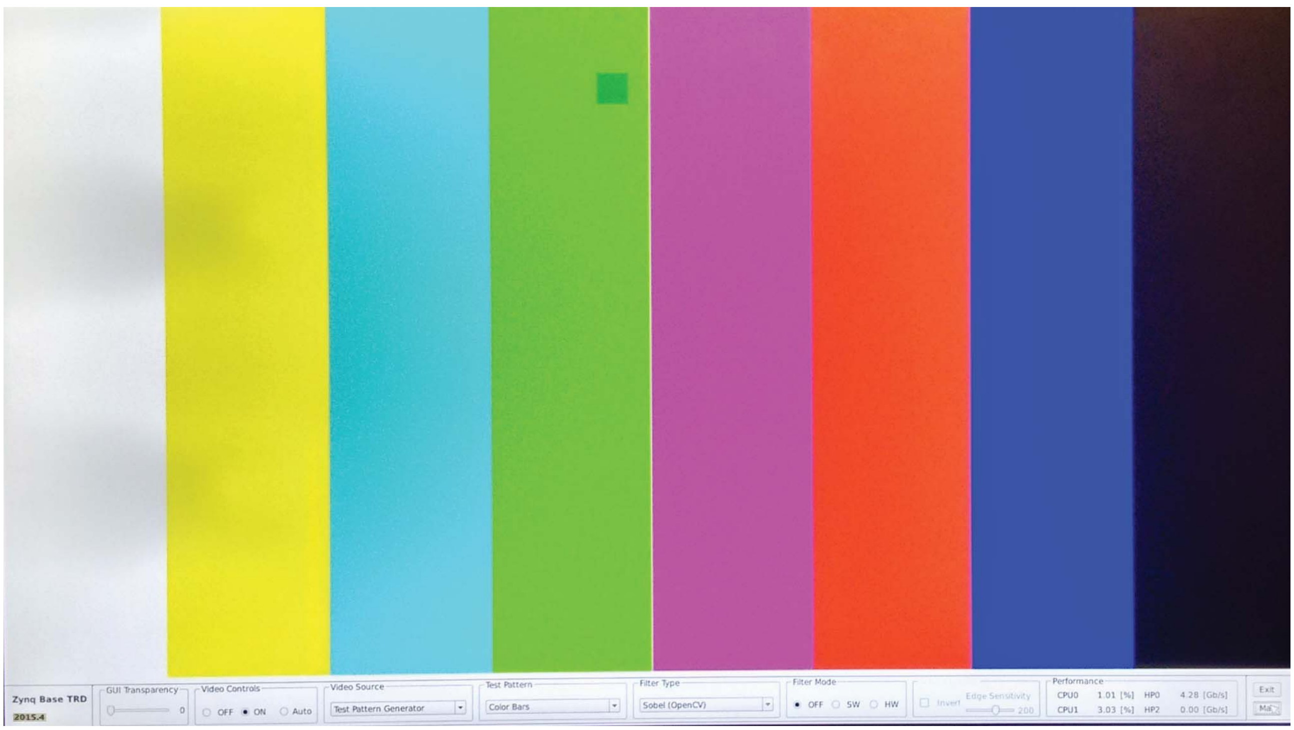
Graphical User Interface

The GUI for this Base TRD is designed using the Qt framework (see Figure 2-7).

UG925_c2_07_121415

Figure 2-7: **GUI for TRD Application**

In Figure 2-8, the gray screen at the bottom is the GUI. The GUI can be minimized or maximized with the **MIN/MAX** button on the right side. The complete screen is the Video (or display) area, where the pattern or video is displayed. There is a transparency slider that makes the GUI semi-transparent. The bottom portion of the GUI consists of the ZYNQ banner, a pictorial view of Operation Mode and two graphs, which are only available in the MAX GUI mode. All other controls are available in both the MAX and MIN GUI mode. Additionally, there is AUTO mode in the GUI that runs the demo and randomly selects video sources and filter type without user intervention.



UG925_c2_08_121415

Figure 2-8: **Minimized GUI Mode**

Figure 2-8 shows the minimized (MIN) GUI mode.

The main functionality of the GUI includes:

- Getting user inputs
- Plotting graphs
- Displaying the video area

Get User Inputs

The Qt framework provides for having the mouse as input device (it internally uses Linux USB-HID class drivers). The input from the user includes **video Enable/Disable**, **Input Source Select**, **Sobel Control Settings**, and **Mode Select** for the video pipeline.

Plot Graph

Two graphs are plotted using the Qt framework. The first graph demonstrates CPU utilization for each ARM core, and the second demonstrates AXI memory bandwidth utilization on HP0 and HP2 ports.

In the CPU utilization graph, the horizontal axis is for time and the vertical axis is for the percentage of CPU utilization.

In the memory bandwidth graph, the horizontal axis is for time and the vertical axis is for the Gb/s of read and write transactions on AXI.

Along with the graphs, the utilization and bandwidth numbers are also displayed above the graphs. These numbers are available in both MIN and MAX GUI mode.

Display Video Area

This is the full screen area, where the output of the video pipeline is displayed.

Control and Decision Making

This block receives input from the GUI and maintains the state transition for the complete application. It communicates with all other blocks of the application and with the kernel drivers to change the state of hardware.

Data Flow Use Cases

The video source can be either the video TPG (internal) or an external video source (HDMI IN). The six combinations of data flow use cases are listed in this section. Use cases 1, 2, and 3 use the TPG and use cases 4, 5, and 6 use external video as video source.

The Sobel filter can either be a software or hardware implementation or it is bypassed entirely (turned off). Use cases 1 and 4 have the filter turned off, use cases 2 and 5 refer to the software implementation, and use cases 3 and 6 refer to the hardware implementation. Figure 2-9 combines TPG/external video in one functional block. Figure 2-10 combines software and hardware implementation in one functional block, therefore, use cases 4, 5, and 6 refer to both Figure 2-9 and Figure 2-10.

There are six combinations of the data flow:

1. The TPG creates and writes the pattern in reserved video memory in DDR. The display controller displays video memory (TPG pattern) as shown in Figure 2-9.

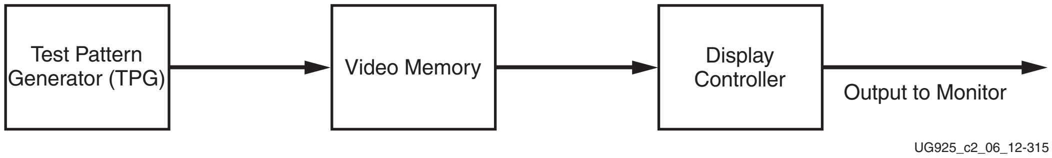


Figure 2-9: **Video Display Pipeline Data Flow**

2. Figure 2-10 shows the software Sobel filter configuration. The TPG creates and writes the pattern in intermediate DDR memory. The software Sobel filter reads the intermediate DDR memory, detects the edges, and writes the filtered image in the reserved video memory in DDR. The display controller displays video memory (filtered TPG pattern) as shown in Figure 2-10.

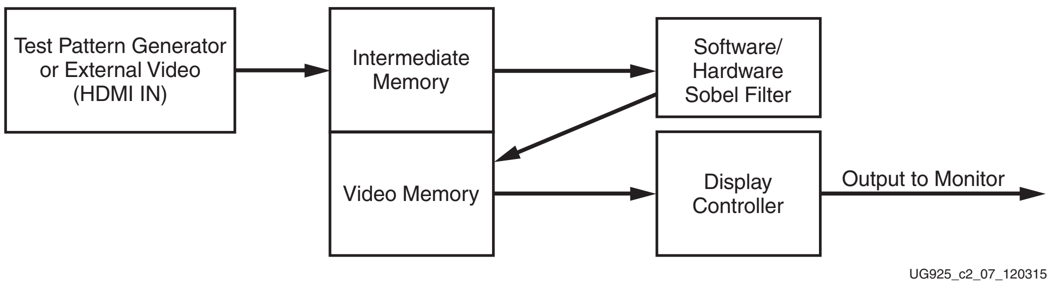


Figure 2-10: **Video Processing and Display Pipeline Data Flow**

3. Figure 2-10 shows the hardware Sobel filter configuration. The TPG creates and writes the pattern in intermediate DDR memory. The hardware Sobel filter reads the intermediate DDR memory (using VDMA), detects the edge, and writes the filtered image in the reserved video memory in DDR (using VDMA). The display controller displays video memory (filtered TPG pattern).

Similarly, there are three more cases (4, 5, and 6) where instead of an internally generated pattern, external video is used. See Figure 2-9 and Figure 2-10 with external video (HDMI IN) used instead of TPG for the video source. Cases 4, 5, and 6 are the same as cases 1, 2, and 3 with the only difference being the input source. In cases 1, 2, and 3, the input comes from TPG, whereas in cases 4, 5, and 6, the input comes from external video.

User Space Device Control

The UIO framework is designed to allow light-weight, low-overhead access to hardware devices directly from user space.

The Xilinx APM driver is added as a UIO driver in the kernel that provides access to device address space and configuration parameters that are passed through the device-tree. It provides the device base address in map[0] (xilinx_apm) and a structure containing APM parameters in map[1] (axi-pmon). The APM driver also implements interrupt handling and the interrupt status register is updated in APM structure.

AXI performance monitor IP is configured and controlled from the user space.

The APIs to configure AXI performance monitor IP are explained in the perfmon_lib which is part of the application source code provided with the Base TRD.

Software Sobel Filter Processing

This section describes the implementation of the software Sobel filter algorithm for edge detection implemented using OpenCV. The algorithm runs in a separate thread and is turned On and Off by the controlling block. The application is compiled with the highest compiler optimization level (O3), so that the algorithm gives best performance. This algorithm takes the buffer address of the original frame as input and writes the filter image on the buffer address provided as output.

The following API is used for software Sobel filter processing:

```
void opencv_sobel(IplImage *_src, IplImage *_dst)
```

An object's edge detection filter algorithm for imaging/video looks for gradient magnitude changes in the image background and processing is performed in both the horizontal and vertical directions. The algorithm takes the YUV raw formatted image and detects the image object's edges based on the pixel luminance values.

Register Description

This appendix describes the details about configuration and control registers most commonly accessed by the Linux driver and application. The registers implemented in hardware are memory-mapped to the PS address range directly.

Sobel Filter Registers

The Sobel filter registers are used to configure and control various internal features within the Sobel filter logic.

Base address	`0x400D0000`
Version	N/A

Control and Status Register

The relative address of the Control and Status register is `0x00`. Table A-1 describes this register's structure.

Relative address	`0x00`

Table A-1: **Control and Status Register**

Bit Position	Mode	Default Value	Description
31:8-6:4	-	-	Reserved.
7	RW	`0x0`	Auto Restart
3	R	`0x1`	IP is ready to accept new data.
2	R	`0x1`	IP is idle.
1	COR	`0x0`	Frame process is done.
0	RW/SC	`0x0`	Start processing the frame.

Global Interrupt Enable Register

The relative address of the Global Interrupt Enable register is `0x04`. Table A-2 describes this register's structure.

Relative address 0x04

Table A-2: **Global Interrupt Enable Register**

Bit Position	Mode	Default Value	Description
31:1	-	-	Reserved.
0	RW	0x0	Global interrupt enable.

Interrupt Enable Register

The relative address of the Interrupt Enable register is 0x08. Table A-3 describes this register's structure.

Relative address 0x08

Table A-3: **Interrupt Enable Register**

Bit Position	Mode	Default Value	Description
31:1	-	-	Reserved.
0	RW	0x0	Frame processing done interrupt enable.

Interrupt Status Register

The relative address of the Interrupt Status register is 0x0C. Table A-4 describes this register's structure.

Relative address 0x0C

Table A-4: **Interrupt Status Register**

Bit Position	Mode	Default Value	Description
31:2	-	-	Reserved.
1	RW	0x0	IP ready interrupt status.
0	RW	0x0	Frame processing done interrupt status.

Number of Rows Register

The relative address of the Number of Rows register is 0x14. Table A-5 describes this register's structure.

Relative address 0x14

Table A-5: **Number of Rows RegisterSoC**

Bit Position	Mode	Default Value	Description
31:0	RW	-	Number of rows in a frame.

Number of Columns Register

The relative address of the Number of Columns register is `0x1C`. Table A-6 describes this register's structure.

Relative address 0x1C

Table A-6: **Number of Columns Register**

Bit Position	Mode	Default Value	Description
31:0	RW	-	Number of columns in a frame.

XR0C0 Coefficient Register

The relative address of the XR0C0 Coefficient Register is `0x24`. Table A-7 describes this register's structure.

Relative address 0x24

Table A-7: **XR0C0 Coefficient Register**

Bit Position	Mode	Default Value	Description
31:0	RW	0x0	XR0C0 coefficient for Sobel filtering.

XR0C1 Coefficient Register

The relative address of the XR0C1 Coefficient Register is `0x2C`. Table A-8 describes this register's structure.

Relative address 0x2C

Table A-8: **XR0C1 Coefficient Register**

Bit Position	Mode	Default Value	Description
31:0	RW	0x0	XR0C1 coefficient for Sobel filtering.

XR0C2 Coefficient Register

The relative address of the XR0C2 Coefficient Register is 0x34. Table A-9 describes this register's structure.

Relative address 0x34

Table A-9: **XR0C2 Coefficient Register**

Bit Position	Mode	Default Value	Description
31:0	RW	0x0	XR0C2 coefficient for Sobel filtering.

XR1C0 Coefficient Register

The relative address of the XR1C0 Coefficient Register is 0x3C. Table A-10 describes this register's structure.

Relative address 0x3C

Table A-10: **XR1C0 Coefficient Register**

Bit Position	Mode	Default Value	Description
31:0	RW	0x0	XR1C0 coefficient for Sobel filtering.

XR1C1 Coefficient Register

The relative address of the XR1C1 Coefficient Register is 0x44. Table A-11 describes this register's structure.

Relative address 0x44

Table A-11: **XR1C1 Coefficient Register**

Bit Position	Mode	Default Value	Description
31:0	RW	0x0	XR1C1 coefficient for Sobel filtering.

XR1C2 Coefficient Register

The relative address of the XR1C2 Coefficient Register is 0x4C. Table A-12 describes this register's structure.

Relative address 0x4C

Table A-12: **XR1C2 Coefficient Register**

Bit Position	Mode	Default Value	Description
31:0	RW	0x0	XR1C2 coefficient for Sobel filtering.

XR2C0 Coefficient Register

The relative address of the XR2C0 Coefficient Register is 0x54. Table A-13 describes this register's structure.

Relative address 0x54

Table A-13: **XR2C0 Coefficient Register**

Bit Position	Mode	Default Value	Description
31:0	RW	0x0	XR2CO coefficient for Sobel filtering.

XR2C1 Coefficient Register

The relative address of the XR2C1 Coefficient Register is 0x5C. Table A-14 describes this register's structure.

Relative address 0x5C

Table A-14: **XR2C1 Coefficient Register**

Bit Position	Mode	Default Value	Description
31:0	RW	0x0	XR2C1 coefficient for Sobel filtering.

XR2C2 Coefficient Register

The relative address of the XR2C2 Coefficient Register is 0x64. Table A-15 describes this register's structure.

Relative address 0x64

Table A-15: **XR2C2 Coefficient Register**

Bit Position	Mode	Default Value	Description
31:0	RW	0x0	XR2C2 coefficient for Sobel filtering.

YR0C0 Coefficient Register

The relative address of the YR0C0 Coefficient Register is 0x6C. Table A-16 describes this register's structure.

Relative address 0x6C

Table A-16: **YR0C0 Coefficient Register**

Bit Position	Mode	Default Value	Description
31:0	RW	0x0	YR0C0 coefficient for Sobel filtering.

YR0C1 Coefficient Register

The relative address of the YR0C1 Coefficient Register is 0x74. Table A-17 describes this register's structure.

Relative address 0x74

Table A-17: **YR0C1 Coefficient Register**

Bit Position	Mode	Default Value	Description
31:0	RW	0x0	YR0C1 coefficient for Sobel filtering.

YR0C2 Coefficient Register

The relative address of the YR0C2 Coefficient Register is 0x7C. Table A-18 describes this register's structure.

Relative address 0x7C

Table A-18: **YR0C2 Coefficient Register**

Bit Position	Mode	Default Value	Description
31:0	RW	0x0	YR0C2 coefficient for Sobel filtering.

YR1C0 Coefficient Register

The relative address of the YR1C0 Coefficient Register is 0x84. Table A-19 describes this register's structure.

Relative address 0x84

Table A-19: **YR1C0 Coefficient Register**

Bit Position	Mode	Default Value	Description
31:0	RW	0x0	YR1C0 coefficient for Sobel filtering.

YR1C1 Coefficient Register

The relative address of the YR1C1 Coefficient Register is 0x8C. Table A-20 describes this register's structure.

Relative address 0x8C

Table A-20: **YR1C1 Coefficient Register**

Bit Position	Mode	Default Value	Description
31:0	RW	0x0	YR1C1 coefficient for Sobel filtering.

YR1C2 Coefficient Register

The relative address of the YR1C2 Coefficient Register is 0x94. Table A-21 describes this register's structure.

Relative address 0x94

Table A-21: **YR1C2 Coefficient Register**

Bit Position	Mode	Default Value	Description
31:0	RW	0x0	YR1C2 coefficient for Sobel filtering.

YR2C0 Coefficient Register

The relative address of the YR2C0 Coefficient Register is 0x9C. Table A-22 describes this register's structure.

Relative address 0x9C

Table A-22: **YR2C0 Coefficient Register**

Bit Position	Mode	Default Value	Description
31:0	RW	0x0	YR2C0 coefficient for Sobel filtering.

YR2C1 Coefficient Register

The relative address of the YR2C1 Coefficient Register is 0xA4C. Table A-23 describes this register's structure.

Relative address 0xA4

Table A-23: **YR2C1 Coefficient Register**

Bit Position	Mode	Default Value	Description
31:0	RW	0x0	YR2C1 coefficient for Sobel filtering.

YR2C2 Coefficient Register

The relative address of the YR2C2 Coefficient Register is 0xAC. Table A-24 describes this register's structure.

Relative address 0xAC

Table A-24: **YR2C2 Coefficient Register**

Bit Position	Mode	Default Value	Description
31:0	RW	0x0	YR2C2 coefficient for Sobel filtering.

High Threshold Register

The relative address of the High Threshold register is 0xB4. Table A-25 describes this register's structure.

Relative address 0xB4

Table A-25: **High Threshold Register**

Bit Position	Mode	Default Value	Description
31:8	-	–	Reserved.
7:0	RW	0x0	High Threshold for Sobel filtering.

Low Threshold Register

The relative address of the Low Threshold register is 0xBC. Table A-26 describes this register's structure.

Relative address	0xBC

Table A-26: **Low Threshold Register**

Bit Position	Mode	Default Value	Description
31:8	-	–	Reserved.
7:0	RW	0x0	Low Threshold for Sobel filtering.

Invert Output Register

The relative address of the Invert Output register is 0xC4. Table A-27 describes this register's structure.

Relative address	0xC4

Table A-27: **Invert Output Register**

Bit Position	Mode	Default Value	Description
31:1			Reserved.
0	RW	0x0	Inverts output of Sobel filter.

Extended Display Identification Data

The Extended Display Identification Data (EDID) protocol is a standardized means for a display to communicate its operational characteristics, such as native resolution, to a source device. This allows the source to generate the necessary video characteristics to match the needs of the display. Because the source generates the necessary video characteristics, this removes dependency on the user to configure the source and display device, thus reducing the potential for incorrect settings and adjustments.

Storing and Programming EDID data

Contents of EDID data can be modified by the ADVANTIV EDID editor:
http://ez.analog.com/docs/DOC-2143

Directory Structure

This appendix describes the directory structure and organization of the files and folders delivered with the package.

Included Files and Systems

Figure C-1 gives a top-level view of the directories and files included in the Zynq®-7000 SoC ZC702 Base TRD package. Table C-1 summarizes how the directories provided relate to this user guide. This user guide uses the directories in the extracted Base TRD zipped package and assumes that the directories and files are placed or copied onto your local computer.

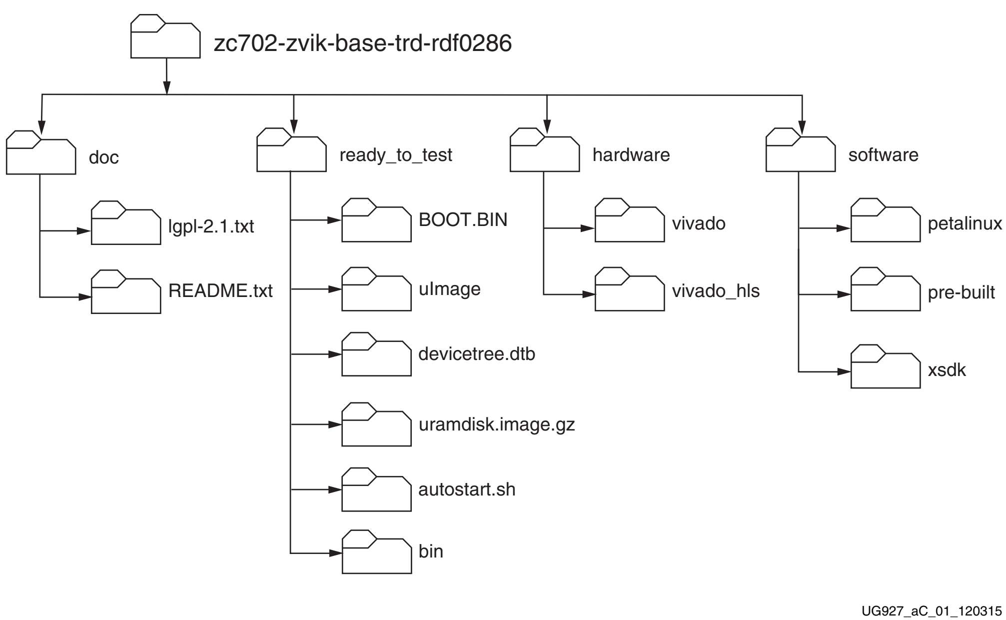


Figure C-1: **Directory Structure**

Table C-1: **Explanation of Directories in the Zynq-7000 SoC ZC702 Base TRD**

Directory	Purpose
`doc`	This directory contains the documents provided with the Zynq-7000 SoC ZC702 Base TRD.
`ready_to_test`	This directory contains a pre-built hardware design and software executables which provide a quick way to run the video demonstration. The application binary images are compatible to boot from the SD MMC and run the Linux application.
`hardware`	This directory includes the Zynq-7000 SoC ZC702 Base TRD hardware design and VHLS project to build Sobel Filter IP.
`software`	This directory includes the Base TRD Petalinux project and application sources. It also includes a pre-built directory that contains third-party binaries /libraries and headers which are used in TRD.

PetaLinux Software Development Kit

The PetaLinux Software Development Kit (SDK) is a complete embedded Linux distribution and development environment that works with the Xilinx® hardware design flow for Xilinx FPGAs and Zynq®-7000 SoCs that implement designs using the MicroBlaze™ processor.

With PetaLinux, you can:

- Synchronize the hardware platform and software platform in one step
- Propagate user applications to embedded Linux systems using Zynq-7000 devices
- Test a Linux system based on a Zynq-7000 device in a virtual machine environment using QEMU.

The PetaLinux SDK will automatically generate a custom, Linux board support package (BSP) including device drivers for Xilinx embedded processing IP cores, kernel and boot loader configurations. This capability minimizes low-level software tasks, and allows engineers to focus on their value-added applications. Development templates, tools, debuggers and trace utilities are integrated into the SDK that allow software teams to create new device drivers, applications, libraries and BSP configurations for their custom design.

The PetaLinux SDK includes tools for the custom configuration of the boot loader, Linux kernel, file system, libraries, and system parameters. These configuration tools are integrated with Xilinx hardware development tools and custom hardware-specific data files so that, for example, device drivers for Xilinx embedded IP cores will be automatically built and deployed according to the engineer-specified address mapping of that device.

The PetaLinux SDK includes:

- Command-line interface and Eclipse-based IDE
- Application, device driver, and library generators
- Development templates
- Bootable system image builder
- Debug and trace tools
- GCC tools
- Integrated QEMU full system simulator
- Automated tools

Developers can use these tools to customize the boot loader, Linux kernel, or Linux applications. They can add new device drivers, applications, libraries, and boot and test software stacks on the included full system simulator (QEMU) or on physical hardware via network or JTAG.

The PetaLinux SDK complements use of the Xilinx SDK by allowing application developers to continue using the Xilinx SDK to build and deploy against any Linux system that is configured with the PetaLinux SDK tools.

- *PetaLinux SDK User Guide: Installation Guide* (UG976) [Ref 10]
- *PetaLinux SDK User Guide: Getting Started Guide* (UG977) [Ref 11]
- *PetaLinux SDK User Guide: Board Bringup Guide* (UG980) [Ref 12]
- *PetaLinux SDK User Guide: Application Development Guide* (UG981) [Ref 13]

PetaLinux SDK documents can be found at:

www.xilinx.com/support/index.html/content/xilinx/en/supportNav/design_tools/petaLinux-sdk.html

Install the Zynq-7000 SoC Design and Development Environment

Install the Xilinx Vivado Design Suite

The user needs to install the Embedded Edition and System Edition of the Xilinx Vivado Design Suite. See *Vivado Design Suite User Guide: Release Notes, Installation* (UG973) [Ref 14] to install and license the Vivado Design Suite.

Additional Resources

Xilinx Resources

For support resources such as Answers, Documentation, Downloads, and Forums, see the Xilinx Support website.

For continual updates, add the Answer Record to your myAlerts.

For definitions and terms, see the Xilinx Glossary.

Solution Centers

See the Xilinx Solution Centers for support on devices, software tools, and intellectual property at all stages of the design cycle. Topics include design assistance, advisories, and troubleshooting tips.

References

The most up to date information related to the ZC702 Evaluation Kit and its documentation is available on these websites:

ZC702 Evaluation Kit

ZC702 Evaluation Kit Documentation

ZC702 Evaluation Kit Master Answer Record (AR 47864)

These Xilinx documents and sites provide supplemental material useful with this guide:

1. *Zynq-7000 SoC Technical Reference Manual* (UG585)
2. *LogiCORE IP Processor System Reset Module Product Specification* (PG164)
3. *LogiCORE IP AXI Interconnect* (PG059)

4. *LogiCORE IP AXI Video Direct Memory Access Product Guide* (PG020)
5. *LogiCORE IP Video Timing Controller Product Guide* (PG016)
6. *LogiCORE IP Test Pattern Generator Product Guide* (PG103)
7. *LogiCORE IP AXI Performance Monitor Product Guide* (PG037)
8. *Implementing Memory Structures for Video Processing in the Vivado HLS Tool* (XAPP793)
9. *LogiCORE IP Video In to AXI4-Stream v1.0* (PG043)
10. *PetaLinux SDK User Guide: Installation Guide* (UG976)
11. *PetaLinux SDK User Guide: Getting Started Guide* (UG977)
12. *PetaLinux SDK User Guide: Board Bringup Guide* (UG980)
13. *PetaLinux SDK User Guide: Application Development Guide* (UG981)
14. *Vivado Design Suite User Guide: Release Notes, Installation* (UG973)
15. *Zynq-7000 SoC Software Developers Guide* (UG821)
16. *Xilinx Design Tools: Installation and Licensing Guide* (UG798)
17. *Zynq-7000 SoC: Concepts, Tools, and Techniques Guide* (UG873)
18. *Zynq-7000 SoC Overview* (DS190)
19. *Vivado Design Suite Migration Methodology Guide* (UG911)
20. Zynq-7000 SoC webpage
21. Zynq-7000 SoCs Product Table
22. Xilinx PetaLinux Software Development Kit webpage

These external websites provide supplemental material useful with this guide:

23. GitHub
    Xilinx Wiki open source ARM Git repository
24. Install Xilinx Tools
    Xilinx Wiki ARM GNU tools
25. Using Git
    A public Git tree for Xilinx open source projects
26. Git Webpage
    Downloads, documentation and information for Git open-source version control
27. Open Source Linux
    Xilinx wiki open source developer resources
28. Device Tree Main Page
    Device tree general information

29. Xylon LogicBRICKS
logiCVC-ML Compact Multilayer Video Controller IP core, description and data sheet
30. Qt Documentation
Documentation for the QT cross-platform UI and software application development framework
31. ARM Information Center
AMBA AXI4-stream protocol specification
32. Silicon Labs
Si570, Si5324C, CP2103GM, USB to UART Bridge, VCP Drivers

Regulatory and Compliance Information

This product is designed and tested to conform to the European Union directives and standards described in this section.

Declaration of Conformity

To view the Declaration of Conformity online, please visit:

http://www.xilinx.com/support/documentation/boards_and_kits/ce-declarations-of-conformity-xtp251.zip

Directives

2006/95/EC, *Low Voltage Directive (LVD)*

2004/108/EC, *Electromagnetic Compatibility (EMC) Directive*

Standards

EN standards are maintained by the European Committee for Electrotechnical Standardization (CENELEC). IEC standards are maintained by the International Electrotechnical Commission (IEC).

Electromagnetic Compatibility

EN 55022:2010, *Information Technology Equipment Radio Disturbance Characteristics – Limits and Methods of Measurement*

EN 55024:2010, *Information Technology Equipment Immunity Characteristics – Limits and Methods of Measurement*

This is a Class A product. In a domestic environment, this product can cause radio interference, in which case the user might be required to take adequate measures.

Safety

IEC 60950-1:2005, *Information technology equipment – Safety, Part 1: General requirements*

EN 60950-1:2006, *Information technology equipment – Safety, Part 1: General requirements*

Markings

This product complies with Directive 2002/96/EC on waste electrical and electronic equipment (WEEE). The affixed product label indicates that the user must not discard this electrical or electronic product in domestic household waste.

This product complies with Directive 2002/95/EC on the restriction of hazardous substances (RoHS) in electrical and electronic equipment.

This product complies with CE Directives 2006/95/EC, *Low Voltage Directive (LVD)* and 2004/108/EC, *Electromagnetic Compatibility (EMC) Directive.*

Appendix H

Warranty

THIS LIMITED WARRANTY applies solely to standard hardware development boards and standard hardware programming cables manufactured by or on behalf of Xilinx (“Development Systems”). Subject to the limitations herein, Xilinx warrants that Development Systems, when delivered by Xilinx or its authorized distributor, for ninety (90) days following the delivery date, will be free from defects in material and workmanship and will substantially conform to Xilinx publicly available specifications for such products in effect at the time of delivery. This limited warranty excludes: (i) engineering samples or beta versions of Development Systems (which are provided “AS IS” without warranty); (ii) design defects or errors known as “errata”; (iii) Development Systems procured through unauthorized third parties; and (iv) Development Systems that have been subject to misuse, mishandling, accident, alteration, neglect, unauthorized repair or installation. Furthermore, this limited warranty shall not apply to the use of covered products in an application or environment that is not within Xilinx specifications or in the event of any act, error, neglect or default of Customer. For any breach by Xilinx of this limited warranty, the exclusive remedy of Customer and the sole liability of Xilinx shall be, at the option of Xilinx, to replace or repair the affected products, or to refund to Customer the price of the affected products. The availability of replacement products is subject to product discontinuation policies at Xilinx. Customer may not return product without first obtaining a customer return material authorization (RMA) number from Xilinx.

THE WARRANTIES SET FORTH HEREIN ARE EXCLUSIVE. XILINX DISCLAIMS ALL OTHER WARRANTIES, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING, WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT, AND ANY WARRANTY THAT MAY ARISE FROM COURSE OF DEALING, COURSE OF PERFORMANCE, OR USAGE OF TRADE. (2008.10)

Do not throw Xilinx products marked with the “crossed out wheeled bin” in the trash. Directive 2002/96/EC on waste electrical and electronic equipment (WEEE) requires the separate collection of WEEE. Your cooperation is essential in ensuring the proper management of WEEE and the protection of the environment and human health from potential effects arising from the presence of hazardous substances in WEEE. Return the marked products to Xilinx for proper disposal. Further information and instructions for free-of-charge return available at: http:\\www.xilinx.com\ehs\weee.htm.