

# Joint ICTP-IAEA School on FPGA-based SoC and its Applications for Nuclear and Related Instrumentation | (smr 3562) 25 Jan – 19 Feb, 2021

## Organizers:

Mladen Bogovac (IAEA)

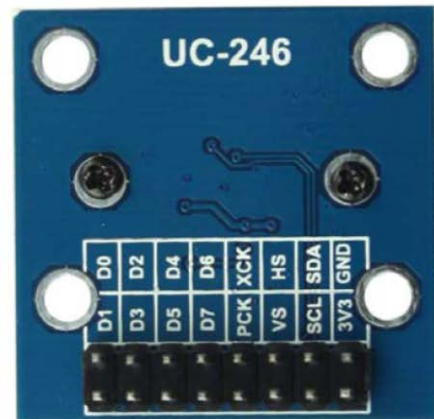
Andres Cicuttin (ICTP)

Local Organizer: Maria Liz Crespo (ICTP)

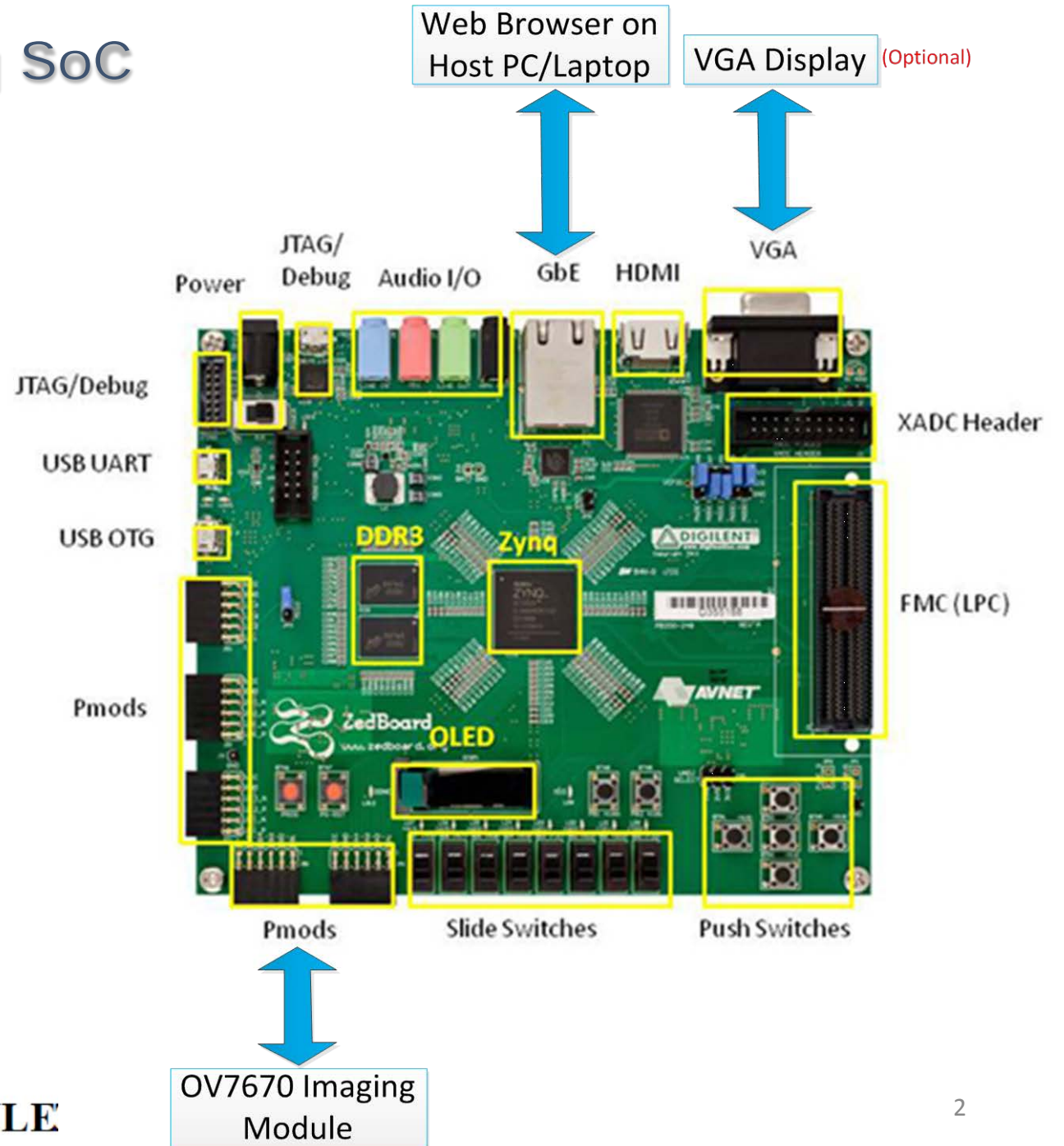
## WebCam Development using Zynq SoC

Presenter: Nasir Ahmed

# Webcam Development using Zynq SoC System Hardware

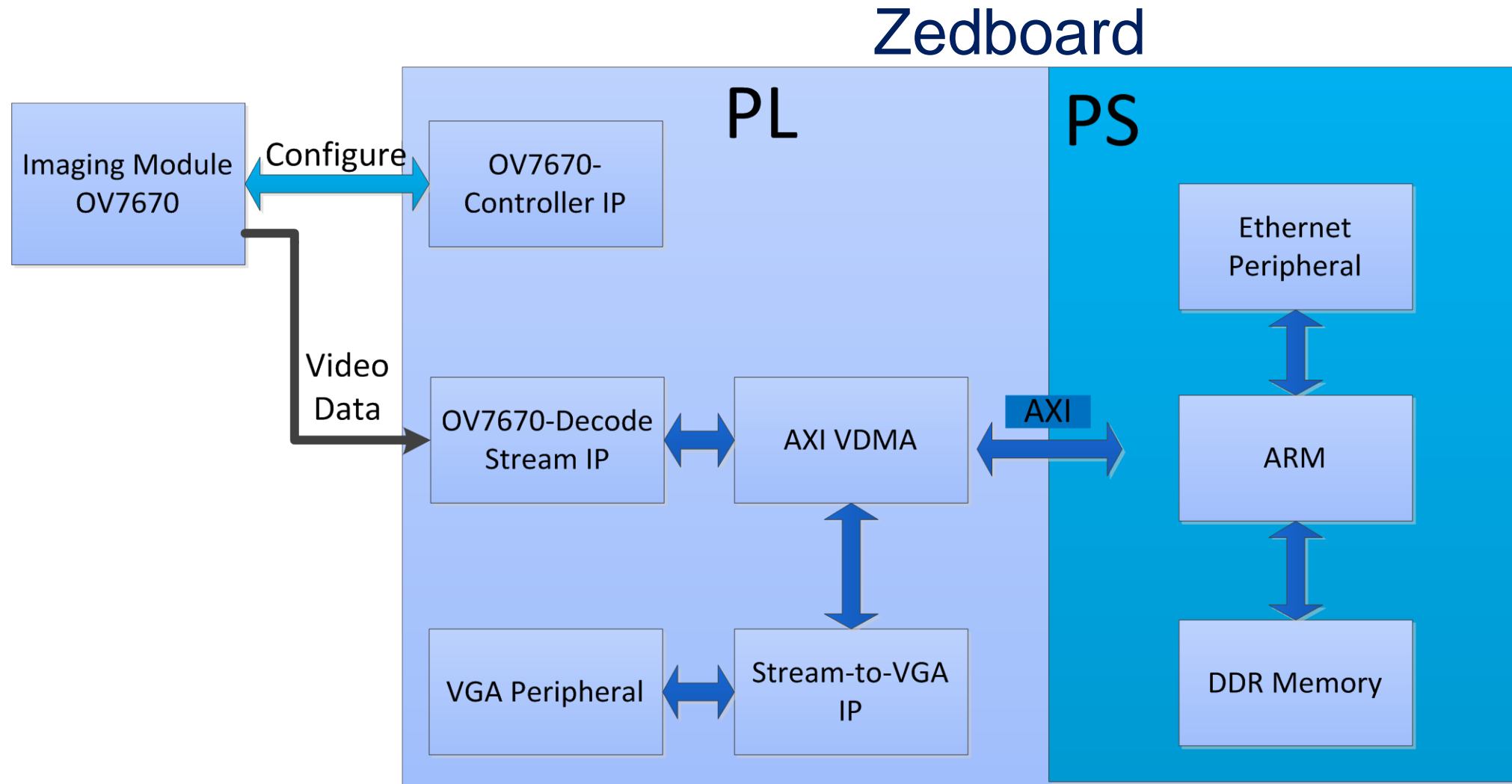


**0.3M Pixels CMOS OV7670 CAMERA MODULE**



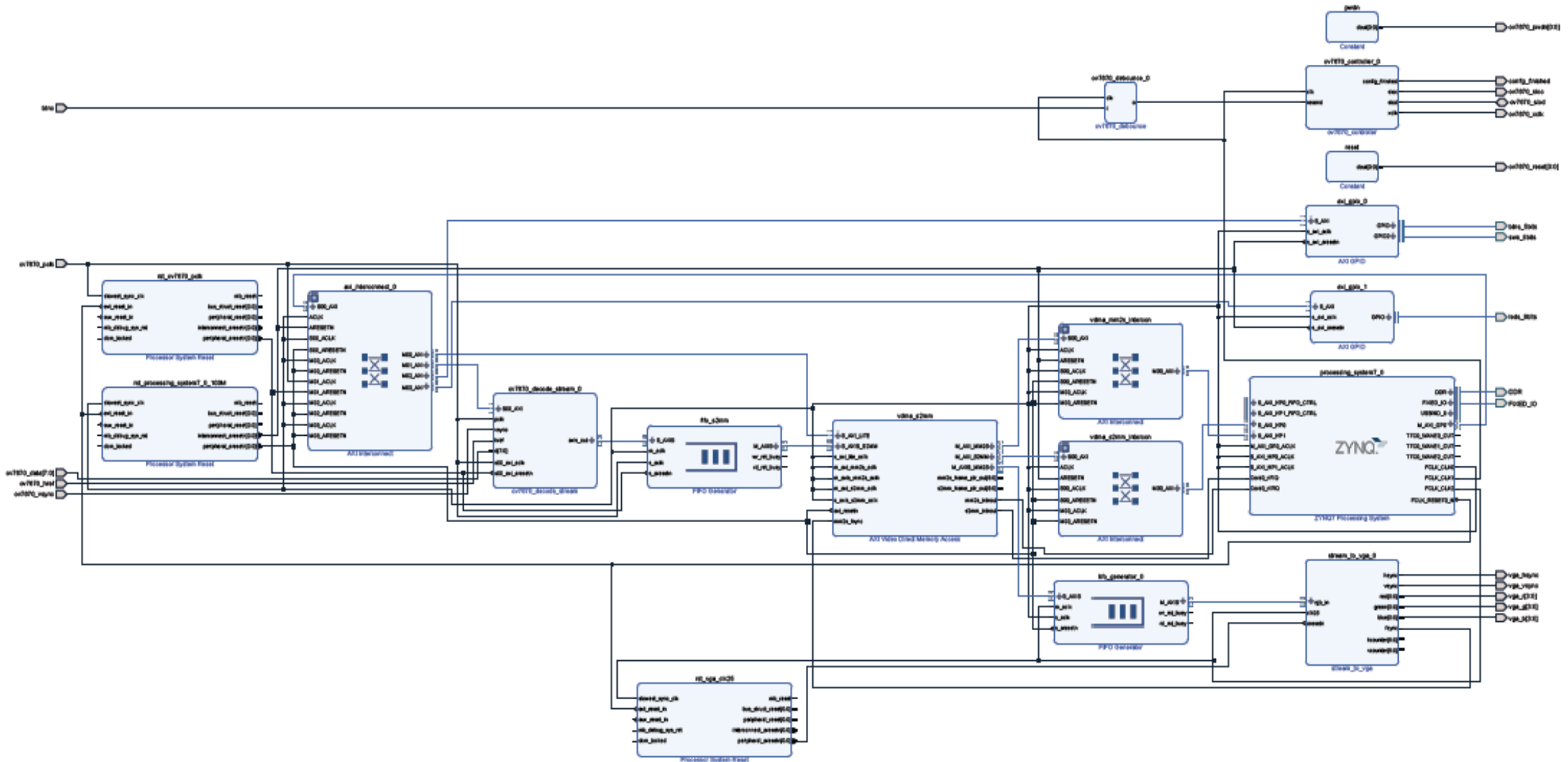
# Webcam Development using Zynq SoC

## Block Diagram



# Webcam Development using Zynq SoC

## Vivado Block Diagram



# Webcam Development using Zynq SoC

## Vivado Address Editor

The screenshot displays the Vivado 2019.1 IDE interface for a project named 'ov7670\_vdma'. The main window is the 'Address Editor', which shows a hierarchical view of memory segments. The 'Sources' window on the left shows the project structure, including 'vdma\_ov7670\_wrapper' and 'vdma\_ov7670\_i'. The 'Address Segment Properties' window shows the selected segment 'SEG\_ov7670\_decode\_stream\_0\_S00\_AXI\_reg'. The 'Address Editor' window displays a table of memory segments with columns for Cell, Slave Interface, Base Name, Offset Address, Range, and High Address. The 'Tcl Console' window at the bottom shows the output of the synthesis process, including the addition of various component instances.

**Address Editor Table:**

Cell	Slave Interface	Base Name	Offset Address	Range	High Address
processing_system7_0					
Data (32 address bits : 0x40000000 [ 1G ])					
axi_gpio_0	S_AXI	Reg	0x4120_0000	4K	0x4120_0FFF
axi_gpio_1	S_AXI	Reg	0x4121_0000	4K	0x4121_0FFF
ov7670_decode_stream_0	S00_AXI	S00_AXI_reg	0x4000_0000	4K	0x4000_0FFF
vdma_s2mm	S_AXI_LITE	Reg	0x4300_0000	4K	0x4300_0FFF
vdma_s2mm					
Data_MM2S (32 address bits : 4G)					
processing_system7_0	S_AXI_HP1	HP1_DDR_LOWOCM	0x0000_0000	512M	0x1FFF_FFFF
Data_S2MM (32 address bits : 4G)					
processing_system7_0	S_AXI_HP0	HP0_DDR_LOWOCM	0x0000_0000	512M	0x1FFF_FFFF

**Tcl Console Output:**

```
Adding component instance block -- xilinx.com:ip:axi_dwidth_converter:2.1 - auto_us
Adding component instance block -- xilinx.com:ip:axi_vdma:6.3 - vdma_s2mm
Adding component instance block -- xilinx.com:ip:axi_interconnect:2.1 - vdma_s2mm_intercon
Adding component instance block -- xilinx.com:ip:axi_protocol_converter:2.1 - auto_pc
Adding component instance block -- xilinx.com:ip:axi_dwidth_converter:2.1 - auto_us
Adding component instance block -- xilinx.com:ip:axi_gpio:2.0 - axi_gpio_0
```

# Webcam Development using Zynq SoC

## SDK2019.1: Standalone OS

The screenshot shows the Xilinx SDK IDE interface. The main window displays the source code for `helloworld.c`. The code includes headers and defines constants for stream address, VDMA S2MM address, and register offsets. The `#define VDMA_S2MM` line is highlighted in orange.

```
#include "xparameters.h"
#include "xil_io.h"
#include <stdint.h>

#define OV7670_STREAM                0x40000000
#define VDMA_S2MM                    0x43000000

#define HEIGHT                        480
#define WIDTH                         640

/* Register offsets */
#define OFFSET_PARK_PTR_REG           0x28
#define OFFSET_VERSION                0x2c

#define OFFSET_VDMA_MM2S_CONTROL_REGISTER 0x00
#define OFFSET_VDMA_MM2S_STATUS_REGISTER  0x04
#define OFFSET_VDMA_MM2S_VSIZE          0x50
```

The Project Explorer on the left shows the project structure, with the `Hello` directory selected. The Target Connections panel at the bottom left shows the hardware server, Linux TCF Agent, and QEMU TcfGdbClient. The bottom status bar shows the XSCF Process.

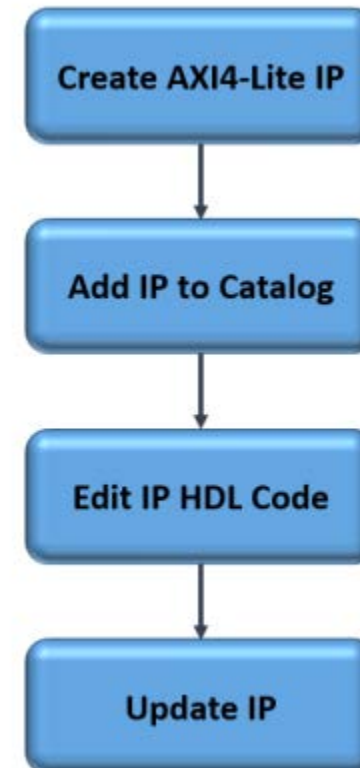
# Webcam Development using Zynq SoC

## ITCP Labs

### Labs at ITCP (Zedboard)

1. Hellow World
2. GPIO\_IP
3. Interrupt SW/HW
4. DMA
5. Custom IP
6. Comblock

Lab5. Custom IP  
Steps to Follow:



# Webcam Development using Zynq SoC

## ITCP Labs: Custom IP

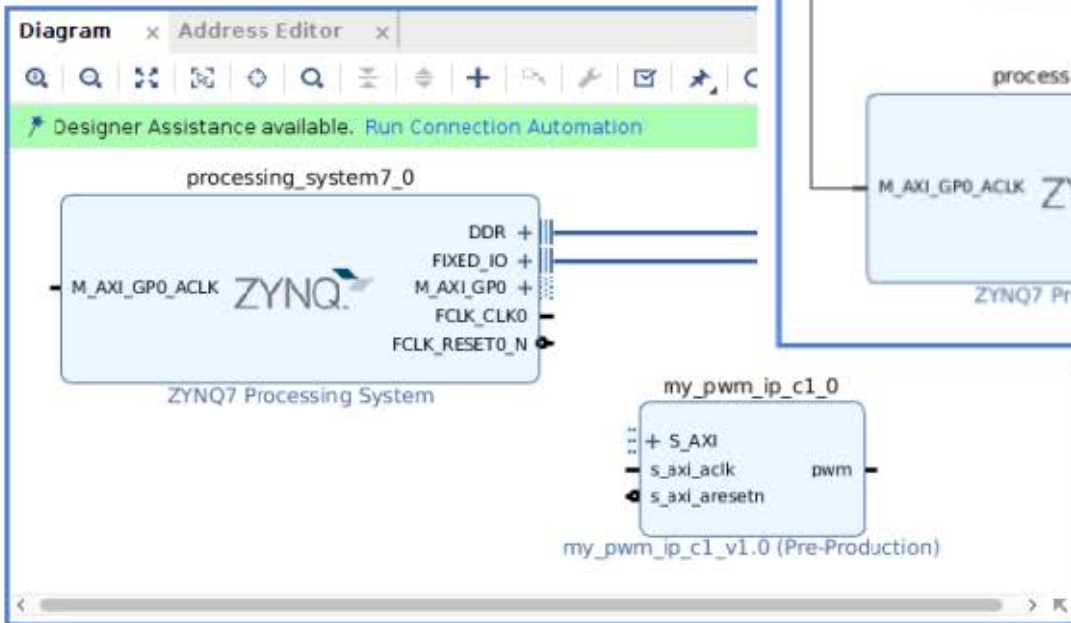
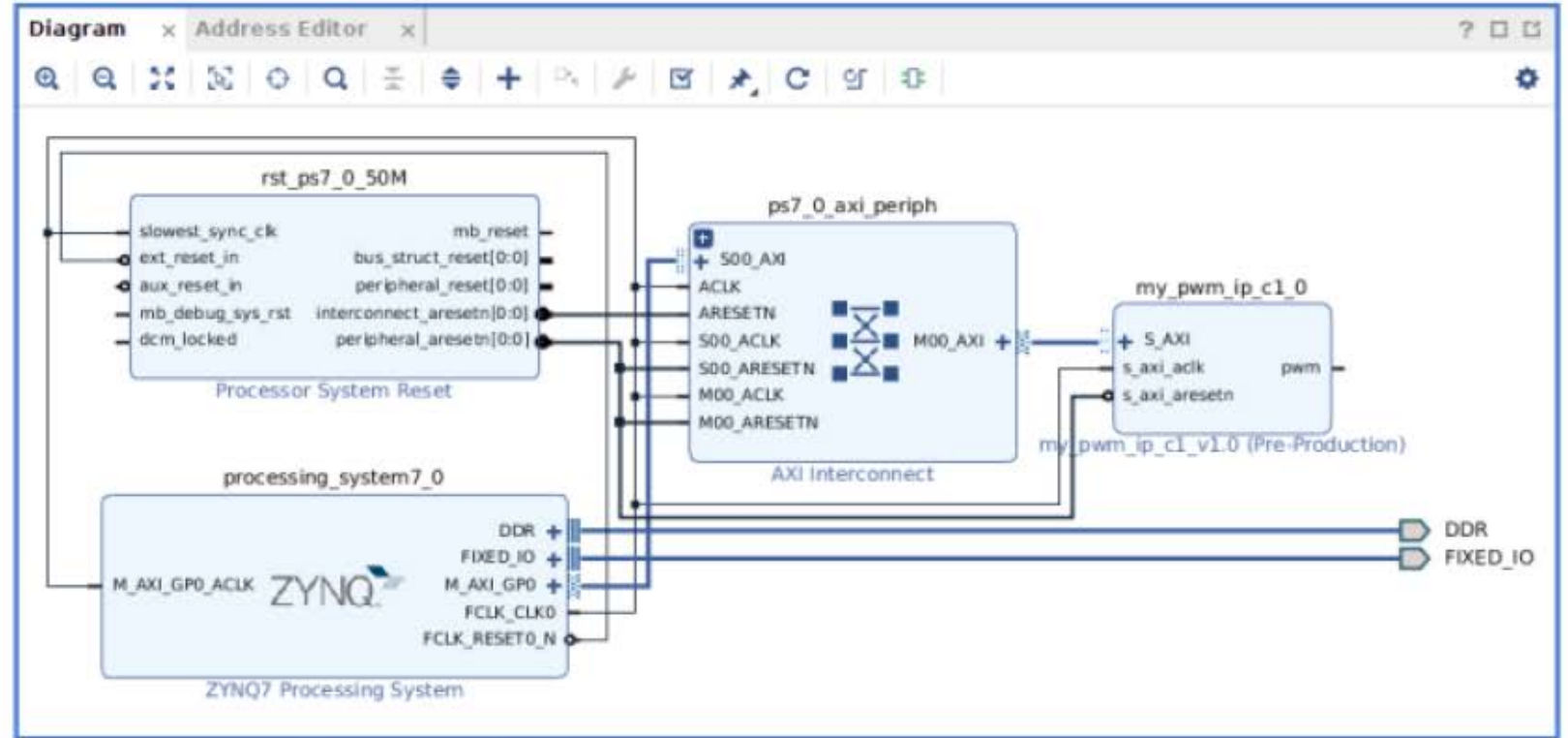
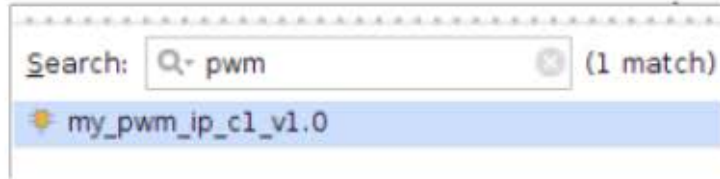


Diagram x Address Editor x lab\_custom\_ip.xdc x

Cell	Slave Interface	Base Name	Offset Address	Range	High Address
processing_system7_0					
Data (32 address bits : 0x40000000 [ 1G ])					
my_pwm_ip_c1_0	S_AXI	S_AXI_reg	0x43C0_0000	64K	0x43C0_FFFF



# Webcam Development using Zynq SoC

## ITCP Labs: Custom IP

```
lab_custom_ip.sdk - C/C++ - lab_custom_ip_wrapper_hw_platform_0/drivers/my_pwm_ip_c1_v1_0/src/my_

Project Explorer
lab_custom_ip_wrapper_hw_plat
  drivers
    my_pwm_ip_c1_v1_0
      data
      src
        Makefile
        my_pwm_ip_c1_selftest.c
        my_pwm_ip_c1.c
        my_pwm_ip_c1.h
lab_custom_ip_wrapper.bit
ps7_init_gpl.c
ps7_init_gpl.h
ps7_init.c
ps7_init.h
ps7_init.html
ps7_init.tcl
system.hdf

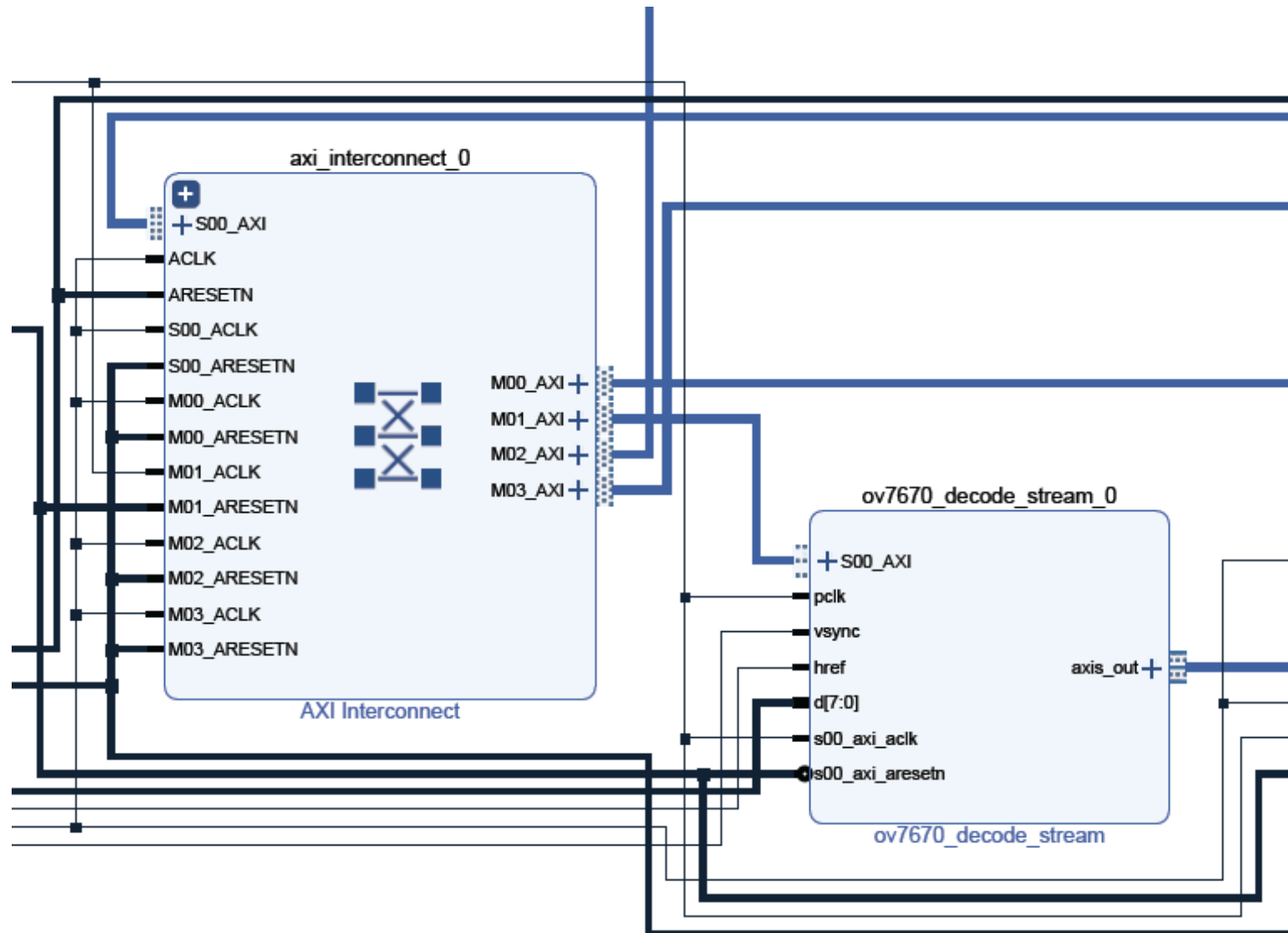
system.hdf  my_pwm_ip_c1.c  my_pwm_ip_c1.h
#include "xstatus.h"

#define MY_PWM_IP_C1_S_AXI_SLV_REG0_OFFSET 0
#define MY_PWM_IP_C1_S_AXI_SLV_REG1_OFFSET 4
#define MY_PWM_IP_C1_S_AXI_SLV_REG2_OFFSET 8
#define MY_PWM_IP_C1_S_AXI_SLV_REG3_OFFSET 12

/***** Type Definitions *****/
/**
 *
 * Write a value to a MY_PWM_IP_C1 register. A 32 bit write is
 * If the component is implemented in a smaller width, only the
 * significant data is written.
 *
 * @param BaseAddress is the base address of the MY_PWM_IP_C1
 * @param RegOffset is the register offset from the base to w
 * @param Data is the data written to the register.
 *
 * @return None.
 *
 * @note
 * C-style signature:
 * void MY_PWM_IP_C1_mWriteReg(u32 BaseAddress, unsigned RegOf
 *
 */
#define MY_PWM_IP_C1_mWriteReg(BaseAddress, RegOffset, Data) \
    Xil_Out32((BaseAddress) + (RegOffset), (u32)(Data))
```

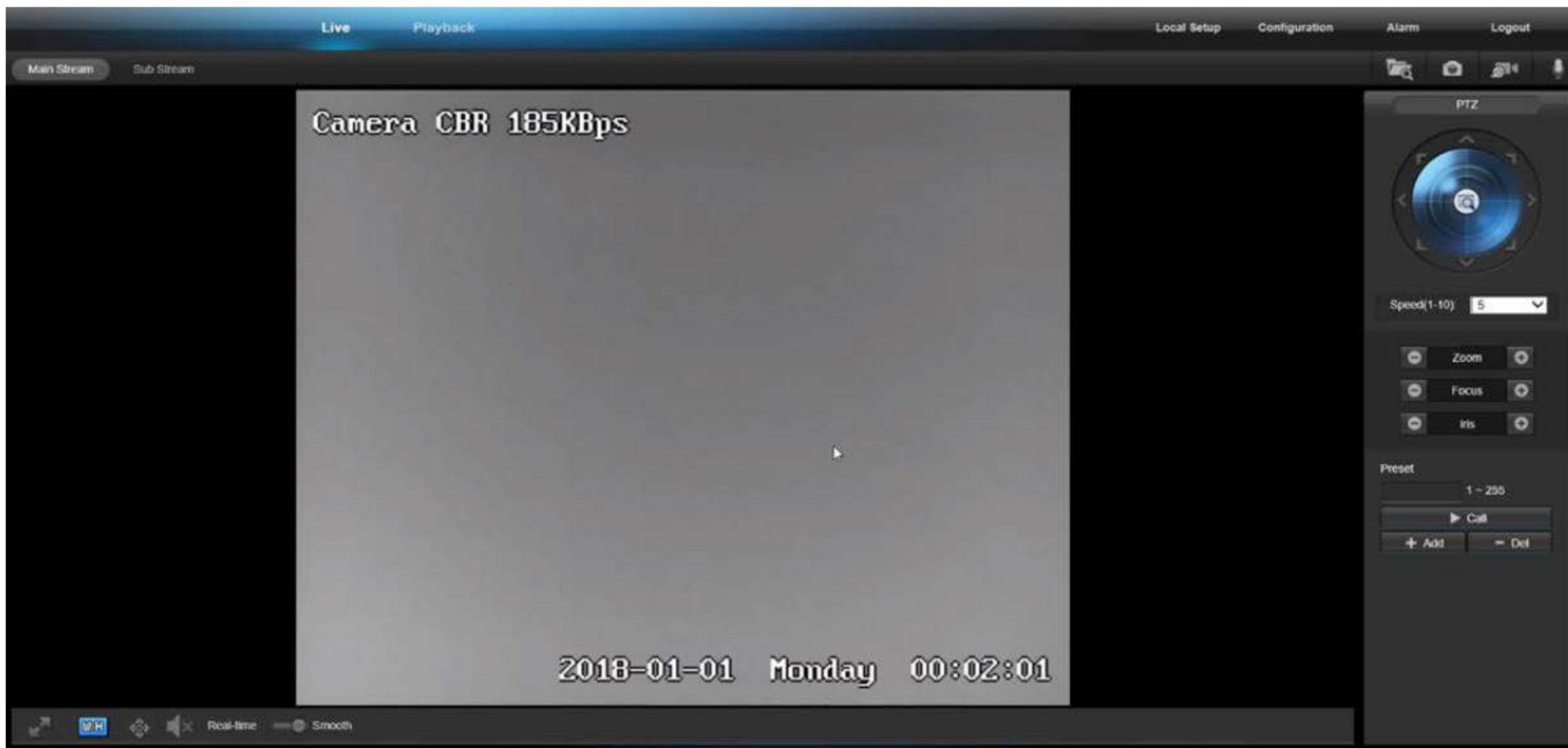
# Webcam Development using Zynq SoC

## ITCP Labs: Custom IP



# Webcam Development using Zynq SoC

## IP Camera Interface

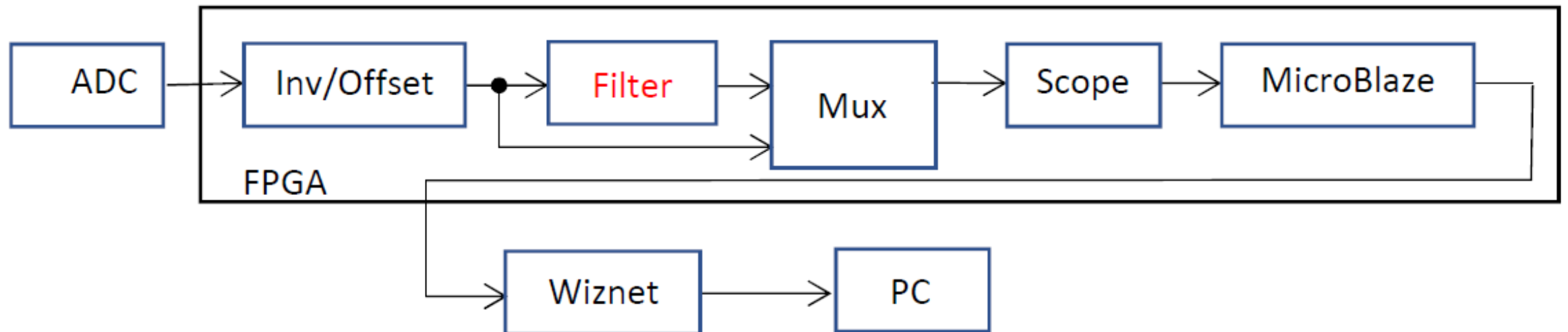


# Webcam Development using Zynq SoC

## Labs IAEA

### Main Labs at IAEA (CMOD-A7 and WIZned boards)

1. ADC data inversion and offset
2. Shaper
3. DC Stabilizer and base line restorer (BLR)
4. Pulse Height Analysis
5. Embedded Programming of WIZnet board



# Webcam Development using Zynq SoC

## Labs IAEA: DPP (Hierarchical Design)

DPP - [E:/ITCP/labs/IAEA\_LABS/smr3562/iaea/xilinx/projects/DPP/DPP.xpr] - Vivado 2019.1

File Edit Flow Tools Reports Window Layout View Help Q- Quick Access write\_bitstream Out-of-date details Default Layout

Flow Navigator

- PROJECT MANAGER
  - Settings
  - Add Sources
  - Language Templates
  - IP Catalog
- IP INTEGRATOR
  - Create Block Design
  - Open Block Design
  - Generate Block Design
- SIMULATION
  - Run Simulation
- RTL ANALYSIS
  - Open Elaborated Design
- SYNTHESIS
  - Run Synthesis
  - Open Synthesized Design
- IMPLEMENTATION
  - Run Implementation
  - Open Implemented Design
- PROGRAM AND DEBUG
  - Generate Bitstream
  - Open Hardware Manager

BLOCK DESIGN - design\_1

Diagram

The diagram shows a hierarchical design with the following components and connections:

- ps\_mb\_0** (Processing System Memory Block):
  - Inputs: Clk, reset, dom\_locked, intr[0:0], S00\_ARESETN[0:0]
  - Outputs: M00\_AXI, M01\_AXI, M02\_AXI, M03\_AXI, M05\_AXI, M06\_AXI, M07\_AXI, M08\_AXI, M09\_AXI, M10\_AXI, M11\_AXI
- dpp\_0** (Digital Processing Platform):
  - Inputs: S00\_AXI, axibusdomain\_s\_axi, S00\_AXI1, axi\_clk\_domain\_s\_axi, S00\_AXI2, AXI\_DC\_STAB, S00\_AXI3, S\_AXI, S00\_AXI4, axibusdomain\_s\_axi1, adc\_data[13:0], clk, s00\_axi\_aclik, s00\_axi\_aresetn
  - Output: full[0:0]
- mb\_periph\_0** (Memory Block Peripheral):
  - Inputs: S\_AXI, s\_axi\_aclik, s\_axi\_aresetn
  - Output: usb\_uart
- clk\_wiz\_0** (Clocking Wizard):
  - Inputs: sys\_clock, reset, clk\_in1, locked
  - Outputs: clk\_cpu, clk\_dpp, clk\_adc

Sub-block Pro

Name: ps\_mb\_0  
Parent name: design\_1

Tcl Console

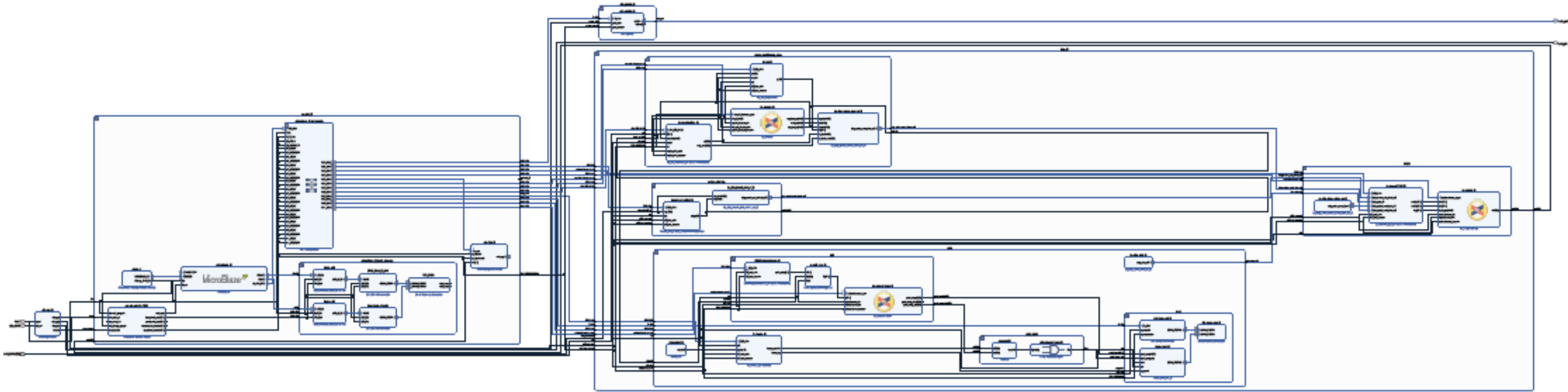
```
Adding component instance block -- xilinx.com:user:ip_timers:2.0 - ip_timers_0
Adding component instance block -- xilinx.com:ip:xlconcat:2.1 - xlconcat_0
```

Type a Tcl command here

System Net: clk\_wiz\_0\_clk\_adc

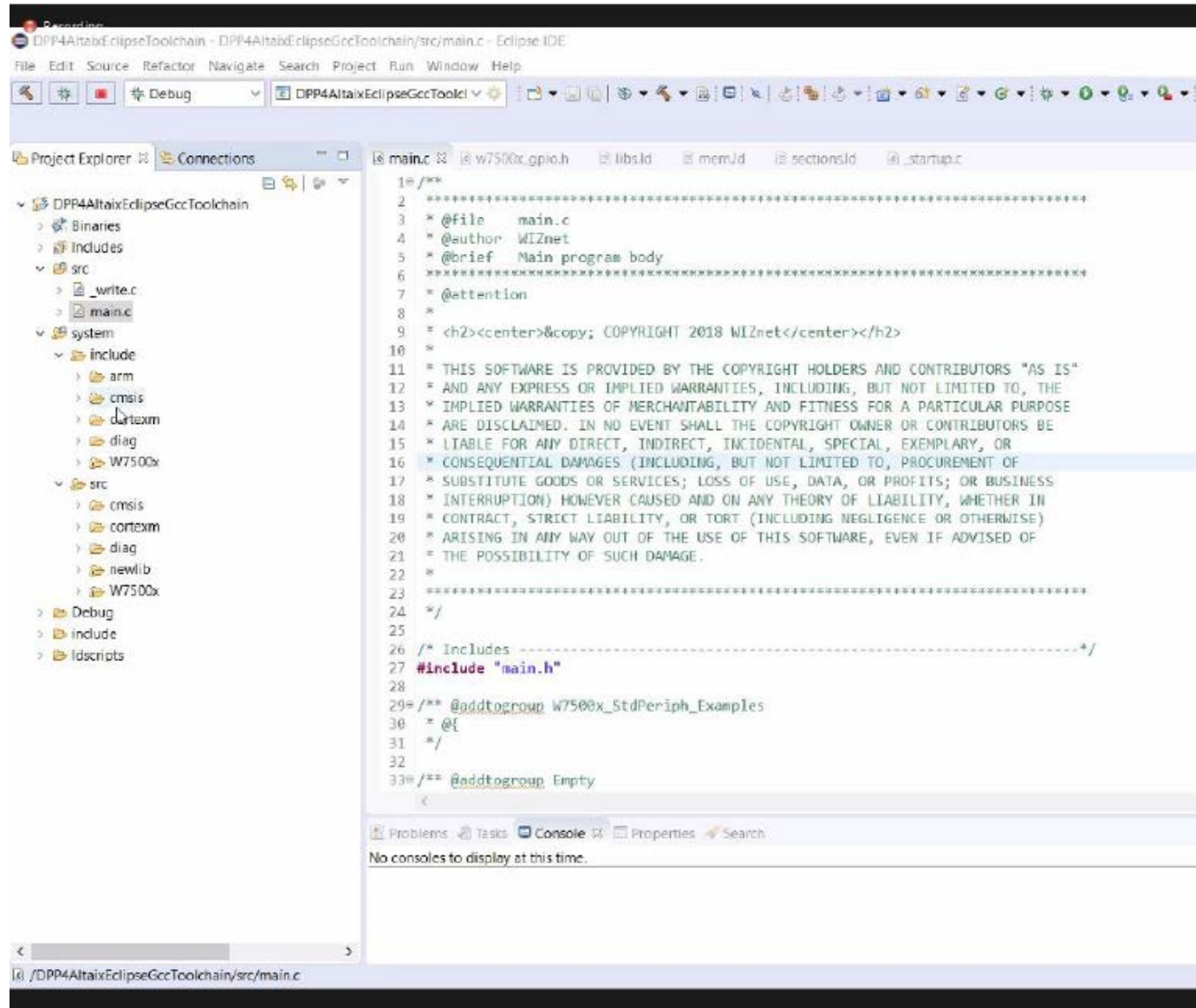
# Webcam Development using Zynq SoC

## Labs IAEA: DPP (Complex Desing)



# Webcam Development using Zynq SoC

## Labs IAEA: Embedded Development (Bare Metal)



The screenshot displays the Eclipse IDE interface for a project named "DPP4AltaixEclipseGccToolchain". The Project Explorer on the left shows a directory structure with folders for "Binaries", "Includes", "src", and "system". The "src" folder contains files like "\_write.c" and "main.c". The "system" folder contains sub-folders for "include" (with "arm", "cmsis", "cortexm", "diag", and "W7500x") and "src" (with "cmsis", "cortexm", "diag", "newlib", and "W7500x").

The main editor window shows the content of "main.c":

```
1  /**
2  *
3  * @file    main.c
4  * @author  WIZnet
5  * @brief  Main program body
6  *
7  * @attention
8  *
9  * <h2><center>&copy; COPYRIGHT 2018 WIZnet</center></h2>
10 *
11 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
12 * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
13 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
14 * ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
15 * LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
16 * CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
17 * SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
18 * INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
19 * CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
20 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
21 * THE POSSIBILITY OF SUCH DAMAGE.
22 *
23 *
24 */
25
26 /* Includes -----*/
27 #include "main.h"
28
29 /** @addtogroup W7500x_StdPeriph_Examples
30 * @{
31 */
32
33 /** @addtogroup Empty
```

# Webcam Development using Zynq SoC

## Embedded Linux SDK: PetaLinux

### Introduction

PetaLinux is an embedded Linux software development kit (SDK) targeting Xilinx® FPGA-based System-on-Chip designs. This guide helps the reader to familiarize with the tool enabling overall usage of PetaLinux.

You are assumed to have basic Linux knowledge, such as how to run Linux commands. You should be aware of OS and host system features, such as OS version, Linux distribution, security privileges, and [basic Yocto concepts](#).

The PetaLinux tool contains:

- Yocto Extensible SDK (eSDK)
- Minimal downloads
- XSCT and toolchains
- PetaLinux CLI tools

**Note:** Xilinx Software Development Kit (SDK) (XSDK) is the integrated design environment (IDE) for creating embedded applications on Xilinx microprocessors.

PetaLinux SDK is a Xilinx development tool that contains everything necessary to build, develop, test, and deploy embedded Linux systems.



# Webcam Development using Zynq SoC

## Embedded Linux SDK: PetaLinux

**Table 5: Design Flow Overview**

Design Flow Step	Tool / Workflow
Hardware Platform Creation (for custom hardware only)	Vivado® design tools
Create PetaLinux Project	<code>petalinux-create -t project</code>
Initialize PetaLinux Project (for custom hardware only)	<code>petalinux-config --get-hw-description</code>
Configure System-Level Options	<code>petalinux-config</code>
Create User Components	<code>petalinux-create -t COMPONENT</code>
Configure the Linux Kernel	<code>petalinux-config -c kernel</code>
Configure the Root Filesystem	<code>petalinux-config -c rootfs</code>
Build the System	<code>petalinux-build</code>
Package for Deploying the System	<code>petalinux-package</code>
Boot the System for Testing	<code>petalinux-boot</code>

# Webcam Development using Zynq SoC

## Embedded Linux SDK: PetaLinux

```
training@training-VirtualBox: ~/Documents/Projects/PetaLinux/web-project1
* newly built u-boot image which is <PROJECT>/images/linux/u-boot.elf
Generate bitstream merged with fsbl
$ petalinux-package --boot --fsbl <FSBL_ELF> --fpga <BITSTREAM> --format DOWNLOAD.BIT
It will generate a download.bit in <PROJECT>/images/linux, with specified <BITSTREAM> and <FSBL_ELF>.

ERROR: Output file "/home/training/Documents/Projects/PetaLinux/web-project1/images/linux/BOOT.BIN" already exists. Please use --force to overwrite it.
training@training-VirtualBox:~/Documents/Projects/PetaLinux/web-project1/images/linux$ petalinux-package --boot --fsbl zynq_fsbl.elf --fpga system.bit --u-boot u-boot.elf
f
INFO: File in BOOT BIN: "/home/training/Documents/Projects/PetaLinux/web-project1/images/linux/zynq_fsbl.elf"
INFO: File in BOOT BIN: "/home/training/Documents/Projects/PetaLinux/web-project1/images/linux/system.bit"
INFO: File in BOOT BIN: "/home/training/Documents/Projects/PetaLinux/web-project1/images/linux/u-boot.elf"
INFO: Generating Zynq binary package BOOT.BIN...

***** Xilinx Bootgen v2019.1
**** Build date : May 11 2019-11:15:10
** Copyright 1986-2019 Xilinx, Inc. All Rights Reserved.

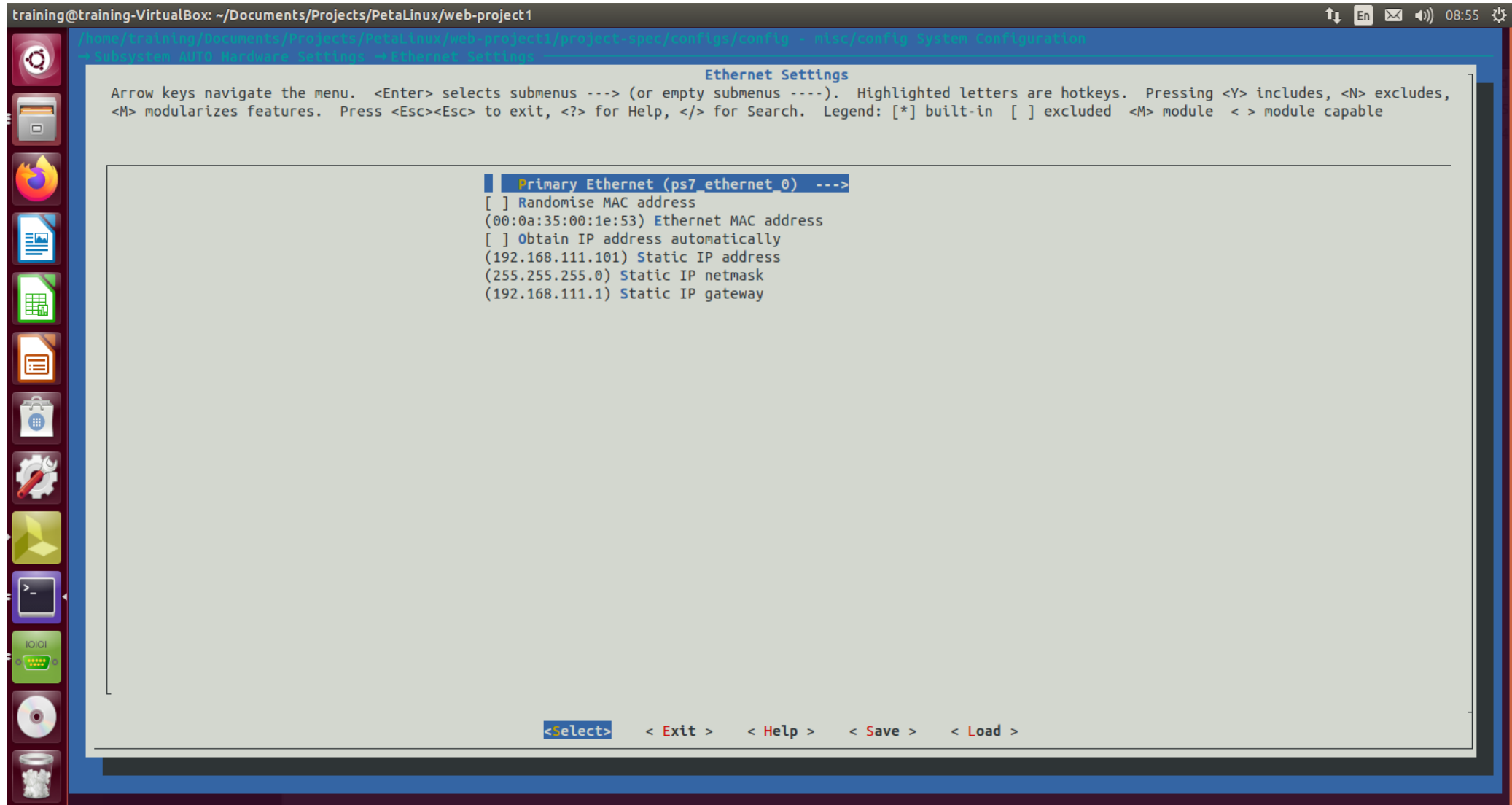
INFO: Binary is ready.
training@training-VirtualBox:~/Documents/Projects/PetaLinux/web-project1/images/linux$ cd ..
training@training-VirtualBox:~/Documents/Projects/PetaLinux/web-project1/images$ cd ..
training@training-VirtualBox:~/Documents/Projects/PetaLinux/web-project1$ petalinux-config
[INFO] generating Kconfig for project
[INFO] menuconfig project

*** End of the configuration.
*** Execute 'make' to start the build or try 'make help'.

[INFO] sourcing bitbake
[INFO] generating plnxtool conf
[INFO] generating meta-plnx-generated layer
[INFO] generating user layers
[INFO] generating machine configuration
[INFO] generating bbappends for project . This may take time !
[INFO] generating u-boot configuration files
[INFO] generating kernel configuration files
[INFO] generating kconfig for Rootfs
[INFO] silentconfig rootfs
[INFO] generating petalinux-user-image.bb
[INFO] successfully configured project
training@training-VirtualBox:~/Documents/Projects/PetaLinux/web-project1$
```

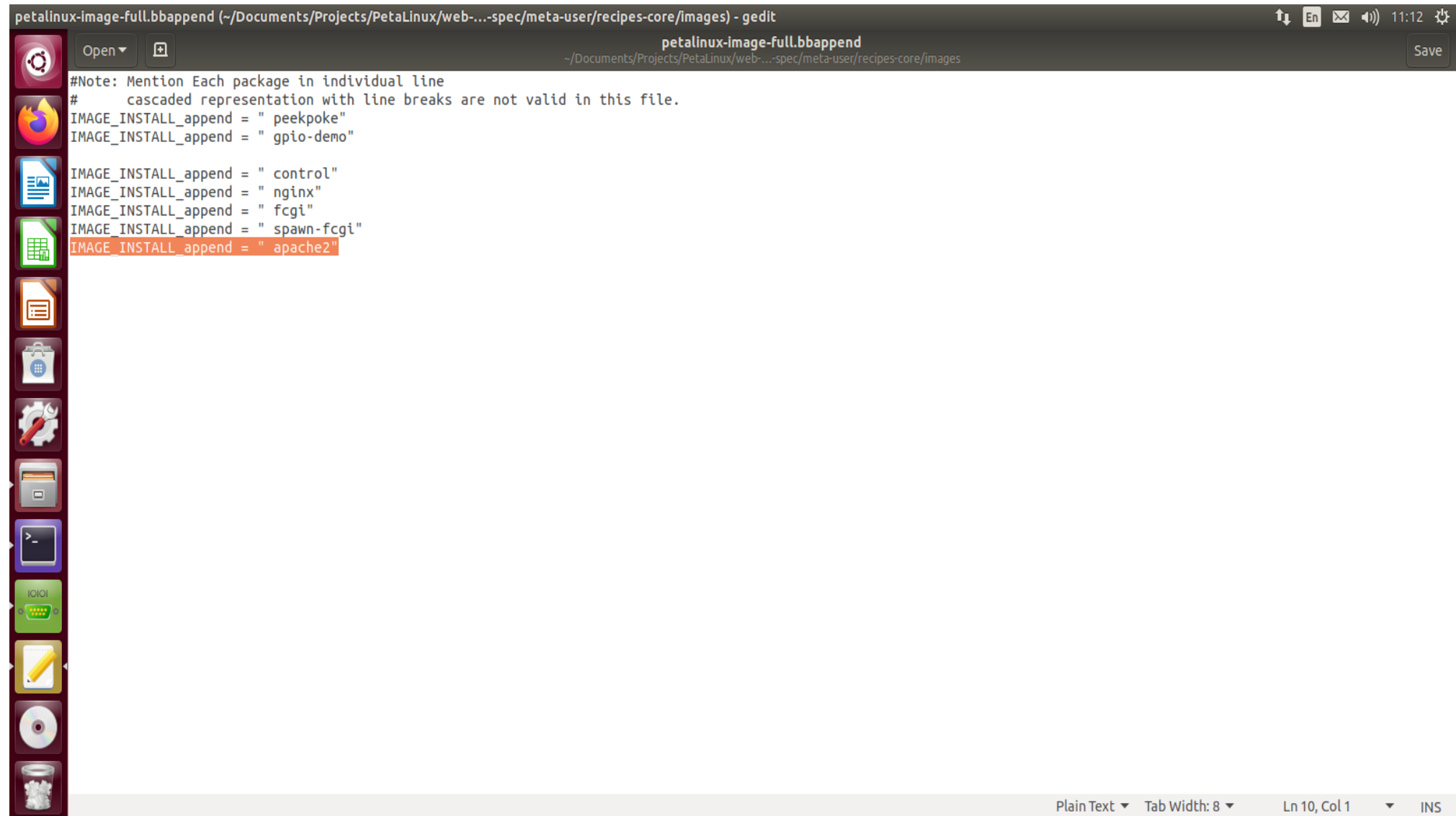
# Webcam Development using Zynq SoC

## Embedded Linux SDK: Ethernet IP



# Webcam Development using Zynq SoC

## Embedded Linux SDK: Apache Web Server



```
petalinux-image-full.bbappend (~/.Documents/Projects/PetaLinux/web-...-spec/meta-user/recipes-core/images) - gedit
petalinux-image-full.bbappend
~/Documents/Projects/PetaLinux/web-...-spec/meta-user/recipes-core/images
Save

#Note: Mention Each package in individual line
# cascaded representation with line breaks are not valid in this file.
IMAGE_INSTALL_append = " peekpoke"
IMAGE_INSTALL_append = " gpio-demo"

IMAGE_INSTALL_append = " control"
IMAGE_INSTALL_append = " nginx"
IMAGE_INSTALL_append = " fcgi"
IMAGE_INSTALL_append = " spawn-fcgi"
IMAGE_INSTALL_append = " apache2"
```

Plain Text ▾ Tab Width: 8 ▾ Ln 10, Col 1 ▾ INS

# Webcam Development using Zynq SoC

## Embedded Linux SDK: Web Server Config. httpd.conf

```
serial port Cypress UART
GtkTerm - /dev/ttyACM0 115200-8-N-1
root@web-project1:/usr/lib/cgi-bin# ls
printenv      printenv.vbs  printenv.wsf  test-cgi
root@web-project1:/usr/lib/cgi-bin# cd ..
root@web-project1:/usr/lib# cd apache2
root@web-project1:/usr/lib/apache2# ls
modules
root@web-project1:/usr/lib/apache2# cd modules
root@web-project1:/usr/lib/apache2/modules# ls
httpd.exp          mod_authz_dbd.so      mod_deflate.so        mod_log_config.so    mod_proxy_hcheck.so  mod_setenvif.so
mod_access_compat.so  mod_authz_dbm.so      mod_dir.so            mod_log_debug.so     mod_proxy_http.so    mod_slotmem_shm.so
root@web-project1:/# cd /etc/apache2
root@web-project1:/etc/apache2# ls
conf.d  extra  httpd.conf  magic  mime.types  modules.d
root@web-project1:/etc/apache2# cd /usr/lib/apache2
root@web-project1:/usr/lib/apache2# ls
modules
root@web-project1:/usr/lib/apache2# cd /usr/lib/cgi-bin
root@web-project1:/usr/lib/cgi-bin# ls
printenv      printenv.vbs  printenv.wsf  test-cgi
root@web-project1:/usr/lib/cgi-bin#
```

/dev/ttyACM0 115200-8-N-1

DTR RTS CTS CD DSR R



# Webcam Development using Zynq SoC

## HTML Web Page Programming

### HTML Web page

```
<html>
  <body>
    <h1>Hello there!</h1>
  </body>
</html>
```

### Equivalent CGI script of HTML Web page

- Perl

```
#!/usr/bin/perl
print "Content-type: text/html\n\n";
print "<html><body><h1>Hello there!";
print "</h1></body></html>\n";
```

- C

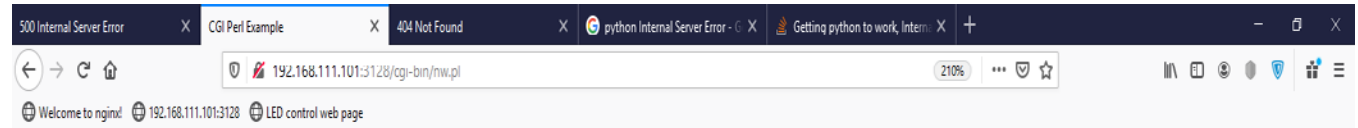
```
#include <stdio.h>
Int main(){
    printf("Content-type: text/html\n\n");
    printf("<html>\n");
    printf("<body>\n");
    printf("<h1>Hello there!</h1>\n");
    printf("</body>\n");
    printf("</html>\n");
    return 0;
}
```

# Webcam Development using Zynq SoC

## HTML Web Page Programming: Perl

### Perl

```
#!/usr/bin/perl
print "Content-type: text/html\n\n";
print <<htmlcode;
<html>
<head>
<title>CGI Perl Example</title>
</head>
<body>
<h1>CGI Perl Example</h1>
<p>CGI Perl Example</p>
</body>
htmlcode
```



## CGI Perl Example

CGI Perl Example



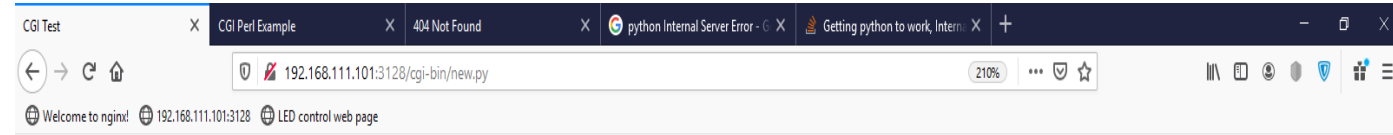


# Webcam Development using Zynq SoC

## HTML Web Page Programming: Python

### Python

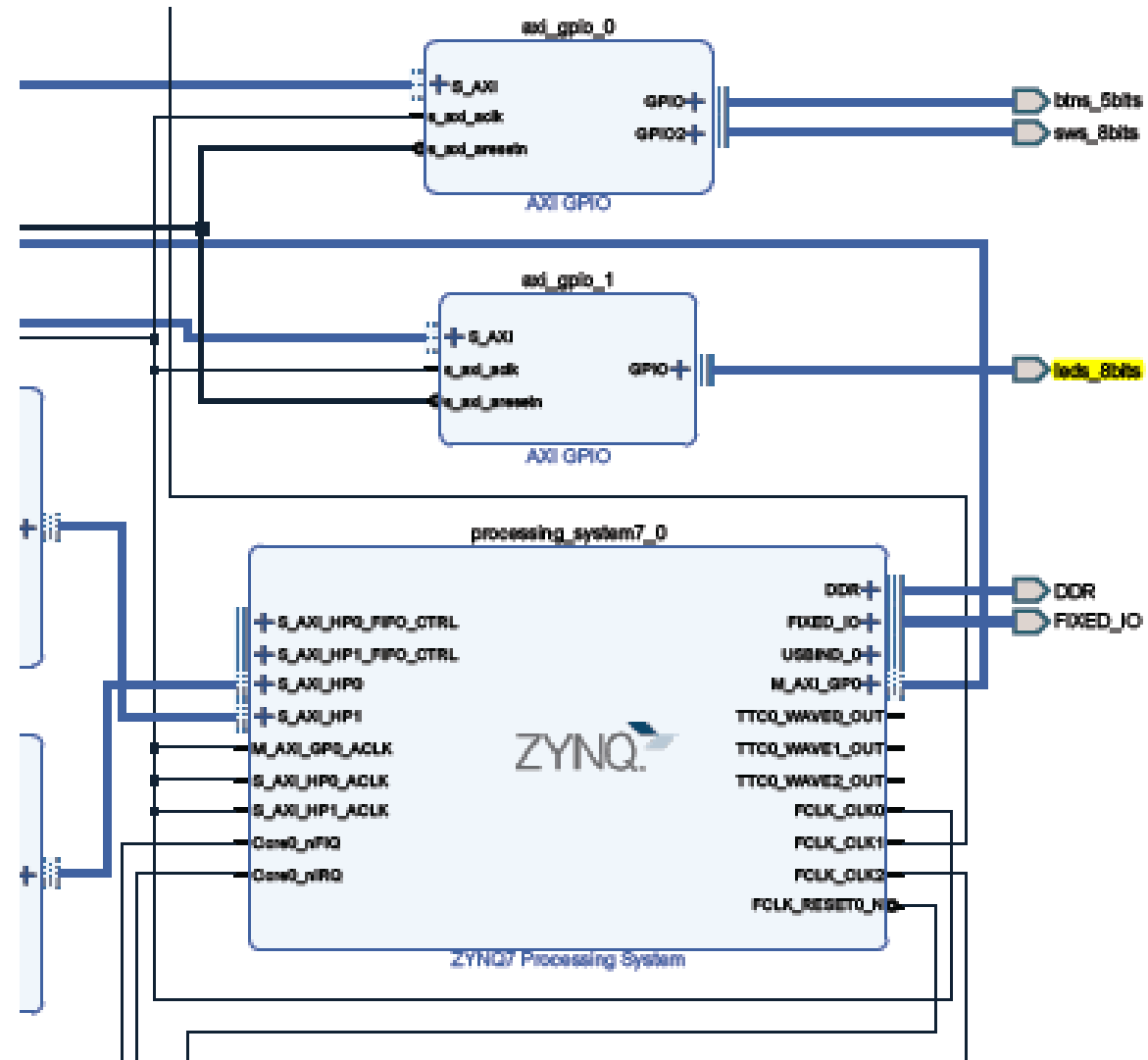
```
#!/usr/bin/python2.7
print "Content-type: text/html\n\n";
print "<html><head>";
print "<title>CGI Test</title>";
print "</head><body>";
print "<p>Test page using Python</p>";
print "</body></html>";
```



Test page using Python

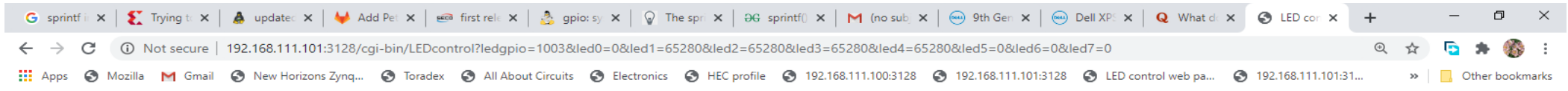
# Webcam Development using Zynq SoC

## Vivado Block Diagram: Zedboard LED



# Webcam Development using Zynq SoC









## Host PC: Zedboard LED Control



## ZedBoard LED Control

LED GPIO ID:

To change LED GPIO, change the value in the "LED GPIO ID" box and press "Clear" button.

led0	led1	led2	led3	led4	led5	led6	led7
							
<a href="#">ON/OFF</a>	<a href="#">ON/OFF</a>	<a href="#">ON/OFF</a>	<a href="#">ON/OFF</a>	<a href="#">ON/OFF</a>	<a href="#">ON/OFF</a>	<a href="#">ON/OFF</a>	<a href="#">ON/OFF</a>



# Webcam Development using Zynq SoC

## Embedded Linux Drivers: Ethernet and USB

Activities Hog\_System - [/home/training/Documents/Projects/Hog\_System\_Zynq/Hog\_System.xpr] - ... 10:21 19 فروری

Re-customize IP

ZYNQ7 Processing System (5,5)

Documentation Presets IP Location Import XPS Settings

Page Navigator

- Zynq Block Design
- PS-PL Configuration
- Peripheral I/O Pins
- MIO Configuration
- Clock Configuration
- DDR Configuration
- SMC Timing Calculation
- Interrupts

Zynq Block Design

Summary Report

64b AXI ACP Slave Ports

Processing System (PS)

Programmable Logic (PL)

OK Cancel

write\_bitstream Complete ✓

IP Integrator

ZYNQ

ma.sdk/vdma\_ov7670\_wrapper.hdf

# Webcam Development using Zynq SoC

## Embedded Linux Drivers: USB Webcam

```
COM5 - Tera Term VT
File Edit Setup Control Window Help
root@web-project1:/dev# usb 1-1: new high-speed USB device number 3 using ci_hdrc
usb 1-1: New USB device found, idVendor=1908, idProduct=2311, bcdDevice= 1.00
usb 1-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
usb 1-1: Product: USB2.0 PC CAMERA
usb 1-1: Manufacturer: Generic
usb 1-1: SerialNumber: 20100331010203
uvcvideo: Found UVC 1.00 device USB2.0 PC CAMERA (1908:2311)
uvcvideo 1-1:1.0: Entity type for entity Processing 2 was not initialized!
uvcvideo 1-1:1.0: Entity type for entity Camera 1 was not initialized!
input: USB2.0 PC CAMERA: USB2.0 PC CAM as /devices/soc0/amba/e0002000.usb/ci_hdrc.0/usb1/1-1/1-1:1.0/input/input1
cd /dev
root@web-project1:/dev# ls
block                loop-control        mtd1                 ram1                 snd                 tty2                tty35               tty50               tty9
bus                  loop0               mtd1ro              ram10               stderr              tty20              tty36              tty51              ttyPS0
char                 loop1               mtd2                 ram11               stdin               tty21              tty37              tty52              udev_network_queue
console             loop2               mtd2ro              ram12               stdout              tty22              tty38              tty53              v4l
cpu_dma_latency     loop3               mtd3                 ram13               tty                 tty23              tty39              tty54              v4l
disk                 loop4               mtd3ro              ram14               tty0                tty24              tty4                tty55              vcs
fd                   loop5               mtdblock0           ram15               tty1                tty25              tty40              tty56              vcs1
full                 loop6               mtdblock1           ram2                 tty10               tty26              tty41              tty57              vcsa
gpiochip0            loop7               mtdblock2           ram3                 tty11               tty27              tty42              tty58              vcsa1
gpiochip1            media0              mtdblock3           ram4                 tty12               tty28              tty43              tty59              vcsu
gpiochip2            mem                 network_latency     ram5                 tty13               tty29              tty44              tty6               vcsu1
gpiochip3            memory_bandwidth   network_throughput ram6                 tty14               tty3                tty45              tty60              vga_arbiter
iio:device0          mmchlk0             null                 ram7                 tty15               tty30              tty46              tty61              video0
initctl              mmchlk0p1          port                 ram8                 tty16               tty31              tty47              tty62              videol
input                mtab                ptmx                 ram9                 tty17               tty32              tty48              tty63              watchdog
kmsg                 mtd0                pts                  random                tty18               tty33              tty49              tty7               watchdog0
log                  mtd0ro              ram0                 shm                   tty19               tty34              tty5                tty8               zero
root@web-project1:/dev#
```



# Webcam Development using Zynq SoC

## Embedded Linux Drivers: AXI VDMA (Generic)

The screenshot shows a file manager window with the following details:

- Top bar: Activities, Files, 12:47 17 فروری, system icons.
- Breadcrumb: web-project > build > tmp > work-shared > plnx-zynq7 > kernel-source > drivers > dma > xilinx
- Left sidebar: Recent, Starred, Home, Desktop, Documents, Downloads, Music, Pictures, Videos, Trash, sf\_VirtualBox\_Share, VBox\_GAs\_6.1.10, Other Locations, SDK.
- Main pane: A table of files with columns Name, Size, Modified, and Star.

Name	Size	Modified	Star
axidmatest.c	17.1 kB	جنوری 15	☆
cdmatest.c	17.2 kB	جنوری 15	☆
Kconfig	1.8 kB	جنوری 15	☆
Makefile	515 bytes	جنوری 15	☆
vdmatest.c	17.7 kB	جنوری 15	☆
xilinx_dma.c	77.2 kB	جنوری 15	☆
xilinx_dpdma.c	65.3 kB	جنوری 15	☆
xilinx_frmdbuf.c	44.0 kB	جنوری 15	☆
xilinx_ps_pcie.h	1.2 kB	جنوری 15	☆
xilinx_ps_pcie_dma_client.c	35.9 kB	جنوری 15	☆
xilinx_ps_pcie_main.c	5.4 kB	جنوری 15	☆
xilinx_ps_pcie_platform.c	94.6 kB	جنوری 15	☆
zynqmp_dma.c	34.4 kB	جنوری 15	☆

2 items selected (95.0 kB)

# Webcam Development using Zynq SoC

## AXI VDMA Driver: Self test

The screenshot displays the Vivado 2019.1 interface for a project named 'ov7670\_vdma'. The main window shows a Block Design diagram for 'vdma\_ov7670'. The diagram includes the following components and connections:

- vdma\_s2mm** (AXI Video Direct Memory Access): A central block highlighted with an orange border. It has ports for S\_AXI\_LITE, S\_AXIS\_S2MM, M\_AXI\_MM2S, M\_AXIS\_S2MM, M\_AXIS\_MM2S, mm2s\_frame\_ptr\_out[5:0], s2mm\_frame\_ptr\_out[5:0], mm2s\_introut, and s2mm\_introut.
- vdma\_mm2s\_intercon** (AXI Interconnect): Connected to the S00\_AXI and M00\_AXI ports of the vdma\_s2mm block.
- vdma\_s2mm\_intercon** (AXI Interconnect): Connected to the S00\_AXI and M00\_AXI ports of the vdma\_s2mm block.
- processing\_system7\_0** (ZYNQ7 Processing System): The main SoC component, connected to the interconnects. It has various ports including S\_AXI\_HP0\_FIFO\_CTRL, S\_AXI\_HP0, S\_AXI\_HP1, M\_AXI\_GPO\_ACLK, S\_AXI\_HP0\_ACLK, S\_AXI\_HP1\_ACLK, and IRQ\_F2P[1:0].
- xlconcat\_0** (In0[0:0]): A component connected to the M00\_AXI port of the vdma\_s2mm block.

The Tcl Console at the bottom shows the output of the `regenerate_bd_layout` command:

```
regenerate_bd_layout: Time (s): cpu = 00:00:00.11 ; elapsed = 00:00:05 . Memory (MB): peak = 6732.023 ; gain = 0.000 ; free physical = 781 ; free virtual = 7229
```



# Webcam Development using Zynq SoC

## Linux Kernel module: vdmatest.c

```
GtkTerm - /dev/ttyACM0 115200-8-N-1
Pv6: ADDRCONF(NETDEV_UP): eth0: link is not ready
done.
Starting Dropbear SSH server: random: dropbearkey: uninitialized uranroot
Password:
root@ov7670_dma:~# modprobe vdmatest
vdmatest: loading out-of-tree module taints kernel.
xilinx_vdmatest: Started 1 threads using dma0chan0 dma0chan1
dma0chan0-dma0c: terminating after 1 tests, 0 failures (status 0)
root@ov7670_dma:~#
```

SDK

/dev/ttyACM0 115200-8-N-1

DTR RTS CTS CD DSR RI

# Thanks

- **Organizers**
- **Faculty Members**
- **Lab Tutors**

Joint ICTP-IAEA School on FPGA-based SoC and its Applications for Nuclear and Related Instrumentation | (smr 3562) 25 Jan – 19 Feb, 2021

**In the name of Allah, the Gracious, the Merciful.**

Read [O Prophet] in the name of thy Lord Who created [96:1], Created man from a clot of blood [96:2]. Read, and your Lord is Most Honorable [96:3], Who taught with the pen [96:4], Taught man what he knew not [96:5].

**Holy Quran [96:1-5]**

# Webcam Development using Zynq SoC

## Embedded Linux: Web Page Programming

```
GtkTerm - /dev/ttyACM0 115200-8-N-1
hd          python2-config  xzcat
head        python2.7         yes
root@web-project1:/usr/bin# cd /usr/lib/cgi-bin
root@web-project1:/usr/lib/cgi-bin# ls
printenv  printenv.vbs  printenv.wsf  test-cgi
root@web-project1:/usr/lib/cgi-bin# chmod 777 cgi-bin
chmod: cgi-bin: No such file or directory
root@web-project1:/usr/lib/cgi-bin# cd test-cgi
-sh: cd: test-cgi: Not a directory
root@web-project1:/usr/lib/cgi-bin# vi new.py
root@web-project1:/usr/lib/cgi-bin# chmod new.py
BusyBox v1.29.2 (2020-09-19 04:25:00 UTC) multi-call binary.

Usage: chmod [-R] MODE[,MODE]... FILE...

Each MODE is one or more of the letters ugoa, one of the
symbols += and one or more of the letters rwxst

-R      Recurse
root@web-project1:/usr/lib/cgi-bin# vi new.py
root@web-project1:/usr/lib/cgi-bin# chmod 777 new.py
root@web-project1:/usr/lib/cgi-bin# httpd -k restart
root@web-project1:/usr/lib/cgi-bin# apache2 -restart
-sh: apache2: command not found
root@web-project1:/usr/lib/cgi-bin# vi nw.pl
root@web-project1:/usr/lib/cgi-bin# chmod 777 nw.pl
root@web-project1:/usr/lib/cgi-bin# httpd -k restart
root@web-project1:/usr/lib/cgi-bin# ls
new.py  printenv  printenv.wsf
nw.pl  printenv.vbs  test-cgi
root@web-project1:/usr/lib/cgi-bin# vi new.py
root@web-project1:/usr/lib/cgi-bin# chmod 777 new.py
root@web-project1:/usr/lib/cgi-bin# httpd -k restart
root@web-project1:/usr/lib/cgi-bin# ls
new.py  printenv  printenv.wsf
nw.pl  printenv.vbs  test-cgi
root@web-project1:/usr/lib/cgi-bin# vi new.py
root@web-project1:/usr/lib/cgi-bin#
```

## Perl

```
#!/usr/bin/perl
print "Content-type: text/html\n\n";
print <<htmlcode;
<html>
<head>
<title>CGI Perl Example</title>
</head>
<body>
<h1>CGI Perl Example</h1>
<p>CGI Perl Example</p>
</body>
htmlcode
```

# Webcam Development using Zynq SoC

## HTML Web Page Programming: C (LED control)

```
LEDcontrol.c (~/Desktop/Working code LEDcontrol 210620) - gedit
LEDcontrol.c
~/Desktop/Working code LEDcontrol 210620
Save

/* LEDcontrol.c */
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <string.h>

/*#include "cgivars.h"*/

/* function prototypes */
int getRequestMethod();
char **getGETvars();
char **getPOSTvars();
int cleanUp(int form_method, char **getvars, char **postvars);
/* local function prototypes */
char hex2char(char *hex);
void unescape_url(char *url);
char x2c(char *what);
#define GET 0
#define POST 1

/*#include "htmllib.h"*/

/* function prototypes */
void htmlHeader(char *title);
void htmlBody();
void htmlFooter();
void addTitleElement(char *title);

/*#include "led_cgi.h"*/
//int nchannel = 0;
int set_leds();
int led_cgi_page (char **getvars, int form_method);
int open_gpio_channel (int gpio_base);
int close_gpio_channel (int gpio_base);
int set_gpio_direction (int gpio_base, int nchannel, char *direction);
int set_gpio_value (int gpio_base, int nchannel, int value);

#include <fcntl.h>
```

# Webcam Development using Zynq SoC

## Vivado VDMA Self test setting

The screenshot displays the Vivado 2019.1 interface for configuring the AXI Video Direct Memory Access (6.3) component. The component name is set to `vdma_s2mm`. The configuration is shown in the 'Advanced' tab, with the following settings:

- Enable Asynchronous Mode (Auto)
- Enable Single AXI4 Data Interface
- Enable Vertical Flip

**Write Channel Options:**

- Fsync Options: `None`
- GenLock Mode: `Dynamic-Slave`
- Allow Unaligned Transfers

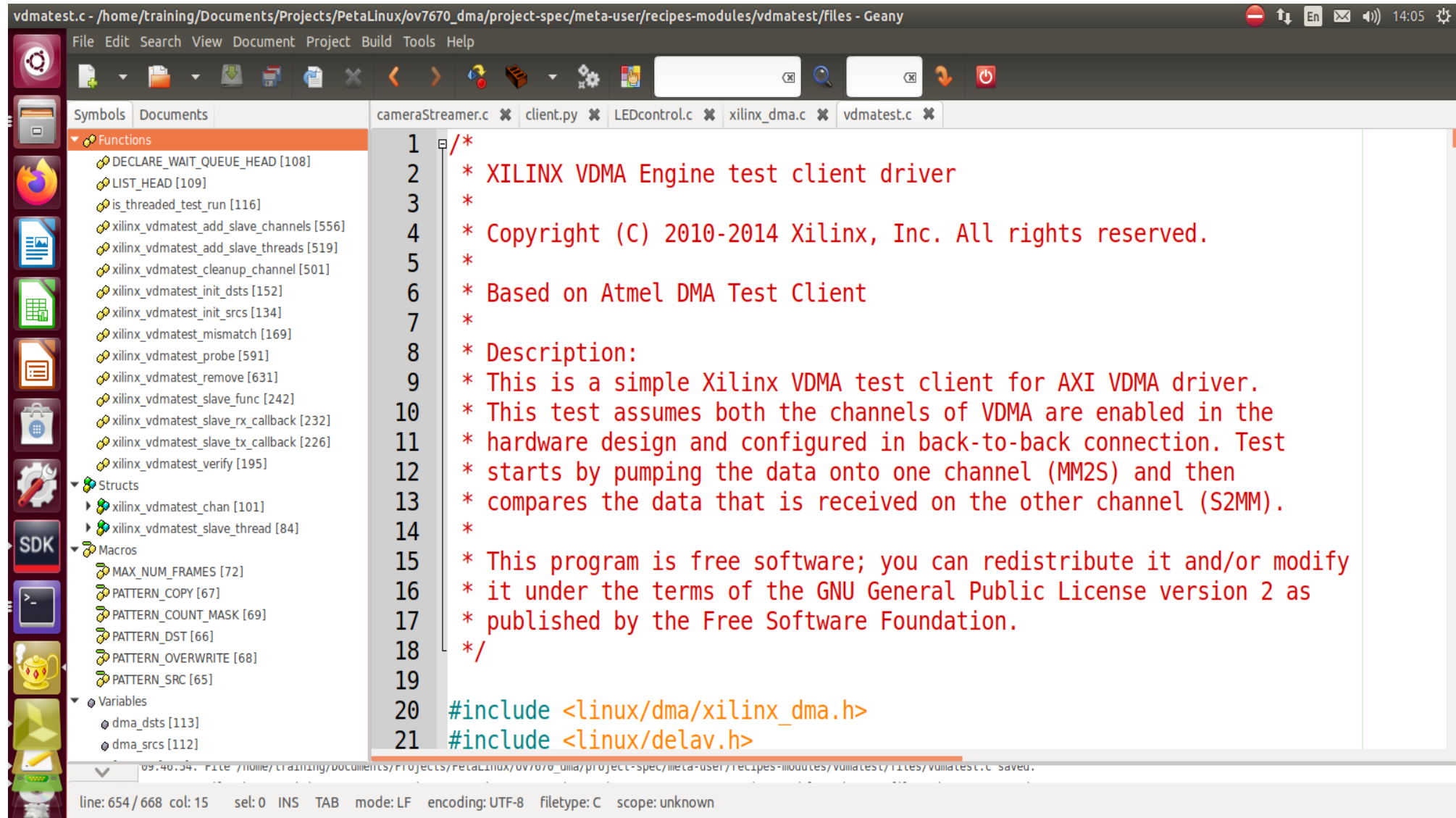
**Read Channel Options:**

- Fsync Options: `None`
- GenLock Mode: `Dynamic-Master`
- Allow Unaligned Transfers

The left sidebar shows the project structure, including the IP Integrator and Simulation sections. The IP Integrator section is expanded, showing the configuration of the AXI Video Direct Memory Access (6.3) component. The right sidebar shows the IP Integrator window, indicating that the bitstream has been written successfully.

# Webcam Development using Zynq SoC

## VDMAtest.c



```
1 /*
2  * XILINX VDMA Engine test client driver
3  *
4  * Copyright (C) 2010-2014 Xilinx, Inc. All rights reserved.
5  *
6  * Based on Atmel DMA Test Client
7  *
8  * Description:
9  * This is a simple Xilinx VDMA test client for AXI VDMA driver.
10 * This test assumes both the channels of VDMA are enabled in the
11 * hardware design and configured in back-to-back connection. Test
12 * starts by pumping the data onto one channel (MM2S) and then
13 * compares the data that is received on the other channel (S2MM).
14 *
15 * This program is free software; you can redistribute it and/or modify
16 * it under the terms of the GNU General Public License version 2 as
17 * published by the Free Software Foundation.
18 */
19
20 #include <linux/dma/xilinx_dma.h>
21 #include <linux/delay.h>
```

line: 654 / 668 col: 15 sel: 0 INS TAB mode: LF encoding: UTF-8 filetype: C scope: unknown