# Lab1: Design and Test of VHDL IP

## HDMI Display Controller + Camera Emulator

3 points

Once a VHDL model for a system is made (often called IP), the next step is to test it. In this lab we want to test a HDMI display controller, designed to display images captured by a camera. Data of the HDMI display controller is finally send to the on-board HDMI transmitter chip. The camera is getting emulated by another VHDL model. The following figure shows the top level RTL design:

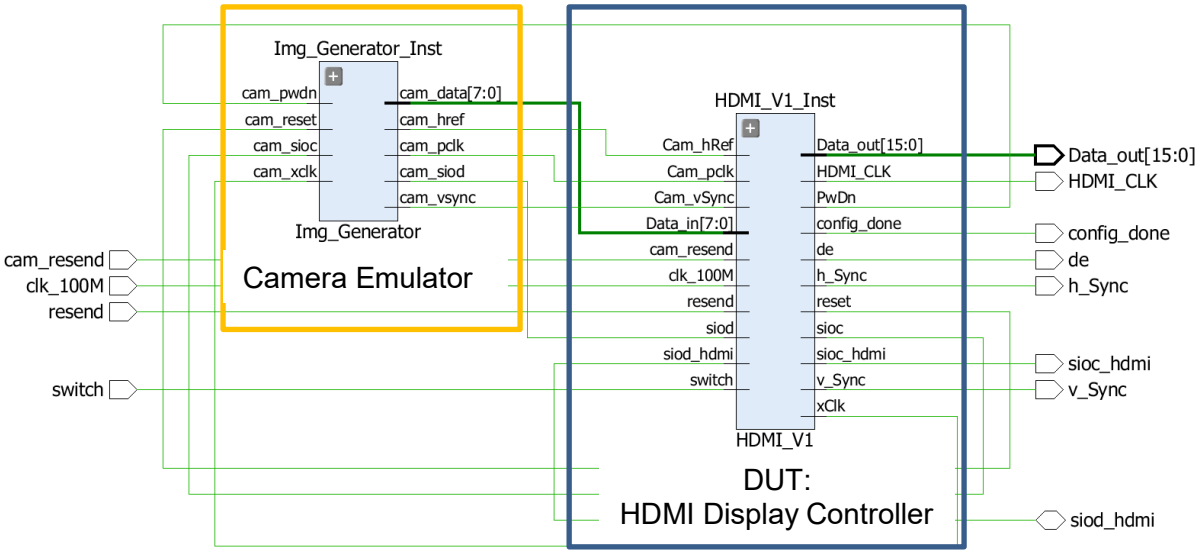



Figure 1: Block Diagram of top level RTL design

As a hardware designer and test expert you need to perform the followings parts in this lab:

1) Study of the top level RTL design

2) VHDL design of the Camera Emulator

3) Simulation of the Camera Emulator

4) Test of HDMI Display Controller on ZedBoard

**Software** required for this lab:
Xilinx Vivado Design Suite 2018.2 WebPACK
ModelSim – Intel FPGA Starter Edition

**Hardware** required for this lab:
ZedBoard, Micro-USB cable for JTAG interface, ASUS screen

**Data-files**:
*Camera_Emulation_Project_v2_lab1.zip*

**Manuals**:
*OV7670CameraChip*
*ZedBoard_HW_UG_v2_2*

Part1: **Study top level RTL design**

Open the project **Camera_Emulation_Project_v2_lab1** in Vivado 2018.2 and study the hierarchical VHDL design of *Camera_Emulator.vhd*! (RTL Analysis -> Open Elaborated Design).

The component **HDMI_V1** is the HDMI display controller, the component **IMG_GENERATOR** emulates the camera.

Task1: Describe the meaning of all output signals of HDMI_V1!! Consider that those output signals are inputs to the HDMI transmitter ADV7511. Do research on ADV7511 of ZedBoard. Use a table format with two columns: signal name and description!

Task2: The component IMG_GENERATOR emulates the camera. Do research on OV7670 camera (data sheet). Describe each signal provided by the camera (assume a resolution setting of 640x480 and an output format of Y/Cb/Cr 4:2:2) and compare with the output ports of IMG_GENERATOR! Use a table format with two columns: signal name and description!

Part2: **Design of camera emulator**

Instead of using a camera to test the HDMI controller you need to create input data normally provided by the camera. For emulation of the camera a component *Img_Generator.vhd* is already designed, but you as a test expert have to provide the pixels you want to display (*Pixel_Generator.vdh*) as well as the camera clock (*Camera_Clock_Generator.vhd*).

When investigating the component Img_Generator you will find two subcomponents:

**Camera_Clock_Generator**
**Pixel_Generator**

Task1: In *Camera_Clock_Generator.vhd* we need to generate both, VSYNC and HREF in order to emulate the camera.
VSYNC is responsible for synchronizing an entire image frame on the screen, and HREF is responsible for synchronizing each line of the image frame. How the HREF signal is handled according to the clock signal is shown in Figure 2. How both the VSYNC signal and HREF signal are handled is shown in Figure 3.

Note: time per pixel is $t_p = 2 \times t_{PCLK}$
Every line takes $t_{LINE} = 784 \times t_p = 1568 \times t_{PCLK}$ , time for 640 pixels per row, as well as 144 buffer ticks between a row and the next.
Image has a fixed size of 640 x 480 pixels. 480 lines plus 30 buffer ticks results in total image creation time: $t_{image} = 510 \times t_{LINE} = 510 \times 784 \times t_p = 799680 \times t_{PCLK}$

**Figure 2. Horizontal timing:**

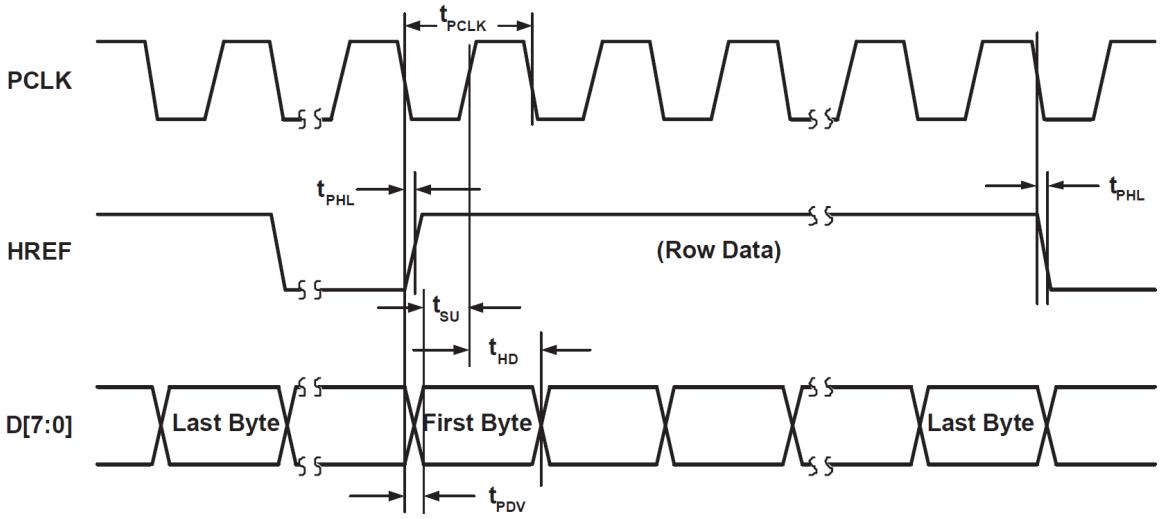
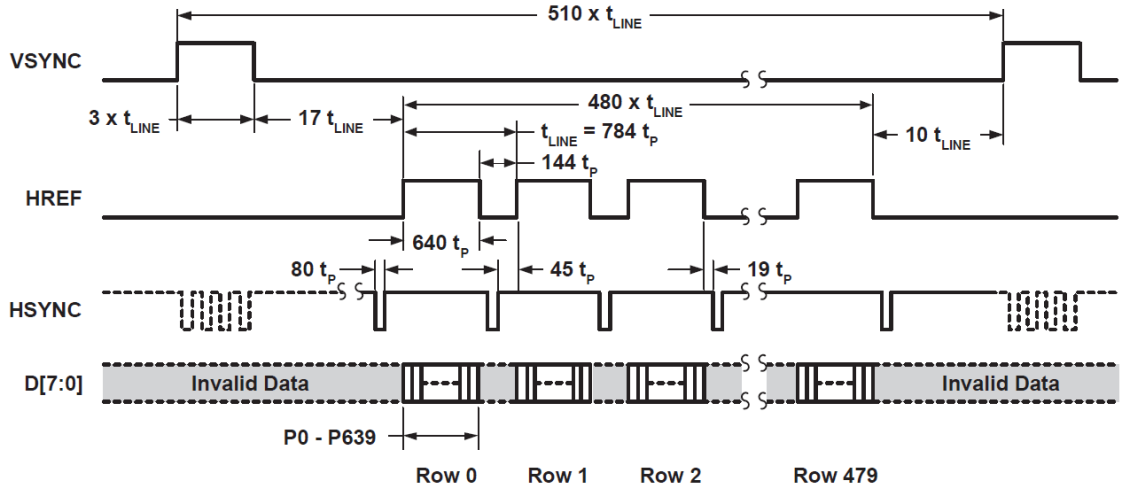



**Figure 3. VGA Frame Timing:**




**NOTE:**
For Raw data, $t_P = t_{PCLK}$
For YUV/RGB, $t_P = 2 \times t_{PCLK}$

Task2: In *Pixel_Generator.vhd* we need to specify Y,CB and CR. Use VHDL *constant* with data type STD_LOGIC_VECTOR (7 downto 0)! Generate **blue** **pixels**!

Format Y/Cb/Cr 4:2:2 means, that a pair of two pixels share the same chrominance Cb and Cr information. This is called chroma subsampling. The order in which the camera send the Y,Cb,Cr information is presented in the table below:

| ***N*** | *1st* | *2nd* | *3rd* | *4th* | *5th* | *6th* | *7th* | *8th* | … |
|---|---|---|---|---|---|---|---|---|---|
| ***Byte*** | CB0 | Y0 | CR0 | Y1 | CB2 | Y2 | CR2 | Y3 | … |

Where each individual pixel is format by:

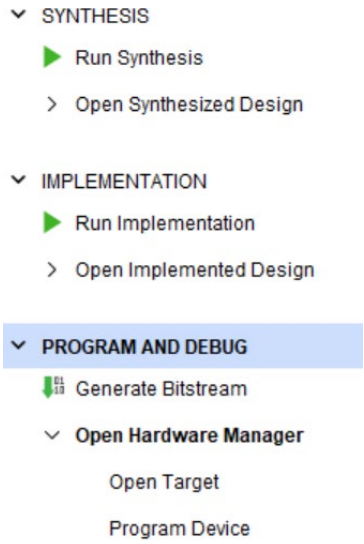
| ***PIXEL*** | ***Y Byte*** | ***CB Byte*** | ***CR Byte*** |
|---|---|---|---|
| *Pixel 0* | Y0 | CB0 | CR0 |
| *Pixel 1* | Y1 | CB0 | CR0 |
| *Pixel 2* | Y2 | CB2 | CR2 |
| *Pixel 3* | Y3 | CB2 | CR2 |
| *Pixel 4* | Y4 | CB4 | CR4 |

Part3: **Simulation of camera emulator**

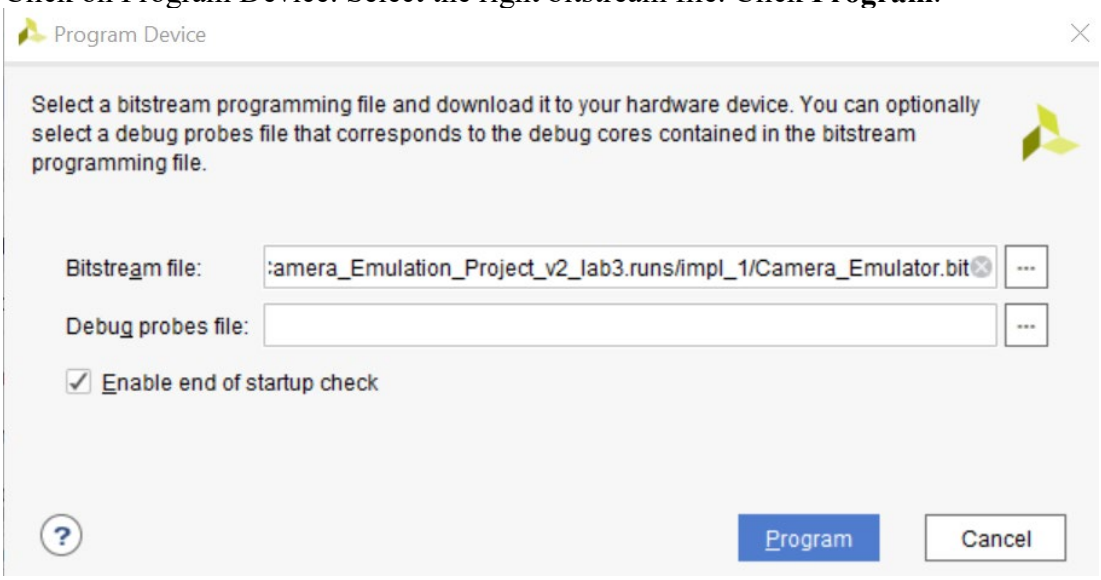
Task1: Run simulation of *Camera_Clock_Generator.vhd* using ModelSim (Copy code to a ModelSim project). No test bench required, because one input signal only. Compare your output with fig.2 and fig.3!
Task2: Run simulation of *Pixel_Generator.vhd* using ModelSim (Copy code to a ModelSim project). No test bench required, because three input signals only. Check 8-bit output data on expected CB,Y,CR,Y,CB,Y,CR sequence for blue pixel!

Part4: **Test of HDMI Display Controller on ZedBoard**

Task1: First check on file *video_project_constraints.xdc* if all the pins are set correctly. Provide a table for all I/O pins and their respective signal names!
Task2: Synthesize your VHDL design and implement it on the ZedBoard. For this you **Run Synthesis**, then **Run Implementation**, finally click on **Generate Bitstream**.




Now switch on power of your ZedBoard and connect to host computer. Connect Screen via HDMI cable to your ZedBoard.
In Open Hardware Manager click on **Open Target**: Auto Connect
Click on Program Device. Select the right bitstream file. Click **Program**.




Take photo of screen output!

Task3: Change *Pixel_Generator.vhd* in a way that you get color strips on your screen as shown in fig.4! Take photo of screen output

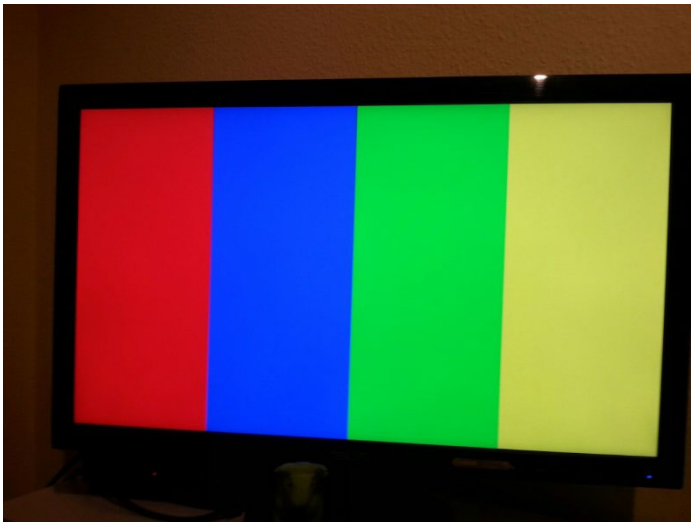


Figure 4: HDMI controller test pattern

**Prepare before lab (send pdf-file to lab tutor):**

1. Part1: Task 1-2
2. Part2: Task2: Complete *Pixel_Generator.vhd*! Prepare values for different colors in YCbCr format.

**Note:** Send pdf-file to tutor no later than **May/8, 8pm**! No preparation means you need to leave lab with no points!

Lab report: deadline **May/24, 8pm**! Submit your lab report to lab tutor (group of 2 students). Include:

1. task 1-2 of part1
2. VHDL-code *Camera_Clock_Generator.vhd* and *Pixel_Generator.vdh* for blue pixels. Comment your code!
3. Simulation results of part 3 (screen shots) with detailed evaluation of outputs.
4. I/O pins used in ZedBoard design (part4, task1) and photo of screen output.
5. VHDL-code *Pixel_Generator.vhd* for color strips and photo of screen output.