Lab3: HDMI Design on ZedBoard

4 points

In this lab you get a task very common in industry: a project developed by another division needs get tested. Unfortunately almost no documentation has been done, just a description on how to get the project run. Data captured by a camera is getting displayed on the HDMI output of the ZedBoard. The camera needs to get connected to the Pmods, the screen to the HDMI transmitter of the ZedBoard (see fig.1). Both interfaces are connected to the PL side of Zynq device.

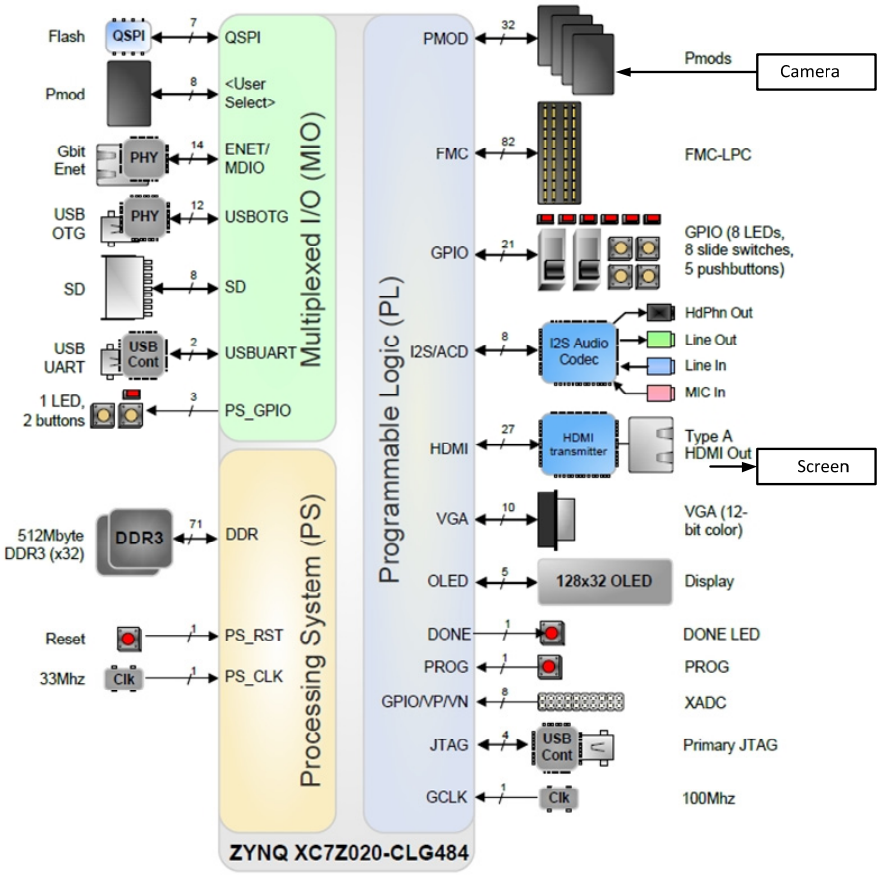


Figure 1: ZedBoard

Software required for this lab:
Xilinx Vivado Design Suite 2018.2 WebPACK

Hardware required for this lab:
ZedBoard
Micro-USB cables for JTAG interface
ASUS Screen

Data-files:
Camera_Emulation_Project_v2_lab3.zip
Manuals:
OV7670CameraChip
ZedBoard_HW_UG_v2_2

Part1: **Study VHDL Design of HDMI project**

The project is implemented on PL only, meaning you have to deal with a VHDL project. Aim is to test the HDMI display controller (component HDMI_V1). Unfortunately the camera is not available to perform a real test scenario. For your convenience a design team in Darmstadt has already created a test setup: For emulation of the camera a component IMG_GENERATOR is designed, but you as an expert have to provide the testpattern (part2).

Open the project **Camera_Emulation_Project_v2_lab3** in Vivado 2018.2 and study the hierarchical VHDL design of *Camera_Emulator.vhd*!

Task1: Draw a block diagram of the top level RTL design. Name all input and output ports and also all internal signals which connect the two components HDMI_V1 and IMG_GENERATOR! (RTL Analysis -> Open Elaborated Design).

Task2: Describe the meaning of all output signals of HDMI_V1!! Use a table format with two columns: signal name and description! For the signals provided to the HDMI transmitter do research on ADV7511.

Task3: 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 (task1)! Use a table format with two columns: signal name and description!

Part2: **Design of camera emulation**

Instead of using a camera to test the project 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*).

Task1: When investigating the component Img_Generator you will find two subcomponents:
Camera_Clock_Generator
Pixel_Generator
Complete those VHDL models so that data is generated to display **blue pixels** on your screen!

A. 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:

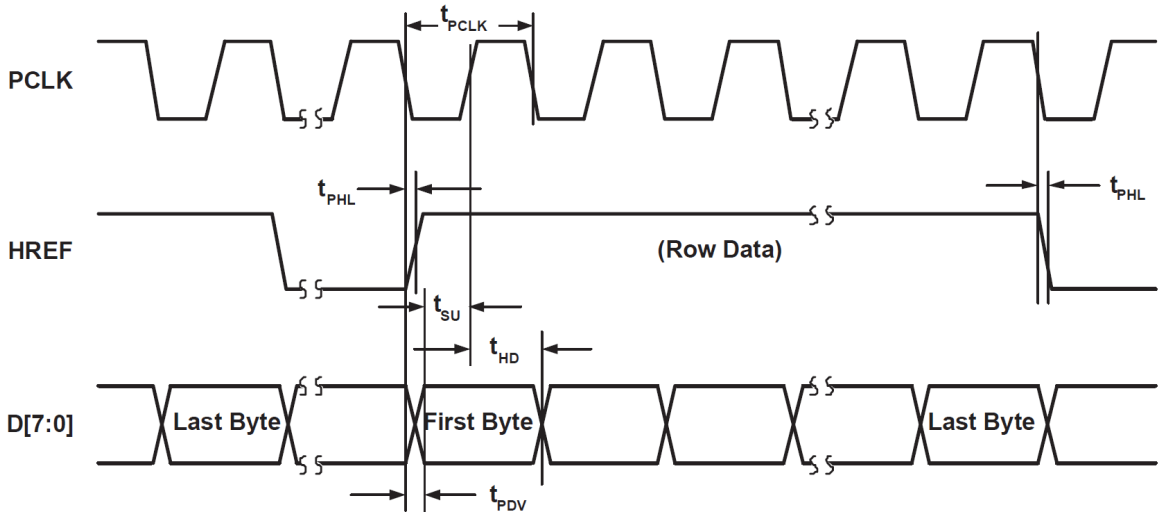
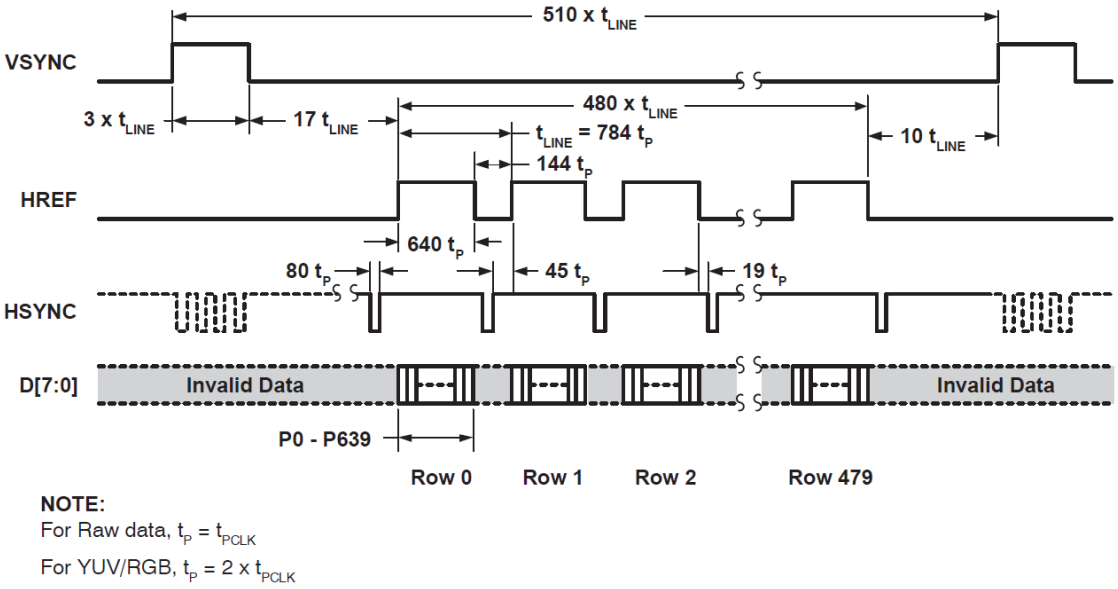


Figure 3. VGA Frame Timing:



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

B. In *Pixel_Generator.vdh* we need to specify Y,CB and CR. Use VHDL *constant* with data type STD_LOGIC_VECTOR (7 downto 0)!

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

Task2: Implement your test design on the ZedBoard! 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!

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

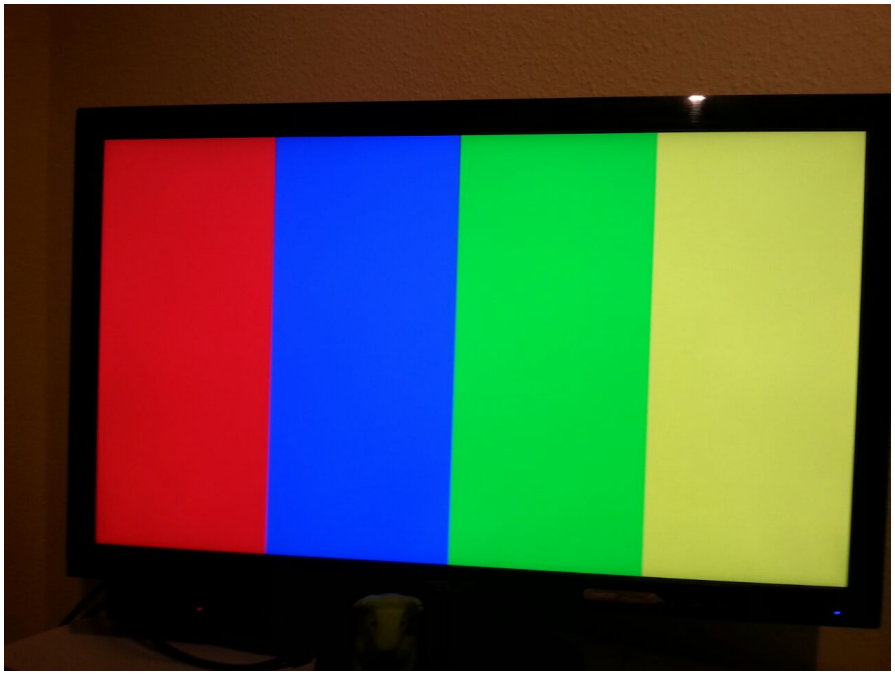


Figure 4: HDMI controller test pattern

Prepare before lab:

1. Part1: Task 1-3
2. Part2: Name all signals you need to define Pixel_Generator.vdh! Prepare values for different colors in YCbCr format. Make sure you understand the camera output sequence and the required timing of PCLK and HREF.

Note: Tutor will check on your preparation at the beginning of lab! No preparation means you need to leave lab with no points!

Lab report: deadline is **Jan/14**! Submit your lab report as hardcopy (group of 2 students). Include:

1. task 1-3 of part1
2. VHDL-code *Camera_Clock_Generator.vhd* and *Pixel_Generator.vdh* for blue pixels and color strips (task1+task3 of part2). Comment your code! Also attach photo of screen output!
3. I/O pins used in ZedBoard design (task2 of part2).