Digital System Design SS2022

Lab 1: Design and Test of VHDL IP

Submitted by: Shusmitha Padamati & Adarsh Rastogi
Matrikelnummer: 1112325 , 1112332
Date of submission: 13/May/2022

1.1 Introduction

Test a HDMI display controller, designed to display images captured by a camera. Data of the HDMI display controller is finally sent to the on-board HDMI transmitter chip. The camera is getting emulated by another VHDL model.

Objectives:

- Study of the top-level RTL design
- VHDL design of the Camera Emulator
- Simulation of the Camera Emulator
- Test of HDMI Display Controller on ZedBoard

1.2 Tasks

Part 1: Study top level RTL design

Task 1: 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!

Answer:

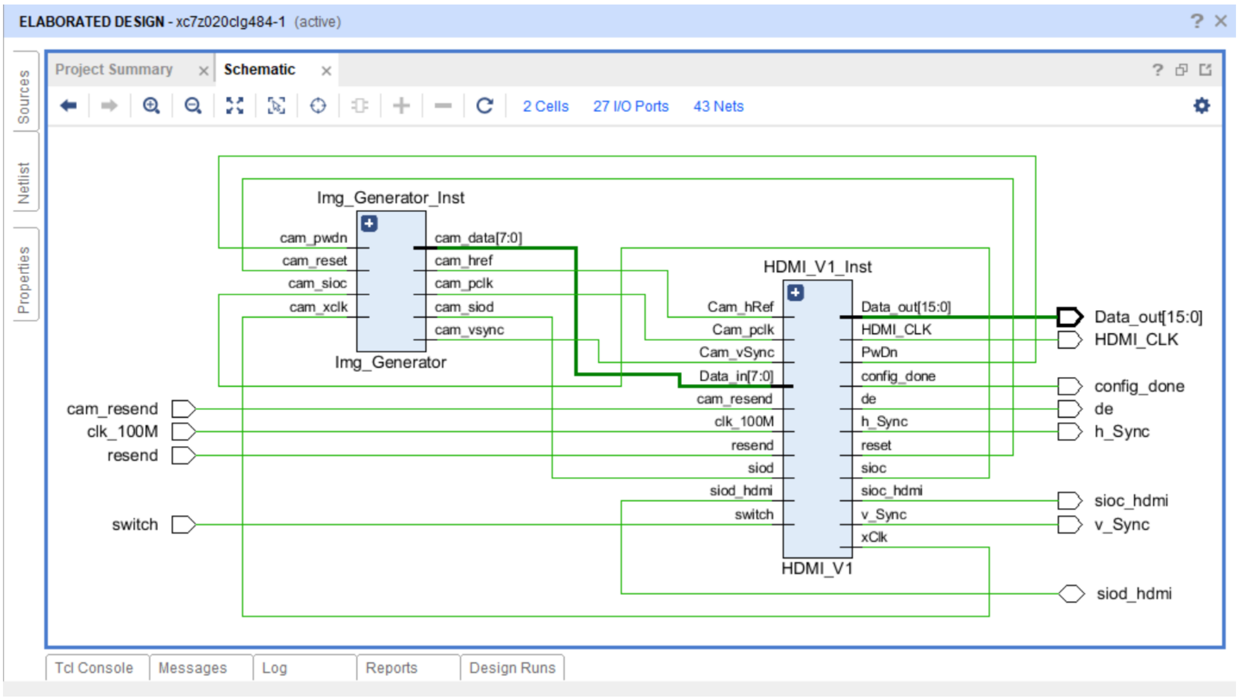

Fig 1.1 High level block diagram of top-level RTL design

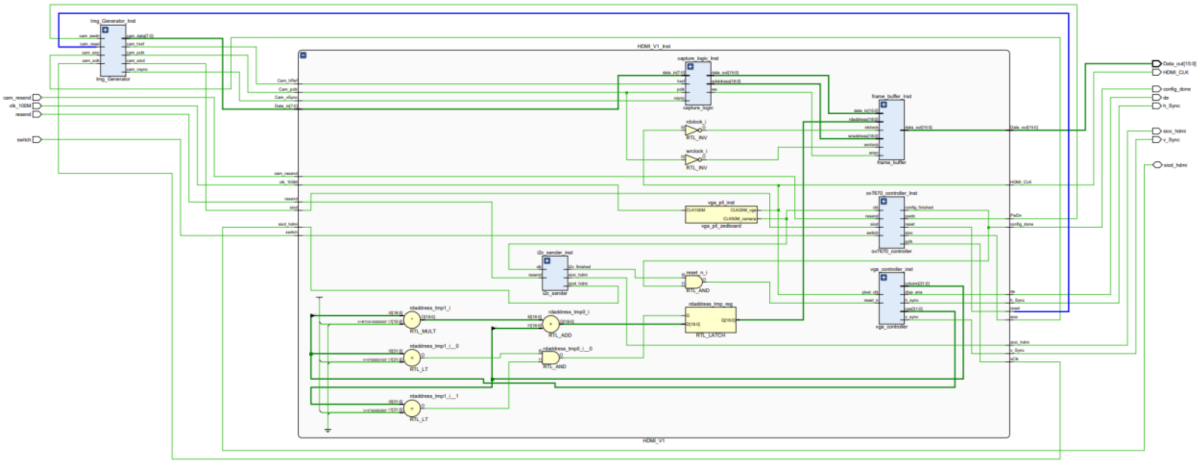

Fig 1.2 In detail block diagram of RTL design

Signal Name	Description
Data_out[15:0]	16-bit Data output of frame buffer.
HDMI_CLK	A clock of 25MHz from vga_pll_zedboard.
PwDn	Power saver mode: -One of the outputs of ov7670_controller (transfers registers to the camera over an I2C like bus). -A LED shows if it is at the power saver mode.
config_done	-One of the outputs of ov7670_controller. -A LED to show when config is finished.
de	-One of the outputs of vga_controller and acts as input to Video data capture of the HDMI transmitter ADV7511. -Display enable where '1' is the display time and '0' is the blanking time.
h_Sync	-Horizontal sync pulse. -One of the outputs of vga_controller and acts as input to Video data capture of the HDMI transmitter ADV7511.
reset	Reset: -Always '1' for normal mode. -One of the outputs of ov7670_controller.
sioc	-IC2 Clock. -One of the outputs of ov7670_controller.
sioc_hdmi	- Can write to the registers.

	-One of the i2c_sender.
v_Sync	-Vertical sync pulse. -One of the outputs of vga_controller and acts as input to Video data capture of the HDMI transmitter ADV7511.
xClk	-Clk Driver for OV7670 camera. -One of the outputs of ov7670_controller.
siod_hdmi	-Can read from/ write to the registers (out). - -One of the i2c_sender.

Table 1.1 Output signals of HDMI_V1

Task 2: 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!

Answer:

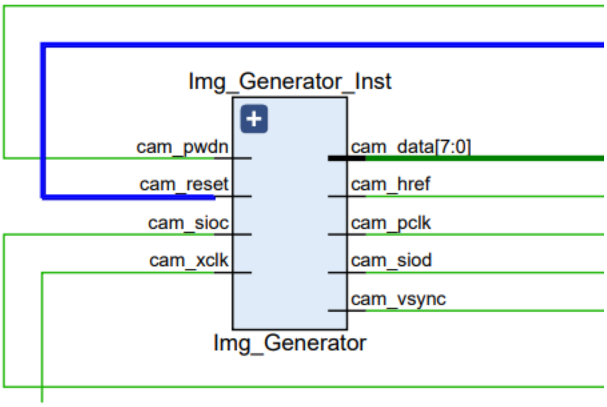

Fig 2.1 High level block diagram of Image Generator

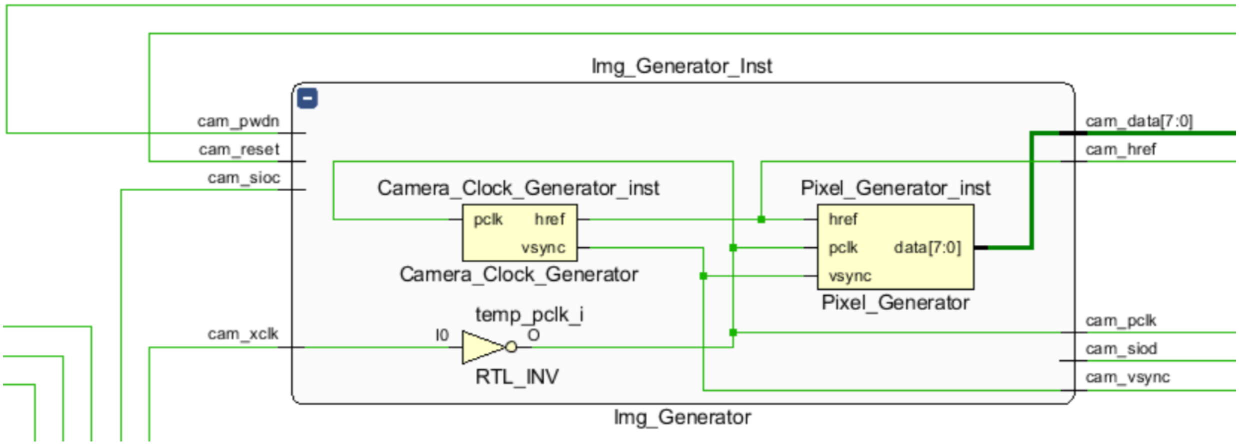

Fig 2.2 In detail block diagram of Image Generator

Signal Name	Description
Inputs	
cam_pwdn	Power Down Mode Selection 0: Normal mode 1: Power down mode
cam_reset	Clears all registers and resets them to their default values. 0: Normal mode 1: Reset mode
cam_sioc	SCCB serial interface clock input
cam_xclk	System clock input
Outputs	
cam_data[7:0]	-8-bit output of Pixel_Generator - YUV video component output
cam_href	- HREF output. - It is responsible for synchronizing each line of the image frame.
cam_pclk	Pixel clock output
cam_siod	SCCB serial interface data I/O
cam_vsync	- Vertical sync output. - It is responsible for synchronizing an entire image frame on the screen.

Table 2.1 Input & Output signals of Image Generator

Part 2: Design of camera emulator

Task 2: 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!

Answer:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity Pixel_Generator is
  Port ( pclk : in STD_LOGIC;
         href : in STD_LOGIC;
         vsync : in STD_LOGIC;
         data : out STD_LOGIC_VECTOR (7 downto 0));
end Pixel_Generator;
```

```
architecture Behavioral of Pixel_Generator is
  --Hex value representation for blue pixel in YCbCr format.
  constant y_blue   : STD_LOGIC_VECTOR (7 downto 0) := x"29";
  constant cb_blue : STD_LOGIC_VECTOR (7 downto 0) := x"F0";
  constant cr_blue : STD_LOGIC_VECTOR (7 downto 0) := x"6E";
  --A counter of 640 as resolution is 640x480
  signal   count        : integer range 0 to 639 := 0;
begin
  --On the falling edge of pixel clock
  blue_pixel_generation : process(pclk)
  begin
    if(falling_edge(pclk)) then
      if((count rem 4) = 0) then
        data <= cb_blue;
      elsif ((count rem 4) = 1) then
        data <= y_blue;
      elsif ((count rem 4) = 2) then
        data <= cr_blue;
      elsif ((count rem 4) = 3) then
        data <= y_blue;
      end if;

      if(count = 639) then
        count <= 0;
      else
        count <= (count + 1);
      end if;
    end if;
  end process;
end Behavioral;
```

Colours	YCbCr Value
Black	(16,128,128)
White	(235,128,128)
Red	(82,90,240)
Green	(145,54,34)
Blue	(41,240,110)
Yellow	(210,16,146)
Cyan/Aqua	(170,166,16)
Magenta	(107,202,222)

Table 2.1 Values for different colors in YCbCr format