Assignment 2 (15%)

CompSys303 S2 2022

Maryam Hemmati

Due: Wednesday 19 October, 11:59 pm

This is a group assignment and should be done as a group of two students, using DE2-115 board.

Submission

- Due: **Wednesday 19 October, 11:59 pm.**
- Late penalties: up to 24h late: -20% of achieved marks, 24h to 48h late: -30% of achieved marks, more than 48h late: 0 marks.
- Upload your **submission** on **Canvas** as a **zip file** including all the codes, and a pdf file which includes your answers for Task 1 to Task 3.
- Submit your completed **Peer Assessment Form**. It should be submitted separately by each member of the team.

Aim:

The aim of this assignment is to implement an embedded system on DE2-115 board using a Nios II processor, some video processing IP cores, and a few other peripherals. The implemented system can apply simple image processing filters on a video stream and output the results through the VGA output on the board. In Task 1 and Task 2, the settings of video processing IP cores are defined at hardware level and are fixed during the run time. In Task 3, the video processing IP cores are accessed and controlled by Nios II processor to provide more flexibility for the system.

Task 1 (3%)

For the first part of the assignment, you need to use the ***assign2_system.qsys*** file which has been provided on Canvas as part of the assignment. Open the ***assign2_system.qsys*** in Platform Designer and check if all the connections are set properly. You may notice that the system is very similar to what you have developed during Lab 1. There are a few other IP cores added to the system with their connections through a proper Avalon bus. Note that two different types of Avalon bus are used in the provided system. Figure 1 shows the block diagram of the video processing part of the system in ***assign2_system.qsys*** (the other parts including Nios II and its peripherals are not shown).

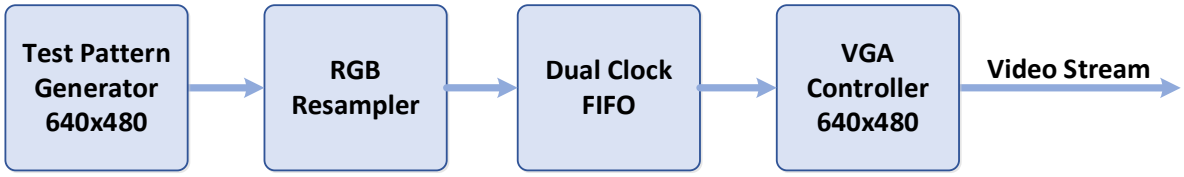

Figure 1: Video Processing Pipeline for Task 1

Note that since the data width of the video stream generated by ***Test Pattern Generator*** does not match with the data width at the ***VGA Controller***, we use an ***RGB Resampler*** IP. The ***Dual Clock FIFO*** is also inserted in the pipeline to handle the data transfer between two clock domains; ***VGA Controller*** clock is 25MHz while the other peripherals use 50MHz clock.

Your task is to:

- Generate a system from the given ***qsys*** file by Selecting ***"Generate" -> "Generate HDL"*** From the menu and adding it to a Quartus project.
- Rename the pins to make sure the pin names match with the names in the given pin assignment (***DE2_115_pin_assignments.csv***).
- Import the pin assignment and compile the design.
- Configure the FPGA on DE2-15 board using the USB Blaster.
- Connect the VGA output of the board to a monitor and check the image on the monitor. Note that if you are using the monitors in the lab, you will need a VGA to HDMI adapter. You can ask for the adapter from the technical staff in the store.
- Refer to ***Video and Image Processing Suite (ug_vip-18.pdf)*** for the detailed information about the video processing IP cores used in the system.
- In your report for Task 1:
  - Add an image of what you have seen on the monitor.
  - Explain if there is any way to change the generated pattern (compulsory). You may also try generating one or more patterns (optional).
  - Discuss the two different Avalon bus type that are used in this system, and explain the reason behind it.

Task 2 (4%)

In the second part of the assignment, we plan to modify our system to implement the video processing pipeline shown in Figure 2.

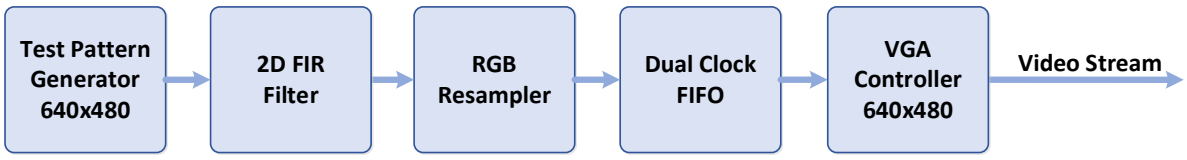

Figure 2: Video Processing Pipeline for Task 2

Note that with the use of 2D FIR filter, we apply some filters on video frames to modify them. There are several filters with different sizes and functionalities. In this assignment we use 3x3 filters for smoothing the images and detecting the edges in X and Y directions. There are a variety of smoothing filters, and a few of them are as follows:

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

The Sobel filters are usually used to detect the edges in X and Y directions. Sobel Edge filter in X direction is as follows:

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

The Sobel Edge filter in Y direction is as follows:

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Your task is to:

- Modify the ***assign2_system.qsys*** file in Platform Designer and add a ***2D-FIR II*** IP to it. When you add ***2D-FIR II,*** keep all the settings to default values other than the following settings:
  - Set the ***Number of color planes*** to ***3.***
  - Set the ***Maximum frame width*** to ***640.***
  - Set the ***Maximum frame height*** to ***480.***
  - Tick the check box for ***Enable output guard bands.***
  - Set the ***Filtering Algorithm*** to ***STANDARD_FIR.***
  - In the ***Runtime Settings***, give the path and name to the coefficient file (***Filter1.txt***) that has been provided to you on Canvas.
- Modify the connections where needed, and generate HDL for your modified qsys file. Update your design in Quartus, compile it and configure the board with the new bitstream (.sof file).
- Observe the VGA output on the monitor.
- Document your observations by taking a photo from the monitor and adding the image to your report.
- Create the coefficient files for **Sobel Edge X** and **Sobel Edge Y** filters and change the settings of ***2D-FIR II*** to use these filters instead. Compile your design for each of these two settings and observe the VGA output on the monitor for these two scenarios.
- In your report for Task 2:
  - Include the images of what you have seen on the monitor for three diffrent filters you have used.
  - Discuss how using each of these filters resulted in the figures added to our report.
  - Discuss how flexible is your design for updating the filter coefficients.

Task 3 (8%)

In the last part of the assignment, we plan to modify our system and improve its flexibility. The final system uses the Nios II processor to switch between different filters through modifying the filter coefficients in ***2D-FIR II***. For this purpose, you need to change another setting in your IP core to provide run-time control for 2D FIR filter. You will notice that by ticking the check box for ***Run-time control,*** another interface will be added to the ***2D-FIR II*** in Platform Designer. You should make proper connections for the new interface and compile your new design. With this added feature, you should create a C application for your Nios II to control the filter settings. Use ***SW[2..0]*** on the board to choose between three filters as below:

SW[2..0]	Filter
001	Smoothing
010	Sobel Edge X
100	Sobel Edge Y
Other Values	Identity

Note that the filter coefficients for identity filter is as follows:

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Your task is to:

- Modify the settings of ***2D-FIR II*** IP to add run-time control to it.
- Modify the connections where needed, and generate HDL for your modified qsys file. Update your design in Quartus, compile it and configure the board with the new bitstream (.sof file).
- Create a C application for Nios II to control the filter values by checking the three lower switches on the board.
- In your C code, access the LCD on the board to show the name of chosen filter on the LCD.
- Observe the VGA output on the monitor. Document your observations and verify that your Nios II application works as per expectations.
- In your report for Task 3:
  - Explain your approach for controlling the filter coefficeints using Nios II processor.
  - Discuss any potential improvement in your final system.
  - The report for this part does not require the images of VGA output on the monitor. However, you should include your projects and a readme file to make sure your results could be replicated when you are marked.