Lab 4: Video Processing using PYNQ-Z2

1 Overview

1.1 Introduction

Last week, we became familiar with the PYNQ overlay for performing filtering of an audio signal. This week, we will be using the PYNQ-Z2 to perform image processing. You will use Vivado HLS to create an IP to perform some image processing functionality (such as edge detection). After creating the IP, you will design an image processing overlay in Vivado and include your designed IP in the overlay.

Sobel filters are used in image and video processing to extract the edges of objects (edge detection). They can be used for noise reduction and as a precursor for applying machine learning classifiers in areas such as medical imaging.

1.2 Learning Outcomes

On completing this lab, you will be able to:

- Create an image processing IP for edge detection using Vivado HLS
- Design a video pipeline in Vivado for conducting image processing on the PYNQ-Z2 board
- Understand the elements required for designing an image processing overlay
- Successfully load the designed overlay onto the PYNQ-Z2 board and perform edge detection on an image of your choice

1.3 Definitions

- Overlay: Overlays are hardware system designs on PYNQ. It's a configurable and reusable class of Programmable Logic Design and can be downloaded into the Programmable Logic at runtime to provide functionality required by the software application. The PYNQ overlay has a Python interface, meaning it can be used like a Python package.

- High Level Synthesis (HLS): The Xilinx Vivado HLS tool allows you to create Intellectual Property (IP) by translating C code (C, C++, SystemC) into a Register Tranfer Level (RTL) implementation. The created IP can be included in your Vivado overlay for performing functions such as video processing.

1 Create the HLS IP

I. Open Vivado HLS
II. Create a new project and save it within your local directory
III. Add the *sobel.cpp* and *sobel.h* as Design Files, type *sobel_accel* as the top function then click *Next*
IV. Add *tb_sobel.cpp* as the TestBench File and click *Next*
V. For Solution Configuration, you can leave the solution name as *solution1* and set the Period to 10. Choose the pynq-z2 as the board by clicking the 3 dots next to Part Selection. Note: If you cannot find the PYNQ-Z2 board, navigate to C:/Xilinx/Vivado/2019.2/data/boards/board_files and ensure the pynq-z2 board files are present. [only on your own machines – these exist on the lab machines]
VI. Click okay to finish creating the project.
VII. Run C Synthesis by clicking the green 'play' button in the toolbar. You will be able to see information such as the timing, latency and the utilization of the FPGA resources when this is complete.
VIII. You will implement the filter on an image of your choice. Upload the chosen image (jpg format) to your project files.
IX. Open tb_sobel.cpp and edit the file paths of the source image and the filter output.
X. To run the TestBench, click the button to the right of the C Synthesis button, labelled 'Run C/RTL Cosimulation'. After this, you should find the result of the filter synthesis in the output directory you specified tb_sobel.cpp.
XI. Export the RTL by clicking *Solution→ Export RTL*

2 Design the Image Processing Overlay

2.1 Opening the project

I. Download the 4C1_Lab4.zip. Right click the downloaded file and select *Extract All…* to extract the PYNQ image to a new folder named 4C1_Lab4_<your_name> (e.g. 4C1_Lab4_AWalsh).

II. Open Vivado 2019.2, select Open Project, navigate to <path_to_project>/PYNQ-image_v2.5.4/boards/Pynq-Z2/base/base and double click on base.xpr.

2.2 Creating the Overlay

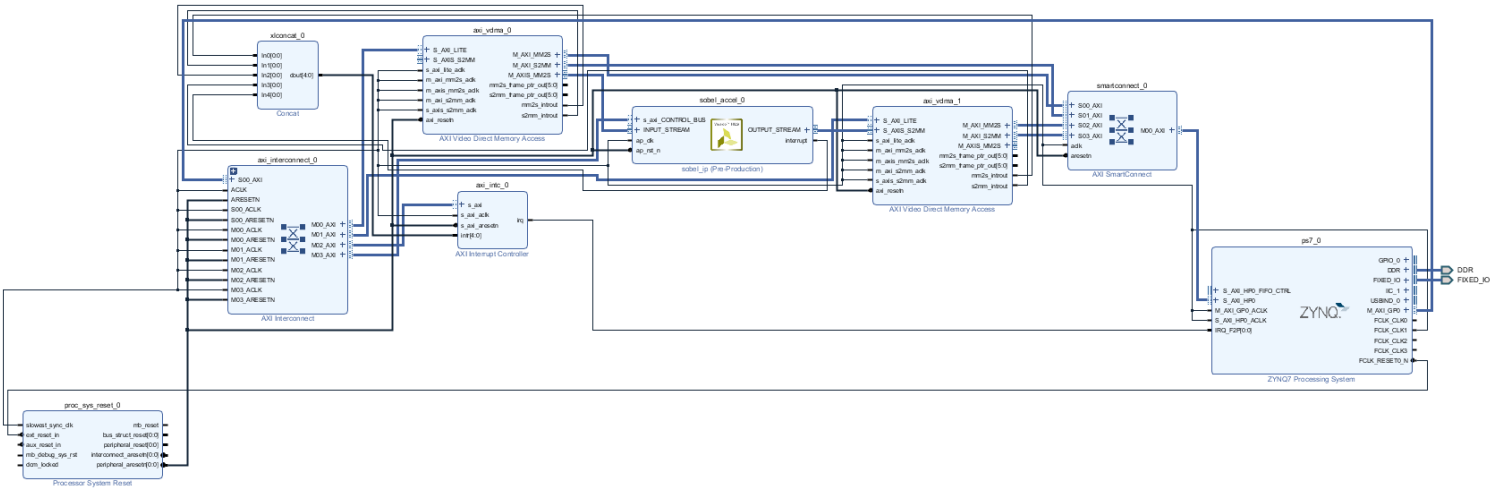
I. You should now see the PYNQ base overlay. You can delete everything as we will be building a much simpler overlay from scratch. There's a *select area* tool within the Diagram tab that you can use to delete everything quickly.

II. To add IPs, click on the plus icon and add the following:

a. ZYNQ7 Processing System
b. AXI Video Direct Memory Access (x2)
c. Processor System Reset
d. Concat
e. AXI Interrupt Controller
f. AXI Interconnect
g. AXI Smart Connect
h.

III. Next we are going to add our Sobel filter to the IP Catalog. In the Flow Navigator within the Project Manager section, select *Settings*.
IV. The Settings window will open. On the left side under Project Settings, go to IP → Repository.
V. Add the IP we created in HLS by clicking the plus icon underneath IP Repositories and select the solution1 folder within the directory of your HLS project.
VI. Click Apply to finish adding the IP to the Catalog and press OK to continue.
VII. Return to your block diagram and add the sobel_accel IP.
VIII. Customise the following changes to the IPs
    a. ZYNQ7 Processor System – double click this block to open the customisation window
        i. Under PS-PL Configuration, HP Slave AXI Interface, enable S AXI HP0 interface
        ii. Under Clock Configuration, PL Fabric Clock, enable FCLK_CLK1 and set the frequency to 100 (MHz)
        iii. Under Interrupts, enable Fabric Interrupts; Going into PL-PS Interrupt Ports, enable IRQ_F2P[15:0]
    b. AXI Video Direct Memory Access (axi_vdma_0)
        i. On the Read Channel Options, select Stream Data Width to 24 bits – This is the M_AXIS_MM2S interface that connects to the INPUT_STREAM pin of your Sobel_accel IP
    c. Concat
        i. Set Number of Ports to 5

IX. Follow the diagram to connect all the IPs together. Some of the connections can be completed by clicking the design automation link that may pop up
X. From Tools, select Validate Design to check for any errors/critical warnings.



2.3 Exporting the Overlay

We will now export the overlay to implement on our PYNQ-Z2 boards:

I. In the Sources tab, underneath the Design Sources, right click on *base* and select *Create HDL Wrapper.*
II. Close the project and open up the Vivado tcl command line.
III. Navigate to <path_to_project>/PYNQ-image_v2.5.4/boards/Pynq-Z2/base and run *source ./build_bitstream.tcl.*
IV. This will make the .bit and .hwh files available within the current directory. We will transfer these files to Jupyter Notebooks so we can implement our designed image processing overlay on the PYNQ-Z2.

3 Implement the Overlay on the PYNQ-Z2

3.1 Connect to Jupyter Notebooks

I. Use the USB dongle and the ethernet cable provided to connect your board to your PC then turn on your board.
II. As in the last lab, your board should automatically be assigned an IP address. Open a web browser and navigate to http://192.168.2.99 to open Juptyer Notebooks on the PYNQ.

3.2 Implement the Overlay

I. Create a new folder for lab 4 and upload your .bit and .hwh files.
II. You will be carrying out processing on an image of your choice. Upload your image (jpeg format) to the folder.

III. Open the provided Jupyter Notebook file (4C1_Lab4.ipynb) and follow the instructions to implement your filter on the PYNQ board.

4 Submission

Please submit a **brief** lab report containing:

- A **brief** description of the filter you created in Vivado HLS
- Screenshots of your overlay and describe the role of each IP.
- Include the image you used before and after processing.

Please submit the following in a zipped folder using your name and lab4 as the file name (e.g. AWalsh_lab4):

5 References

This lab was adapted from the project available at: https://github.com/21stars/pynq_cv