Lab 4

Sean Fahey Student Number: 21360313

Intro

The aim of this lab is to perform edge detection on an image by creating an IP and implementing it on an FPGA platform. I will be using a PYNQ Z2 board with a Jupyter interface, and the IP will be created and implemented using the Xilinx Vivado suite.

Vivado HLS will be used to define an IP block implementing a Sobel operator, which will used in Vivado to generate a bitstream that we can use on the image.

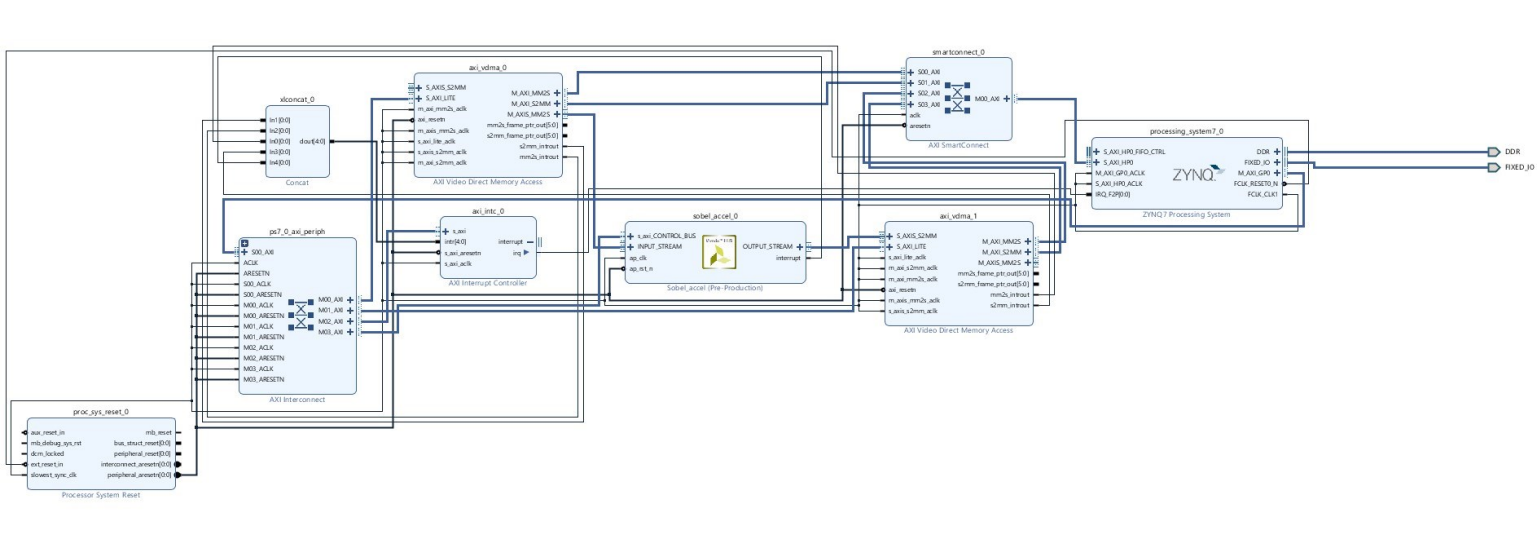
Filter Description

The Sobel operator is a method for approximating the *image gradient*. This is necessary for *edge detection*, which is how computers can isolate (and thus recognize) objects in images.
For each pixel, the Sobel operator returns either the gradient vector, or the norm of the gradient vector. Two 3x3 kernels are used to determine gradients, one for horizontal changes and one for vertical changes. The two gradients are combined ;

$$\|Gradient\| = \sqrt{Gradient_{horizontal}{}^2 + Gradient_{vertical}{}^2}$$

The resulting gradient will determine the brightness of the corresponding point on the filtered image, where a large gradient will be bright and a small gradient will be dark.

Overlay & IP



Processor System Reset

Module for any type of reset. Options selected using pins set to high/low. Supports synchronous, asynchronous reset. An asynchronous reset can also be synchronized with the clock.

AXI Interconnect

The AXI Interconnect IP connects one or more AXI memory-mapped Master devices to one or more memory-mapped Slave devices. This IP can only handle memory-mapped transfers ; it cannot handle AXI4-Stream transfers.

Concat

The Concat IP core is used for concatenating bus signals of varying widths.

AXI Video Direct Memory Access

An IP core allowing high-bandwidth direct memory access between memory and video target peripherals using AXI4 protocol. The core provides 2-D direct-memory-access (DMA) operations with asynchronous read and write channel operation. The capability for 2-D operations are used to perform image processing.

AXI Interrupt Controller

Can map several different interrupt signals to one interrupt signal. Uses AXI4 protocol.

AXI SmartConnect

Connects one or more memory-mapped master devices to one or more memory-mapped slave devices, using the AXI protocol. Can handle AXI3, AXI4 and AXI4-Lite.

ZYNQ7 Processing System

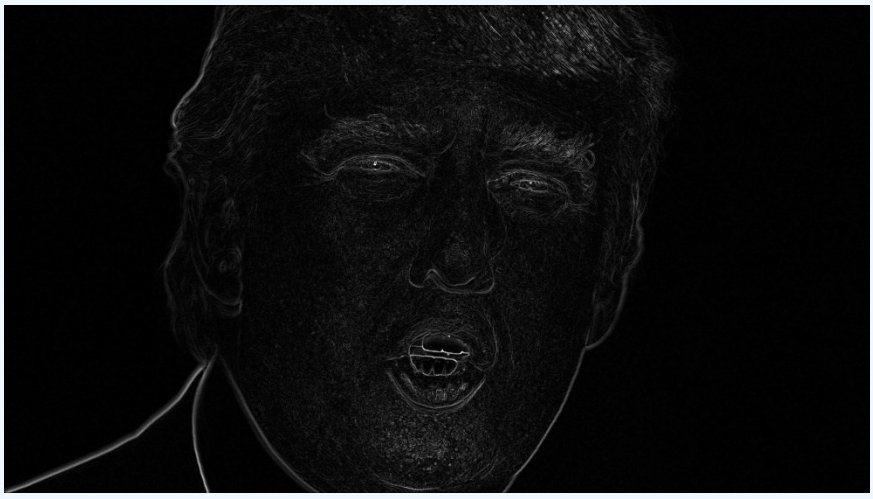
Acts as a wrapper around the ZYNQ7 processing system. Facilitates integration of other IP into the processing system.

Sobel_accel

This is the custom IP block generated in Vivado HLS. It implements a hardware accelerator for the Sobel operator described in the *Filter Description*.

Image Filtering

Original



Software-implemented filter (Vivado HLS)

3. Implement Filter

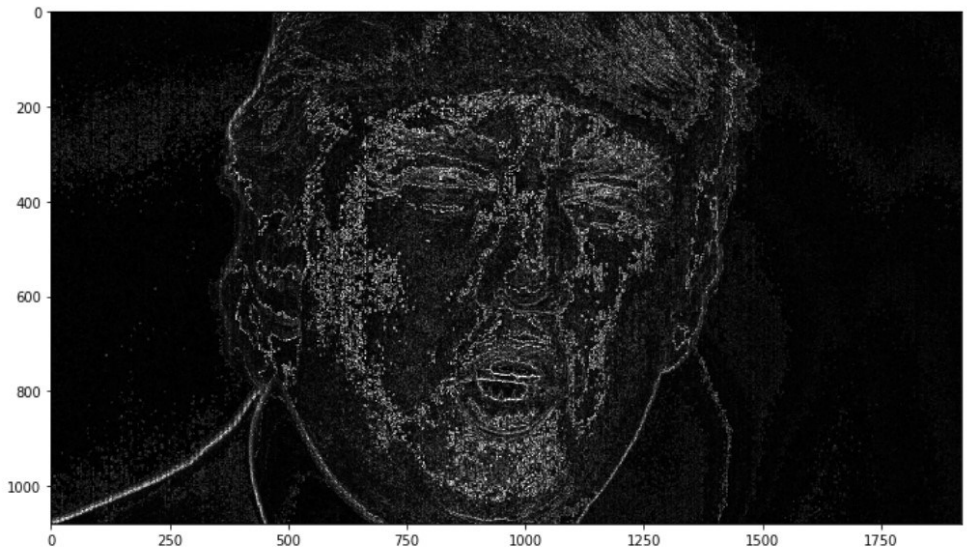
```
In [20]: #set the video mode of the DMA before starting
         framemode = VideoMode(1920,1080,24)

In [21]: vdma_in.writechannel.mode = framemode
         vdma_out.readchannel.mode = framemode
         xlnk = Xlnk()       ####### added

In [22]: #read from DRAM and pass to sobel_accel
         vdma_in.writechannel.start()
         #read from sobel_accel and pass to DRAM
         vdma_out.readchannel.start()

In [23]: # auto restart mode
         sobel_acc.write(0x00,0x81)

In [24]: in_buffer = xlnk.cma_array(shape=(height, width, 3), dtype=np.uint8, cacheable=1)
         out_buffer = xlnk.cma_array(shape=(height, width, 3), dtype=np.uint8, cacheable=1)
         in_buffer[:] = src_rgb
```

The filtered image was returning a completely black image. This issue was solved by importing a function *xlnk()* , which allows NumPy to allocate the size/shape of the array required.

4. Display Filter Output

```
In [26]: canvas = plt.gcf()
         size = canvas.get_size_inches()
         canvas.set_size_inches(size*2)
         print("Image size: {}x{} pixels.".format(width, height))
         _ = plt.imshow(out_buffer)
```

```
Image size: 1920x1080 pixels.
```



The hardware-implemented filter output image