COL215 - Hardware Assignment 3 Report

Abhishek Amrendra Kumar 2022CS11598
Ayush Kumar Gupta 2022CS11114

November 15, 2023

1 Aim of the Assignment

The objective of the assignment is implementation of a 3x3 image filtering operation. The components involved are shift registers, simple MAC units, memories for storing images and filter values, then combining all of them to do filtering process. Then showing filtered values on VGA monitor.

2 Explanation of Code

2.1 Reading of Filter values from Filter ROM

In the starting of project, we generated a ROM containing coe file of filter. Now, we made filter array and stored the pixel values of filter into this array.

2.2 Gradient Calculation using shift Registers

The Image ROM is read sequentially. We made shift registers which acts somewhat like queue. At a time it contains 131 previous values of Image ROM. After each iteration on ROM, the new ROMvalue is pushed in this shift register and the front value of shift register is removed simultaneously. From these 131 values, suitable 9 values are picked and 9 filter values are picked from filter array. Now, we feed these 18 values into compute(MAC) module. This compute unit calculates convolution value and stores it in the ram register.Also, at each iteration we maintained a minimum and a maximum convolution value till there. This same is done for all pixels.

2.3 Normalization through RAM Register

We iterate through the ram register and normalise each value by the formula given in the assignment pdf and store it in the RAM.

2.4 VGA Display Process

This is the final part of assignment. We have done it in the same way as hardware assignment 2. We just changed the values by scaling down from 256 to 64.

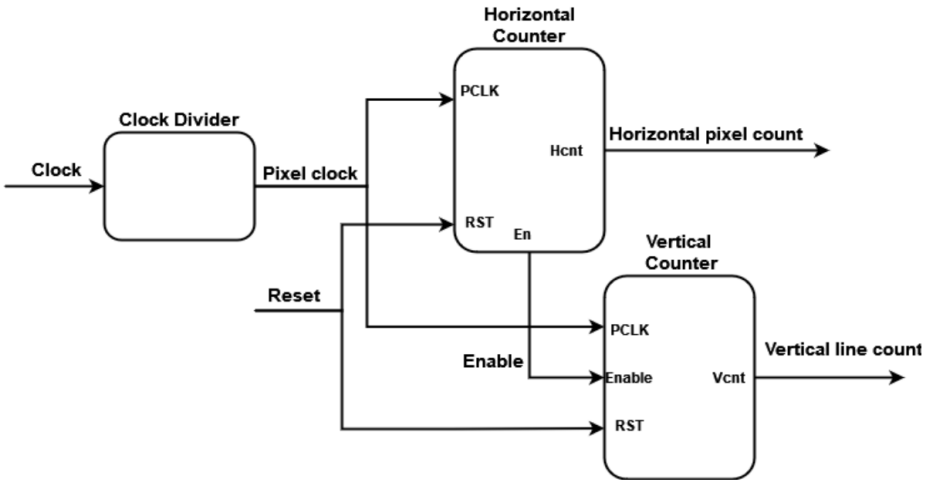
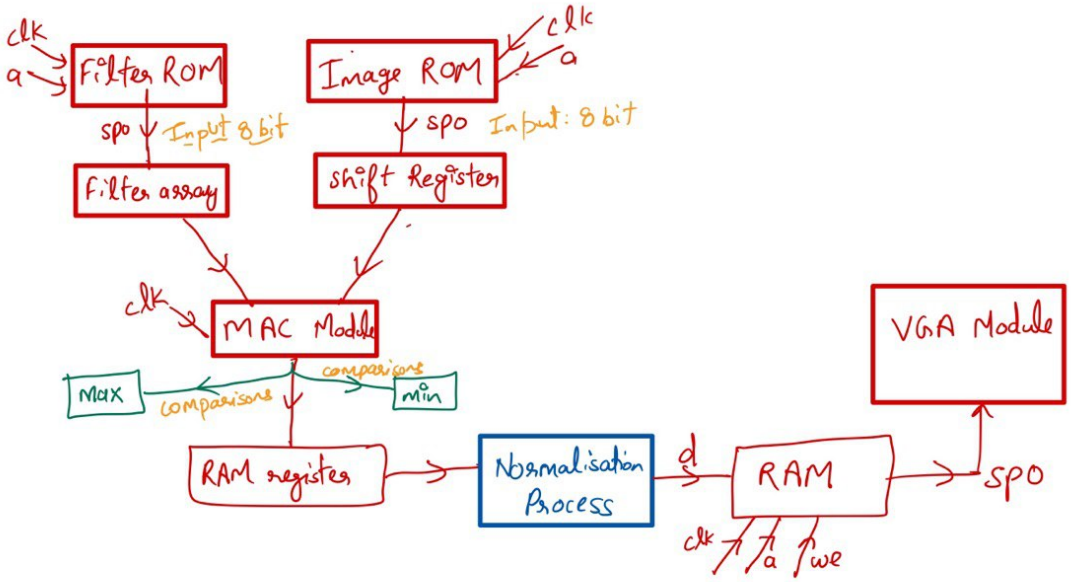


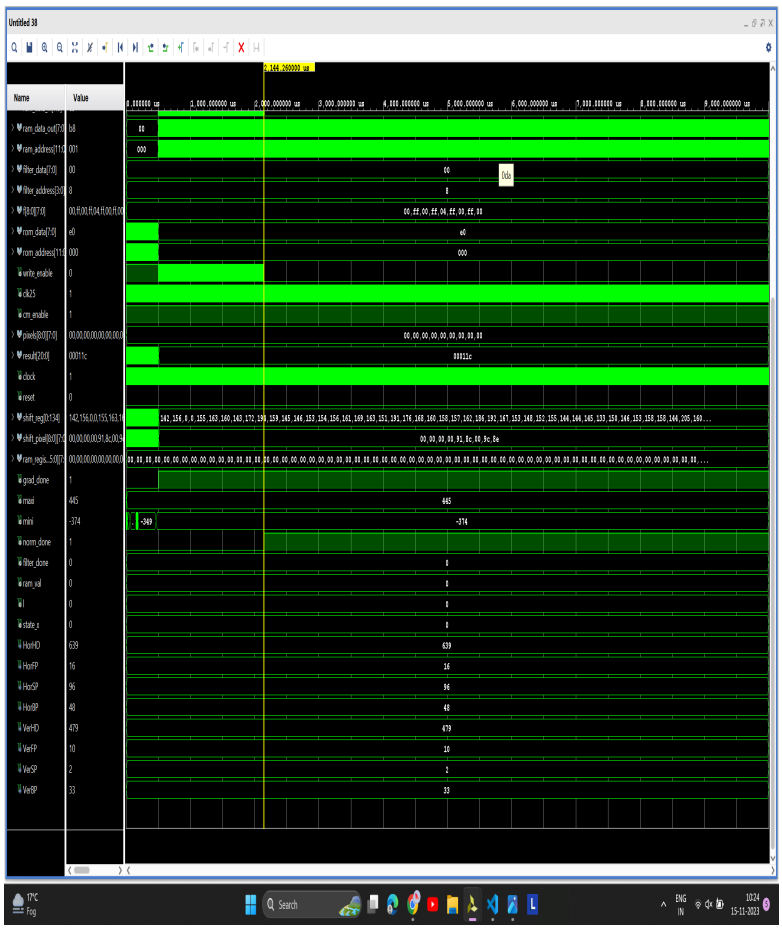
Figure 7: Module diagram for display controller

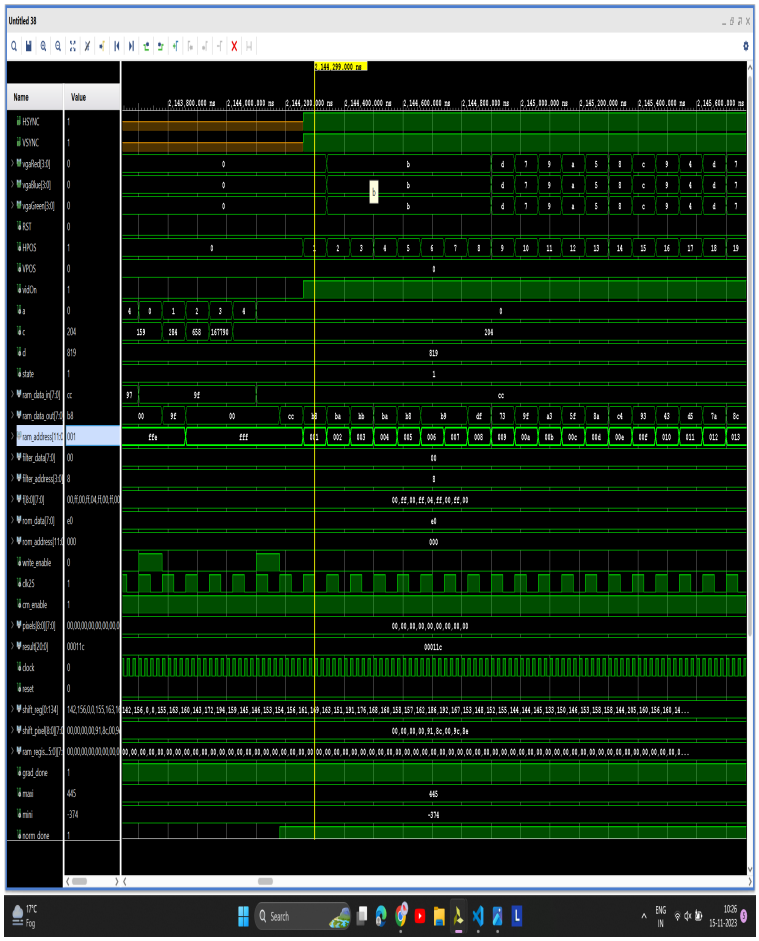
3 Block Diagram

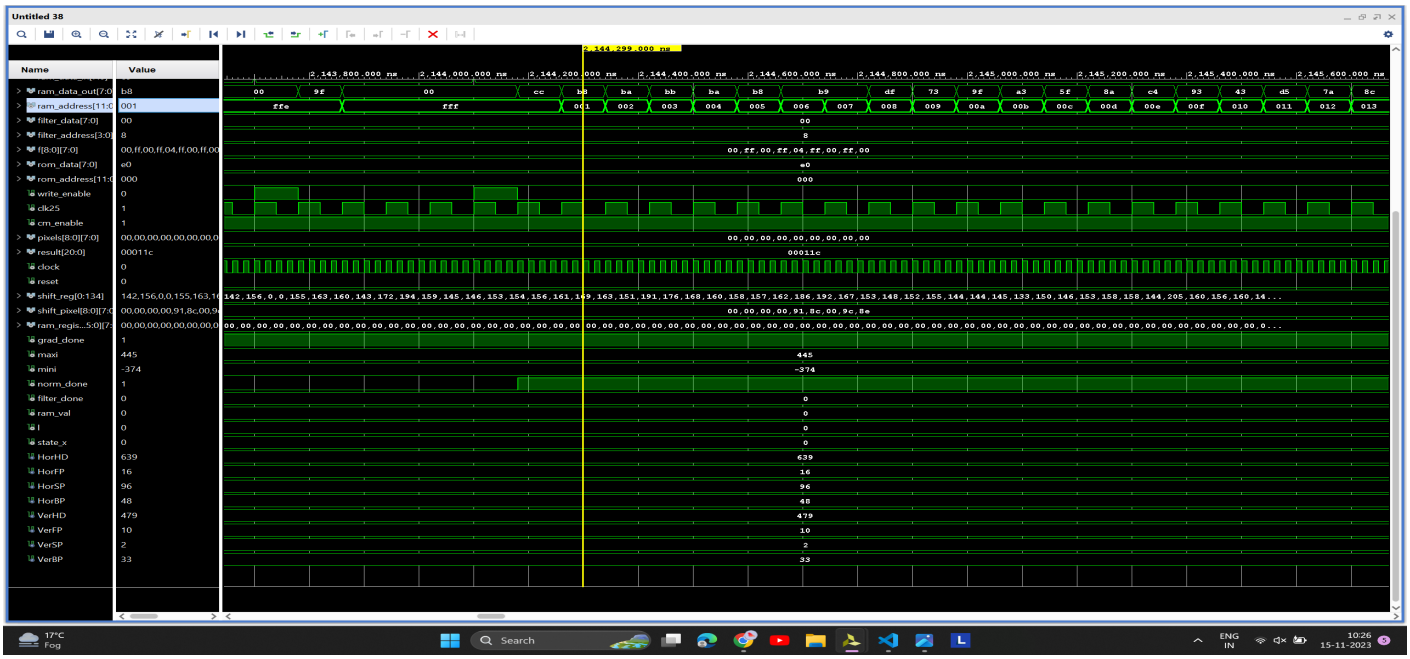


4 Simulation Snapshots

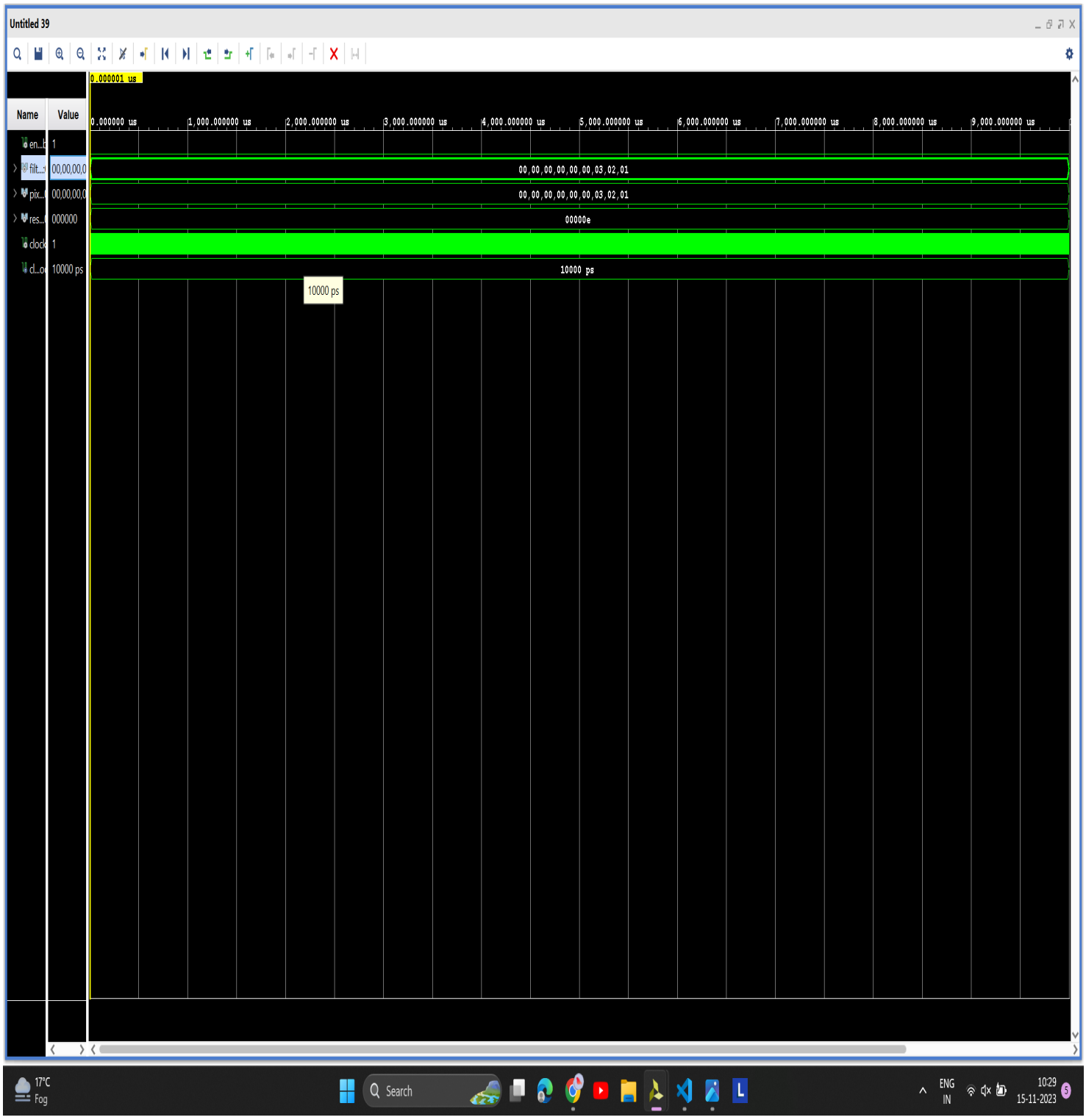
4.1 Simulation of Face file







4.2 Testbench simulation for MAC module



5 Python Tester for filtering

We made a python tester for calculating convolution values by using python libraries.

```
from PIL import Image
import numpy as np
a, b, c, d, e, f, g, h, i = [None] * 9
filter_file_path= 'filter.coe'
image_file_path='lighthouse_bin_64.coe'

with open(filter_file_path,'r') as filter_file:
    lines= filter_file.readlines()[2:]
values=[]
for x in range(9):
    line= lines[x].strip()
    if line.endswith(',') or line.endswith(';'):
        line= line[:-1]
    pixel_value= int(line,2)
    if(line[0]=='1'):
        pixel_value -=256
    values.append(pixel_value)
a,b,c,d,e,f,g,h,i=values

result= np.zeros((64,64))
kernel= np.array([
    [a,b,c] ,
    [d,e,f],
    [g,h,i]
])
image= np.zeros((64,64))
with open(image_file_path,'r') as image_file:
    lines= image_file.readlines()[2:]
    for i in range(64):
        for j in range(64):
            line= lines[j+64*i].strip()
            if line.endswith(',') or line.endswith(';'):
                line= line[:-1]
            image[i,j]= int(line,2)
```

```
padded_image = np.zeros((66, 66))
padded_image[1:65, 1:65] = image
output = np.zeros((64, 64))
for i in range(64):
    for j in range(64):
        output[i, j] = np.sum(kernel*padded_image[i:i+3,j:j+3])

max=np.amax(output)
min=np.amin(output)

image = Image.new('L', (64, 64))

for i in range(64):
    for j in range(64):
        output[i,j]= int(255*(output[i,j]-min)/(max-min))
        output[i,j]= int(output[i,j])

for i in range(64):
    for j in range(64):
        image.putpixel((j,i),int(output[i,j]))

image.show()
```

6 VGA Results vs Python Generated Image

Our Output

Python Generated Image