Report

COL215 Hardware Assignment 3

Jaskaran Singh Bhalla (2021TT11139) Saurabh Arge (2022CS11610)

Problem Statements

- Implementation of a 3x3 image filtering operation.
- Implement FSM (simple or optimised) and integrate it with the hardware design.

Task 1

ROM

First, we have created a **filter_rom** of depth 16 and data width of 8. Then we have created a **gen_rom** of depth 4096 and data width of 8.

RAM

We have created a **gen_ram** of depth 4096 and data width of 8. Here we have stored the final gradient value obtained after clamping.

Explanation

Reading from filter. Coe file :

This process is done when the filter_ready signal is ‘0’ .

First we defined variable f1,f2,f3,f4,f5,f6,f7,f8,f9 and stored the value from the given image gradient 3x3 matrix containing negative values in 2’s complement form (8 bit binary) by reading one by one by using clock cycles.

We defined a signal counter which is integer. In first cycle we gave the address of first element from matrix and in next cycle we stored the data in the defined variable f1,similarly it continues till all 9 value are stored. Then the signal filter_ready is set to ‘1’ .

Gradient

- The output pixel value O(i, j) at location (i, j) is computed as the sum of element-wise multiplications between kernel value and input image pixel, as shown in the equation below. The input image pixel at location (i, j) is denoted I(i, j).

$$O(i, j) = a * I(i - 1, j - 1) + b * I(i - 1, j) + c * I(i - 1, j + 1) + d * I(i, j - 1) + e * I(i, j) + f * I(i, j + 1)$$

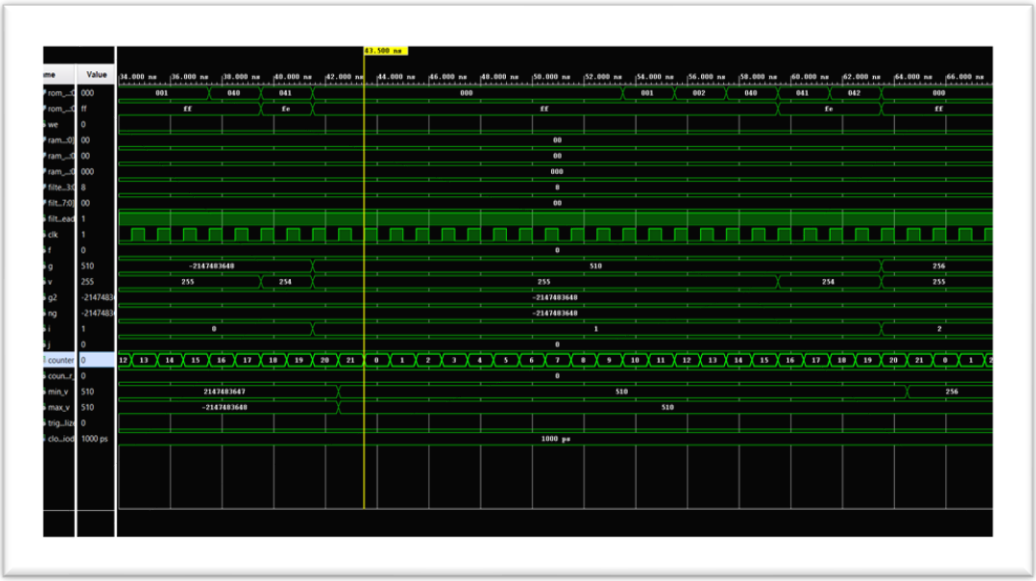
$$+g * I(i + 1, j - 1) + h * I(i + 1, j) + i * I(i + 1, j + 1).$$

For finding output pixel value here we have defined v1, v2, v3, v4, v5, v6, v7, v8, v9 variables and also max_v and min_v which will be used in the normalization part for outer pixel value.

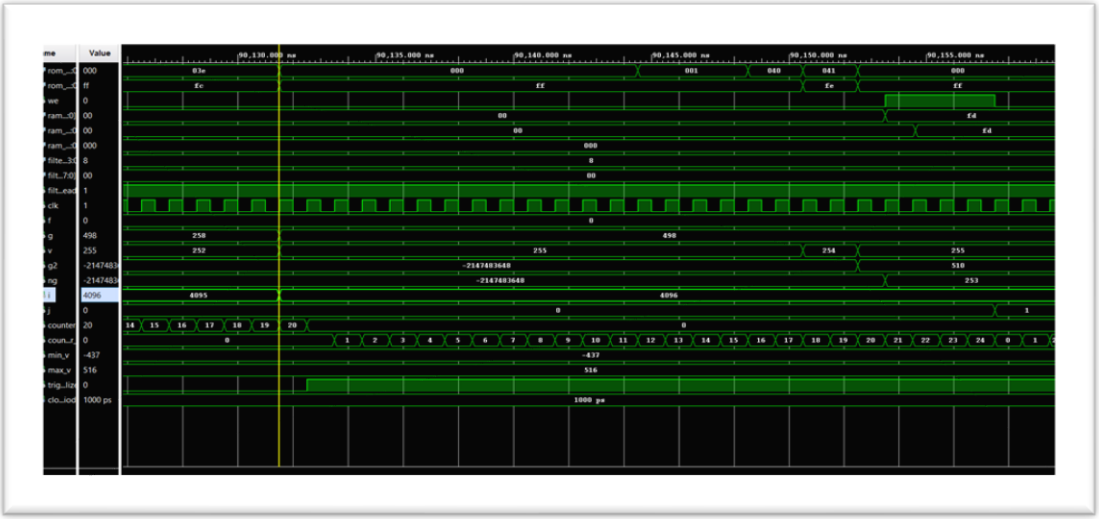
Here v1,v2… represent the corresponding I(i-1, j-1) , I(i-1,j)… values . we started the process from i=0 and when counter is 1 we assign the address of I(i-1,j-1) if it exists then in the next cycle we assign the rom data value to v1 if it exists else it is set as 0 , this is repeated till all the nine values are obtained then in the next value we calculate the output pixel value by the above given formula.

Then in the next cycle we updated the max_v and the min_v values. This process is continued until i=4095 then at the end we get the maximum minimum value of outer pixel values for the given image.

Stimulation for coins. Coe file



Here the output pixel value is stored in signal **g** when i changes to one the g value becomes 510 and this happens when counter is 19 when counter is 20 min_v and max_v is updated then after counter become 21 it set to 0 and this continue and we can see g changes to 256 this is for i=1.



Here in the above image i=4095 means it has calculated all the output pixel of the given image and we got the max_v and min_v pixel values in the given image.

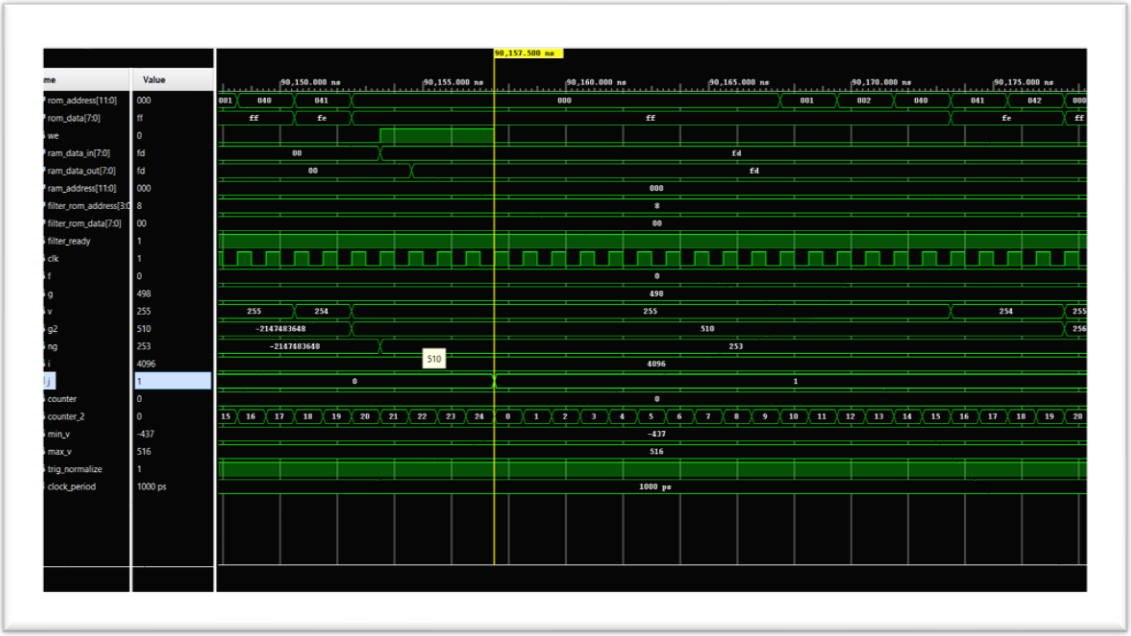
Image normalization:

Here the output value is clamped between 0-255.

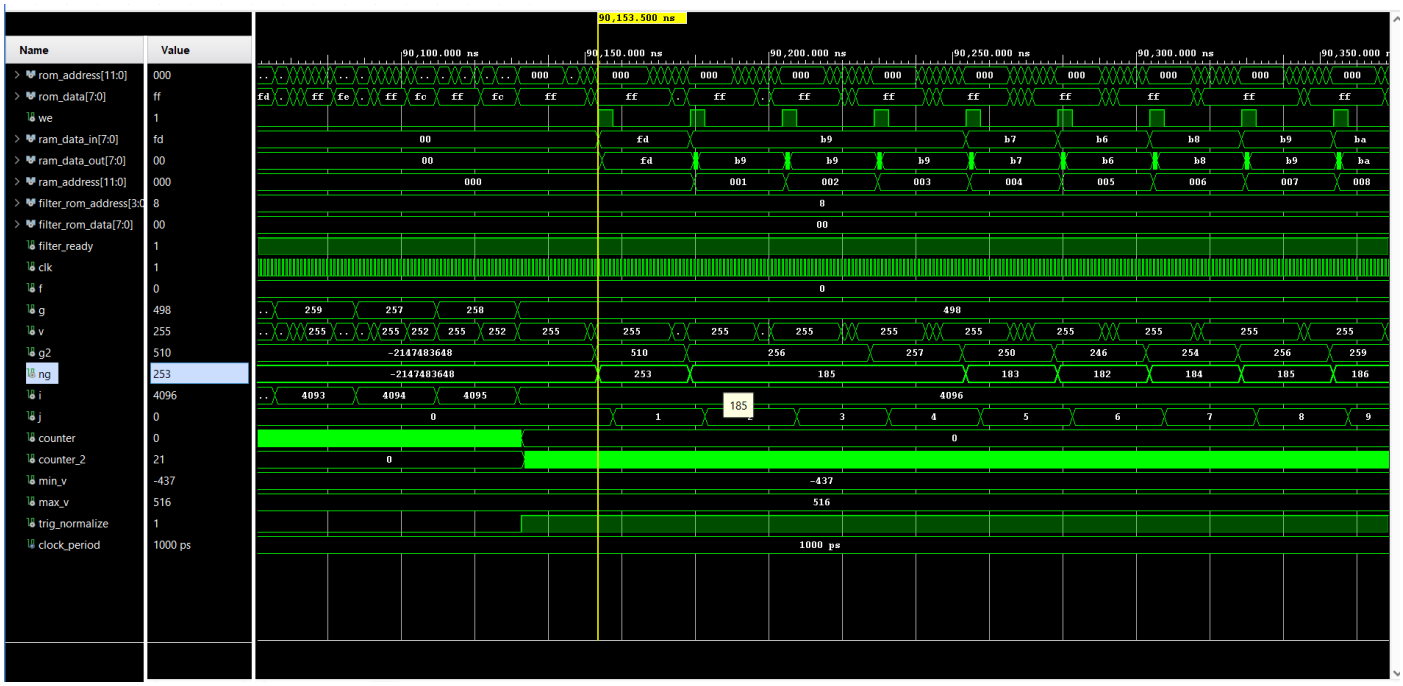
$$New\_I(i, j) = (I(i, j) - min) * 255 / (max - min)$$

When trig_normalize is '1' then the next process will start here it goes from j=0 to j=4095 and depends on counter_2. What we have done in the above gradient we do the same finding the output gradient and then when counter_2 is 20 we apply the normalization formula given above the we enable the we and give the j value to ram address and wait for 2 cycle then j is incremented.

Stimulation



Here **ng** signal represents the final pixel value. When the counter_2 is 20 it is updated to 253 for j=0 then we have waited for three cycle to write in ram then the process continuous till j=4095



Here ng is final output pixel for corresponding j value and the difference of three clock cycle because ng is calculated when counter_2 is 20 and j is incremented when counter_2 is 23.