Digital Image Filter

Anirudh Gupta

2019CS10328

Design Overview

We take 3 inputs namely clock (used to trigger process synchronous to clock edge), push button (to start the process of filtering) and the switch(used to determine whether image has to be smoothened or sharpened)

We store the image pixels in RAM which uses 8 bit registers. We store the values of filtering coefficient in ROM which uses 9 bit registers. To multiply and add we have MAC which uses 18 bit registers.

Initial state is idle. And when the push button is pressed the state is changed to S and filtering process starts.

According to the switch we choose if the image has to be sharpened or smoothening has to be done. Changing the switch mode when the filtering is going on won't have any effect. The state of switch is read when the push button is pressed.

The data of original image should be already stored in the given RAM64Kx8 memory as unsigned integer pixel values. We first calculate the pixel values of the output image one by one using the filtering coefficients and MAC. Then after 12 clock cycles the value is stored in the RAM from 32768 address onwards. The filtered image has size 118*158 and when these much values are calculated we go back to the idle state. While storing the values we check if output given by MAC is negative or not. If its negative then we change it to 0. Else we downscale the output value to 8 bits and store it in the RAM.

Now the filtered image can be loaded as desired and to again filter the image push button has to be pressed again.

Note that pressing the push button while filtering process is going on will not have any effect. These design decisions helps that there will be consistency in delivering the output.

The value stored in the ROM from address 0 is assumed to be for smoothening and from address 16 is assumed to be for sharpening.

And in the RAM the input image is from 0 and from 0 to 159 the first row is written and then from 160 to 319 next row and so on. Hence data is stored row wise in the RAM. In the output image first 158 addresses from 32768 store the 1^{st} row and so on we store 118 rows. In total we store 118*158 pixel values for output image.