COL215: Assignment 3

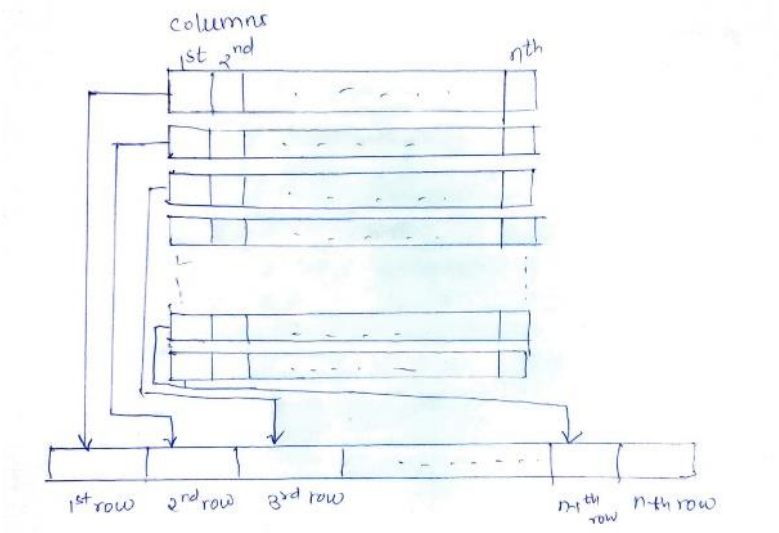
-Anirudha Kulkarni
2019CS50421

Directory structure:

/2019CS50421_A3--

1. Design Overvies.pdf
2. ASM charts –
   - 2.1 ASM_data+control.jpg
   - 2.2 ASM_control only.jpg
3. VHDL—
   - 3.1 overall_design.vhdl #assembling all components together
   - 3.2 part_d.vhdl #actual fsm and datapath
   - 3.3 readme.txt #to run simulation using ghdl and gtkwave

---



Counters:

- P: p corresponds to the current pixel that is to be processed. P is initialized to 161 as previous inputs are invalid and do not have all 9 pixels to take the product with coefficients. Similarly, we iterate till 160*119-2th pixel.
  Also, pixels in 1st and last columns are also invalid. So keep another counter by name r which keeps track of reminder of current pixel number. If we get reminder as 158 it corresponds to next pixel in last column and we increment p by 3 instead of normal increment of 1.
- R: r corresponds to column number of current pixel. It helps to cover wrapping around condition as described above. As we are not allowed to use modulo operator directly this counter does this job. R is initialised to 1 initially and goes till 158 as p rises and then comes to 1 indicating current pixel is in 2nd column.
- C: c corresponds to coefficient that is being processed. It rises from 0 till 8 taking 9 coefficients and then reinitialised to 0 in next pixel processing

States:

COL215: Assignment 3

- **Active**: When reset is given enters into active state. Checks for any push button input. If push button is presses it starts the process to process the image
- **Idle**: This state is to take care of pseudo multiple pressing due to larger pressing time compared to clock frequency. When image processing is done we reach into this state. As pressing time is much larger than processing time button will still be pressed. So this keep it in non active state till button is released. Then traverse to active state which will respond to button pressing as new image processing operation
- **proc_p:** This state corresponds to processing the pth pixel in matrix. This state will also initialize counter c corresponding to 1 of 9 coefficients.
- **proc_c**: This state will give control signals corresponding to current coefficient. If current coefficient is between 0 to 3 , we need upper row w.r.t. current pixel. This corresponds to p-160+c-1th pixels. similarly p+i-4 for middle row and p+160+i-7th for lowest row. This also takes care of 1 clock cycle delay in each coefficient access via ROM, each pixel value access via RAM and multiplication via MAC.

P-160-1 C=0	p-160 C=1	p-160+1 C=2
P-1 C=3	**P** C=4	P+1 C=5
P+160-1 C=6	P+160 C=7	P+160+1 C=8

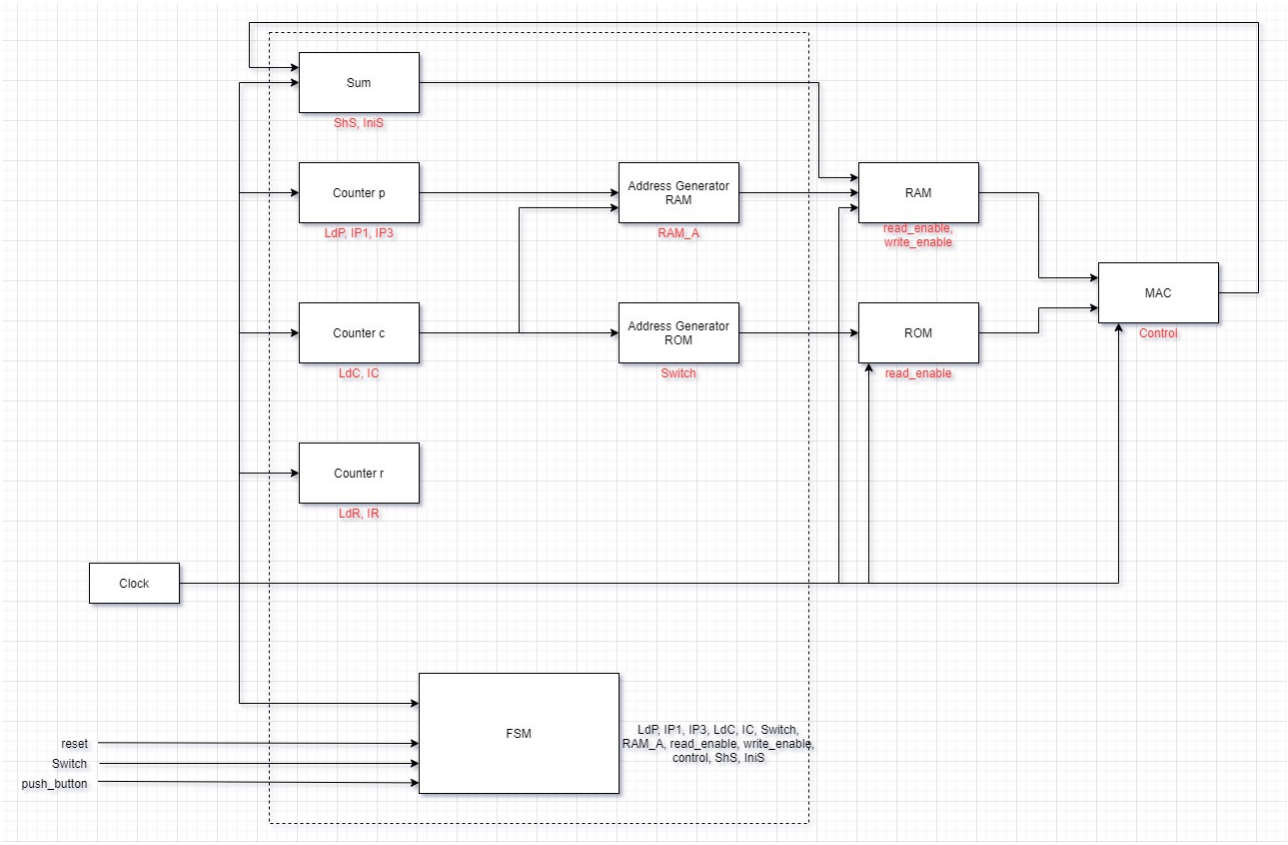
- **slc_s**: When all sums of products are processed, we reach this state. Select sum to be written in RAM as 0 when negative else chose as it is. Also vectors are resized to appropriate sizes. This is also to accommodate 1 clock cycle required to write the image pixel in the RAM.
- **increment**: When sum is written in RAM module we reach in this state and increment counters as per conditions mentioned above.

Control Signals:

- LdR: loads initial value of r=1
- IR: increment r by 1
- LdC: Load initial value of c as 0
- IC: Increment c by 1
- LdP: Load initial value of p as 161
- IP1: Increment p by 1
- IP3: Increment p by 3 in case of corner case of column
- RAM_A: chose address to be given to RAM depending on conditions
- Switch: chose address of coefficient based on switch. If switch is on or off.
- Read_enable: decides if we are accessing the coefficients and pixel
- Write_enable: decides if we are writing final sum
- Control: decides which iteration among 9 pixel is going on. If c=0 then control is 0 to initialise sum then control is kept 1 to add new products to original sum.

COL215: Assignment 3

Datapath



(Drawn by using draw.io app)

(ASM charts drawn by creatly.com)

Assumptions:

1. row wise storage of pixels
2. Coefficients are already multiplied by 128 to scale up and hence only sum of products is scaled down by 7 bits
3. switch=0 indicates smoothing operations and corresponding coefficients are stored at 0 to 8 and when switch =1 we take 16 to 24. As switch button don't require sequential checking it is excluded from asm chart