COE4DS4 – Lab #1 Exercise 1&2 Report

Group 22

Mengjia Li (400011362)
Nicole Wu (400089778)
lim14@mcmaster.ca
wuz78@mcmaster.ca
Jan 20th, 2020

Exercise 1:

The main objective of this exercise is to get familiar with the design for LCD display and touch panel operation. First of all, the 800x480 display screen is divided into 4 equal sections that are two black and two white based on the Colourbar_X (400) and Colourbar_Y (240) registers, each is assigned with a unique color counter starting from either 0 or 7. As for touch panel representation (4096x4096), TP_Y_coord (2048) and TP_X_coord (2048) are used to set a unique area_code for each region.

Furthermore, a second and millisecond counter are created to help execute the counting process. In an always_ff logic, a buffer is used to register the previous area_code every clock cycle. Meanwhile, if TP_touch_en is high and area_code_buf = area_code, counter_1s will count until 49999999 (1s/20ns = 5x10^6). Every time counter_1s is 49999999, the unique color counter will either choose to decrease or increase or rollover based on the requirement. counter_1ms follows a similar fashion with counting to 49999 this time. Each time when it reaches 49999, 4 digits BCD logic starts.

To determine the color code displayed, compare every two regions firstly.For each comparison, if two regions have the same color code, and the other two are different, take the color code appears in two regions. Otherwise, compare the value of two color codes and pick the larger one. If there is no same color code, the color code will pick the maximum color code.

Exercise 2:

The objective of this exercise is to experience the interaction between RGBY data from memory to the LCD display through the I/O line buffer. The first issue is to identify the timing to assign Y_calc data from memory to the Y_p2 register for filters 6 and 7 and synchronize in/out line buffers. Therefore, the WE for out line buffer is delayed one clock cycle since now an extra register is introduced. Similarly, Y[i+2] only takes greyscale data at the third clock cycle, while Y[i-1] and Y[i-2] need to take Y[i+1] in the corner case.

For the edge detector (6), bit concatenation is used to perform the equation of *Y_edge = 2*Y[i+2] + 2*Y[i+1] - 2*Y[i-1] - 2*Y[i-2]*. Instead of using multipliers, the sign bit from each input is extended 3 times and a 1'b0 is added to the LSB in order to shift the result left by 1. Then using switch 4 to identify whether the range is from 2^6 or 2^7. For the negative range (when Y_edge[11] is high), the bits of Y_edge[10:6/7] are ANDed together in order to check if there is any 0 exist since one 0 means it's less than -64/-128. On the other hand, for the positive range, the bits of Y_edge[10:6/7] are ORed together in order to check if there is any 1 exist since one 1 means it's greater than +63/+127.

It takes two processes to determine the median. Each sample is assigned to a register Y_medium_# and position [#]. For a single sample, if there is at least one value in other three samples that is bigger or equal to this sample and another value that is smaller or equal to this sample, then this sample is the median. The position [#] is marked as 1, otherwise as 0. Depend on the position [#] value, two mediums are summed up and shifted right by 1 bit.