**COE4DS4 - Lab #4 Report**

**Group 6**

**Jing Nan Chen (1073196) Piravinth Puvanesarajah (1055744)**

[**chenj39@mcmaster.ca**](mailto:chenj39@mcmaster.ca)[**puvanp@mcmaster.ca**](mailto:puvanp@mcmaster.ca)

**Monday, February 09, 2015**

**Exercise 1:**

The first task in this exercise to was to display the 32x32 boxes correctly on the screen. Once we reach the end of the first row, we had to increment the colour on to the next row instead of resetting the colour. After this task we implemented the cursor into the draw\_box function directly. To do this we used an ‘if’ statement in the draw\_box function. This ‘if’ statement took care of complementing the colour within a set region of a specific box. We complemented the colour and used the new complemented value as an input to the RGB\_colour. Preceding this, we had to implement the portion of the exercise where the cursor moves on different commands. To do this we used case statements according to what key was pressed. Not only does the cursor move on different key strokes, but the range of the y\_pos and x\_pos had to be met. This was also done within the key case statements. One thing worth mentioning, after each key stroke, it will take a long time to prepare for the next key stoke.

**Exercise 2:**

From NIOS II, we pass the different sets of coefficients that are converted to 8-bit 2's compliment format. That means we need 3 registers (Register 10, 11 and 12)for one set of coefficients. From the imageline interface, we get the coefficients and assign them to 10 different 8-bit signals. The signals are used as new inputs to the filter pipe. The calculations are done on unsigned 14-bit format, and afterwards, we do the clipping on the results. For 32, we check if any bit is high from Bit 12 to Bit 5. If so, we clip the result to 255. For -32, we check if any bit is low from Bit 12 to Bit 5, or the value is equal to 0x3FE0. If so, we clip the result to 255. For 40, we check if any bit is high from Bit 12 to Bit 6, if so we clip the result to 255; else we check if Bit 5 is high AND if Bit 4 or Bit 3 is high, if so we clip the result to 255. For -40, we check if any bit is low from Bit 12 to Bit 6, if so we clip the result to 255; else we check if the result is equal to 0x3FD8, or if Bit 5 is low AND if Bit 4 OR Bit 3 is low, if so we clip the result to 255. For 48, we check if any bit is high from Bit 12 to Bit 6, if so we clip the result to 255; else we check if Bit 5 is high AND if Bit 4 is high, if so we clip the result to 255. For -48, we check if any bit is low from Bit 12 to Bit 6, if so we clip the result to 255; else we check if the result is equal to 0x3FD0, or if Bit 5 is low AND if Bit 4 is low, if so we clip the result to 255. Otherwise, we clip the results to 0.