CSE P567 - Winter 2010 Name_________________

Lab 3 – Adding a Frame Counter and Imaging Threshold

Please turn in this sheet with the TA signatures along with printouts of your frame counter and threshold filter designs.

Overview

In this lab you will add a frame counter module to the camera pipeline, which counts the camera frames as they pass by and displays a frame number on the 7-segment displays. You will also modify the pipeline to display a 1-bit thresholded image.

Step 1 – Frame Counter Simulation

For homework assignment #2, you designed and simulated a circuit that implements a frame counter. First, demonstrate the frame counter simulation to the TA.

Frame Counter Simulation Signoff ______________________________TA

Step 2 – Frame Counter Integration

Now you must integrate the frame counter into the remainder of the camera pipeline. This pipeline already instantiates the frame counter module in frame_counter.v, which is empty. Replace this file with your own files. Make sure that the call to frame_counter in cameraPipeline.v matches your interface. For example, you may have to change the argument names to match your interface.

Now compile and download the modified camera pipeline project and test to see that it works correctly on the FPGA board.

Frame Counter Demo Signoff __________________________________TA

Step 3 – 1-bit Thresholded Image

In the next lab, you will design an adaptive filter that adjusts the image according to the current lighting. You will design this filter in the homework assignment and then test/debug/modify it in the next lab session.

As a way of getting started, implement a simple filter in the style of the first lab assignment. First, convert RGB to grayscale using the following equation: Lum = .30 Red + .59 Green + .11 Blue. Using this grayscale value, implement a threshold filter: for values less than or equal the threshold, the output value is black, and for values greater than the threshold, the output value is white. You may choose your own threshold value.

1-bit Thresholded Image Signoff________________________________TA

Step 4 – Adaptive 1-bit Thresholded Image

The threshold value in the previous step was arbitrary whereas it should be a function of the image itself. We will assume that images don't change much from one frame to the next and will use the average pixel value of one frame as the threshold value for the next frame. To do this, you need to accumulate the pixel values for an entire frame, and then compute the average at the end of the frame. This average is then saved and used as the threshold value for the next frame, while a new average is computed using the new frame.

There are two very important things you need to know. First, the bayPixelValid signal is asserted (i.e. set to 1) only if the pixel value is valid on the current clock cycle. Only valid pixels should be averages of course. Second, the signal "ccdNewFrame" is asserted for one clock cycle after one frame ends and before the next frame begins. You should use this signal to compute and save the average pixel value as the threshold value for the next frame, and to clear the accumulator in preparation for the next frame.

Adaptive Thresholded Image Progress Checkoff__________________________TA

(This part of the Lab may be completed in the next Lab period.)