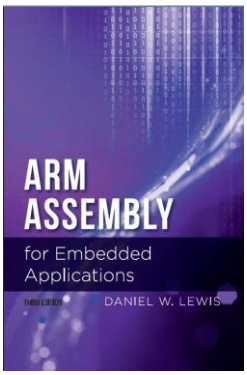

Programming Lab #9

Multimedia Processing

Prerequisite Reading: Chapters 1-11
Revised: October 29, 2017

Function `SIMD_USatAdd` (see Listing 11-5, page 215) increases the brightness of an image by adding a constant to each RGB color component of every pixel. However, the loop only processes 4 bytes per iteration and most images contain hundreds of thousands of bytes. To improve performance, use the technique discussed near the end of section 11.1.1 to modify the function so that it processes 40 bytes per iteration. The total number of bytes in the image is only guaranteed to be a multiple of four, so you will need a "cleanup" loop to process up to 36 bytes of additional RGB data not processed by the main loop.

Processing *more* than 40 bytes per iteration decreases the time per pixel in the main loop, but it also increases the (shorter) execution time of the cleanup loop. You may want to experiment with more than 40 bytes per iteration to see how it affects the processing time per pixel. If so, you may find this spreadsheet helpful. Although it's not usually a significant issue, note that increasing the number of bytes processed in one iteration of the main loop inherently limits the minimum size of the image.

Test your function using the main program downloaded from here. Press the reset (black) pushbutton to start program execution, which should display the image properties. Pressing the blue button once displays the initial version of the image, precomputes a 4% brighter image, and displays the number of clock cycles required to execute your function. Each subsequent button press displays the next image and precomputes another 4% brighter version.