ELEC-H-473 Microprocessor Architectures
SIMD labs, support document

2015–2016

1 Objectives

This document explains what are the minimum requirements for the code you develop during the SIMD labs of the Microprocessor Architectures practical sessions.

2 Lab 1: threshold on image

This lab is about applying a threshold to a gray[1] image.

We expect you to demonstrate you are able to:

- open a file in C/C++ (and verify no error occurred)
- allocate dynamically memory for a buffer (and verify the memory space has been allocated)
- copy a (section of the) file to the buffer
- process it with pure C code (*i.e.* architecture independent code)
- process it with inline assembly code (gcc of visual studio style) demonstrating the use of SIMD instructions
- measure accurately the time spend for processing for both codes, conclude
- write the processed data back to a file
- display the result.

[1] byte coded 256 shades, because 50 is not enough

3 Lab 2: morphological image filtering

This lab is about morphological image filtering using a min/max application on a 3 × 3 neighbourhood.

We expect you to demonstrate (again) you are able to:

- open a file in C/C++ (and verify no error occurred)
- allocate dynamically memory for a buffer (and verify the memory space has been allocated)
- copy a (section of the) file to the buffer
- process it with pure C code (*i.e.* architecture independent code)
- process it with inline assembly code (gcc of visual studio style) demonstrating the use of SIMD instructions
- process the corner cases accordingly
- measure accurately the time spend for processing for both codes, conclude
- write the processed data back to a file
- display the result.

4 Lab 3: multi-threading

This lab is about using multi-threading to improve further the processing time for the exercises of the previous labs.

We expect you to demonstrate (again) you are able to:

- open a file in C/C++ (and verify no error occurred)
- allocate dynamically memory for a buffer (and verify the memory space has been allocated)
- copy a (section of the) file to the buffer
- create multiple threads
- divide wisely the data, the goal is to process them using the code from the previous labs
- process the corner cases accordingly
- measure accurately the time spend for processing for both codes, conclude
- write the processed data back to a file
- display the result and verify correctness