



Universidade de Aveiro

Mestrado Integrado em Engenharia Computacional

Computação Paralela

Lesson 2: Vector computing using MMX and SSE

Academic year: 2022/2023

Nuno Lau/Rui Costa

The use of multimedia instruction extensions of current processors allows increasing the performance of multimedia applications and of some common operations in general computing. The **gcc** compiler allows the insertion of assembly instructions embedded in C code through the `__asm__` directive (for more details see [1]), and this feature will be used in this lab assignment for various experiments using instructions from MMX and SSE multimedia extensions.

1. In this first exercise we are going to test the performance of routines to add arrays. The **cp_simd.tgz** archive contains the files: **Makefile**, **sumarray.cpp**, **TheMMXinstructionSet.pdf** and **Intel-IA32-Manual-Vol-1.pdf**. For the next exercises, compile the programs with and without optimization options and take note of the execution times.
 - 1.1. Analyze the **sumarray.cpp** program and identify the implemented functionality.
 - 1.2. Compile and run the **sumarray.cpp** program. Determine the speedup of the MMX sum version. Draw conclusions about the relative performance of the sum functions.
 - 1.3. Replace the **paddb** instruction by the **paddusb**. Run the program, observe the produced results and explain them.
 - 1.4. Implement a function to add the contents of two arrays by using the **movdqa** and **paddb** instructions and SSE registers. Name that function **sumarray_sse**. Test the implemented function and compare its performance with the performance of the other two available functions.
 - 1.5. Create a copy of the **sumarray.cpp** file and change it so that the three available functions operate on **char** type arrays.
 - 1.6. Change the program obtained in exercise 1.4 so that the **sumarray** functions can operate with arrays of any size.
2. Consider now a program to sum all elements of an array.
 - 2.1. Implement 3 versions (C, MMX and SSE) of a function to add all elements of an array. The function prototype should be: **int sumelems(int *a, int size)**.
 - 2.2. Change the SSE version of the function implemented in exercise 2.1, in order to use the horizontal add instruction available in SSE3.

Bibliography

- [1] <http://www.ibiblio.org/gferg/ldp/GCC-Inline-Assembly-HOWTO.html>
- [2] Intel® 64 and IA-32 Architectures Software Developer's Manual, Volume 1: Basic Architecture, Intel Corporation, October 2019
- [3] The Art of Assembly Language Programming (AoA), Randy Hyde, chapter 11, 2001