Computer Science
C.Sc. 342
Final Take Home Test No. 3
Test Title:
OPTIMIZATION OF DOT PRODUCT COMPUTATION OF TWO VECTORS USING VECTOR INSTRUCTIONS
CSc   or   CPE

Submit report and ready to demo video, working source code files at 11:58 PM May 18, 2021

Objective:

The objective of this final take home test is to optimize compiler generated code to compute dot product using vector instructions.

Tasks to perform:

1. *Use CPUID instruction to determine your processor vector processing capabilities.*
2. *Write C++ function to compute dot product in Visual Studio environment. Place the function in a separate file from main() that calls this function. Vector sizes should be powers of 2 (e.g. 16, 32, 64, ...512, ..2^{16} etc.). Disable Automatic Parallelization, /Qpar, and Automatic Vectorization, /arch.*
   Use QueryPerformanceCounter function to measure execution time.
   Plot graph: time versus vector size.
3. Compile code in §2. *Enable Automatic Parallelization, /Qpar, and Automatic Vectorization, /arch.*
   Use QueryPerformanceCounter function to measure execution time.
   Plot graph: time versus vector size.
   Inspect compiler generated assembly code. Observe if compiler vectorized code for very large vector sizes. Try to optimize compiler generated code. Based on compiler generated assembly code (or your optimized code) create an assembly code for dot product computation function (in the same way as shown in the text book for "clear-array example for MIPS"). Please refer to Tutorial in a separate post.
   Use QueryPerformanceCounter function to measure execution time.
   Plot graph time versus vector size.
4. *To optimize the code further, please try to use vector instruction DPPS to compute dot product.*
   Use QueryPerformanceCounter function to measure execution time.
   Plot graph: time versus vector size.
5. Compare all plots in one figure.
6. Submit a detailed report and complete source code listing. If requested be ready to demo working project.
7. Perform this take home test in LINUX using gcc. No MIPS, No ARM is required in this test.
8. What to submit: 1. Write a report, 2.Create less than 2 min video on this project, 3. Source code files used in this project + Readme file with instructions.
   DO NOT SUBMIT PROJECT FILES SHOWN IN TUTORIAL!
   TUTORIAL was posted on slack on a separate post on Monday May 3, 2021.