263-2300-00: How To Write Fast Numerical Code

Assignment 7: 70 points
Due Date: May 9th , 17:00
http://www.inf.ethz.ch/personal/markusp/teaching/263-2300-ETH-spring13/course.html
Questions: fastcode@lists.inf.ethz.ch

General submission instructions (read carefully):

- (Submission)
  We set up a SVN Directory for everybody in the course. The Url of your SVN Directory is
  https://svn.inf.ethz.ch/svn/pueschel/students/trunk/s13-fastcode/YOUR.NETZH.LOGIN/
  Submit this homework to the hw06 folder!
- (Late policy)
  You have 3 late days, but can use at most 2 on one homework. Late submissions have to be emailed to fastcode@lists.inf.ethz.ch.
- (Formats)
  If you use programs (such as MS-Word or Latex) to create your assignment, convert them to PDF and submit to svn in the top level of the respective homework directory. Call it homework.pdf.
- (Plots)
  For plots/benchmarks, be concise, but provide necessary information (e.g., compiler and flags) and always briefly discuss the plot and draw conclusions. Follow (at least to a reasonable extent) the small guide to making plots (lecture 5).
- (Neatness)
  5% of the points in a homework are given for neatness.

Exercises:

1. *(Warmup Vectorization 35 pts)* The goal of this exercise is to get you started with vectorization. For the code below

```
void warmup (float * x, float * y, int size ,float alpha)
{
 for (int i = 0; i < size; i++)
        y[i] = x[2*i] * x[2*i] + x[2*i+1]/alpha;
}
```

   do the following

   (a) (Baseline) Create a timing setup and measure a base line performance of the code with vectorization disabled. Allocate x and y aligned with sizes x = 1600 and y = 800 respectively.
   (b) (Assembly output) Create assembly output of our base line implementation. Inspect it and make sure there are no vectorized instructions in the code. Name it base.asm and submit it to the svn.
   (c) (Auto vectorization) Enable vectorization and let your compiler give you a vectorization report. Create again an assembly output and commit it to the svn naming it auto.asm. Give a very short report on the result.
   (d) (Manual vectorization) Even if the compiler manages the auto vectorization - to get you familiar with intrinsics, implement a version of the code using them. Name that code manual.c and commit it to the svn. (the code does not need to be general size, but only needs to handle the size specified for the base line)
   (e) (Performance comparison) Create a small plot reporting the speedup of the auto-vectorized and your manual implementation compared to the base line.

2. *Vectorization (35 pts)* Given the following code

```
void FIR(float * y, float * x, float h0, float h1, float h2, float h3, int size)
{
 for (int i = 0; i < size-3; i++)
  y[i] = h3 * x[i] + h2 * x[i+1] + h1 * x[i+2] + h0 * x[i+3];
}
```

   Repeat the steps of the previous example. Test with input size x = 800 and align x and y to 16 bytes. Commit the same files as in the previous example, but add prefix fir_.