Computer Systems Architecture
Assignment 4
Due: Tuesday, May 13

This assignment is meant to test your understanding of cache designs and the interaction between certain cache configuration parameter values and the effect they have on cache performance.

For this assignment work in teams of two. If possible, I'd like you to pair up with someone you haven't worked with yet. But as was the case for the last assignment, this can be your decision.

Part I: Cache simulation

Write application programs that enable you to discover the cache size, associativity, line (block) size and memory update policy of the **L1 data cache** being simulated in a special version of the SimpleScalar tool set. Design the programs so that each one determines a particular configuration parameter. The order in which you run your applications might be important, so think about this as you are designing the program algorithms.

The cache size, associativity and line size of the L1 data cache will all be powers of 2. The cache will be no larger than 64KB, the associativity will be no greater than 4-way and the line size will be a minimum of 4B and no larger than 64B.

You will use a modified version of Blis, called *Secret*, that has been altered so that the cache configuration parameters of interest have been set to particular values, but are not printed as part of the output. *Secret* will live in Evan's home directory on the instructional machines; you can reach it at:

~/evan/assign4.tgz

The tar file will contain instructions on how to use it. Evan will also send email with additional explanation.

To compile your application programs for Blis, use the alpha cross-compiler discussed on the Blis website (linked from the course website). You can install this compiler on your home Linux machine, or use it on the instructional machines (it's already installed there). Evan will also email any details to these general directions for compiling your application programs.

Secret has been protected from your prying eyes. There are undoubtedly techniques that would enable you to discover the cache configuration we have used, but it is probably not worth your time trying. Whether you do or not, the graphs and/or tables you construct from your simulation output will still have to show data that "proves" that you have discovered the correct configuration, and your application programs must be able to run with the protected configuration file. To test your programs, we may run them ourselves, to confirm your output.

You may use the standard version of Blis to verify the configuration parameter values you have chosen. However, it is not necessary to use this software; it is your applications that will discover the configurations and your results that will validate your choice.

Part II: The report

Your report should follow the guidelines for the last report, except that it should be shorter -- only 2-3 pages of text should do it (your figures and tables can be extra). You won't need all sections of a normal research paper this time. Sample sections might be (1) an introduction that simply provides in a nutshell the problem you are trying to solve, (2) a section that describes the algorithms in your programs, and (3) a results section that presents your graphs/tables, explains them and analyzes the data. No summary will be needed. And don't forget to clearly state what you think the configuration is!

Learning to write concisely is one of the goals of an assignment like this. We are looking for a concise description of the techniques you used to discover the cache configuration parameter values and your analysis of the data that led to this conclusion. If your report is much longer than the recommended length, we may not read the whole thing.

Part III: Turning it in

Turn in your report via e-mail or paper handouts. Use postscript or pdf (it s hard for us to read MS Word documents). Evan will send out directions for turning in the code if he wants to see it.