

CIS 775: Computer Architecture  
Fall 2003

Lab Assignment #1 (Performance and Instruction Set Architecture Issues)

Instructor: Srinivas Parthasarathy

Due: Thursday, 16th October before class.

This exercise carries 100 points.

**You need to do this lab on a SUN Solaris system, the default system in the CIS environment.**

The purpose of this exercise is to get yourself familiarized with simple scalar simulator environment. There are seven individual simulators *sim-bpred*, *sim-cache*, *sim-cheetah*, *sim-fast*, *sim-outorder*, *sim-profile* and *sim-safe*. In this lab we will be focusing on issues related to instruction statistics and its corresponding bearing on execution time. Your first task is to select the appropriate simulator required to solve this task.

The simulators are available at `/usr/class/cis775/simplesimbin`. There are other files available there as well. The first thing to do is to create a LAB1 directory somewhere in your directory space. Once done and once you are in this directory, an easy way to invoke these programs from your directory is by setting symbolic links to the particular executable desired using the `ln -s` command. For example the command `"ln -s /usr/class/cis775/simplesimbin/sim-cache ."` (do not forget the `."` at the end) will establish a soft-link to the `sim-cache` simulator in this directory.

Once you have linked in the appropriate simulator(s), the next thing to do is to copy the benchmark programs required for this assignment to this directory. Do not link the programs but copy them to your directory space. Note you will require to clear about 15MB of space in your user space to do so. Now you are ready to run the simulator programs **MAKE SURE THAT ALL PERMISSIONS TO READ/WRITE/EXECUTE IN THIS DIRECTORY ARE CORRECT**. Incorrect permissions will result in the reporting of incorrect profiles.

The tutorial handouts that were distributed earlier should have given you an idea about the simulator as well as the PISA-big (MIPS-based) instruction set. This is the instruction format we will be using. If you did not receive the tutorial, a copy is available from me or the TA. Finally, a word to the wise, last year several people tried to run the simulator at the last minute. Guess what? For some of the programs the simulation takes a while to run and they could not submit the labs on time. **Lesson in a nutshell: do not procrastinate!**

1. Given that the goal of this assignment is to focus on issues related to instruction statistics, instruction class behavior, and its corresponding bearing on execution time (for the PISA-big instruction set). Which of the seven simulators did you pick to achieve your objectives? [5 points]
2. For the four SPEC benchmarks, *cc1*, *anagram*, *compress95* and *go*, located in `/usr/class/cis775/benchmarks` and for the inputs prescribed in `/usr/class/cis775/benchmarks/README` (you should have copied this so this should be now available in your directory too), your objective is to construct a dynamic instruction mix table (similar to figure 2.32/2.33 in the textbook). Please note that the examples in the README file *may* not contain the

correct simulator to use for this assignment – you will need to figure that out. There should be one column per benchmark. In addition there should be two additional average columns with the following mixtures (MIX1-1:4:4:1 and MIX2-4:3:2:1) for the benchmark programs *cc1*, *anagram*, *compress95* and *go* respectively.

[25 points]

3. Construct a similar table for instruction classes and addressing modes. Note that the addressing mode option in the simulator only provides information relating to the addressing modes for loads and stores. However, your table will need to include information relating to register direct addressing (assume that all non-memory references are register direct) as well. [20 points]
4. Assuming we are working with a chip in which all instructions take 1 clock cycle to execute and that there is no pipelining. Due to a dramatic technological breakthrough a research unit has determined that the time to execute floating point and integer ALU instructions can be halved. However, the cost to this new approach requires a 50% increase in the time to execute memory operations. Assess the impact of this breakthrough on the individual benchmarks and on the two mixtures. [15 points]
5. Another breakthrough, independent of the above one, is reported for all add operations (floating and integer) where the speed of just those operations has been increased by a certain factor. Unfortunately due to errant internal mail delivery problem the CEO of the company does not have the exact factor of improvement. Since the CEO requires an estimate of the maximal benefit of such an operation he asks his technical wizard, you, to outline a best-case-worst-case scenario based on the two mixtures. In the worst-case assume no improvement (but assume that it does not do any worse than the existing system). [15 points]
6. It is determined that three of the PISA-big addressing modes, offset global pointer, offset stack pointer and offset frame pointer can be replaced. Instructions using special purpose registers containing the frame pointer, stack pointer and global data pointer can be replaced by other instructions using a more intelligent compiler. In 80% of the cases the replacement can be with a single displacement addressing mode instruction (basically the frame, stack, and global data pointers are in a general purpose register). In 20% of the cases one needs two instructions (one to load the corresponding pointer, the second to access the memory reference via displacement mode). The additional space gained by doing this speeds up the average time to execute an instruction from 1 cycle per instruction to 0.90 cycles per instruction.

Being the company tech-wiz it is your job to determine which configuration is faster for each of the two benchmark mixtures. [20 points]

The following materials need to be turned in hardcopy.

1. Listing of the simulator runs (one for each benchmark program).
2. Detailed work as part of the solutions for above questions.