The SP.4 modelling assignment

Paul J. Drongowski (Instructor)
Computer Engineering and Science
Case Western Reserve University
ECMP 424 Advanced Computer Architecture
Spring 1990

Introduction.

This assignment is a practical experience in the modelling and measurement of an instruction set architecture (ISA.) It is not intended to be a "no brainer." The informal specification, such that it is, is ambiguous, sometimes incomplete and sometimes erroneous. This is not meant to torment you, but you should view these situations as opportunities to improve upon the architecture and its description. For example, the subroutine return behavior can (and should) be modified to make coding easier and to make the return behavior more consistent with the Polish postfix evaluation of SP.4 programs.

The job.

Your job is to construct a C language program which simulates and instruments the operation of the SP.4. The act of writing the C language simulator will force you to tighten-up the informal SP.4 description.

Once you have written the SP.4 simulator in C, you must write a "benchmark" program for the SP.4. This benchmark program should multiply two 10 by 10 integer matrices together. You must have at least two subroutines – one for the matrix multiplication itself and one for the inner loop of the matrix multiplication. You are free to choose the values for the two matrices. (If one of the matrices is the identify matrix, then check-out should be simplified!) The becnhmark and matrices may be compiled into the simulator image as we will discuss in class.

Instrumentation.

Three quantities are of particular interest and should be measured during simulation.

1. **Dynamic instruction frequency.** Your simulation should measure the dynamic frequency with which SP.4 instructions are executed, e.g., how many times was *Add* executed, etc. You will need two tables here showing the dynamic frequency of operand and descriptor calls, and stack operations, and the frequency of individual stack operations. This information will help designers to select certain critical operations for "optimization."

2. **Stack depth.** The stack size must eventually be fixed. The minimum, maximum and average stack depth plus standard deviation must be computed. It is useful to know how much performance can be "bought" at the cost of additional register space and power.

3. **Locality of reference.** Locality statistics (for instructions and data separately) will help the engineers to design the instruction and data caches for the SP.4. Descriptive statistics and a histogram on the relative distance between instruction fetches and memory data accesses would be helpful here. If one can break out collection on Program Reference Table accesses, it would also be possible to size a translation look-aside buffer (TLB.)

The statistics should form the basis for your discussion of the ISA simulation project and conclusions. (Please read on.)

Discussion.

You should include a brief report (roughly 2 to 3 pages) which summarizes the empirical results obtained from the simulation. You should address the machine design issues outlined above (e.g., critical operations, stack and cache sizing.)