LSU EE 4720 Homework 1 Due: 11 February 2005

Problem 1: POWER is an IBM ISA developed for engineering workstations, PowerPC is an ISA developed by IBM, Apple, and Motorola for personal computers and is based on POWER. POWER and PowerPC have instructions in common but each has instructions the other lacks (and some of the common instructions behave differently). Therefore a POWER implementation could not run every PowerPC program and vice versa.

(*a*) Show the gcc 3.4.3 compiler switches used to compile code for a POWER implementation. *Hint: Google is your friend, look for gcc documentation.*

(*b*) Show the gcc 3.4.3 compiler switches used to compile code for a PowerPC implementation.

(*c*) Is it possible to use gcc 3.4.3 to compile a program that will run on both? If yes, show the switches.

Problem 2: From the SPEC Web site, `http://www.spec.org`, find the fastest result on the SPECFP2000 (that's FP, not INT) benchmark for each of the following implementations: IBM POWER5, Intel Itanium2, Intel Pentium 4, Fujitsu SPARC64 v, and AMD FX-55. (Use the configurable search form and have it display the processor name.)

(*a*) The non-IA-32 implementations (POWER5, Itanium2, and SPARC64 V) blow away the IA-32 implementations on one benchmark. Which one? Which company (of those listed above) would want that benchmark removed?

(*b*) The POWER5 can decode five instructions per clock, the Itanium 2 can decode six instructions per clock, the Pentium 4 and FX-55 each can decode three (what are essentially) instructions per clock, and the SPARC64 V can decode four per clock. Based on the SPECFP2000 results used in the first part, which processor is making best use of these decode opportunities? In other words, if one processor could decode 10^{12} instructions during execution of the suite and another could decode 5×10^{12} instructions during execution of the suite, the first would be more efficient since it ran the suite using fewer instructions. (See last semester's Homework 1 for a similar problem.)

Problem 3: As pointed out in class a processor's CPI varies depending on the program being executed. For the questions below write a program in MIPS assembler (see `http://www.ece.lsu.edu/ee4720/mips32v2.pdf` for a list of instructions), some other assembly language, or assembly pseudocode, as requested below.

(*a*) Write a program that might be used to determine the minimum possible CPI. Suppose you actually used the program to determine the minimum CPI on processor X. How would the CPI be computed? Show an example using made up numbers based on your program an hypothetical processor X. Explain why the result would be the minimum CPI (or close to it).

(*b*) Write a program that might be used to determine the maximum possible CPIand as with the previous part, show how CPI is computed. Your answer should include information about instructions in processor X used in your program. Explain why the result would be the maximum CPI (or close to it).