LSU EE 4720 **Homework 1** Solution **Due: 26 September 2005**

Problem 1: Suppose the *base* and *result (peak)* SPEC CINT2000 benchmark scores were identical on company X's new processor. Make up an advertising slogan based on the fact that they were identical. A catchy tune is optional.

Performance you don't have to work for!

Meaning you don't need to spend hours trying out different compiler optimization options to get the advertised performance.

Grading Notes: Many answered that "extreme optimizations were not necessary," which is not correct (or at least misleading). As used in class "extreme" referred to the effort by the *programmer* to get the compiler to produce the fastest code. Many programmers will not make an extreme effort at optimization and so the performance they see might be along the lines of the base number, which is fine on systems in which it's the same as peak. In a system with identical base and peak numbers the compiler may well be doing optimizations that could be described as extreme, however since the compiler can do them when given only basic optimization flags their benefits are be seen in the base numbers.

Problem 2: According to the CPU performance equation increasing the clock frequency (ϕ) by a factor of x without changing instruction count (IC) or cycles per instruction start (CPI) will reduce execution time by a factor of x. Find two SPEC CINT2000 disclosures (benchmark results) that provide good evidence for this.

(a) Give the CPU, clock frequency, and the base and result CINT2000 scores.

To show the effect of clock frequency pick two systems which are close to identical in every way except clock frequency. For example, the 2.16 GHz and 1.87 GHz Fujitsu SPARC64 V chips:

`http://www.spec.org/osg/cpu2000/results/res2005q2/cpu2000-20050419-04024.html`
`http://www.spec.org/osg/cpu2000/results/res2005q1/cpu2000-20050208-03825.html`

The 1.87 system scores 1594 peak and 1456 base, the 1.87 GHz system scores 1341 peak and 1254 base.

Other than clock frequency the major differences between the systems are in the L2 cache size (3 MiB v. 4 MiB) and the number of CPUs the system can handle (2 v. 16, though the systems tested each had one).

(b) Explain why for these disclosures ϕ is different (obvious) but IC and CPI are probably the same (requires some thinking). It may not be possible to determine this for certain and it may not be possible to find a pair for which they are exactly the same, it's sufficient to find a pair in which they are arguably close.

The clock frequency is listed in the disclosure and they are different. The IC is probably the same because the same compiler was used, Fujitsu Parallelnavi 2.3 with Sun Studio 9. The CPI is probably the same because the compiled code is the same (for reasons just given) and because the code is running on chips of the same microarchitecture (that's assumed because of the same processor name).

(c) Based on the assumption of IC and CPI equality, show how closely the CPU performance equation predicts the performance of one of the systems. Suggest reasons for any difference.

If clock frequency were the only differing factor then based on the 1341 peak performance of the 1.87 GHz chip one would predict a peak performance of $\frac{2.16\,\text{GHz}}{1.87\,\text{GHz}}1341 = 1549$ on the 2.16 GHz chip. The performance is actually higher, a possible reason might be the larger L2 cache.

Problem 3: In section 1.2 of the SPEC CPU 2000 run and reporting rules, `http://www.spec.org/cpu2000/docs/runrules.html`, there is a bullet item that states, "The vendor encourages the implementation for general use." Explain what that means and why it is there. Why would it be bad if the "implementation" were not "for general use."

In the disclosure "implementation" refers to a tool or library used to build (compile, etc) the benchmarks. (It might also refer to the hardware but most of that section talks about compilers and related tools.) The statement says that SPEC expects that anything used to build the benchmark should be a product of the company (or offered by others) and

that the company should make a serious attempt to sell it. It would be bad if the "implementation" were not for general use because that might mean it was too unreliable for customer use and so the benchmark scores achieved using it are higher than a typical user, who would avoid unreliable products, could expect to achieve.

Another danger is that a compiler could be too benchmark specific. In an extreme case an assembly language programmer could hand-code parts of the benchmarks and the compiler would insert that code wherever it recognized the benchmark. That would only work on those specific programs (even minor modifications to the benchmark would render such an optimization useless) and would not work on other code.