

**Problem 1:** Section 2.2.2 of the run and reporting rules for SPECcpu2006, <http://www.spec.org/cpu2006/Docs/runrules.html>, specifies that the optimization flags and options used to obtain the base result must be the same for each benchmark (compiled with the same language, say C). Why must they be the same?

Base scores are supposed to reflect the amount of optimization effort a programmer would make under ordinary circumstances. In that case the programmer might not find a best set of optimization flags for each program. Even if they did, there is no workable way to write a rule that would distinguish between compiler switches found with ordinary and extraordinary effort, and so requiring switches to be the same on all benchmarks is a reasonable substitute.

**Problem 2:** Section 1.2.3 of the run and reporting rules for SPECcpu2006, <http://www.spec.org/cpu2006/Docs/runrules.html>, assumes that the tester is honest. Provide an argument that many of the run and reporting rules ignore this assumption, or at best are based on the assumption that the tester is honest but sloppy or unmotivated.

The rules are written to ensure that anyone (with the budget, skills, and time) can reproduce any disclosed result. So the rules don't assume the tester is honest, they assume that the tester doesn't want to be caught lying.

**Problem 3:** Find the SPECcpu2000 CINT2000 disclosure for the fastest systems using each of the chips below. All chips implement some form or superset of IA-32 (also known as 80x86). All of the implementations are superscalar, meaning they can sustain execution of more than one instruction per cycle. In particular, an  $n$ -way superscalar processor can sustain execution of  $n$  instructions per cycle on ideal code, on real code the sustained execution rate is much lower (for reasons to be covered later in the course, such as cache misses). Some of the implementations are multi-cored. (A core is an entire processor and so a 2-core chip has two complete processors.)

- Pentium III, 1-core, 2-way
- Pentium 4, 1-core, 3-way
- Pentium Extreme, 2-core, 3-way
- Intel Core 2 Extreme X6800, 2-core, 4-way
- Opteron 256, 1-core, 3-way
- Athlon FX-62, 2-core, 3-way

(a) For each system list the following information:

- The peak (result) ratio (for the suite).
- The clock frequency.
- The gcc peak (result) run time (in seconds).
- The maximum number of instructions the system could have executed during the run of gcc assuming all cores were used.
- The maximum number of instructions the system could have executed during the run of gcc assuming one core was used.
- Execution efficiency assuming all cores were used: number of instructions executed divided by maximum number of instructions that could have been executed in the same amount of time. Assume that all systems run the same binary (executable) of gcc and make a guess at how many instructions would be executed when running the binary for the SPEC inputs.

- Execution efficiency assuming a single core was used: Same as previous value, except assume only one core used.

The information is listed in the table below. Those viewing this with Adobe Reader can view the SPECint2000 disclosure by clicking the names in the *Chip* column.

The *Cores* column shows the number of cores on the tested system, the *Width* column shows the decode width per core (the  $n$  in  $n$ -way superscalar), the *Peak* column shows the SPECint2000 result (peak) ratio, the *Clock* column shows the clock frequency, and the *gcc time* column shows the execution time of the gcc benchmark. Values for the columns mentioned so far are found in the SPEC disclosure or in this assignment.

The next two columns, both headed *Max Insn*, show the maximum number of instructions that the respective processor could have executed in the time need for the run of the gcc benchmark using all cores and one core. Those are found by multiplying the maximum number of instructions per second for the processor by the benchmark run time for each case; the formulæ are shown below the column heading.

To find the efficiency one must estimate the number of instructions executed in a run of the benchmark, call that number  $I$ . It can't be larger than any of the entries under *Max Insn*, and realistically will be much smaller (due to stalls and squashes). Since there is no way to tell exactly what  $I$  is with what is given here,  $I$  will be set to the minimum *Max Insn* of the single-core cases (because multiple cores don't help gcc). That is  $I = 366 \times 10^9$  instructions based on the Core 2 X6800 at 2.933 GHz.

The efficiencies are computed by dividing  $I$  by *Max Insn*. Using this method the efficiency of the Core 2 X6800 is 1, meaning only that the X6800 is the most efficient, not that it's perfect.

Chip (Clickable in PDF)	Cores $c$	Width $n$	Peak	Clock $\phi$	gcc time $t$	All Cores		One Core	
						Max Insn $cn\phi t 10^{-9}$	Eff $\frac{I}{cn\phi t}$	Max Insn $n\phi t 10^{-9}$	Eff $\frac{I}{n\phi t}$
Pentium III	1-core	2-way	665	1.400 GHz	154 s	431	.849	431	.849
Pentium 4	1-core	3-way	1863	3.800 GHz	49.2 s	561	.653	561	.653
Pentium Extreme	2-core	3-way	1872	3.733 GHz	50.8 s	1138	.322	569	.643
Core 2 Extreme X6800	2-core	4-way	3119	2.933 GHz	31.2 s	732	.500	366	1.0
Opteron 256	1-core	3-way	2009	3.000 GHz	49.4 s	445	.823	445	.823
Athlon FX-62	2-core	3-way	2061	2.800 GHz	50.9 s	855	.428	428	.856

(b) The execution efficiency computation was based on the assumption that the number of executed instructions was the same in all systems. Identify two systems for which this was more likely to be true and two systems where this was less likely to be true.

In all cases gcc was compiled using the same source code and run using the same inputs, as provided by SPEC. What differs is the compilers (and other build tools) used to test each system, as well as how those compilers were used. The number of instructions are most likely identical where the compiler and options are most similar.

The Pentium Extreme and Core 2 X6800 systems both use the Intel Compiler 9.1 and MicroQuill SmartHeap Library 8.0. These are the exact same compiler and heap (malloc and friends) libraries. The compile flags for gcc are: `-fast shlw32m-80.lib PASS1=-Qprof gen PASS2=-Qprof use` for the Pentium Extreme and `-fast shlw32m.lib PASS1=-Qprof gen PASS2=-Qprof use` for the Core 2 X6800. These are as similar as one could expect, the factor that might nevertheless result in different instruction counts are how the compiler will respond to the `-fast` flag. That tells the compiler to emit the fastest code for the host system, since the two chips are different that might result in different instruction counts.

The number of instructions are less likely identical when the compilers and options are less similar. The Pentium III system uses older version of the Intel compiler, 5.0, and heap library, 5.0, so the generated code is more likely to differ.

(c) How much does a dual-core implementation improve the performance of gcc?

Not much, based on a comparison of the single-core Pentium 4 and the dual-core Pentium Extreme. The performance is nearly the same. (The microarchitectures of the two processors' cores are very similar, so it's not just that each core of the Pentium Extreme is half the speed of the single core of the Pentium 4.)