LSU EE 4720 **Homework 1** Solution **Due: 15 September 2004**

Problem 1: Select two pairs of disclosures (that's four total) from the CPU2000 benchmark results posted at `www.spec.org`. A pair should be for machines using the same ISA but having different implementations. Make the implementations as different as possible. Explain why you think the implementations are very different.

Some ISAs and implementations are listed in lecture set 1, but the solution is not restricted to those. Feel free to ask if you're not sure what ISA a processor implements or whether two ISAs are considered the same or different.

For each disclosure list: the ISA, the implementation, the peak (result) performance, and file name of the HTML-formatted disclosure.

First pair: ISA IA-32, implementations:

Xeon, Peak 1402, file res2004q3/cpu2000-20040727-03291.pdf and Athlon, Peak 1395, results/res2003q3/cpu2000-20030908-02502.pdf.

The two implementations are very different because they come from two different companies (and one was not simply licensing the processor design from another). One obvious evidence of difference is that the two achieve similar performance though having vastly different clock frequencies.

Second pair: ISA SPARC V9, implementations:

SPARC64 V, Fujitsu, Peak 1345, 1.89 GHz, res2004q2/cpu2000-20040518-03044.pdf and UltraSPARC III Cu, Sun Microsystems, Peek 722, 1.2 GHz, `http://www.spec.org/cpu2000/results/res2003q2/cpu2000-20030326-01999.html`

The two are from different companies. Some evidence of their difference is that the performance of the SPARC64 is faster than what one would expect by scaling clock frequency, so something about the systems other than clock frequency must be different. (The material has not been covered yet, but the Sun chip is statically scheduled while the Fujitsu chip uses a more advanced dynamically scheduled organization.)

Problem 2: The processors below have roughly the same SPEC CINT2000 peak (result) scores but are very different. (The links should be clickable in Acrobat Reader.)

ISA: Power, Implementation: POWER5, Decode: 5-way Superscalar*
Disclosure: `http://www.spec.org/osg/cpu2000/results/res2004q3/cpu2000-20040804-03314.pdf`

ISA: Itanium (IA-64), Implementation: Itanium 2, Decode: 6-way Superscalar*
Disclosure: `http://www.spec.org/osg/cpu2000/results/res2004q1/cpu2000-20040126-02775.pdf`

ISA: IA-32, Implementation: Xeon, Decode: 3-way Superscalar*
Disclosure: `http://www.spec.org/osg/cpu2000/results/res2004q3/cpu2000-20040727-03291.pdf`

ISA: ≈IA-32, Implementation: Athlon, Decode: 3-way Superscalar*
Disclosure: `http://www.spec.org/osg/cpu2000/results/res2003q3/cpu2000-20030908-02502.pdf`

ISA: SPARC V9, Implementation: SPARC64 V, Decode: 4-way Superscalar*
Disclosure: `http://www.spec.org/osg/cpu2000/results/res2004q2/cpu2000-20040518-03044.pdf`

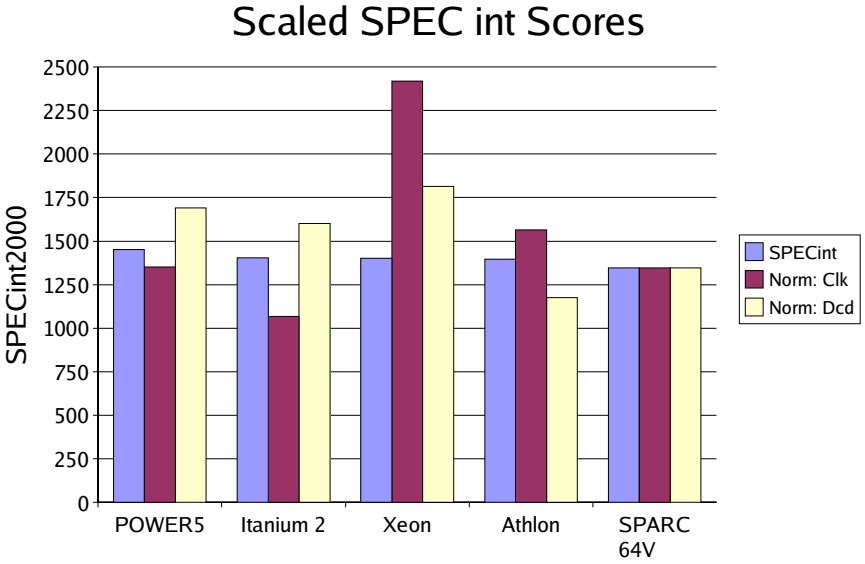
(*a*) The performance of the processors, based on the peak result, are roughly the same. On the same graph plot the performance in the following ways:

- Using the SPEC peak (result) scores.
- Assume that performance is proportional to clock frequency. Determine the score of a processor by comparing its clock frequency to that of the SPARC64 and using that to scale the SPARC64 peak result.
- The table above shows how many instructions a processor can decode per cycle. (Four-way superscalar means four per cycle, see explanation below.) Determine the performance by comparing the number of instructions fetched per second to the SPARC64 and use that to scale the SPARC64 peak result.

What conclusions can be drawn from the plotted data?

**The following information is not needed to solve this assignment. The decode widths shown above are the maximum number of instructions that can be decoded per cycle. For any real program the number will be much lower due to a variety of factors, which will be covered later in the semester. One relatively minor factor is the instruction mix. The POWER5 (implementation), Itanium (ISA), and to a lesser extent the others limit the kinds of instructions that can be decoded together. More on this later in the semester. The decode widths for the Xeon and Athlon don't refer to IA-32 instructions: these processors take IA-32 instructions and break them into simpler instructions called micro-ops by Intel and (favoring marketing over descriptive accuracy) macro-ops by AMD. The three instructions per cycle for the Xeon and Athlon are actually three micro- or macro-ops per cycle.*

The "Norm Clk" plot shows performance scaled using clock frequency. The value is $s'_X = s_{\text{sparc}} \frac{\phi_X}{\phi_{\text{sparc}}}$, where s'_X is the scaled performance of system X and ϕ_X is the clock frequency of system X. The "Norm Dcd" plot shows the performance scaled using decodes per second (not the same as decodes per cycle). The value is $s''_X = s_{\text{sparc}} \frac{\phi_X d_X}{\phi_{\text{sparc}} d_{\text{sparc}}}$, where s''_X is the scaled performance of system X and d_X is the decode width (decodes per cycle) of system X.



Conclusions:

The obvious conclusion is that clock frequency is a poor indicator of performance since the predictions based on clock frequency are way off. Taking into account the number of instructions decoded per cycle is a better predictor of performance (than just clock frequency) but still far from perfect.

Taking into account decode rate, the performance of the Athlon system is underpredicted, whereas the others are overpredicted. This hints that the Athlon makes the best use of each instruction decode slot. The worst use of slots is made by the Xeon. Note that there are many other differences, for example, the number of instructions in a program, and the way the IA-32 processors split instructions, so that one can't conclude for sure that the Athlon is making the best use of its slots.

(*b*) The Xeon and Athlon systems in the disclosures above have about the same performance. AMD might argue that those disclosures don't show the full potential of the Athlon. Find a system that uses an Athlon and scores much better, and explain what accounts for the difference.

AMD might argue that the gcc compiler used in the test system does not make the best use of the processor, and so the tested system does not show the full potential. The AMD system using a ASUS SK8N motherboard uses an identical Athlon FX-51 processor, but scores a higher 1447. The difference is probably due to the compiler. The higher-scoring system uses an Intel IA-32 compiler, the lower-scoring system uses gcc which is not known for speed. (What it lacks in speed it makes up for in portability.)

(*c*) There is a system characteristic that affects the performance of benchmark mcf. What is it?

Cache size. The systems with the larger cache size do much better.

(*d*) Nominate a disclosure for The Most Desperate Peak Tuning award.

Of the five above I nominate the Primepower 650 for the award, because of the longest list of compiler flags.