LSU EE 4720 **Homework 1** Solution **Due: 17 September 2003**

Problem 1: Look at the following SPEC CINT2000 disclosures for these Dell and HP Itanium 2 systems:
HP: `http://www.spec.org/osg/cpu2000/results/res2003q3/cpu2000-20030711-02389.html`
Dell: `http://www.spec.org/osg/cpu2000/results/res2003q3/cpu2000-20030701-02367.html`. (Note: Links are clickable within Acrobat reader.)

The CPU performance equation decomposes execution time into three components, clock frequency, ϕ, instruction count, IC, and CPI. For each component determine if its value on the two systems is definitely the same, probably the same, probably different, definitely different. *Hint: The answer for the clock frequency is easy, the others require a little understanding of what IC and CPI are.* Briefly justify your answers.

Clock frequency: The same, the clock frequency is given in the disclosures.

Instruction count (IC): definitely different because the benchmarks were compiled with different compilers.

CPI: probably different. The HP system is faster, some of that difference may be due to the compiler and some may be due to the system (amount of memory, speed of memory bus, etc.). Both system details (such as memory speed) and compiler details (instruction mix and scheduling) affect CPI and since both are different on the two systems the only way CPI could be identical would be by coincidence. (If the CPIs' were identical then the HP systems's better performance would be due solely to the compiler.)

Problem 2: Though one may normally think of an implementation as a microprocessor chip, the definition can also include other parts of the system, such as memory and even disk. Why is that important in the problem above?

Because the two systems used the same implementation, a 1.5 GHz Itanium 2, but other parts of the system, such as memory and disk, differed and that could lead to different execution times (and CPIs).

Problem 3: Differences in ISA, compiler, and implementation all affect the execution time of programs, and the impact of these factors can vary from program to program. For example, an implementation with faster floating point will have a larger impact on programs that do more floating-point computations.

From a look at the SPEC disclosures one can see that the fastest program on one system may not be the fastest program on another. (Use the int2000 results. From the spec CPU2000 results page find the configurable query form and request a page sorted by "Result" in descending order. It would be helpful to include the processor and compiler in the results. If your system is slow omit results before 2002.) For example, the Dell system from the first problem ran vortex fastest (of all the benchmarks), while the HP system ran mcf fastest. In this case the difference in fastest benchmark could not be the ISA, but it could be the compiler or the implementation. Call the speed ranking of benchmarks for a system its *character*. The character of the Dell system is vortex, gcc, eon, ... (benchmarks from fastest to slowest) and the character of the HP system is mcf, vortex, gcc, ...

The differences in character are due in part to the ISA, compiler, and implementation. Using the SPEC CINT2000 disclosures determine which is most important in determining character. Please do not try to look at all disclosures, just enough to determine an answer, even if that answer might change if you were to look at more.

In your answer, state which (ISA, compiler, or implementation) is most important, which disclosures you looked at, and how you drew that conclusion. *This question is easy to answer (once it's understood).*

As best you can explain why a particular factor is most important and why it is least important. *You are not expected to answer this question very well, most of the material has not been covered yet. Don't take too much time and do your best with what has been covered and what you already know.*

Additional Information:

Here are some ISAs and implementations of processors listed:

IA-32 ISA: (includes variations) implemented by Pentium 4 (and other Pentia) Xeon, Athlon, Opteron. Power Architecture ISA: Implementated by POWER4, RS64IV. Itanium ISA: Implementated by Itanium 2. Alpha ISA: Implementated by Alpha 21164, 21264, 21364. SPARC V9 ISA: Implementated by SPARC64V, UltraSPARC III Cu MIPS ISA: Implementated by R14000

The solution compares the characters of pairs of disclosures, one pair to bring out ISA effects, a pair for compiler effects, and a pair for implementation effects. When selecting a pair to compare to systems are found which differ on the thing being looked at (ISA, compiler, or implementation) but are as similar as possible with everything else. Once a pair is found the ranking for the integer benchmarks in each system is found.

To show how different the character is, the sum of the absolute difference in ranks of the first five programs for one system to the other is found. (This is probably easier then it sounds.) For example, when looking at ISA effects, the ranking of the first three Xeon programs is the same as in the Itanium system. The fourth program for Xeon is ranked 12th for Itanium, so the distance is 8. The fifth Xeon program is ranked eighth in the Itanium system, for a difference of 3. The sum of differences is 11.

Note: To draw any real conclusions from this analysis one would need to look at more than three pairs of systems, which is too much work for one homework assignment.

ISA Effects. Compare two systems with different ISAs but otherwise as similar as possible. One system below implements Itanium 2, the other IA-32. They both use the Intel Intel C/C++ compiler.

Dell PowerEdge 3250, Itanium 2:
`http://www.spec.org/osg/cpu2000/results/res2003q3/cpu2000-20030701-02367.html`.
Intel Xeon (3.06GHz, 533MHz bus)
`http://www.spec.org/osg/cpu2000/results/res2003q3/cpu2000-20030630-02338.html`

```
Program Ranks: (First is fastest.)
Xeon:     vortex, gcc, eon, gap, perlbmk, twolf, gzip, crafty, parser, bzip2, vpr, mcf
Itanium:  vortex, gcc, eon, mcf, crafty,  twolf, bzip2, perlbmk, vpr, gzip, parser, gap
Rank Diffs: 0      0    0    8    3 = 11
```

The fastest three programs in the two systems (above) are the same, but the order of the other programs is very different.

Compiler affects: Look at two systems with the same implementations (and of course the same ISA), but different compilers. A possible pair are the two systems used in Problem 2 (the difference in memory and IO are still there, systems with identical hardware would be better, but even so there would be differences due to the operating system).

Both systems use the same implementation of Itanium, Itanium 2. Differences: compiler, also operating system and hardware (other than Itanium 2). (In a better comparison the same operating system would be used. However it seems that for most (maybe all) systems, only one compiler is used for a particular operating system [gcc for Linux, Intel for Windows, etc]).

```
Top programs on HP (Unix): mcf,    vortex, gcc, eon, twolf
Top programs on Dell:      vortex, gcc,    eon, mcf, crafty, twolf
Rank Differences:          3       1       1    3    1  =  9
```

Implementation Effects. Compare two systems with the same ISA but different implementations. Two implementations from the same manufacturer may be similar, but one can be pretty certain that Intel's lawyers made sure the AMD Athlon had little in common with Intel's Xeon (similar to the Pentium). Therefore, will compare Xeon and Athlon.

AMD Athlon FX-51, 2.2 GHz,
`http://www.spec.org/osg/cpu2000/results/res2003q3/cpu2000-20030908-02472.html`
Intel Xeon (3.06GHz, 533MHz bus)
`http://www.spec.org/osg/cpu2000/results/res2003q3/cpu2000-20030630-02338.html`

Common IA-32 ISA (with small variations), Intel C/C++ 7.0 compiler (and libraries). Differences: Implementation. (The Athlon FX actually implements a 64-bit extension of IA-32, however it's unlikely that the Intel compiler is emitting any AMD 64-bit instructions.)

```
Top programs on Athlon: vortex, eon, twolf, gcc, perlbmk
Top programs on Xeon:   vortex, gcc, eon,   gap, perlbmk, twolf, gzip
Distance:               0       1    3      2    0   = 6
```

Using just this data, it would appear that ISA is most important for character, followed by compiler, then implementation. As stated before, there are way too few data points to draw any real conclusion.

Problem 4: The two procedures below are compiled with optimization on but with no special optimization options. Why might the second one run faster? (This is very similar to the classroom example.)

```
void add_array_first_one(int *a, int *b, int *x)
{
  for( i = 0;  i < 100;  i++  ) a[i] = b[i] + *x;
}
void add_array_second_one(int *a, int *b, int *x)
{
  int xval = *x;

  for( i = 0;  i < 100;  i++  ) a[i] = b[i] + xval;
}
```

First program has to de-reference `x` (load value from memory) each iteration while the second program can load the value into a register before entering the loop, then just use the value in the register.