**LSU EE 4720 Homework 1 Due: 10 February 2003**

*At the time this was assigned computer accounts and solution templates were not ready. If they become available they can be used for the solution, either way a paper submission is acceptable.*

**Problem 1:** When compiling code to be distributed widely one should be conservative when selecting the target ISA but less caution needs to be taken with the target implementation. Explain what "conservative" and less caution mean here, and explain why conservatism and in one case less caution in the other can be taken.

**Problem 2:** Based on the SPECINT2000 results for the fastest Pentium and the fastest Alpha, which programs would a shameless and unfair Alpha advocate choose if the number of programs in the suite were being reduced to five. Justify your answer.

**Problem 3:** The Pentium 4 can execute at a maximum rate of three instructions (actually, microops, but pretend they're instructions) per cycle (IPC), the Alpha 21264 can execute at most 4 IPC and the Itanium 2 can execute at most 6 IPC. Assume that the number of instructions for perlbmk, one of the SPECINT2000 programs, is the same for the Alpha, Itanium 2, and Pentium 4 (pretending micro-ops are instructions, if you happen to know what micro-ops are).

(*a*) Based upon the SPECINT2000 results (not base) for the perlbmk benchmark, which processor comes closest to executing instructions at its maximum rate? ("Its", not "the".)

(*b*) Are these numbers consistent with the expected tradeoffs for increasing clock frequency (mentioned in class) and for increasing the number of instructions that can be started per cycle?

**Problem 4:** Complete the `lookup` routine below so that it counts the number of times an integer appears in an array of 32-bit integers. Register `$a0` holds the address of the first array element, `$a1` holds the number of elements in the array, and `$a2` holds the integer to look for. The return value should be written into `$v0`.

```
lookup:
        # Call Arguments
        #
        # $a0: Address of first element of array.  Array holds 32-bit integers.
        # $a1: Number of elements in array.
        # $a2: Element to count.
        #
        # Return Value
        #
        # $v0: Number of times $a2 appears in the array starting at $a0

        # [ ] Fill as many delay slots as possible.
        # [ ] Avoid using too many instructions.
        # [ ] Avoid obviously unnecessary instructions.

        # A correct solution uses 11 instructions, including 6 in
        # the loop body.  A different number of instructions can be used.

        # Solution Starts Here




        # Use the two lines to return, fill the delay slot if possible.
        jr $ra
        nop
```