Computer Engineering 431
Computer Architecture
Fall 2015

Homework #1

Distributed: August 26, 2015
Due: September 3, 2015 via Angel by 11:55pm EDT
Points: 15

1. (2 pts) Computer A executes the MIPS ISA (a RISC machine) and computer B executes the x86 IS (a CISC machine). On average, programs execute 1.5 times as many MIPS instructions as x86 instructions. Computer A has an average CPI of 1.5 and computer B an average CPI of 3. If computer B runs at a 2GHz clock frequency, what speed does computer A have to run at to be at least as fast as computer B?

2. (5 pts) The following measurements have been made using a simulator for a machine design that is projected to have a clock rate of 1GHz (1nsec).

Instruction Class	Cycles/Instr Class (CPI)	Frequency of Instr Class
R-types	1	45%
Loads	4	25%
Branches	2	20%
Stores	3	10%

(a) What is the machine's CPI?
(b) How much faster would the machine be if a better data cache reduced the load CPI by one cycle?
(c) How does this compare to using branch prediction to shave one cycle off the branch CPI?
(d) An optimized version of the machine design has been done which doubles the projected clock rate to 2GHz (500psec) but also doubles the CPI of each instruction class. Which is the faster design, the 1GHz version or the 2GHz version, and by how much?

3. (4 pts) Suppose a program segment consists of a purely sequential part which takes 10 cycles to execute and a code loop part, shown below, which can be distributed across multiple cores for improved performance. The CPI for each instruction in the loop is the same as for the 1GHz machine design in problem 2.

```
lp:   lw   $t0, 0($s1)
      add  $t0, $t0, $s2
      sw   $t0, 0($s3)
      addi $s1, $s1, -4
      addi $s3, $s3, 4
      bne  $s1, $0, lp
```

   If the loop is to be executed 10 times, what is the maximum possible speedup if there are 100 cores available (compared to a single core)? What is the maximum possible speedup if the loop is to be executed 100 times?

4. (4 pts, short essay (no more than 500 words)) As discussed in class, Moore's Law (the doubling of the number of transistors that can be put on one chip every 2 years) has held for over 50 years and has contributed significantly to the amazing improvement in the performance of computers during that period. Assume that, using current technology (CMOS), Moore's Law will stop five years from now (so in 2020) when we reach 10nm transistors. Assuming that the next generation technology which will allow further scaling is still twenty years out and that people will still expect/demand the sort of performance improvements that they have seen over the last 50 years, what can computer architects do to help and what will be the primary design challenges? How about software developers?