CSE 240A Winter 2012 Assignment 1

Assigned Tuesday 1/17
Due Tuesday 1/24 (In class at start of lecture)
Homework should be typed. Do not forget to put your name.

1. Program A runs 20 billion instructions on a 2 GHz processor, and achieves a CPI of 1.5. Introduction of a new instruction to the ISA (and recompiling the code) would allow a reduction in the instruction count to 19 billion instructions, resulting in a speedup of 1.2. What is the CPI for the new code on the improved processor?

2. Machine B runs at 2 GHz and has a CPI of 1.3 for a particular program. Machine C runs at 5 GHz and has a CPI of 2.4 for that program, while executing 20% more instructions. Which machine is faster? What is the speedup over the slower machine?

3. You have an optimization that speeds up floating point operations by a factor of 2, but does not help other instructions. You have another optimization that only speeds up non-FP instructions by 10% (that is, it speeds up those instructions by a factor of 1.1), and you want to decide which to use. Suppose your favorite program has 10% of its instructions FP operations. Further, assume FP operations have a CPI of 3.0, while non-FP instructions have a CPI of 1.0.

4. (Amdahl's law backwards) You improve your memory subsystem so that memory latencies are sped up by a factor of 2.4. After applying the optimization, you observe that you now spend half your time on waiting for memory. What percentage of the original execution (before the optimization) was spent waiting for memory?

5. You are trying to design a machine for recognition, mining and synthesis. As popularly defined by Intel (google recognition mining and synthesis if you are interested in Intel’s vision of workloads for 2015), Recognition is a type of machine learning which enables computers to model objects or events of interest to the user or application. Given such a model, the computer must be able to search or mine instances of the model in complex, often massive, static or streaming datasets. Synthesis is discovering “what if” cases of a model. If an instance of the model doesn’t exist, a computer should be able to create it in a virtual world.

   a. Pick a subset of 5 SPEC benchmarks that might be appropriate as suitable benchmarks for this machine. You can find description of SPEC benchmarks at www.spec.org/cpu2006/publications/CPU2006benchmarks.pdf Justify the inclusion of each benchmark you choose in 1 or 2 lines.

   Next you are designing another machine for scientific computations such as those that go on at CERN or national labs.

   b. Pick a subset of 5 SPEC benchmarks that might be appropriate as suitable benchmarks for this machine. Y Justify the inclusion of each benchmark you choose in 1 or 2 lines.

6. Give me the code for \( D = \frac{A + B}{C} \) in an accumulator, stack, GPR, and load-store ISA. For each give me the total size (in bits) of the code, assuming (a) fixed-length instructions, and then (b) variable-length instructions. Assume opcodes are 6 bits, memory addresses are 32 bits, and register specifiers are 4 bits. You'll probably make other assumptions -- just tell me what they are.

7. Suppose we wanted to add the auto-increment addressing mode to the MIPS ISA -- e.g., lw R1, 1000(R2++). This saves an instruction every time we observe a load followed by an increment of the address register. Is this a good idea? Consider only performance, and assume that we have to increase cycle time by 5% to accommodate the new instruction, that 20% of our instructions are loads, that we can apply this change to 40% of all loads, and that the CPI doesn't change.

8. For the following code:
   
   ```
   for (i=0; i<= 100; i++)
   {A[i] = B[i] + C;}
   ```
   
   give the assembly in MIPS code. Assume i is in register R1, that the starting address of Array A is in R2, the address of array B is in R3, that the address of C is in R4. What is the dynamic instruction count of this code?

9. P8: Assume we are executing MIPS code where 15% of instructions are conditional branches, 20% are load instructions, 5% are stores, and the rest are arithmetic. For this code, (a) what percentage of all memory accesses are for data? (b) what percentage of all data accesses are reads? (c) what percentage of all memory accesses are reads? [minor hint -- what do we access memory for besides loads and stores to data memory?]