COMP4300 Spring 2021
Homework 1
Due 11:59pm, Feb 12, 2021

Each problem worth 20 points

1. For a PDP-8, generate assembly code to multiply the number in hex address 0x200 by 4, and store the result in address 0x201. The program should start in address 0x100. You can assume the number in 0x200 is positive and less than 0x100. You will need to consult the PDP-8 programming card for mnemonics and instruction formats. Note in particular that the PDP-8 has no multiply instruction. Be sure to give the address of each instruction.

2. From the following assembly language code for a 32-bit MIPS processor, generate the binary machine language for this code fragment. Consult the Internet, for example:

   http://max.cs.kzoo.edu/cs230/Resources/MIPS/MachineXL/InstructionFormats.html

   for the bit patterns for opcodes and register numbers (5 points/instruction).

```
Start:    LW R1,40(R2)

          LW R3,1000(R0)

          ADDU R1, R2, R3

          J Start:
```

(where Start is at 32-bit address 0x00001000)

** ORDER IS BIG-ENDIAN **

   Tell what bits go in each memory address. Remember that an address holds 8 bits.

3. Suppose a given optimization to the ALU speeds up execution for the system as a whole by a factor of 1.75. After optimization, ALU operations take up 1/6 of the total execution time. What fraction of execution time BEFORE OPTIMIZATION was taken up by ALU operations? What was the speedup factor to ALU operations due to the optimization?

4. For a particular computer, the CPI for certain types of instructions is as follows:

   ALU operations, 2 cycles, make up 25% of dynamic (run-time) instruction count

Load/store operations, 10 cycles, 30% of dynamic instruction count
Control flow, 3 cycles, 20% of dynamic instruction count
All other instructions, 1 cycle

What is the average CPI?
Suppose there is an optimization in which the CPI of load/store is reduced to 5, but cycle time is lengthened by 20%. What is the speedup due to this optimization?

5. Suppose for the problem in question 4, Load/store operations were made to take 1 cycle, without lengthening the cycle time. What would be the speedup due to that optimization?