CIS 451 Single Cycle Performance Homework

Remember: $t_{cpu} = n \cdot p \cdot CPI$ where

- n is the number of instructions in the program,
- p is the CPU's clock period, and
- CPI is cycles per instruction. (For this homework, CPI is always 1.)

The general approach to solving these problems is to

1. determine how the proposed hardware change will affect p. (Call it p'.)
2. determine how the proposed hardware change will affect n. (Call it n'.)
3. Compare the two equations. (In many, but all, cases, this means calculating the conditions under which $n'p' \leq np$.)

Problem from Lecture:

- Consider the single-cycle CPU with a 925ps clock period. Suppose that doubling the size of the register file from 32 to 64 registers would reduce the number of load and store instructions, but would increase the delay of the register file from 150 to 175. What percent of these instructions must be removed in order for the CPU with the larger register file to have better performance that the original CPU? Assume that loads and stores account for 35% of the original workload.

Time for original CPU: $925n$
Time for new CPU: $(925 - 150 + 175)n'$

Divide the instructions into two groups: "memory instructions" (`lw` and `sw`) and "non-memory instructions" (everything else). Every "non-memory" instruction will appear in programs compiled for the new CPU. Only some of the memory instructions will appear; the additional registers will make some memory instructions unnecessary. Suppose the new CPU will retain x of the memory instructions. Then

$$n' = .65n + .35xn$$

.

For the new CPU to have better performance

$$950(.65n + .35xn) \leq 925n$$

This happens when $x \leq .9248$. This means that we can keep at most 92.48% of the loads and stores and must get rid of at least 7.5%.

Updated Fri 11th Jan, 2019 at 12:47

Name: _____

Questions:

1. Consider a CPU. Suppose we were given the option of decreasing the cycle time by 20% in exchange for a 20% increase in the number of instructions. Is this a good tradeoff? (Note: In this problem, both n and p cancel, so you don't need specific values for either.)

2. Your term works with the MIPS CPU presented in the Harris and Harris textbook (with the 925 ps clock period.) You are trying to persuade your hardware design team to remove the offset parameter from load and store instructions. As we discussed in class, removing the offset will require an extra instruction for *some* of the load and store instructions. If you assume that 35% of the instructions are loads and stores, what is the maximum percentage of the loads and stores that can require an extra instruction before your proposed CPU is slower than the current design? (Hint, let x be the percentage of loads and stores that requires an extra instruction when the offset is removed.)

3. Consider the single-cycle CPU with a 925ps clock period. Suppose that doubling the size of the register file from 32 to 64 registers would reduce the number of load and store instructions by 20%, but would increase the delay of the register file from 150 to 175. What is the minimum percentage of instructions that can be loads or stores in order for this new 64-register CPU to perform better than the orignal 32-register CPU?

4. (From Patterson and Hennessy ARM edition) When processor designers consider a possible improvement to the processor datapath, the decision usually depends on the cost/performance trade-off. Assume that 25% of all instructions are loads, and 10% are stores. Also assume the following costs for the MIPS CPU components:

I-Mem	Reg. File	Mux	ALU	Adder	D-Mem	Single Reg.	Sign ext.	Single gate	Control
1000	200	10	100	30	2000	5	100	1	500

Suppose doubling the number of general purpose registers from 32 to 64 would reduce the number of load and store instruction by 12%, but increase the latency of the register file from 150ps to 160ps and double the cost from 200 to 400.

   (a) What is the speedup achieved by adding this improvement?

   (b) Compare the change in performance to the change in cost.

   (c) Given the cost/performance ratios you just calculated, describe a situation where it makes sense to add more registers and describe a situation where it doesnt make sense to add more registers.

Name: _____

5. Problem 2.43 from the Patterson and Hennessy textbook (3rd edition): As you know, MIPS is a "load-store" architecture. In contrast, the Intel architecture allows ALU operations to access memory. In this exercise, you will examine quantitatively the pros and cons of adding an "IA-32 style" addressing mode to MIPS.

   Consider adding a new instruction:

   ```
   addm $t2, 100($t3)  #  $t2 = $t2 + Memory[$t3 + 100]
   ```

   Assume that the new instruction will cause the cycle time to increase by 10%. Use the instruction frequencies given below:

   - Arithmetic / Logic: 42%.
   - Loads: 24%.
   - Stores: 12%
   - Branch and Jump: 22%.

   (these values are for for the SPEC2000int and are given in Figure 2.48). What percentage of loads must be eliminated for a single-cycle machine with the new instruction to have at least the same performance as the single-cycle machine?

6. Consider the following proposed instruction: `incmp` (*inc*rement and co*mp*are). This instruction increments the value of a register before comparing it. In other words, it performs an operation similar to this: `if (++R1 != R2)`. It would be most useful in combining the condition check and increment in a typical `for` loop. (i.e., combining `addi, R1, 1` and `bne R1, R2, TOP_OF_LOOP` into a single `incmp` instruction). Adding this instruction will remove some `addi` instructions at the cost of adding 30ps to the ALU time.

   Given that

   - The current CPU's clock period is 925ps.
   - The current CPU's ALU requires 200ps.
   - Adding `incmp` to the ALU will increase its latency by 30ps.
   - Branch statements comprise 11% of a typical workload.

   (a) What percent of branch instructions must be consolidated with an `addi` into a single `incmp` instruction in order for the new CPU to have better performance than the current CPU?

   (b) Suppose the CPU had a clock period of 725 instead of 925. Would you need to consolidate more or fewer branch instructions to "break even"? Explain how you can answer this question without doing any additional calculations.