Jason Millette

ECE 473 Homework 1
(40 points)
Due: September 26, Thursday

1. (15 points) Several enhancements say, E_1, E_2, …, E_k, are being proposed to improve the performance of a machine. Let FE_i and SE_i denote the *fraction_enhanced* and *speedup_enhanced* for the i^{th} enhancement E_i, i = 1, 2, …, k. Assume that all these enhancements have been implemented in the upgraded machine, but they are usable only one at a time, i.e., they are non-overlapping.

   a. (5 points) Write a generalized expression for *speedup_overall* in terms of FE_i's and SE_i's reflecting Amdahl's Law.

   $$Speedup_i = \frac{1}{1 - FE_i + FE_i/SE_i}$$

   b. (5 points) Three non-overlapping enhancements with the following values of *speedup_enhanced*, have been implemented for a new architecture.

	Speedup_enhanced
Enhancement-1	30
Enhancement-2	20
Enhancement-3	10

   If enhancement 1 and 2 are each usable for 30% of the time, what fraction of the time must enhancement 3 be used to achieve an overall speedup of 10?

   150%

   c. (5 points) Assume for some benchmark, the fraction of use is 15% for each of enhancements 1 and 2, and 70% for enhancement 3. If only one enhancement can be implemented, which one among the three should be chosen in order to maximize performance?

   $SP_1 = \frac{1}{1-.15+\frac{.15}{30}} = 1.17$ $\quad SP_2 = \frac{1}{1-.15+\frac{.15}{20}} = 1.17$

   $SP_3 = \frac{1}{1-.70+\frac{.70}{10}} = 2.70$ $\quad$ 3

2. (10 points) Several researchers have suggested that adding a register-memory addressing mode to a load-store machine might be useful. The idea is to replace sequences of
   Code Sequence

   LOAD R1, 0(Rb)
   ADD R2, R2, R1

   with

   ADD R2, 0(Rb)

   Assume that the new instruction will cause the clock cycle time to increase by 10% and will not affect the CPI. In the SPECint200 benchmark suite, the frequency distribution of different instructions executed in the gcc benchmark is as follows.

load	store	add	sub	mul	compare	load imm	cond branch	cond move
25.1%	13.2%	19%	2.2%	0.1%	6.1%	2.5%	12.1%	0.6%

jump	call	return	shift	and	or	xor	other logic
0.7%	0.6%	0.6%	1.1%	4.6%	8.5%	2.1%	0.4%

What percentage of the loads must be eliminated for the machine with the new instruction to have at least the same performance?

SP = $\frac{}{1 - .226 + \frac{12\%}{10}} = 1.33$

3. (15 points) In the load-store architecture of MIPS, operands of arithmetic and logical instruction must be from registers. For a typical integer program, the instruction distribution and CPI of 4 groups are given in the following table.

Type	Frequency	CPI
ALU	50%	1
Load	25%	2
Store	15%	2
Branch	10%	4

a. (5 points) Calculate the average CPI of the integer program.

1.70 clock CPI

b. (5 points) Now, assume that a set of new memory-register type of arithmetic and logical instructions are added into the ISA. Each memory-register ALU instruction combines one Load and one original ALU instruction together. It takes 4 cycles to execution this new type of instruction. Assume 60% of the load instructions can be combined for the program; calculate the new CPI of the integer program.

85%?

Type	Frequency	CPI
ALU	35%	1
Load	10%	2
Store	15%	2
Branch	10%	4
ALU-load	15%	4

1.85 average CPI

c. (5 points) Assume the modification makes the overall cycle time increased by 5%. Is this modification really worthwhile?

NO it was never worthwile