Computer Architecture and Assembly Language Assignment

Due: On the day of final Exam

- **The main purpose of this assignment is to help you attain an understanding of the course material and to help you prepare for the exams.**

1. You are a computer Architect at company XYZ to investigate the effect on CPU performance when an instruction that immediately follows another instruction depends on the result of the first instruction. For example, a ***"load"*** instruction which loads a value from memory into a register might be followed by an ALU instruction which uses this value from the register. A technique called pipelining overlaps the execution of instructions so that the second instruction starts to execute before the first one is finished.

If the output from the first instruction is not available in time to be used as input to the second instruction then the machine has to stall the execution of the second instruction until the result from the first is available. Assume that in a particular machine, load instructions create 1 clock cycle stalls when the instruction that immediately follows depends on the result of the load (this is called a hazard).

Statistics show that **25%** of the instruction mix are loads and that hazards occur for **50%** of loads. Furthermore, all "branch" instructions create 1 clock stalls, and **10%** of the instruction mix are branches. A combination of architectural and software techniques would make it possible to reduce the stall rate for loads from **50% to 25%**, and reduce the stall rate for branches from **100% to 35%.** Assume all other types of instructions take 1 clock cycle to execute (while loads and branches also take 1 clock cycle when they do not have to stall).

What is the overall speedup that would result from the application of the enhancements? Find the speedup using two different approaches. In the first approach, use the CPU time equation directly. For the second approach, use Ahmdahl's law as a different way to get the same answer.

2. In this problem you are required to answer design questions based on the following information. Certain company has two implementation, P1 and P2, of the same instruction set. There are five classes of instructions (A, B, C, D, and E) in the instruction set. The clock rate and CPI of each class is given below.

Machine	Clock rate	CPI A	CPI B	CPI C	CPI D	CPI E
P1	1.0 GHz	1	1	2	3	2
P2	1.5 GHz	1	2	3	4	3

2.1 The company measures a peak performance as the fastest rate that a computer can execute any instruction sequence. What are the peak performances of P1 and P2 expressed in instructions per second?

2.2 If the number of instructions executed in a certain program is divided equally among the classes of instructions except for class E, which occurs twice as often as each of the others: Which computer is faster? How much faster is it?

2.3 If the number of instructions executed in a certain program is divided equally among the classes of instructions except for class A, which occurs twice as often as each of the others: Which computer is faster? How much faster is it?

3. The table below shows instruction-type breakdown for different programs. Using this data, you will be exploring the performance trade-offs with different changes made to a MIPS processor.

Program	Instructions				
	Compute	Load	Store	Branch	Total
Program 4	1500	300	100	100	1750

3.1 Assuming that computes take 1 cycle, loads and store instructions take 10 cycles, and branches take 3 cycles, find the execution time of this program on a 3 GHz MIPS processor.

3.2 Assuming that computes take 1 cycle, loads and store instructions take 2 cycles, and branches take 3 cycles, find the execution time of this program on a 3 GHz MIPS processor.

3.3 Assuming that computes take 1 cycle, loads and store instructions take 2 cycles, and branches take 3 cycles, what is the speed-up of a program if the number of compute instructions can be reduced by one-half?

4. *Using compiler optimization can greatly improve the execution times of your programs. In this question, you compare the performance of a machine with and without compiler optimization. you use two benchmark programs for this performance comparison.*

(a) Write the following two benchmark programs in C++: ***Program A*** *fills an* ***N x N*** *array of double precision numbers with random values in the range* ***[−π, π]****. Program A then replaces each array entry with its* ***cosine****. Program B fills two* ***N X N*** *arrays in the same way as Program A (i.e., with cosines of random doubles), then transposes one of the arrays. Finally, compute the matrix product of the two arrays in a third array.*

In order to generate random numbers, use both ***srand() and rand()****, as well as* ***RAND_MAX*** *(you can use the current time as the seed).*

The array sizes for both programs will have to be large (N = hundreds to thousands, as explained later), therefore make sure to declare all arrays as global variables. Your programs do not have to be efficient (and don't bother using dynamic memory allocation for the arrays), ***just be correct.*** *Compile each program with* ***Devc++,*** *both with and without* ***optimization:*** *you can change compiler setting in DevC++ the following way,*

alt-P compiler tab, code generation sub-tab then optimization level. select medium that will give the -O2 option. Try out other options for faster or smaller code.

Adjust the array dimension N so the user CPU times are between ***1 and 10 seconds.*** *You may have to choose different values of N for programs A and B in order to get reasonable CPU times.*

(b) Repeat the time measurement ***10*** *times for each program and tabulate the resulting* ***user CPU times*** *and* ***system CPU times*** *(read how to profile a program on Dev c++). Calculate the mean and standard deviations of the CPU times. Please use the format of the sample table given below (or a very similar one) for your tabulation. Make sure to write down the values of N for programs A and B. Also, specify your operating system name and version, and the compiler version used.* ***Which CPU time is more affected by optimization, system CPU time or user CPU time? Why?***

format of the sample table

RUN #	PROG A	PROG A	PROG B	PROG B

	(optimized)		(without Optimization)		(optimized)		(without Optimization)	
	USER	SYSTEM	USER	SYSTEM	USER	SYSTEM	USER	SYSTEM
1	5.6	1.23						
2	5.42	1.31						
.	.	.	.					
.	.	.	.					
10	5.50	1.19						
MEAN	***5.51***	***1.24***						
S.D.	***0.06***	***0.04***						

(c) For this question consider only the user CPU time as a program's execution time. With the tabulated mean of user CPU times, we can now compare the performance of your machine with and without optimization. Calculate the weighted arithmetic mean execution times for the machine with and without optimization. ***What is the speedup in execution time with optimization?***