York University
Dept. of Computer Science and Engineering
CSE4201
HW 1
Due Oct 2, 2007

1. For a given program two compilers produced the following instruction counts: The machine is assumed to run at a clock rate of 100 MHz

	Instruction counts (in millions) for each instruction class		
Code from:	A	B	C
Compiler 1	5	1	1
Compiler 2	10	1	1

With the following CPI per class

Instruction class	CPI
A	1
B	2
C	3

Calculate the MIPS rating, CPI, and total time to execute each program

2. A processor spends 20% of its time in ALU operations, 50% of its time in memory operations, the rest in I/O. There are 2 ways to speedup the processor
   - The first is to speed the memory by a factor of 1.2 and the I/O by a factor of 1.5.
   - Or, we can spend the same amount of money to design a faster ALU that is three times as fast as the original one.

Which design is faster?

3. A program runs in 100 seconds on a machine with multiply operations responsible for 80 seconds of this time. By how much must the speed of multiplication be improved to make the program four times faster? What is the percentage of the time spent in multiplication after improvement?

4. Discuss How to implement forwarding among the FP units and memory in the R4000 mentioning the difficulties you face compared to the integer unit only. Consider only the forwarding from the output of the memory unit.

5. Consider adding a new index addressing mode to MIPS. The addressing mode adds two registers and an 11-bit signed offset to get the effective address.

Our computer will be changed so that code sequence on the form

ADD R1, R1, R2

LW Rd, 100(R1) or store

Will be replaced with a load (or store) using the new addressing mode.

a Assume that the addressing mode can be used for 10% of the displacement loads and stores What is the ratio of the instruction count of the old and new machine

b If the new addressing mode lengthen the clock cycle by 5%, which machine is faster and by how much?

Consider the following instruction MIPS (taken from HP3rd ED

Instruction	Gap	gcc
Load	26.5%	25.1%
Store	10.3%	13.2%
Add	21.1%	19.0%
Sub	1.7%	2.2%
Mul	1.4%	0.1%
Compare	2.8%	6.1%
Load imm	4.8%	2.5%
Cond branch	9.3	12.1%
Cond move	0.4	0.6%
Jump	0.8%	0.7%
Call	1.6%	0.6%
Return	1.6%	0.6%
Shift	3.8%	1.1%
And	4.3%	4.6%
Or	7.9%	%8.5
xor	1.8%	2.1%
Other	0.1%	0.4%