ICS 233: Computer Architecture & Assembly Language
Fall Semester 2021 (211) – Section 01
Quiz 6

Name: *ID#*

Q1. (5 points) Consider two different implementations, **M1** and **M2**, of the same instruction set. There are three classes of instructions (A, B, and C) in the instruction set. M1 has a clock rate of **6 GHz** and M2 has a clock rate of **3 GHz**. The CPI for each instruction class on M1 and M2 is given in the following table:

Class	CPI on M1	CPI on M2	C1 Usage	C2 Usage
A	2	1	40%	60%
B	3	2	40%	15%
C	5	3	20%	25%

The above table also contains a summary of the usage of instruction classes generated by two different compilers: C1 and C2. Assume that each compiler generates the same number of instructions for a given program. Which computer and compiler combination give the best performance?

Execution time based on M1 and C1=IC×(2×0.4+3×0.4+5×0.2)×1/ 6×10^9 =IC×3×1/6×10^9 =IC×0.5ns.
Execution time based on M2 and C1=IC×(1×0.4+2×0.4+3×0.2)×1/3×10^9 =IC×1.8×1/3×10^9 =IC×0.6ns.
Execution time based on M1 and C2=IC×(2×0.6+3×0.15+5×0.25)×1/6×10^9 =IC×2.9×1/6×10^9 =IC×0.48ns.
Execution time based on M2 and C2=IC×(1×0.6+2×0.15+3×0.25)×1/3×10^9 =IC×1.65×1/3×10^9 =IC×0.55ns.

The best performance is achieved using compiler C2 and machine M1

Q2. (5 points) Calculate the MIPS rating for each compiler with respect to both implementations.

MIPS of C1 w.r.t M1 = 6×10^9 / (2×0.4+3×0.4+5×0.2) $\times10^6$ = 2×10^3
MIPS of C1 w.r.t M2 = 3×10^9 / (1×0.4+2×0.4+3×0.2) $\times10^6$ = 1.667×10^3
MIPS of C2 w.r.t M1 = 6×10^9 / (2×0.6+3×0.15+5×0.25) $\times10^6$ = 2.069×10^3
MIPS of C2 w.r.t M2 = 3×10^9 / (1×0.6+2×0.15+3×0.25) $\times10^6$ = 1.818×10^3

ICS 233: Computer Architecture & Assembly Language
Fall Semester 2021 (211) – Section 02
Quiz 6

Name: *ID#*

Q1. (4 points) A benchmark program runs for 100 seconds. We want to speedup the benchmark by a factor of 3. We enhance the floating-point hardware to make floating point instructions run 6 times faster. How much of the initial execution time would floating-point instructions have to account for to show an overall speedup of 3 on this benchmark?

Speedup = 1 / (f/s + (1-f))=> 3 = 1 / (f/6+(1-f)) => f/6 + 1-f = 1/3 => f + 6 – 6f = 6/3 => 5f = 4 => f =4/5=0.8
Thus, floating-point instructions must account for 80% of the initial execution time, i.e. 80 seconds, to show an overall speedup of 3 on this benchmark

Q2. (6 points) Consider the following fragment of MIPS code. Assume that a and b are arrays of words and the base address of a is in $a0 and the base address of b is in $a1. How many instructions are executed during the running of this code? If ALU instructions (addu and addiu) take 1 cycle to execute, load/store (lw and sw) take 5 cycles to execute, and the branch (bne) instruction takes 3 cycles to execute, how many cycles are needed to execute the following code (all iterations). What is the average CPI?

```
       addu $t0, $zero, $zero   # i = 0
       addu $t1, $a0, $zero     # $t1 = address of a[i]
       addu $t2, $a1, $zero     # $t2 = address of b[i]
       addiu $t3, $zero, 100    # $t3 = 100 (max i)
loop:  lw $t4, 0($t2)           # $t4 = b[i]
       addu $t5, $t4, $s0       # $t5 = b[i] + c
       sw $t5, 0($t1)           # a[i] = b[i] + c
       addiu $t0, $t0, 1        # i++
       addiu $t1, $t1, 4        # address of next a[i]
       addiu $t2, $t2, 4        # address of next b[i]
       bne $t0, $t3, loop       # loop if (i != 100)
```

The loop body will be executed 100 times. Thus, the total number of instructions executed per class is:
Addu and addiu will be executed 4 + 100*4 = 404 times
Lw and Sw will be executed 100*2 = 200 times
Beq will be executed 100 times
Hence, the total number of instruction executed is 404 + 200 + 100 = 704
Total number of execution cycles = 404*1 + 200 * 5 + 100 * 3 = 1704 cycles
CPI = 1704 / 704 = 2.42