Lecture 3 Performance/ISA Exercises

Q1:

Consider Processor A, which has an average CPI of 5.0 for a specific program and a clock speed (i.e., frequency) of 2GHz.
(a) If a code segment, S, takes 15 seconds to run, how many instructions are in S?

(b) Suppose we were able to optimize 40% of instructions in S by a factor of 2, what is the speedup from the optimization for segment S?

(c) With the changes in (b), can we compute the overall average CPI of the processor? If YES, please state why and calculate the new CPI. If NO, please state why not.

Solution:

(a) CPI = 5
Clock frequency = 2*10^9 Hz
Execution Time: ET = 15s
Cycles = ET * Clock frequency = 15 * 2*10^9 = 30*10^9
Instructions * CPI = Cycles => Instructions = Cycles / CPI
So Instructions = 30*10^9 / 5 = 6 billion

(b) Amdahl's Law: 1/ ((1-40%) + 40%/2) = 1.25 Or
Base CPI = 5, New CPI = 0.4*2.5 + 0.6*5 = 4, So Speedup = Base CPI / New CPI = 1.25

(c) No, we cannot. CPI is not only a characteristic of the processor. It is also impacted by program, compiler, ISA.

Q2:
Given the following assembly code:

```
L3:  addu $7, $4, $3
lw $7, ($7)
addu $8, $5, $3
lw $8, ($8)
mul $7, $7, $8
addu $2, $2, $7
addiu $3, $3, #4
bne $3, $6, L3
```

Where registers are written as $(reg number), e.g., $1.

In a non-pipelined CPU with the following instruction categories and execution latencies:

Instruction	Latency (Cycles)
add (addu, addiu)	4
load (lw)	10
multiply (mul)	20
branch (bne)	8

A. What is the CPI of the loop for one iteration (the bne instruction included)?
B. Which of the following optimizations would produce bigger CPI improvement for one iteration (the bne instruction included)?

- Implement prefetching to reduce the latency of loads from 10 cycles to 7 cycles.
- Implement branch prediction to reduce the latency of branch instructions from 8 cycles to 2 cycles.

Solution:
Q2-A:
There are a total of 8 instructions in the loop. The percentage breakdown by type is:

- add: 50%
- load: 25%
- multiply: 12.5%
- branch: 12.5%

So the CPI is: 0.5*4 + 0.25*10 + 0.125*20 + 0.125*8 = 8

Q2-B:
The performance improvement is the same for both optimizations:
CPI for prefetching: 0.5*4 + 0.25*7 + 0.125*20 + 0.125*8 = 7.25
CPI for branch prediction: 0.5*4 + 0.25*10 + 0.125*20 + 0.125*2 = 7.25

Q3:

Consider a program which takes 900 seconds to execute, broken into 3 phases: A, B, and C. Without any optimizations, each phase takes one-third of the total execution time.
(a) What is the speedup if we improve Phase B by a factor of 1.5 and Phase C by a factor of 3?
(b) Now, suppose that Phase B and C are both 90% parallelizable, and Phase A is 0% parallelizable. Assuming that there are infinitely many available processors, and zero overhead from parallelization, what is the maximum speedup that can be obtained?

Solution:

(a) ET – Execution Time

$$ET_{old} = 900$$

$$ET_{new} = ET_A + ET_B + ET_C = 300 + \frac{300}{1.5} + \frac{300}{3} = 300 + 200 + 100 = 600$$

$$Speedup = \frac{ET_{old}}{ET_{new}} = \frac{900}{600} = 1.5$$

(b) We know that B and C compose ⅔ of the program and they are improved infinitely.

Parallelizable portion of the program is ⅔ * 90%,
So unparallelizable portion of the program is 1- (⅔ * 90%),
Improvement factor is infinite, because there are infinite number of processors and each processor can run one piece of program in parallel

$$Speedup = \frac{1}{(1 - Fraction_{enhanced}) + \frac{Fraction_{enhanced}}{Speedup_{enhanced}}}$$

$$Speedup = \frac{1}{(1 - \frac{9}{10} * \frac{2}{3}) + \frac{\frac{9}{10} * \frac{2}{3}}{\inf}}$$

$$Speedup = \frac{1}{1 - \frac{6}{10}} = \frac{1}{\frac{4}{10}} = 2.5$$