Lecture 2 Performance Exercises

Q1:
Short answers
(a) If a given program runs on a processor with a higher clock frequency, does it imply that the processor always executes more instructions per second (compared to a processor with a lower clock frequency)? Why?
(b) If a processor executes more of a given program's instructions per second, does it imply that the processor always finishes the program faster (compared to running a program that executes fewer instructions per second)? Why?

Solution:

(a) No. Instructions per second is also impacted by program, compiler, ISA.
(b) No. Instruction count can be different.

Q2:
Assume a program has the following instruction type breakdown:
40% memory, latency is 4 cycles
50% adds, latency is 2 cycles
10% multiplies, latency is 18 cycles
If you could pick one type of instructions to make twice as fast (half of latency) in the next-generation of this processor, which instruction type would you pick? What is the speedup?

Solution 1:
Base CPI: 40%*4 + 50%*2 + 10%*18 = 4.4

Pick one type of instructions to make twice as fast (half of latency):
Make memory instructions 2x faster: New CPI = 40%*2 + 50%*2 + 10%*18 = 3.6
Make add instructions 2x faster: New CPI = 40%*4 + 50%*1 + 10%*18 = 3.9
Make multiplies 2x faster: New CPI = 40%*4 + 50%*2 + 10%*9 = 3.5

Therefore, I will pick to optimize multiply instructions.

Speedup = Base CPI / New CPI = 4.4 / 3.5 = 1.257

Solution 2:
Calculate Base CPI: Assume a program has a total of 100 instructions, then
40 are memory instructions, will take 40 * 4 = 160 cycles
50 are adds, will take a total of 50 * 2 = 100 cycles
10 are multiplies, will take 10 * 18 = 180 cycles

The total number of cycles to run the program with 100 instructions = 160 + 100 + 180 = 440 cycles

So Base CPI = 440 / 100 = 4.4

Pick one type of instructions to make twice as fast (half of latency):
Make memory instructions 2x faster: New total cycles = 40*2 + 50*2 + 10*18 = 360,
So New CPI = 360 / 100 = 3.6

You may calculate New CPIs with other optimizations in the same way…and get the same results as solution 1.

Solution 3:
You may calculate IPC instead of CPI, and should reach the same conclusion.

Q3:
Assume a typical program has the following instruction type breakdown:
30% loads, latency is 4 cycles
10% stores, latency is 4 cycles
50% adds, latency is 2 cycles
8% multiplies, latency is 16 cycles
2% divides, latency is 50 cycles

If you could pick one type of instruction to make twice as fast (half the latency) in the next-generation of this processor, which instruction type would you pick? Why?

Solution:
The same as Q2, you may pick either one of the methods to solve the problem. In the following, I show one of the solutions.

Base CPI = ?

Pick one type of instructions to make twice as fast (half of latency):
Make loads 2x faster: New CPI = ?
Make stores 2x faster: New CPI = ?
Make adds 2x faster: New CPI = ?
Make multiplies 2x faster: New CPI = ?
Make divides 2x faster: New CPI = ?

	Base	loads	stores	adds	mults	divs
Loads 30%*4	1.2	0.6	1.2	1.2	1.2	1.2

Stores 10%*4	0.4	0.4	0.2	0.4	0.4	0.4
Adds 50%*2	1.0	1.0	1.0	0.5	1.0	1.0
Mults 8%*16	1.28	1.28	1.28	1.28	0.64	1.28
Divs 2%*50	1.0	1.0	1.0	1.0	1.0	0.5
CPI	**4.88**	**4.28**	**4.68**	**4.38**	**4.24**	**4.38**

Therefore, I would pick the multiplies to make twice as fast, because it reduces the CPI the most,

by 15% over the Base case, because
Speedup = Base CPI / New CPI = 4.88 / 4.24 = 1.15