McGovern, Jeff
CPE 315, Section 01
January 22, 2016
Homework 1

Assignment 2

1. Suppose we have two implementations of the same instruction set architecture (ISA). For some program,

   a. Machine A has a clock cycle time of 10 ns and a CPI of 2.0

   b. Machine B has a clock cycle time of 20 ns and a CPI of 1.2

   Answer the following:

   - What machine is faster for this program, and by how much?
   - If two machines have the same ISA for a given program which of our quantities (e.g., clock rate, CPI, execution time, # of instructions, instructions per second) will always be identical?

   Answer:

   - Given a program implemented under machine A and machine B, we calculate its execution time:

     $$ExecTime_A = CCT_A \cdot CPI_A \cdot \#Instructions = 10\frac{ns}{\cancel{cycle}} \cdot 2.0\frac{\cancel{cycle}}{instruction} \cdot \#instructions$$

     $$ExecTime_B = CCT_B \cdot CPI_B \cdot \#Instructions = 20\frac{ns}{\cancel{cycle}} \cdot 1.2\frac{\cancel{cycle}}{instruction} \cdot \#instructions$$

     Alternatively, since we don't know exactly how many instructions the program is, we can say:

     $$ExecTime_A = 20\frac{ns}{instruction}$$
     $$ExecTime_B = 24\frac{ns}{instruction}$$

     Since both machines have the same ISA, the number of instructions will be identical. Thus, the performance comparison is:

     $$\frac{Perf_A}{Perf_B} = \frac{CCT_B \cdot CPI_B \cdot \cancel{\#Instructions}}{CCT_A \cdot CPI_A \cdot \cancel{\#Instructions}} = \frac{20\cancel{ns} \cdot 1.2}{10\cancel{ns} \cdot 2.0} = 1.2$$

     Thus, the implementation for A is $1.2\times$ faster than for B.

   - Anything relating to the processor is *not* identical. When we talk about having the same Instruction Set Architecture (ISA), we mean that the instructions used to logically write programs are the same. Thus, between two machines with the same ISA, the *number of instructions* are the only thing that remain identical. All other aspects depend on how that processor works (e.g. clock rate) and implements the instructions set (e.g. CPI, execution time, instructions per second).

2. A compiler designer is trying to decide between two code sequences for a particular machine. Based on the hardware implementation, there are three different classes of instructions: Class A, Class B, and Class C. They require one, two, and three cycles (respectively).

   a. The first code sequence has 5 instructions: 2 of A, 1 of B, and 2 of C.

   b. The second sequence has 6 instructions: 4 of A, 1 of B, and 1 of C.

   Answer the following:

   - Which sequence will be faster? How much?
   - What is the CPI for each sequence?

   Answer:

Table 1: 3 Classes of Instructions

	A	B	C
Cycles	1	2	3
Program 1	2 of A	1 of B	2 of C
Program 2	4 of A	1 of B	1 of C

$$
\begin{aligned}
rt &= CCT \cdot CPI \cdot I \\
\frac{rt_{p1}}{rt_{p2}} &= \frac{CCT \cdot CPI_{p1} \cdot I_{p1}}{CCT \cdot CPI_{p2} \cdot I_{p2}} \\
CPI_{p1} &= \frac{2}{5} \cdot 1 + \frac{1}{5} \cdot 2 \frac{2}{5} \cdot 3 = \frac{10}{3} = 2 \\
CPI_{p2} &= \frac{4}{6} \cdot 1 + \frac{1}{6} \cdot 2 \frac{1}{6} \cdot 3 = \frac{9}{6} = 1.5 \\
I_{p1} &= 5 \\
I_{p2} &= 6 \\
\frac{rt_{p1}}{rt_{p2}} &= \frac{5 \cdot 2}{6 \cdot 1.5} = \frac{10}{9}
\end{aligned}
$$

Thus, $p2$ is $\sim 12\%$ faster than $p1$.

3. Two different compilers are being tested for a 3 GHz machine with three different classes of instructions: Class A, Class B, and Class C, which require one, two, and three cycles (respectively). Both compilers are used to produce code for a large piece of software.

   a. The first compiler's code uses 5 million Class A instructions, 1 million Class B instructions, and 1 million Class C instructions.

b. The second compiler's code uses 10 million Class A instructions, 1 millionClass B instructions, and 1 million Class C instructions.

Answer the following:

- Which sequence will have a greater number of instructions per second?
- Which sequence will have the faster runtime?

Answer:

	A	B	C
Compiler/Cycles	1	2	3
First	5×10^6	1×10^6	1×10^6
Second	10×10^6	1×10^6	1×10^6

- Consider the following conversions:

$$\frac{\text{instr}_A}{\text{second}} = \frac{1 \text{ instr}_A}{1 \cancel{\text{ clock cycle}}} \times \frac{3 \times 10^9 \cancel{\text{ clock cycles}}}{\text{second}} = \frac{3 \times 10^9 \text{ instr}_A}{\text{second}}$$
$$\frac{\text{instr}_B}{\text{second}} = \frac{1 \text{ instr}_B}{2 \cancel{\text{ clock cycle}}} \times \frac{3 \times 10^9 \cancel{\text{ clock cycles}}}{\text{second}} = \frac{1.5 \times 10^9 \text{ instr}_B}{\text{second}}$$
$$\frac{\text{instr}_C}{\text{second}} = \frac{1 \text{ instr}_C}{3 \cancel{\text{ clock cycle}}} \times \frac{3 \times 10^9 \cancel{\text{ clock cycles}}}{\text{second}} = \frac{1 \times 10^9 \text{ instr}_C}{\text{second}}$$

For the First sequence:

$$\begin{aligned} 5 \times 10^6 \cancel{\text{ instr}_A} \cdot \frac{\text{second}}{3 \times 10^9 \cancel{\text{ instr}_A}} &+ \\ 1 \times 10^6 \cancel{\text{ instr}_B} \cdot \frac{1.5 \times 10^9 \cancel{\text{ instr}_B}}{\text{second}} &+ \\ 1 \times 10^6 \cancel{\text{ instr}_C} \cdot \frac{1 \times 10^9 \cancel{\text{ instr}_C}}{\text{second}} & \\ = \frac{1}{300} \text{ second} & \end{aligned}$$

For the Second:

$$\begin{aligned} 10 \times 10^6 \cancel{\text{ instr}_A} \cdot \frac{3 \times 10^9 \cancel{\text{ instr}_A}}{\text{second}} &+ \\ 1 \times 10^6 \cancel{\text{ instr}_B} \cdot \frac{1.5 \times 10^9 \cancel{\text{ instr}_B}}{\text{second}} &+ \\ 1 \times 10^6 \cancel{\text{ instr}_C} \cdot \frac{1 \times 10^9 \cancel{\text{ instr}_C}}{\text{second}} & \\ = \frac{1}{200} \text{ second} & \end{aligned}$$

Clearly, the First takes less time to run ($\frac{1}{300} = 0.033\ldots < \frac{1}{200} = 0.050$). however, the number of instructions per second at the end of their run is thus:

$$IPS_A = \frac{7 \times 10^6}{\frac{1}{300}} = 2.1 \times 10^9$$
$$IPS_B = \frac{13 \times 10^6}{\frac{1}{200}} = 2.6 \times 10^9$$

Thus, B runs through more instructions per second than A does per second.

- The first sequence will have a faster runtime, simply because it has fewer instructions to run.

4. For the following mix of instructions:

Table 2: Mix of instructions for Question 4.

Operation Type	Frequency	Cycles
ALU	50%	1
Load	20%	5
Store	10%	3
Branch	20%	2

Answer the following questions:

a. What's the CPI?

b. How much faster would the machine be if the average load time was reduced to 2 cycles?

c. How does this compare with shaving a cycle off the branch time?

d. What if two ALU instructions could (always) be executed at once?

Answer:

a.

$$CPI = \frac{50}{100} \cdot 1 + \frac{20}{100} \cdot 5 + \frac{10}{100} \cdot 3 + \frac{20}{100} \cdot 2 = 2.2$$

b.

$$CPI = \frac{50}{100} \cdot 2 + \frac{20}{100} \cdot 2 + \frac{10}{100} \cdot 2 + \frac{20}{100} \cdot 2 = 2$$
$$\frac{2.2}{2} = 1.1 \times \text{ speedup}$$

c.

$$CPI = \frac{50}{100} \cdot 1 + \frac{20}{100} \cdot 5 + \frac{10}{100} \cdot 3 + \frac{20}{100} \cdot 1 = 2$$
$$\frac{2.2}{2} = 1.1 \times \text{ speedup} \Rightarrow \text{ same amount}$$

d.

$$CPI = \frac{50}{100} \cdot 0.5 + \frac{20}{100} \cdot 2.5 + \frac{10}{100} \cdot 1.5 + \frac{20}{100} \cdot 1 = 1.1$$

$$\frac{2.2}{1.1} = 2 \times \text{ speedup}$$

5. Suppose a program runs in 100 seconds on a machine, with multiply responsible for 80 seconds of this time. Answer the following:

   a. How much do we have to improve the speed of multiplication if we want the program to run 4 times faster?

   b. How about making it 5 times faster?

Answer:

$$p_1 = \frac{80}{100} \cdot c_1 + \frac{20}{100} = \frac{80c_1 + 20x}{100}$$
$$p_2 = \frac{80}{100} \cdot c_2 + \frac{20}{100}x = \frac{80c_2 + 20x}{100}$$

According to Ahmdahl's Law, we need to consider the equation

$$S = \frac{1}{(1-F) + \frac{F}{E}}$$

where S is the speedup, E is the speedup of the portion that we are modifying, and F is the proportion of the whole task benefiting from the modification.

a. For a 4× speedup, we need find E to satisfy the following equation:

$$4 = \frac{1}{(1-.80) + \frac{.80}{E}}$$
$$\Rightarrow (1-.80) + \frac{.80}{E} = \frac{1}{4}$$
$$\Rightarrow \frac{.80}{E} = \frac{1}{4} - 1 + .80$$
$$\Rightarrow E = \frac{.80}{.05}$$
$$\Rightarrow E = 16$$

Thus, we need to increase the speed of our multiply operator by 16×

b.

$$5 = \frac{1}{(1-.80) + \frac{.80}{E}}$$
$$\Rightarrow (1-.80) + \frac{.80}{E} = \frac{1}{5}$$
$$\Rightarrow \frac{.80}{E} = \frac{1}{5} - 1 + .80$$
$$\Rightarrow E = \frac{.80}{0} \quad *gasp*$$

We can't. We've reached our theoretical limit of what that portion of program can do. The very highest is can speedup the program by is $5\times$, but that's only if speedup that operation by ∞. In other words, so you don't break maths, consider that when we increase the speedup to ∞, i.e.

$$\frac{F}{E}_{E\to\infty},$$

the term $\frac{F}{E} \to \infty$, giving us a speedup, S of

$$S = \frac{1}{1-F} = \frac{1}{1-.80} = 5$$

Thus, in order to get $5x$ speedup practically, we need to look at the other portion of the program, or magically make multiply infinitely faster.

6. Suppose we enhance a machine making all floating-point instructions run five times faster. If the execution time of some benchmark before the floating-point enhancement is 10 seconds, what will the speedup be if half of the 10 seconds is spent executing floating-point instructions?

   Answer: Coming back to Ahmdahl's Law, we must consider the following equation:

$$S = \frac{1}{(1-F)+\frac{F}{E}}$$

   where S is the speedup we wish to find, E is the speedup to the floating point operation, and F is the proportion of the whole runtime that the floating point operation takes up.

   Thus,

$$S = \frac{1}{(1-.5)+\frac{.5}{5}} = \frac{1}{.5+.1} = 1.66\ldots\times \text{ speedup}$$

   with a $1.66\ldots\times$ speedup, our runtime will be 6 seconds instead of 10.

7. We are looking for a benchmark to show off the new floating-point unit described in Problem 6, and want the overall benchmark to show a speedup of 3. We are considering a benchmark that runs for 100 seconds with the old floating-point hardware. How much of the execution time would floating-point instructions have to account for in this program in order to yield our desired speedup on this benchmark?

   Answer: Consider again Ahmdahl's Law:

$$S = \frac{1}{(1-F)+\frac{F}{E}}$$

   where we want S to show a $3\times$ speedup, E to remain the speedup of the floating point operation, and F the proportion of the whole runtime that we want to control for in order to get an overall $3\times$ speedup.

Solving accordingly,

$$\begin{aligned}
S &= \frac{1}{(1-F)+\frac{F}{E}} \\
\Rightarrow \frac{1}{S} &= 1 - F + \frac{F}{E} \\
\Rightarrow F - \frac{F}{E} &= 1 - \frac{1}{S} \\
\Rightarrow \frac{E \cdot F - F}{E} &= \frac{S-1}{S} \\
\Rightarrow F &= \frac{E(S-1)}{S(E-1)}
\end{aligned}$$

we can now get the answer we seek:

$$F = \frac{5(3-1)}{3(5-1)} = \frac{5}{6} = 83\%$$

In order to get a 3× speedup, let the now faster floating point operation take up a little over 83.3 seconds of the 100 seconds runtime.

8. For the following mix of instructions:

Table 3: Mix of instructions for Problem 4.

Operation Type	Frequency	Cycles
ALU	50%	1
Load	20%	2
Store	10%	2
Jump/Call	10%	1
Branch	10%	3

a. Calculate CPI for this program.

b. If this runs on a 3.3 GHz processor, how many instructions per second (IPS) execute?

c. Your compiler team reports they can eliminate 20% of ALU instructions (i.e. 10% of all instructions). What is the speedup?

d. With the compiler improvements, what is the new CPI and IPS?

Answer:

a.

$$CPI = \frac{50}{100} \cdot 1 + \frac{20}{100} \cdot 2 + \frac{10}{100} \cdot 2 + \frac{10}{100} \cdot 1 + \frac{10}{100} \cdot 3 = 1.6$$

b. The value a. gave us the cycles per instruction. Converting that to instructions per second:

$$\frac{1\text{instr}}{1.6\cancel{\text{cycles}}} \times \frac{3 \times 10^9 \cancel{\text{cycles}}}{\text{second}} = 625 \times 10^6 \frac{\text{instr}}{\text{second}}$$

c.

$$S = \frac{1}{(1 - .5) + \frac{.5}{5}} = 1.66\ldots \times \text{ speedup}$$

d.

$$CPI = \frac{30}{80} \cdot 1 + \frac{20}{80} \cdot 2 + \frac{10}{80} \cdot 2 + \frac{10}{80} \cdot 1 + \frac{10}{80} \cdot 3 = 1.625$$