Steven Yu

ECE 4300

February 15, 2021

Homework 1

1.3 Describe the steps that transform a program written in a high-level language such as C into a representation that is directly executed by a computer processor.
A high-level language program is converted to a assembly language program by a compiler. A assembly language program is converted to a machine language program by an assembler. The machine language program is interpreted by the machine and execute by the computer processor to send control signals.

1.4 Assume a color display using 8 bits for each of the primary colors (red, green, blue) per pixel and a frame size of 1280 × 1024.

a. What is the minimum size in bytes of the frame buffer to store a frame?
3 [byte/pixel] × (1280 × 1024) [pixel/frame] = 3932160 [byte/frame]

b. How long would it take, at a minimum, for the frame to be sent over a 100 Mbit/s network?
(3932160 × 8) [bit] / 100 [Mbit/s] = 0.3146 [s]

1.5 Consider three different processors P1, P2, and P3 executing the same instruction set. P1 has a 3 GHz clock rate and a CPI of 1.5. P2 has a 2.5 GHz clock rate and a CPI of 1.0. P3 has a 4.0 GHz clock rate and has a CPI of 2.2.

a. Which processor has the highest performance expressed in instructions per second?
P1: 1/1.5 [instruction/clock cycles] × 3 G [clock cycles/s] = 2×10^9 [instruction/s]
P2: 1/1 [instruction/clock cycles] × 2.5 G [clock cycles/s] = 2.5×10^9 [instruction/s]
P3: 1/2.2 [instruction/clock cycles] × 4 G [clock cycles/s] = 1.82×10^9 [instruction/s]

P2 executes the highest number of instructions per second.

b. If the processors each execute a program in 10 seconds, find the number of cycles and the number of instructions.
P1: 3 G [clock cycles/s] × 10 [s] = 3×10^10 [instruction/s]
P1: 2×10^9 [instruction/s] × 10 [s] = 2×10^10 [instruction]

P2: 2.5 G [clock cycles/s] × 10 [s] = 2.5×10^10 [instruction/s]
P2: 2.5×10^9 [instruction/s] × 10 [s] = 2.5×10^10 [instruction]

P3: 4 G [clock cycles/s] × 10 [s] = 4×10^10 [instruction/s]
P3: 1.82×10^9 [instruction/s] × 10 [s] = 1.82×10^10 [instruction]

We are trying to reduce the execution time by 30% but this leads to an increase of 20% in the CPI. What clock rate should we have to get this time reduction?
0.7 = (new clock cycles × new clock cycle time) / (old clock cycles × old clock cycle time) = (instructions × 1.2 × [clock/instruction] × new clock cycle time) / (instructions × [clock/instruction] × old clock cycle time) = 1.2 × old clock rate / new clock rate

new clock rate = 1.2 × old clock rate / 0.7

1.6 Consider two different implementations of the same instruction set architecture. The instructions can be divided into four classes according to their CPI (class A, B, C, and D). P1 with a clock rate of 2.5 GHz and CPIs of 1, 2, 3, and 3, and P2 with a clock rate of 3 GHz and CPIs of 2, 2, 2, and 2.
Given a program with a dynamic instruction count of 1.0E6 instructions divided into classes as follows: 10% class A, 20% class B, 50% class C, and 20% class D, which implementation is faster?
A: 0.1 × 10^6 = 10^5 [instructions]
B: 0.2 × 10^6 = 2×10^5 [instructions]
C: 0.5 × 10^6 = 5×10^5 [instructions]
D: 0.2 × 10^6 = 2×10^5 [instructions]

P1: ((1 × 10^5) + (2 × 2×10^5) + (3 × 5×10^5) + (3 × 2×10^5)) / 2.5 GHz = 1.04 ms
P2: ((2 × 10^5) + (2 × 2×10^5) + (2 × 5×10^5) + (2 × 2×10^5)) / 3 GHz = 0.6667 ms

P2 executes the program faster.

a. What is the global CPI for each implementation?
P1: (1 × 10^5) + (2 × 2×10^5) + (3 × 5×10^5) + (3 × 2×10^5) / 10^6 = 2.6 [clock cycles/instruction]
P2: (2 × 10^5) + (2 × 2×10^5) + (2 × 5×10^5) + (2 × 2×10^5) / 10^6 = 2 [clock cycles/instruction]

b. Find the clock cycles required in both cases.
P1: (1 × 10^5) + (2 × 2×10^5) + (3 × 5×10^5) + (3 × 2×10^5) = 2.6×10^6 [clock cycles]
P2: (2 × 10^5) + (2 × 2×10^5) + (2 × 5×10^5) + (2 × 2×10^5) = 2×10^6 [clock cycles]

1.7 Compilers can have a profound impact on the performance of an application. Assume that for a program, compiler A results in a dynamic instruction count of 1.0E9 and has an execution time of 1.1 s, while compiler B results in a dynamic instruction count of 1.2E9 and an execution time of 1.5 s.

a. Find the average CPI for each program given that the processor has a clock cycle time of 1 ns.
A: (1.1 / 10^-9) / 10^9 = 1.1 [clock cycles/instruction]
B: (1.5 / 10^-9) / 1.2×10^9 = 1.25 [clock cycles/instruction]

b. Assume the compiled programs run on two different processors. If the execution times on the two processors are the same, how much faster is the clock of the processor running compiler A's code versus the clock of the processor running compiler B's code?
A: 10^9 [A instructions] × 1.1 [A clock cycles/instruction] × [A clock cycle time] = [execution time]
B: 1.2×10^9 [B instructions] × 1.25 [B clock cycles/instruction] × [B clock cycle time] = [execution time]

10^9 [A instructions] × 1.1 [A clock cycles/s] × [A clock cycle time] = 1.2×10^9 [B instructions] × 1.25 [B clock cycles/s] × [B clock cycle time]

(A clock cycle time / B clock cycle time) = (1.2×10^9 [B instructions] × 1.25 [B clock cycles/instruction]) / (10^9 [A instructions] × 1.1 [A clock cycles/instruction]) = 1.36

The clock of the processor running compiler A's code is 1.36 times faster than the clock of the processor running compiler B's code.

c. A new compiler is developed that uses only 6.0E8 instructions and has an average CPI of 1.1. What is the speedup of using this new compiler versus using compiler A or B on the original processor?
C: 6×10^8 [C instructions] × 1.1 [C clock cycles/instruction] × [C clock cycle time] = [C execution time]

(A execution time / C execution time) = (6×10^8 [C instructions] × 1.1 [C clock cycles/instruction]) / 10^9 [A instructions] × 1.1 [A clock cycles/instruction]) = 0.6 [speedup]

(B execution time / C execution time) = (6×10^8 [C instructions] × 1.1 [C clock cycles/instruction]) / (1.2×10^9 [B instructions] × 1.25 [B clock cycles/instruction]) = 0.44 [speedup]

1.11 The results of the SPEC CPU2006 bzip2 benchmark running on an AMD Barcelona has an instruction count of 2.389E12, an execution time of 750 s, and a reference time of 9650 s.

a. Find the CPI if the clock cycle time is 0.333 ns.
750 [execution time] / 0.333×10-9 [clock cycle time] / 2.389×10^12 [instructions] = 0.9428 [clock cycles/instruction]

b. Find the SPEC ratio.
9650 [s] / 750 [s] = 12.8667

c. Find the increase in CPU time if the number of instructions of the benchmark is increased by 10% without affecting the CPI.
[new cpu time] = 1.1 × [instruction] × [clock cycles/instruction] × [clock cycle time]
CPU time increases by 1.1 times.

d. Find the increase in CPU time if the number of instructions of the benchmark is increased by 10% and the CPI is increased by 5%.
[new cpu time] = 1.1 × [instruction] × 1.05 × [clock cycles/instruction] × [clock cycle time]
CPU time increases by 1.155 times.

e. Find the change in the SPEC ratio for this change.
9650 [s] / (1.155 × 750 [s]) = 11.14

f. Suppose that we are developing a new version of the AMD Barcelona processor with a 4 GHz clock rate. We have added some additional instructions to the instruction set in such a way that the number of instructions has been reduced by 15%. The execution time is reduced to 700 s and the new SPEC ratio is 13.7. Find the new CPI.
(4 GHz [clock rate] × 700 [execution time]) / (0.85 × 2.389×10^12 [instructions]) = 1.3789 [clock cycles/instruction]

g. This CPI value is larger than obtained in 1.11.1 as the clock rate was increased from 3 GHz to 4 GHz. Determine whether the increase in the CPI is similar to that of the clock rate. If they are dissimilar, why?
4 GHz / 3 GHz = 1.33
1.3789 / 0.9428 = 1.46
The clock rate was increased by 33% while the CPI increased by 46%. The difference is likely the result of the 15% reduction of instructions and reduced execution time.

h. By how much has the CPU time been reduced?
50 [s] / 750 [s] = 6.67%

i. For a second benchmark, lib quantum, assume an execution time of 960 ns, CPI of 1.61, and clock rate of 3 GHz. If the execution time is reduced by an additional 10% without affecting to the CPI and with a clock rate of 4 GHz, determine the number of instructions.
0.9 × 960×10^-9 [execution time] / (1 [cycle time] / 4×10^9) = 3456 [clock cycles]
3456 [clock cycles] / 1.61 [clock cycles/instruction] = 2146.5839 [instructions]

j. Determine the clock rate required to give a further 10% reduction in CPU time while maintaining the number of instructions and with the CPI unchanged.
2146.5839 [instructions] × 1.61 [clock cycles/instruction] / 0.9 × 864×10^-9 [execution time] = 4.44 GHz [clock rate] (another 10% reduction after the previous 10% reduction)

k. Determine the clock rate if the CPI is reduced by 15% and the CPU time by 20% while the number of instructions is unchanged.
2146.5839 [instructions] × 0.85 × 1.61 [clock cycles/instruction] / 0.8 × 960×10^-9 [execution time] = 3.83 GHz [clock rate] (20% reduction rather than a 10% reduction)

1.12 Section 1.10 cites as a pitfall the utilization of a subset of the performance equation as a performance metric. To illustrate this, consider the following two processors. P1 has a clock rate of 4 GHz, average CPI of 0.9, and requires the execution of 5.0E9 instructions. P2 has a clock rate of 3 GHz, an average CPI of 0.75, and requires the execution of 1.0E9 instructions.

a. One usual fallacy is to consider the computer with the largest clock rate as having the largest performance. Check if this is true for P1 and P2.
P1: 5×10^9 [instructions] × 0.9 [clock cycles/instruction] / 4×10^9 [clock rate] = 1.125 s [execution time]
P2: 10^9 [instructions] × 0.75 [clock cycles/instruction] / 3×10^9 [clock rate] = 0.25 s [execution time]

P2 has a much faster execution time than P1 despite having a lower clock rate.

b. Another fallacy is to consider that the processor executing the largest number of instructions will need a larger CPU time. Considering that processor P1 is executing a sequence of 1.0E9 instructions and that the CPI of processors P1 and P2 do not change, determine the number of instructions that P2 can execute in the same time that P1 needs to execute 1.0E9 instructions.
P1: 10^9 [instructions] × 0.9 [clock cycles/instruction] / 4×10^9 [clock rate] = 0.225 s [execution time]
P2: instructions × 0.75 [clock cycles/instruction] / 3×10^9 [clock rate] = 0.225 s [execution time]

P2: instructions = 9×10^8

c. A common fallacy is to use MIPS (millions of instructions per second) to compare the performance of two different processors, and consider that the processor with the largest MIPS has the largest performance. Check if this is true for P1 and P2.
P1: MIPS = 4×10^9 [clock rate] × 10^-6 / 0.9 [clock cycles/instruction] = 4.44×10^3
P2: MIPS = 3×10^9 [clock rate] × 10^-6 / 0.75 [clock cycles/instruction] = 4×10^3

P2 has a much faster execution time than P1 despite having a lower MIPS.

d. Another common performance figure is MFLOPS (millions of floating-point operations per second), defined as
MFLOPS = No. Floating Point operations / (execution time × 1E6)
but this figure has the same problems as MIPS. Assume that 40% of the instructions executed on both P1 and P2 are floating-point instructions. Find the MFLOPS figures for the programs.
P1: MFLOPS = 0.4 × 5×10^9 / (1.125 × 10^6) = 1.7778×10^3
P2: MFLOPS = 0.4 × 10^9 / (0.25 × 10^6) = 1.6×10^3

P2 has a much faster execution time than P1 despite having a lower MFLOPS.