CS1216 - Monsoon 2022 - Homework 1

Jivansh Sharma
UG 24 1020211193

Collaborators: None

1. The clock of a processor runs at 800 MHz. The following table provides the instruction distributions for various types of instructions for Benchmark A, as well as the number of cycles one instruction of each type takes, for the different classes of instructions. Assume that the processor only executes one instruction at a time

(a) Calculate the average CPI for Benchmark A.

Let us assume we ran Benchmark A using 100 instructions. This would help simplify our problem.
According to our assumption, `Loads and Stores` takes `20 Instructions`, with a `CPI = 8`
`Arithmetic` takes `40 Instructions`, with a `CPI = 2`
Finally `Branch` takes `40 Instructions`, with a `CPI = 2`

Hence, we can also say that `Average Cycles Per Instruction` would be, `Total Cycles / Total Instructions`, ie

$$\frac{(20*8)+(40*2)+(40*2)}{100}$$
$$\implies \frac{(160)+(80)+(80)}{100}$$
$$\implies \frac{320}{100}$$
$$\implies 3.2$$

Therefore, the `Average Cycles Per Instruction` for Benchmark A is `3.2`

(b) Suppose we design a new compiler, which compiles Benchmark A such that the number of loads and stores in the new version of A is halved, while the number of the rest of the instructions remain unchanged. What will be the new CPI of A?

According to the question, the new compiler would halve the number of `Loads and Stores` to take `10 Instructions`. This would also result in the total Instructions reducing from 100 to 90.

Hence, we can also say that `Average Cycles Per Instruction` would be, `Total Cycles / Total Instructions`, ie

$$\frac{(10*8)+(40*2)+(40*2)}{90}$$

$$\implies \frac{(80)+(80)+(80)}{90}$$

$$\implies \frac{240}{90}$$

$$\implies 2.667$$

Therefore, the `New Average Cycles Per Instruction` for Benchmark A is `2.667`

2. For a given program from a benchmark suite, Processor A has a clock cycle time of 100 ps. and a CPI of 2.5. Processor B has a clock cycle time of 150 ps. and a CPI of 1.8. Which machine is faster for this program, and by how much? Report your results as a ratio.

We already know that,
$\texttt{CPU Execution Time} = (\texttt{Clock Cycle Time} \times \texttt{Avg CPI} \times \texttt{Number Of Instructions})$
And, $\texttt{Speedup of A over B} = \texttt{Performance(A)/Performance(B)}$
Also, $\texttt{Performance} = \texttt{1/CPU Execution Time}$

Now, considering the number of instructions to be equal for A and B. We can say the following

$$\texttt{Speedup of A over B} = \texttt{CPU Execution Time(B)/CPU Execution Time(A)}$$

From this we can see,

$$\texttt{CPU Execution Time}_A = (\texttt{Clock Cycle Time}_A \times \texttt{Avg CPI}_A \times \texttt{Number Of Instructions}_A)$$

$$\texttt{CPU Execution Time}_A = (\texttt{100 PS} \times \texttt{2.5 CPI} \times \texttt{1 Instruction})$$

$$\text{And, } \texttt{CPU Execution Time}_B = (\texttt{150 PS} \times \texttt{1.8 CPI} \times \texttt{1 Instruction})$$

$$\implies \texttt{Speedup of A over B} = \frac{150\times1.8}{100\times2.5}$$

$$\implies \texttt{Speedup of A over B} = \frac{270}{250}$$

$$\implies \texttt{Speedup of A over B} = \texttt{1.08}$$

Hence, we conclude that A is faster than B by `1.08`

3. We have designed a new processor that runs at 1500 million cycles per second. What is the frequency of the processor in GHz? What is its clock cycle time? If a processor has a frequency of 6 GHz, what is the clock cycle time of this processor? Show your work.

We are given the `Clock Frequency` of a processor is `1500 Million Cycles Per Second`. And we know that, $1GHz = 10^9 Hz$
So, the frequency of the processor in GHz would be;

$$\text{Frequency} = \frac{1500\times10^6}{10^9} GHz$$
$$\implies Frequency = \frac{15\times10^8}{10^9} GHz$$
$$\implies Frequency = \frac{15}{10^{-1}} GHz$$
$$\implies Frequency = \texttt{1.5}\ \text{GHz}$$

So, the frequency of the processor in GHz is `1.5 GHz`

Following this, we already know that *ClockTime* is represented as `1/Freq`. We can use this to calculate the clock time of a Processor A.

$$\text{Clock Time} = 1\ /\ \text{Frequency}$$
$$\implies ClockTime_A = \frac{1}{Frequency_A}$$
$$\implies ClockTime_A = \frac{1}{1.5\times10^9}$$
$$\implies ClockTime_A = 0.667 \times 10^{-9}$$

Therefore, the `Clock Cycle Time` of the first processor is $0.667 \times 10^{-9} seconds$

We may also use this to calculate the clock time of a Processor B.

$$\text{Clock Time} = 1\ /\ \text{Frequency}$$
$$\implies ClockTime_B = \frac{1}{Frequency_B}$$
$$\implies ClockTime_B = \frac{1}{6\times10^9}$$
$$\implies ClockTime_B = 0.166 \times 10^{-9}$$

Therefore, the `Clock Cycle Time` of the second processor is $0.166 \times 10^{-9} seconds$

4. A Program X takes 9 seconds to complete on an Intel processor. The same program takes 12 seconds to complete on a RISC-V processor

(a) What is the speedup of the Intel processor over the RISC-V one?

We already know that `Speedup of A over B` = `Performance(A)/Performance(B)`
And, `Performance` = `1/CPU Execution Time`
Therefore, `Speedup of A over B` = `CPU Execution Time(B)/CPU Execution Time(A)`

$$\texttt{Speedup of A over B} = \texttt{CPU Execution Time(B)/CPU Execution Time(A)}$$
$$\texttt{Speedup of A over B} = \texttt{CPU Execution Time(RISC-V)/CPU Execution Time(Intel)}$$

$$\texttt{Speedup of Intel over RISC-V} = \frac{12}{9}$$
$$\texttt{Speedup of Intel over RISC-V} = 1.334$$

Therefore the `Speedup of Intel over RISC-V` = 1.334

(b) What is the performance improvement of the Intel processor over the RISC-V one?

We already know that `Performance Improvement of A over B` = $\frac{Performance(A) - Performance(B)}{Performance(B)}$
And, `Performance` = `1/CPU Execution Time`
So,

$$\texttt{Performance Improvement of A over B} = \frac{\frac{1}{ExecutionTime_A} - \frac{1}{ExecutionTime_B}}{\frac{1}{ExecutionTime_B}}$$
$$\texttt{Performance Improvement of Intel over RISC-V} = \frac{\frac{1}{ExecutionTime_{Intel}} - \frac{1}{ExecutionTime_{RISC-V}}}{\frac{1}{ExecutionTime_{RISC-V}}}$$

$$\texttt{Performance Improvement of Intel over RISC-V} = ET_{RISC-V} \times \frac{ET_{RISC-V} - ETIntel}{ET_{RISC-V} \times ET_{Intel}}$$

$$\texttt{Performance Improvement of Intel over RISC-V} = 12 \times \frac{12-9}{12 \times 9}$$

$$\texttt{Performance Improvement of Intel over RISC-V} = 12 \times \frac{3}{108}$$

$$\texttt{Performance Improvement of Intel over RISC-V} = 12 \times 0.02778$$

$$\texttt{Performance Improvement of Intel over RISC-V} = (0.333 \times 100)\%$$

Therefore the `Performance Improvement of Intel over RISC-V` = 33.33%

5. Program A runs on processors P and Q with clock frequencies 2 GHz and 4.5 GHz respectively. Processors P and Q have a CPI of 1.5 each. Assuming that Program A is broken down into 9 billion instructions on each processor, calculate the performance improvement of Processor Q over Processor P.

We already know that `CPU Execution Time` = (`Clock Cycle Time`×`Avg CPI`×`Number Of Instructions`)
And, Clock Time = 1 / Frequency. So, CPU Time for Processor P would be,

$$ET_P = 1/Freq \times \texttt{No of Instructions} \times CPI$$
$$ET_P = \frac{1}{2\times10^9} \times 9 \times 10^9 \times 1.5$$
$$ET_P = \frac{1}{2} \times 13.5$$
$$ET_P = \texttt{6.75 seconds}$$

Similarly, CPU Time for Processor Q would be,

$$ET_Q = 1/Freq \times \texttt{No of Instructions} \times CPI$$
$$ET_Q = \frac{1}{4.5\times10^9} \times 9 \times 10^9 \times 1.5$$
$$ET_Q = \frac{1}{4.5} \times 13.5$$
$$ET_Q = \texttt{3 seconds}$$

Now, using

$$\texttt{Performance Improvement of Q over P} = ET_P \times \frac{ET_P - ETQ}{ET_P \times ET_Q}$$

$$\texttt{Performance Improvement of Q over P} = 6.75 \times \frac{6.75-3}{6.75\times3}$$

$$\texttt{Performance Improvement of Q over P} = 6.75 \times \frac{3.75}{20.25}$$

$$\texttt{Performance Improvement of Q over P} = 6.75 \times 0.185$$

$$\texttt{Performance Improvement of Q over P} = 1.25$$

$$\texttt{Performance Improvement of Q over P} = 125\%$$

Therefore the `Performance Improvement of Q over P` = 125%

6. MyAwesomeProcessor™ had a power rating of 120 Watts. Due to the microarchitectural enhancements that I made to the processor, I could reduce the power rating to 100 Watts. In the process of carrying out these optimizations, I had to reduce the frequency of the processor from 3.6 GHz to 3 GHz. Also during this process, due to features added on the chip, the execution time of Program A increased from 20 seconds to 22 seconds. If the energy consumption for Program A is my primary metric, are these microarchitectural enhancements worthwhile? Explain briefly.

We know that, `Total Power` = `Dynamic Power` + `Leakage Power` and, `Energy` = `Power` × `Time`

Finally, `Frequency` is directly proportional to `Dynamic Power` but inversely proportional to `Time`

Let us assume E_1 as the energy consumed by `MyAwesomeProcessor` before enhancements and E_2 as the energy consumed by `MyAwesomeProcessor` after enhancements.

So,

$$\begin{gathered}
E_1 = \texttt{Power} \times \texttt{Time} \\
E_1 = (\texttt{Dynamic Power} + \texttt{Leakage Power}) \times \texttt{Time} \\
E_1 = (120 \times 3.6 + x) \times \tfrac{20}{3.6} \text{ (Assume } LP = x) \\
E_1 = (432 + x) \times 5.55 \text{(Assume } x = 0) \\
E_1 = (432) \times 5.55 \\
E_1 = 2397.6J
\end{gathered}$$

Similarly,

$$\begin{gathered}
E_2 = \texttt{Power} \times \texttt{Time} \\
E_2 = (\texttt{Dynamic Power} + \texttt{Leakage Power}) \times \texttt{Time} \\
E_2 = (100 \times 3 + x) \times \tfrac{22}{3} \text{ (Assume } LP = x) \\
E_2 = (300 + x) \times 7.33 \text{(Assume } x = 0) \\
E_2 = (300) \times 5.55 \\
E_2 = 2199J
\end{gathered}$$

We, see that E_2 consumes less energy than E_1. Considering energy consumption is the primary concern, these microarchitectural enhancements are worthwhile.

7. My rudimentary processor consists of 4 circuits, an integer adder,a floating point adder, a multiplier and a divider. These units take 850 ps, 1200 ps, 2800 ps and 3000 ps, respectively. The processor is functioning under the constraint that it has to finish an instruction every cycle, irrespective of the type of instruction, and that the units receive their inputs only at the rising edge of the clock.

(a) Which unit would decide the frequency of the processor? The processor has to complete an instruction every cycle, so the circuit with the slowest Clock Cycle Time will be the deciding factor for the frequency. In this case, the Divider as it takes the greatest amount of time to finish.

(b) What will the frequency of the processor be?
We already know that Frequency = 1/Clock Cycle Time. Using this,

$$\text{Frequency} = \frac{1}{3000\times10^{-12}}$$
$$\text{Frequency} = 0.000333 \times 10^{12}$$
$$\text{Frequency} = 3.33 \times 10^{8}\ \text{GHz}$$

So, the frequency of your rudimentary processor would be `3.33` $\times10^{8}$ `GHz`

8. My Intel processor has 4 cores, each running at 1 GHz. A single instance of a program, Y, can run on any of the cores in 20 seconds, and the processor is fully capable of running multiple copies of the same program in parallel on each core, without slowing down any of the individual instances. I have another processor from AMD, which has only a single core, but is running at 4 GHz. As a result, program Y can finish on this core in 3 seconds. If I was to pick a processor so as to optimize the latency of a single instance of program Y, which processor should I pick? My friend, who is in charge of selecting processors for the datacenter servers, wants to pick a processor to optimize for system throughput. In both our cases, when measuring throughput, the unit of work is a batch of 4 instances of Program Y. Which processor should my friend pick (and why)? Explain your answer in both cases by providing quantitative arguments. Show your work.

We already know,

$$\begin{gathered}
\text{Clock Speed Intel} = 1\text{ GHz}\\
\text{Cores in Intel} = 4\\
\text{Execution Time of Program Y Intel} = 20\text{ seconds}
\end{gathered}$$

$$\begin{gathered}
\text{Clock Speed AMD} = 4\text{ GHz}\\
\text{Cores in AMD} = 1\\
\text{Execution Time of Program Y AMD} = 3\text{ seconds}
\end{gathered}$$

We already know that when optimizing for latency, we want the execution time taken for the program to be minimal. The `Intel` Processor takes 17 seconds more to complete, therefore you should pick the `AMD` processor.

For your friend in charge of selecting processors for the datacenter servers, we already know that `Throughput` can be defined as `Work Done` per unit `Time`.

Considering, we define the work done here as `Programs/Second`.

$$\begin{gathered}
\text{Intel can run } 4/20 \texttt{ Programs/Second}\\
\implies \text{Intel can run } 1/5 \texttt{ Programs/Second}\\
\implies \text{To run 4 instances, Intel takes } 5 \times 4 = 20 \texttt{ Seconds}\\
\text{So, } \texttt{Throughput of Intel} = \frac{WorkDone}{Time}\\
\implies \texttt{Throughput of Intel} = \frac{\frac{1}{5}}{20}\\
\implies \texttt{Throughput of Intel} = \frac{0.2}{20}\\
\implies \texttt{Throughput of Intel} = \frac{0.2}{20}\\
\implies \texttt{Throughput of Intel} = 0.01
\end{gathered}$$

Similarly,

$$\begin{gathered}
\text{AMD can run } 1/3 \texttt{ Programs/Second} \\
\implies \text{To run 4 instances, AMD takes } 3 \times 4 = 12 \texttt{ Seconds} \\
\text{So, } \texttt{Throughput of AMD} = \frac{WorkDone}{Time} \\
\implies \texttt{Throughput of AMD} = \frac{\frac{1}{3}}{12} \\
\implies \texttt{Throughput of AMD} = \frac{0.33}{12} \\
\implies \texttt{Throughput of AMD} = 0.0275
\end{gathered}$$

Under the assumption that Intel can run all 4 instances simaltaneously and there is no delay between re running the AMD processor for each instance, we can see that

Throughput for AMD is greater than that of Intel. Hence, AMD would again be the better choice.

9. Assume that I have designed a processor with 4 cores. The table below depicts the completion times and CPIs of each program. Assume that each program can be individually run on each core, without interference from other cores. All programs are started on their respective cores at the same time. Provided the information below, what is the combined throughput of the processor for the completion of all four programs?

We know that,

$$\begin{gathered}\text{CPU Execution Time} = \text{Clock Cycle Time (1/Clock Speed)} \times \text{CPI} \times \text{Number of Instructions} \\ \implies \text{CPU Execution Time} = \text{(1/Clock Speed)} \times \text{CPI} \times \text{Number of Instructions} \\ \implies \text{Number of Instructions} = \text{CPU Execution Time} \div \text{((1/Clock Speed)} \times \text{CPI)}\end{gathered}$$

Calculating the Number Of Instructions for Core 1,2,3,4

$$\begin{gathered}\text{No. Instructions Core \#1} = \text{CPU Execution Time} \div \text{((1/Clock Speed)} \times \text{CPI)} \\ \implies \text{No. Instructions Core \#1} = 4 \div (1 \times 2) \\ \implies \text{No. Instructions Core \#1} = 4 \div 2 \\ \implies \text{No. Instructions Core \#1} = 2\end{gathered}$$

Similary,

$$\begin{gathered}\implies \text{No. Instructions Core \#2} = 1 \\ \implies \text{No. Instructions Core \#3} = 2 \\ \implies \text{No. Instructions Core \#4} = 7\end{gathered}$$

To calculate the throughput, we have to consider the following. Firstly, the work done here refers to the `Total Number of Instructions` and, the `Unit Time` would be the `Max Core Time` ie, the time taken by the slowest Core, as that is when the process is completed. Using this, we can say that:

$$\texttt{Throughput} = \frac{\texttt{Total Instructions}}{\texttt{Max Core Time}}$$

$$\texttt{Throughput} = \frac{2+1+2+7}{8}$$

$$\texttt{Throughput} = \frac{12}{8}$$

$$\texttt{Throughput} = 1.5 \text{ Instructions Per Second}$$

Therefore the combined throughput is `1.5` Instructions Per Second.