Problem 1 (25%): Compute the Clocks Per Instruction (CPI) of a machine, which has an average CPI for ALU operations of 1.1, a CPI for branches/jumps of 3.0, and a hit rate of 60% in the cache. A hit in the cache takes 1 cycle and a cache miss takes 120 cycles. Assume 22% of instructions are loads, 12% are stores, 20% are branches/jumps and the rest are ALU operations.

$CPI_{ALU} := 1.1 \qquad CPI_{BJ} := 3.0$

$r_h := 60\% \qquad t_h := 1 \qquad t_m := 120 \qquad AMAT := r_h \cdot t_h + (1 - r_h) \cdot t_m = 48.6$

$CPI := (22 + 12)\% \cdot AMAT + (20)\% \cdot CPI_{BJ} + (100 - (22 + 12) - 20)\% \cdot CPI_{ALU} = 17.63$

$CPI = 17.63$

Problem 2 (25%): You are a processor designer and have to make a decision between building a processor, which executes at 1GHz and has an average CPI 1.2 and a processor, which executes at 2GHz, but has a CPI of 2. Which is better to build and why?

$f_1 := 1\ GHz \qquad CPI_1 := 1.2 \qquad L_1 := CPI_1 \cdot \frac{1}{f_1} = (1.2 \cdot 10^{-9})\ s$

$f_2 := 2\ GHz \qquad CPI_2 := 2 \qquad L_2 := CPI_2 \cdot \frac{1}{f_2} = (1 \cdot 10^{-9})\ s$

$speedup := \frac{L_1}{L_2} = 1.2$

As we can see in the calculations above, processor 1's latency is greater than processor 2's latency. Throwing it into the speedup equation, we see that it's 1.2. Since the speedup is greater than 1, it also indicates that L_1 is greater than L_2 which is consistent with our calculations.

In terms of latency, we want lower because it means that the processor is able to complete tasks more quickly, therefore we want processor 2.

Problem 3 (25%): A revolutionary new technology in memory improves your memory subsystem so that memory latencies are reduced by a factor of 3.5. After replacing your memory with the new ones, you observe that you now spend half your time waiting for memory. What percentage of the original execution (with the older memory system) was spent waiting for memory?

Let's say that with the new tech implemented, program execution takes 1 minute. From this we know that 30s is spent waiting on memory and the other 30s is spent on other operations

$L_2 := 60\ s \qquad t_{m2} := 30\ s \qquad t_r := 30\ s$

With a speedup of 3.5, we know the following:

$t_{m1} := 3.5 \cdot t_{m2} = 105\ s \quad t_{t1} := t_{m1} + t_r = 135\ s$

$$\frac{t_{m1}}{t_{t1}} = 77.778\%$$

Problem 4 (25%): You're currently using a single core machine but you want to figure out if it's worth investing in a dual-core machine. Assuming your application is 60% parallelizable, by how much could you decrease the frequency and get the same performance?

$x := 60\%$ $p_1 := 1$ $p_2 := 2$

$$S_{par} := \frac{1}{\left(\frac{x}{p_2} + (1 - x)\right)} = 1.429$$

$$L_1 := CPI_1 \cdot \frac{1}{f_1} = \left(1.2 \cdot 10^{-9}\right)\ s$$

$$L_2 := CPI_2 \cdot \frac{1}{f_2} = \left(1 \cdot 10^{-9}\right)\ s$$

Assuming our IC is the same, we know latency can be calculated as follows: $L = CPI \cdot \frac{1}{f}$

Assuming CPI is also the same, we'll solve for frequency:

Guess Values

$f_1 := 1\ GHz$ $f_2 := 1\ GHz$

Constraints

$$S_{par} = \frac{\frac{1}{f_1}}{\frac{1}{f_2}}$$

Solver

$$\begin{bmatrix} f_1 \\ f_2 \end{bmatrix} := \text{find}\left(f_1, \mathbf{f_2}\right) = \begin{bmatrix} 1 \\ 1.429 \end{bmatrix}\ GHz$$

$$\frac{1}{1.429} = 69.979\%$$

You'd have to reduce the frequency by 0.7