

Homework 2

1. CPU Time = (Instructions/program) * (Clock Cycles/Instruction) * (seconds/clock cycle)
 - a. Instructions/program – this refers to the number of instructions contained in the program being run
 - i. A factor that can impact this variable is if the program is using a lot of instructions to complete what it needs to do.
 - b. Clock Cycles/Instruction – this refers to the number of clock cycles needed to complete a single instruction; this is usually a weighted average based on the clock cycles to completion of each instruction multiplied by the frequency of that instruction
 - i. A factor that can impact this variable is if the instruction set that is being used is inefficient or if the program uses a majority of instructions that take many cycles to complete
 - c. Seconds/Clock Cycle – this refers to the number of seconds it takes for a single clock cycle
 - i. This major factor that can impact this variable is the clock speed of the processor
2.
 - a. $21/14 = 1.5$, Computer A is 1.5 times faster than computer B.
 - b.
 - i. One factor could be the processor speed. If both systems are running off the same instruction set then computer A would have a faster processor than B allowing it to complete more instructions per cycle
 - ii. Another factor could be the instruction set being used. Depending on the instruction set used on computer B, it could be requiring more cycles to complete each instruction thereby making it complete the entire program slower than computer A.
3.
 - a. $M1, 2*(.6) + 3*(.3) + 3*(.1) = 2.4 \text{ CPI}$
 - b. $M2, 3*(.6) + 2*(.3) + 1*(.1) = 2.5 \text{ CPI}$
4.
 - a. $M1\text{-MOD CPI} = 2*(.6) + 2*(.3) + 3*(.1) = 2.1 \text{ CPI}$, CPU Time = $IC * 2.1 * 1 / (2.0 * (10^9))$
 - i. $(IC / 2 * 10^9) / (IC / 2 * 10^9) * (2.4 / 2.1) = 1.143$
 - ii. M1-MOD is 1.143 times faster than M1
 - b. $M2\text{-MOD CPI} = 2*(.6) + 2*(.3) + 1*(.1) = 1.9 \text{ CPI}$, CPU Time = $IC * 1.9 * (1 / (2.2 * 10^9))$
 - i. $(IC / 2.2 * 10^9) / (IC / 2 * 10^9) * 2.5 / 1.9 = 1.316$
 - ii. M2-MOD is 1.316 times faster than M2
5. By running two instructions per cycle of type A, you would be able to cut that in half.
 - a. $M1\text{-MOD CPI} = 2.1$, $M1\text{-MOD w/Hardware CPI} = 1*(.6) + 2*(.3) + 3*(.1) = 1.5 \text{ CPI}$
 - i. $(IC / 2 * 10^9) / (IC / 2 * 10^9) * (2.1 / 1.5) = 1.4$
 - ii. M1-MOD w/Hardware is 1.4 times faster than M1-MOD
 - b. $M1 \text{ CPI} = 2.4 \text{ CPI}$

- i. $(IC/2 \times 10^9)/(IC/2 \times 10^9) * (2.4/1.5) = 1.6 \text{ CPI}$
 - ii. M1-MOD w/Hardware is 1.6 times faster than M1
- 6. CPU-A - CPI_a = 1.3, Clock Rate = 1.2×10^9 , compiled instructions 10^6
 CPU-B - CPI_b = 2.1, Clock Rate = 2.5×10^9
 - a. $(1 \times 10^6)(1.3)(1/(1.2 \times 10^9)) = \text{instructions}(2.1)(1/(2.5 \times 10^9))$
 - i. Instructions = 1.289×10^6
 - b. Assuming IC = 1×10^6
 - i. CPU Time A = $10^6 * 1.3 * (1/(1.2 \times 10^9)) = 0.00108333s$
 - ii. CPU Time A * .5 = $0.00541667s$
 - iii. CPU Time Bnew = $10^6 * CPI_{Bnew} * (1/(2.5 \times 10^9)) = CPI_{Bnew} * 0.0004$
 - iv. CPU Time Bnew = $CPI_{Bnew} * 0.0004 = 0.00541667s$
 - v. $CPI_{Bnew} = 1.354$
 - c.
 - i. One factor may be the that it could be very data intensive so there is a low cpu time but a higher time overall because of the transfer speeds from memory
 - ii. Another factor could be that the processor was required to handle other programs or aspects of the system at the same time meaning that it would push the processes of foo.exe aside temporarily to complete them.
- 7. Assuming these all have the same CPI as it is not otherwise stated, applying Amdahl's Law would give the following:
 - a. If you were to make the Floating point multiplication run 10x faster:
 - i. $T_{imp} = (.5 \times T)/10 + (.5 \times T) = .55T$
 - ii. Applying the relative performance equation gives $T/.55T = 1.818$
 - b. If you were to make the Floating point Division 4x faster:
 - i. $T_{imp} = (.3T)/4 + (.7 \times T) = 0.775T$
 - ii. Applying the relative performance equation gives $T/0.775T = 1.290$
 - c. As such, neither of these options will result in the machine running this program 2x faster than before.