(c) [6 points] 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?

exec time = CPU time = instructions x CPI / clock rate

= exe-time (new) = 0.7 exec-time (old)

same goes for the instructions

Just moving variables around

we know that the number the program takes to execute is 10 seconds

30% of 10 seconds is 7

So that means that new CPI → CPI (new) = 0.7 · CPI (old)

new clock rate old clock rate

CPI (new) = 1.2 CPI

$\frac{1.2}{\text{new clock-rate}} = \frac{0.7}{\text{old...}}$ $= \frac{1.2}{0.7}$ x old clock rate = 1.71 x old clock rate

Clock rate must increase by 71% to get time down (reduced)

3. 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 2.5GHz and CPIs of 1, 2, 3, and 3, and P2 with a clock rate of 3GHz 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.

(a) [6 points] What is the global CPI for each implementation?

CPI = clock cycle / Instruction count

Came to find out that $1.0E6 = 10^6 = 1{,}000{,}000 = \sum_{n=1}^{n} (CPI_i \times \text{instruction count}_i / \text{instruction count})$

$10^6 \times 10\% = 10^5 = 100{,}000$

$10^6 \times 20\% = 2 \times 10^5$

$10^6 \times 50\% = 5 \times 10^5$

$10^6 \times 20\% = 2 \times 10^5$

P1 clock cycle: $(1 \times 10^5) + (2 \times 2 \times 10^5) + (3 \times 5 \times 10^5) + (3 \times 2 \times 10^5)$

$= 2.6 \times 10^6$

P2 clock cycle: $(2 \times 10^5) + (2 \times 2 \times 10^5) + (2 \times 5 \times 10^5) + (2 \times 2 \times 10^5)$

$= 2 \times 10^6$

CPI = clock cycles / # of instructions

$P_1 \rightarrow CPI = \frac{2.6 \times 10^6}{10^6} = 2.6$

$P_2 \rightarrow CPI = \frac{2 \times 10^6}{10^6} = 2$