5 Performance Evaluation [35 points]

A multi-cycle processor $P1$ executes *load instructions* in **10 cycles**, *store instructions* in **8 cycles**, *arithmetic instructions* in **4 cycles**, and *branch instructions* in **4 cycles**. Consider an application A where 20% of all instructions are load instructions, 20% of all instructions are store instructions, 50% of all instructions are arithmetic instructions, and 10% of all instructions are branch instructions.

(a) [5 points] What is the CPI of application A when executing on processor $P1$? Show your work.

$CPI = 0.2 \times 10 + 0.2 \times 8 + 0.5 \times 4 + 0.1 \times 4$
$CPI = 6$

(b) [10 points] A new design of the processor doubles the clock frequency of $P1$. However, the latencies of the load, store, arithmetic, and branch instructions increase by 2, 2, 2, and 1 cycles, respectively. We call this new processor $P2$. The compiler used to generate instructions for $P2$ is the same as for $P1$. Thus, it produces the same number of instructions for program A. What is the CPI of application A when executing on processor $P2$? Show your work.

$CPI = 0.2 \times 12 + 0.2 \times 10 + 0.5 \times 6 + 0.1 \times 5$
$CPI = 7.9$

(c) [10 points] Which processor is faster ($P1$ or $P2$)? By how much? Show your work.

$P2$ is 1.52× faster than $P1$.

Explanation.
$Execution\_Time\_P1 = instructions \times CPI_{P1} \times clock\_rate$
$Execution\_Time\_P2 = instructions \times CPI_{P2} \times \frac{clock\_rate}{2}$
$clock\_rate = \frac{1}{clock\_frequency}$

Assuming that $Execution\_Time\_P2 < Execution\_Time\_P1 \implies \frac{Execution\_Time\_P1}{Execution\_Time\_P2} > 1$. Thus:
$\implies \frac{instructions \times CPI_{P1} \times clock\_rate}{instructions \times CPI_{P2} \times \frac{clock\_rate}{2}}$
$\implies \frac{6 \times clock\_rate}{7.9 \times \frac{clock\_rate}{2}}$
$\implies \frac{6}{3.95}$
$\implies 1.52$

(d) [10 points] There is some extra area available in the chip of processor $P1$, where extra hardware can fit. You can decide to include in your processor a faster branch execution unit or a faster memory device. The faster branch execution unit reduces the latency of branch instructions by a factor of 4. The memory device reduces the latency of the memory operations by a factor of 2. Which design do you choose? Show your work.

A faster memory device.

Explanation.
Application A executes 10% of branch operations and 40% of memory operations (load and stores).
By Amdahl's Law, we have:

$Speedup_{branch} = \frac{1}{(1-0.1)+\frac{0.1}{4}} = 1.08$
$Speedup_{memory} = \frac{1}{(1-0.4)+\frac{0.4}{2}} = 1.25$

Therefore, the new memory device provides more speedup than the faster branch execution unit, for this particular application.

Alternative Solution.
In case we decide to reduce the latency of the branch operations, the new CPI of processor $P1$ will be:
$CPI_{branch} = 0.2 \times 10 + 0.2 \times 8 + 0.5 \times 4 + 0.1 \times \frac{4}{4}$
$CPI_{branch} = 5.7$

In case we decide to reduce the latency of the memory operations, the new CPI of processor $P1$ will be:
$CPI_{memory} = 0.2 \times \frac{10}{2} + 0.2 \times \frac{8}{2} + 0.5 \times 4 + 0.1 \times 4$
$CPI_{memory} = 4.2$

Since $CPI_{memory} < CPI_{branch}$, improving the memory device will provide shorter cycles-per-instructions.