Q1 When parallelizing an application, the ideal speedup is speeding up by the number of processors. This is limited by two things: percentage of the application that can be parallelized and the cost of communication. Amdahl's law takes into account the former but not the later.

a. What is the speedup with N processors if 80% of the application is parallelizable, ignoring the cost of communication?

b. What is the speedup with 8 processors if, for every processor added, the communication overhead is 0.5% of the original execution time?

c. What is the speedup with 8 processors if, for every time the number of processors is doubled, the communication overhead is increased by 0.5% of the original execution time?

Q2 Consider the unpipelined processor introduced previously. Assume that it has a 1 ns clock cycle and it uses 4 cycles for ALU operations and branches, and 5 cycles for memory operations, assume that the relative frequencies of these operations are 40%, 20%, and 40%, respectively. Suppose that due to clock skew and setup, pipelining the processor adds 0.2ns of overhead to the clock. Ignoring any latency impact, how much speedup in the instruction execution rate will we gain from a pipeline?

Que 3: What are different techniques for the pipeline optimization? Explain all techniques.

Que 4: What are different compiler techniques for improving the performance? Explain all techniques.