**Questions on Chapter 1**

1. Give some examples of applications that need high computational power.
2. Why parallel systems are built?
3. Describe how n processors can be used to compute and add n values in an efficient way.
4. Explain the difference between task and data parallelism.
5. Describe the difference between shared and distributed memory systems.
6. Explain the concurrent, parallel, and distributed concepts.

**Questions on Chapter 2**

1. Draw and explain the von Neumann architecture.
2. Explain the concepts: processes, multitasking, and threads.
3. How cache can speed up a program execution?
4. What is the difference between cache hit and cache miss?
5. Suppose the main memory consists of 16 lines with indexes 0–15, and the cache consists of 4 lines with indexes 0–3. Where lines should be stored using direct, fully associative, and 2-way mapping.
6. With an example show how matrix processing can be efficient by considering how cache works.
7. What is the drawback of page table and how this drawback can be solved?
8. With examples, describe the pipelining and speculation concepts.
9. What is the difference between coarse-grained and fine-grained parallelism?
10. Describe SIMD system.
11. What is the characteristic of vector processers?
12. Describe the MIMD systems.
13. Draw the architecture for UMA and NUMA multicore systems and describe them.
14. Describe the crossbar as interconnect for shared memory system.
15. Draw ring, toroidal mesh, fully connected network, hypercube, crossbar, and omega network for eight processors and compute the bisection width for each interconnect.
16. Show the difference between crossbar and omega network switches.
17. Describe the latency concept.
18. Describe with example how cache coherence problem can happen.
19. Explain the concepts: snooping cache coherence, directory-based cache coherence, and false sharing.
20. Which approach to use: distributed memory or shared memory?
21. Give the structure of SPMD programs.
22. Describe the difference between static and dynamic threads.
23. Give examples for nondeterminism when processors execute asynchronously.
24. How mutex and busy waiting can be used to ensure only one thread executes certain instructions at a time?
25. Calling functions designed for serial programs can be problematic in parallel program, give an example.
26. Write lines of code for sending a message from process 0 to process 1.
27. Give examples of functions for collective communication.
28. What one sided communication means?
29. How remote memory access affects the program performance?
30. A lot of issues can appear with input and output in parallel system, give some rules to avoid these issues.
31. Explain the concepts speed up and efficiency.
32. What is meant by Amdahl’s low?
33. When a program is weakly or strongly scalable?
34. How running time of serial and parallel programs can be computed?
35. Outline the foster’s methodology.
36. Apply the foster’s methodology for making histogram out of data.