B.E / B.Tech (Full Time) DEGREE EXAMINATIONS, APRIL / MAY 2013

INFORMATION TECHNOLOGY, VII Semester

CS9043 MULTI-CORE PROGRAMMING

(Regulation 2008)

Time : 3 Hours | Answer ALL Questions | Max. Marks 100

PART-A (10 x 2 = 20 Marks)

1. State Gustafson's law. What is its significance in the context of multi-core processors ?
2. What is the main reason for the shift towards multi-core processors ? What is the major challenge because of this shift ?
3. How does a live-lock occur ?
4. What is the difference between a fence and a barrier ?
5. What are the constructs/functions you will use in openMP to do the following :

(a) To specify the global variables that need to be private for each thread.

(b) To ensure that any one thread executes a piece of code.

6. How is fine-grained locking useful ? Give an example for any data structure.
7. What is the ABA problem in non-blocking algorithms ?
8. What are the four different types of sends supported in MPI ?
9. What would the various MPI collective functions do if the communicator contains a single process ?
10. Consider a program that has been cache-optimized for a single-core processor. How would the performance be affected if a parallel version of the same were run on a multi-core processor?

PART-B (5 X 16 = 80 Marks)

11. (i) Increasing the number of threads does not always lead to better performance. Give reasons for why this is so. (4)

(ii) Discuss the cache and memory related issues that need to be addressed when writing parallel programs. (6)

(iii) How can the sieve of Eratosthenes program be optimized to be cache-friendly? (6)

12. (a) (i) Discuss Flynn's classification of parallel systems. Which category do common multi-core processors fall into ? (5)

(ii) Distinguish between software and hardware multithreading. (6)

(iii) What are the specific architectural features added in multi-core architectures ? Discuss with respect to any multi-core processor. (5)

OR

(b) (i) State Amdahl's law. As per Amdahl's law how many cores would be required to achieve a speedup of 4 if 80% of the task was parallelizable ? (5)

(ii) Distinguish between ILP, TLP and DLP. (5)

(iii) Discuss the cache related issues in shared memory multi-processor architectures. (6)

13. (a) (a) (i) Explain each of the different parallel programming patterns commonly used with an example. (10)

(ii) Show how each of the three basic atomic operations of a condition variable can be implemented using semaphores. (6)

OR

(b) (i) Explain the different synchronization primitives. Specify applications suitable for each. (10)

(ii) Parallelize the following loop. Which programming pattern does this fall into ?

```
for (i=1, i<= 1024, i++) {
        Sum[i] = 0;
        for (j=1, j<=i; j++)
                Sum[i] = sum[i] + b[j];
}
```

(6)

14. (a) (i) Write a parallel program using openMP to multiply two nxn matrices. Estimate the speed-up expected with k processors. (10)

(ii) Explain the problem of data races and how they can be handled. (6)

OR

(b) (i) Write a cache-friendly parallel code using openMP for sorting a large set of numbers. Provide comments to explain your code. (10)

(ii) Discuss the various work-sharing constructs used in OpenMP. (6)

15. (a) (i) Write parallel code for counting the number of words in a huge document using MPI. (10)

(ii) What would be the behaviour of the following send/receive operations in an MPI program.

```
Process X :
        MPI_Send(msg1, ..., Y, 200,..)
    MPI_Rsend(msg2, ..., Y, 250,..)
Process Y :
        MPI_Recv(store@a, ..., X, 200)
    MPI_Recv(store@b, ..., X, 250)
```

(6)

OR

(b) (i) Write parallel code using MPI for a shortest path algorithm of your choice. Explain how you are partitioning the problem, and estimate the speed-up expected. (10)

(ii) Explain the different collective communication functions in MPI. (6)

***** GOOD LUCK *****