REG NO 7/10/15

B.E / B.Tech (Full Time) DEGREE EXAMINATIONS, OCT/NOV 2013 15

INFORMATION TECHNOLOGY, VI Semester

CS531 MULTI-CORE ARCHITECTURE AND PROGRAMMING/

CS9043 MULTI-CORE PROGRAMMING

(REGULATIONS 2004/2008)

Time : 3 Hours Answer ALL Questions Max. Marks 100

PART-A (10 x 2 = 20 Marks)

1. State Amdahl's law. What is its significance in the context of multi-core processors ?
2. What are the challenges in UMA and NUMA types of architectures ?
3. List four parallel programming design patterns commonly used.
4. How are critical sections to be handled when programming for multi-core processors ?
5. What are the constructs/functions you will use in openMP to do the following :
   (a) To wait for all threads to come to a point.
   (b) To ensure that all threads get an equal share of the computation.
6. Show how an openMP program may end-up in a deadlock.
7. What are non-blocking algorithms ?
8. When would you use non-blocking send in MPI ?
9. What are the parameters to be specified for the scatter operation in MPI ?
10. Consider a program that has been cache-optimized for a single-core processor. How would the performance be affected if multiple copies of the same program were run on a multi-core processor?

PART-B (5 X 16 = 80 Marks)

11. (i) Differentiate ILP, DLP from TLP. Which of these are supported on current multi-core processors ? (10)
    (ii) Discuss the challenges that need to be handled in distributed shared memory architectures. (6)

12. (a) (i) Explain the use of various types of locks for synchronization. (10)
    (ii) Give an example program that uses condition variable for the producer-consumer problem. (6)

    OR

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

(ii) Analyze the following loop for parallelism. Which programming pattern does this fall into ?

```
for (i=1, i<= 1024, i++) {
        for (j=1, j<=1024; j++)
                if (a[i][j] < 128)
                b[i][j] = 0;
                else b[i][j]=1;
        c=a[i][j] – b[i][j];
        a[i+1][j+1] += c/16;
        }
}
```

(6)

13. (a) (i) Write a parallel program using openMP to find the maximum number in an nxn matrix. Estimate the speed-up expected with k processors. (10)

(ii) Explain the problem of livelock and how it can be handled. (6)

OR

(b) (i) Explain the OpenMP constructs for the following – reduction, schedule, and work-sharing. (10)

(ii) Explain when cache-line ping-ponging occurs in non-blocking algorithms. (6)

14. (a) (i) Write parallel code for obtaining the frequency of words in a huge document using MPI. (10)

(ii) Explain the different point-to-point communications functions used in MPI. (6)

OR

(b) (i) Write parallel code using MPI for performing binary search. Explain how you are partitioning the problem, and estimate the speed-up expected. (10)

(ii) Explain the use of scatter and gather operations in MPI. (6)

15. (a) (i) Explain the memory and cache related issues to be considered when optimizing code for multi-core processsors. (10)

(ii) How would you determine the optimum number of threads to be used ? (6)

OR

(b) (i) Explain the challenges faced in debugging openMP and MPI programs. What support is required from debug tools for these programs ? (10)

(ii) What are the parameters that affect the performance of an MPI program ? Explain. (6)

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