University of Hyderabad, Department of Computer and Information Sciences

CS726-Parallel Computing

Max. Marks: 60 **Date: November 25, 2009**

Duration: 3 Hrs.

Note:
1. Attempt **all** questions, each carry equal marks
2. Each question contains **3** sub questions. Attempt **any and only 2** questions
3. Answer sub questions of a question before attempting the next question
4. In case of any doubt, mention your assumptions in the answer-book and proceed for your answers

Q.1 (A) How can we improve the performance of a single processor system? What kind of automatic parallelisms are available in the single processor system?

(B) Write a C/OpenMP program to Merge two sorted lists. What happens if number of elements equals to number of threads?

(C) How the cache coherent problem addressed in a UMA based multiprocessor system?

Q.2 (A) Propose a control parallel solution for the Sieve of Eratosthenes. How is it different from data parallelism? Comment on parallel time and processors complexity.

(B) Explain the butterfly network of processing elements.

(C) Propose a parallel algorithm strategy for the Bucket Sorting. Comment on parallel time and processors complexity.

Q.3 (A) What is Amdahl's law? What is the draw back of his law? If 5% of a program's execution is done sequentially, how many processors are needed to achieve a speedup of 20 and how?

(B) What is Hyper threading? How is it different from Chip Multiprocessing?

(C) Write a parallel C/MPI program to calculate value of **pi** by numerical integration.

Q.4 (A) Explain how to achieve simultaneous writes in a PRAM model? Is it possible to convert any CREW algorithm to CRCW model? If yes, how?

(B) Compare Front Side Bus, HyperTransport and Quick Path Interconnects for Multicore machines.

(C) Explain the process of finding global multiplication of ***n*** numbers using Foster's design methodology.

Q. 5(A) n = 2^m numbers stored in an array A of dimension (2n-1) from A(n), A(n+1),...,A(2n-1). Write a parallel algorithm for obtaining the prefix sum of these numbers, at the end A(i), $1\leq i \leq n$ stores the result. Comment on parallel time and processors complexity.

(B) What mechanisms are used for ensuring mutual exclusion of critical sections?

(C) Explain the working behavior of the following program segment? Assume no syntax errors in the code.

```
#include <stdio.h>
#include <omp.h>
#define CHUNKSIZE 5
#define N 10
int main(int argc, char *argv[])
{
    int i, chunk_size;
    float a[N], b[N], total[N];
    for (i = 0; i < N; i++)
         a[i] = b[i] = i * 1.0;
    chunk_size = CHUNKSIZE;
    #pragma omp parallel shared(a,b,total,chunk_size) private(i)
    {
        #pragma omp for schedule(dynamic,chunk_size) nowait
        for (i=0; i < N; i++)
             total[i] = a[i] + b[i];
    }
    for (i = 0; i < N; i++)
       printf(“The total value is = %f\n”, total[i]);
}
```

------Best of Luck------