University of Hyderabad, Department of Computer and Information Sciences

Parallel Computing

Max. Marks: 60 **Date: January 10, 2008**

Duration 3.00 Hrs.

Note: 1. Attempt **any and only ten** questions.
2. Draw neat diagrams, if needed.

Q.1 Explain the role of ***diameter*** and ***bisection-width*** that is used to understand effectiveness in implementing efficient parallel algorithms on cube connected multiprocessor system? **[6]**

Q.2 Explain the ***P-RAM*** model of computation. Which realistic machine uses this model of computation? Why does it reduce the complexity of a sequential algorithm? **[6]**

Q.3 Explain with an example how do we achieve decentralize dynamic load balancing in a message passing program. **[6]**

Q.4 Consider the following program segment of an Open_MP program and comment on the execution. **[6]**

```
#include <stdio.h>
int main(int argc, char *argv[])
{
    int i = 0, j = 0;
    int result = 0;
    #pragma omp parallel for private(i) reduction(+:result)
    for (i = 0; i < 3; i++) {
       for (j = i + 1 ; j < 4; j++) {
           printf("Hello.\n");
          #pragma omp critical
           result = result + 1;
        }
     }
    printf("Number of times printed Hello = %d\n", result);
}
```

Q.5 If $n = 2^m$ numbers stored in an array A of dimension (2n-1) from A[n], A[n+1],..., A[2n-1]. Write a PRAM algorithm to compute prefix product such that at the end A[i] stores A[1]*A[2]*....*A[i]. **[6]**

Q.6 What are the qualifiers used to explicitly qualify variables in a parallel loop of an OpenMp program. Explain working of any two. **[6]**

Q.7 What are the alternatives for programming shared memory multiprocessors . Which one is better and why? **[6]**

Q.8 Explain Foster’s design methodology with an example. **[6]**

Q. 9 Explain the working behavior of the following program segment and write your comments. Assume that there is no syntax error in the program segment. **[6]**

```
int a[10], b[10], npes, myrank;
MPI_status status;
…
MPI_Comm_size(MPI_COMM_WORLD, &npes);
MPI_Comm_rank(MPI_COMM_WORLD, &myrank);
MPI_Send(a, 10, MPI_INT, (myrank+1)%npes, 1, MPI_COMM_WORLD);
MPI_Recv(b, 10, MPI_INT, (myrank-1+npes)%npes, 1, MPI_COMM_WORLD);
…
```

Q.10 Devise a parallel algorithm for sorting **n** numbers. What is the parallel time and processors complexity? **[6]**

l How do you compute the integration of a given function using mean value theorem? Propose a multithreaded program for your algorithm. **[6]**

Max

Note

Q.1

Q.2

Q.3

Q.4

Q
.
1
1

Q.12 List Advantages and **[6]**
Disadvantages of using asymmetrical multi-computers.

Q.13 Define: **[6]**

i. Efficient parallel algorithm
ii. Optimal parallel algorithm
iii. Brent's theorem
iv. Amdhal's law

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