[This question paper contains 4 printed pages.]

**Your Roll No...............**

| | | | |
|---|---|---|---|
| **Sr. No. of Question Paper** | : | **5322** | **L** |
| Unique Paper Code | : | 2344000033 | |
| Name of the Paper | : | Introduction to Parallel Programming | |
| Name of the Course | : | **Computer Science General Elective** | |
| Semester | : | VIII | |

Duration : 3 Hours Maximum Marks : 90

**Instructions for Candidates**

1. Write your Roll No. on the top immediately on receipt of this question paper.
2. **SECTION A** is compulsory.
3. Attempt any **4** questions from **SECTION B**.
4. Parts of a question must be answered together.

**SECTION A**

1. (a) What is cache coherence? Give any two reasons why it is required in shared-memory parallel systems. [3]

(b) List any three differences between Concurrent Computing and Parallel Computing. [3]

(c) What is Single Program Multiple Data (SPMD) model? List any two capabilities of SPMD model. [4]

(d) Define *speedup* and *efficiency* in parallel computing. Given a computational problem with 200 operations, determine the *speedup* and *efficiency* using 50 processors, where 150 operations are executed in parallel and 50 operations are executed sequentially. [4]

(e) Describe critical section in OpenMP with the help of an example. Distinguish between *named* and *unnamed* `#pragma omp critical` directives. [4]

(f) Explain the role of the following constructs: [4]

i. `MPI_Sendrecv`

ii. `nowait`

*P.T.O.*

(g) Consider a 5-stage pipelined processor consisting of Instruction Fetch (IF), Instruction Decode (ID), Operand Fetch (OF), Execute (EX), and Write Back (WB) stages. The processor supports dual instruction issue per cycle. [4]
Construct the pipeline execution schedule for the following instruction sequence:

```
Load R5, @4000
Load R6, @4008
Sub R5, @4004
Mul R6, @400C
Add R5, R6
Store R5, @5000
```

(h) Explain message passing protocol for buffered communication in which buffering is handled exclusively by the sending process. [4]

## SECTION B

2. (a) Explain superscalar execution with the help of an example. List three types of dependencies involved in superscalar instruction execution. [7]

(b) Consider a system with a 2 GHz processor that uses DRAM with a latency of 80 ns and a 64 KB cache with a latency of 0.5 ns. The system performs multiplication of two matrices A and B, each of size 64×64, to produce a resultant matrix C, where each element occupies 8 bytes. The processor is capable of executing 8 instructions per cycle. [8]

i. Compute the total number of floating-point operations required to multiply matrices A and B.
ii. Determine the peak computational throughput of the processor in GFLOPS.
iii. Estimate the total memory required to store matrices A, B, and the resultant matrix C.
iv. Justify whether the given multiplication problem is compute-bound or memory-bound.

3. (a) Discuss the structure of `MPI_Status` with the help of an example. How does it enable dynamic message handling in MPI? [7]

   (b) Explain the following OpenMP functions: [8]
      i. `omp_get_num_threads()`
      ii. `omp_get_max_threads()`
      iii. `omp_in_parallel()`
      iv. `omp_get_dynamic()`

4. (a) What is the role of latency hiding in parallel computing? Give any four differences between prefetching and multithreading techniques of latency hiding. [7]

   (b) Consider a memory system with a single cycle cache and 120 cycle latency DRAM with the processor operating at 2 GHz. If the block size is one word, the processor takes 120 cycles to fetch each word. For each pair of words, the operations performs 2 FLOPs. [8]
   Answer the following:
      i. Calculate the peak FLOP rate (in MFLOPS) for block size of four words.
      ii. Calculate the speedup achieved when block size is increased from four words to six words. Also, explain the reason for this improvement.

5. (a) Explain the `private`, `firstprivate`, and `shared` data-sharing clauses in OpenMP. Using these clauses, write a parallel program to multiply each element of an array by a constant. Clearly specify the variables associated with each clause and justify your choices. [7]

   (b) Consider the following code: [8]

```
int count = 0;
#pragma omp parallel for
for(int i = 1; i <= 50; i++)
  count++;
```

      i. What will be the final value of `count` in serial execution?
      ii. Identify the problem, if the above code is executed in parallel.
      iii. Rewrite the code using an appropriate OpenMP clause to fix the issue identified in part ii.

iv. Justify, whether a critical section or a reduction clause provides better performance for the given code.

6. (a) Write the pseudocode of Odd–Even Sorting algorithm and show its step-by-step execution on the array `[10, 7, 2, 9, 4, 6]` using six processors. [7]

(b) Explain the role of following MPI routines: [8]

```
i.    MPI_Init
ii.   MPI_Comm_size
iii.  MPI_Comm_rank
iv.   MPI_Finalize
```

Also, write an MPI program illustrating their usage.

7. (a) What is the role of the *schedule* clause in OpenMP in loop parallelization? Illustrate static and dynamic scheduling strategies of loop execution with an example. [7]

(b) Explain the following interconnection network topologies: [8]

i. Crossbar network
ii. K-d Mesh
iii. Hyper Cube
iv. Completely Connected Network