13

Roll No. □□□□□□□□□□

B.E (FT) END SEMESTER EXAMINATIONS – APR / MAY 2025

Computer Science and Engineering
Eighth Semester
CS6011 GPU Computing
(Regulation 2018 - RUSA)

Time: 3 Hours Answer ALL Questions Max. Marks 100

PART-A (10 x 2 = 20 Marks)

1. How does memory coalescing improve the performance of global memory accesses in CUDA?
2. Why is it important to consider warp size when designing CUDA programs?
3. How does CUDA handle overlapping data transfer and computation?
4. List down of CUDA tools that help developers in analyzing application's performance.
5. What are common problems faced by CUDA programmers ?
6. Why use #pragma omp for in OpenMP?
7. How does MPI_Comm_size help in MPI programming?
8. Why is MPI_Barrier used in MPI programs?
9. What does the #pragma acc kernels directive do in OpenACC?
10. Explain the concept of a 'gang' in OpenACC task granularity.

PART – B (8 x 8 = 64 marks)

(Answer any 8 questions)

11. Explain the CUDA programming model with emphasis on threads, blocks and grids.
12. Explain the differences between CUDA global memory, shared memory, constant memory and texture memory.
13. Explain the use of pinned memory in overlapping data transfer and computation in CUDA.
14. Describe APOD cycle in detail.
15. Describe CUDA error handling APIs and explain how they can be used for error checking.
16. Explain synchronization problems along with possible solutions.
17. Compare static and dynamic scheduling in OpenMP with examples.
18. Discuss on Parallel patterns using Convolution.
19. Discuss OpenMP task creation, task synchronization, and task dependencies. Include appropriate code examples.
20. Explain the message-passing model in MPI with an example.

21. Describe the role of non-default streams in CUDA programming.

22. Discuss data management in OpenACC. Write an example program to demonstrate copyin, copyout, and create clauses.

PART – C (2 x 8 = 16 marks)

23. Discuss the concept of thread synchronization in CUDA. Write a CUDA program that demonstrates the use of __syncthreads() in a matrix addition operation. Explain why synchronization is necessary and how it ensures correct results.

24. Write an MPI program to calculate the sum of ranks using MPI_Reduce. Explain the code.