PARALLEL COMPUTER ARCHITECTURE AND PROGRAMMING

[Revised Credit System]

(Effective from the academic year 2024-2025)

SEMESTER – VI

Subject Code	**CSE 3221**	**IA Marks**	**50**
Number of Lecture Hours/Week	**03**	**Exam Marks**	**50**
Total Number of Lecture Hours	**36**	**Exam Hours**	**03**

CREDITS - 03

Course objectives: This course will enable students to

- Describe parallel computer structures, architectural classification schemes, architecture of GPU, and different programming languages and models.
- Illustrate point-to-point and collective communication primitives in MPI.
- Understand the basic programming concepts and write efficient kernel functions using OpenCL and CUDA
- Implement CUDA programs using shared memory, synchronization of threads, image processing functions and parallel patterns

Module -1	**Teaching Hours**
INTRODUCTION TO PARALLEL ARCHITECTURES: Introduction to Parallel processing, Parallel Computer Structures: Pipeline Computers, Array Computers, Multiprocessor Systems, Architectural Classification Schemes: Multiplicity of Instruction-Data streams, Introduction, GPUs as parallel computers, Architecture of a modern GPU, Need for parallelism, Parallel programming languages and models **Textbook 4:** Chapter 1.1.2,1.3.1-1.3.3, 1.4.1 **TextBook 1**: 1.1, 1.2, 1.3, 1.4	**5 Hours**
Module -2	
MESSAGE PASSING PROGRAMMING: Introduction, Message passing model, MPI basic data types and functions, Point-to-point communication- MPI_Send, MPI_Recv, MPI_Ssend, MPI_Bsend, Collective communication- MPI_Bcast, MPI_Scatter, MPI_Gather, MPI_Reduce, MPI_Allgather, MPI_Alltoall, MPI_Scan, Benchmarking parallel performance, MPI error handling functions. **Text Book 2:** Chapter 4.1, 4.2, 4.4.1 - 4.4.5, 4.5, 4.6, 6.5	**6 Hours**
Module – 3	
OpenCL ARCHITECTURE AND PROGRAMMING: Introduction, OpenCL standard, OpenCL specification, Kernels and openCL execution model, Platform and Devices, Execution Environment- Context, Command Queues, Buffers, Program Object and Kernel Object, Program layout, Memory model, Writing Kernels, OpenCL APIs, OpenCL programs for vector-vector addition **Text Book 3 :** Selected Topics from Chapter 2	**3 Hours**

Module-4	
INTRODUCTION TO CUDA AND COMPUTATIONAL PATTERNS: Introduction, Data Parallelism, CUDA Program Structure, Vector-Vector addition, Device global memory and Data transfer, Kernel functions and Threads, Runtime APIs and Error Handling, 1D Sequential convolution, 1D Parallel Convolution, Atomic and Arithmetic functions, Constant Memory and caching, Tiled 1D Convolution with halo elements, A simple parallel scan algorithm, Sequential Sparse – Matrix vector multiplication, Parallel SPVM using CSR **Text Book 1:** Chapter 3.1, 3.2, 3.3, 3.4, 3.5, 8.1, 8.2, 8.3, 8.4, 8.5, 9.2, 10.1, 10.2	**13 Hours**
Module-5	
CUDA THREADS AND MEMORY ORGANIZATION: Introduction, CUDA Thread Organization, Mapping threads to multidimensional data, Matrix-Matrix multiplication, Importance of Memory Access Efficiency, CUDA Device Memory types, Synchronization and transparent scalability, Assigning resources to blocks, Querying device properties, Thread scheduling and latency tolerance, A strategy for reducing global memory traffic, A tiled matrix-matrix multiplication kernel, Parallel image processing applications **Text Book 1:** Chapter 4.1-4.7, 5.1, 5.2, 5.3, 5.4 **Text Book 5** :Chapter 3.1, 3.2	**9 Hours**

Course outcomes:

After studying this course, students will be able to:

1. Explain the concepts of parallel computer architectures and programming models.
2. Develop MPI programs using point-to-point and collective communication primitives.
3. Solve parallel programming tasks using OpenCL and CUDA.
4. Apply CUDA programming concepts for different parallel patterns.
5. Design optimized parallel solutions using thread and memory organization in CUDA Parallel image processing concepts.

Text Books:

1. D. Kirk and W. Hwu , "*Programming Massively Parallel Processors –A Hands-on approach*", Elsevier Inc.,2nd Edition, 2013.
2. Michael J. Quinn, "*Parallel Programming in C with MPI and OpenMP*", McGraw Hill Edition, 2003.
3. Benedict R. Gaster, Lee Howes, David R, Perhaad Mistry, Dana Schaa, "*Heterogeneous Computing with OpenCL*", Elsevier Inc., 1st Edition, 2012.
4. Kai Hwang and Faye A. Briggs, *Computer Architecture and Parallel Processing*, (2e), TMH Private Ltd., 2012.
5. Gonzalez, Rafael C., and Richard E. Woods. "Digital image processing" *Publishing house of electronics industry*141.7 (2002).

Reference Books:

1. V.Rajaraman, C. Siva Ram Murthy, "*Parallel Computers Architecture and Programming*" Prentice-Hall India, 2000.
2. Shane Cook, "*CUDA Programming: A developer's guide to parallel computing with GPUs*", Morgan Kaufman Publication, Elsevier, 2013.