PARALLEL COMPUTER ARCHITECTURE AND PROGRAMMING
[Revised Credit System]
(Effective from the academic year 2021-2022)
SEMESTER - V

Subject Code	**CSE 3174**	**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

- Understand the concept of parallelism and parallel computers
- Illustrate point-to-point and collective communication primitives in MPI
- Explain the architecture of GPU and understand OpenCL APIs
- Discuss thread and memory organization in CUDA and write parallel programs for 1D and 2D arrays
- Discuss important patterns of parallel computation

Module -1	**eaching Hours**
INTRODUCTION TO PARALLEL COMPUTERS: Introduction, Need for high speed computing, how to increase the speed of computers, Features of parallel computers, Structure of parallel computers, Classification of parallel computers, Array Processors, Shared Memory Parallel Computers, Utilizing temporal parallelism, Utilizing data parallelism, Comparison of temporal and data parallel processing **Text Book 1:** Chapter: 1, Chapter 4 – 4.1-4.2, 4.5, 4.7, Chapter: 2 – 2.1 – 2.3	**6 Hours**
Module – 2	
MESSAGE PASSING PROGRAMMING: Introduction, Message passing model, Basic data types and functions, Collective communication, Benchmarking parallel performance, MPI error handling functions. **Text Book 2:** Chapter 4: 4.1, 4.2, 4.4.1 - 4.4.5, 4.5, 4.6, Chapter 6: 6.5 and Appendix	**6 Hours**
Module – 3	
GPU ARCHITECTURE AND OVERVIEW OF OpenCL APIs Introduction, GPUs as parallel computers, Architecture of a modern GPU, Need for parallelism, Parallel programming languages and models, Overview of OpenCL APIs for Platform and Devices, Execution Environment- Context, Command Queues, Buffers, Program Object and Kernel Object, Program layout, Writing kernels, OpenCL program for vector-vector addition	**5 hours**

Text Book 3 : Chapter 1: 1.1-1.3,1.6, **Text Book 4 :** Selected Topics from Chapter 2	
Module - 4	
CUDA THREAD AND MEMORY ORGANIZATION: Data Parallelism, CUDA Program Structure, Vector-addition kernel, Device memories and Data transfer, Kernel functions and Threads, Runtime APIs and Error Handling, CUDA thread organization, Mapping threads to multi-dimensional data, Importance of memory access efficiency, Matrix Multiplication, CUDA device memory types, Tiling for Reduced Memory Traffic, A Tiled Matrix Multiplication Kernel **Text Book 3:** Chapter 2 Chapter 3: 3.1,3.2 Chapter 4: 4.1, 4.5	**9 Hours**
Module - 5	
PARALLEL PATTERNS: 1D Parallel Convolution—A Basic Algorithm, Constant Memory and Caching, Sparse Matrix Computation, Background, Parallel SpMV Using CSR, Merge Sort, Sequential Merge Algorithm, A Parallelization Approach, Co-Rank Function Implementation, A Basic Parallel Merge Kernel, Case study- Machine Learning, Convolutional Neural Networks, ConvNets: Basic Layers, ConvNets: Backpropagation, Convolutional Layer: A Basic CUDA Implementation of Forward Propagation, Reduction of Convolutional Layer to Matrix Multiplication **Text Book 3:** Chapter: 7.1-7.3, 10.1,10.2, 11.1-11.5, 16.1-16.4	**10 Hours**
Course outcomes:	
After studying this course, students will be able to: 1. Explain the structure and classification of parallel computers. 2. Write MPI programs using point-to-point and collective communication primitives. 3. Outline the GPU architecture and OpenCL APIs 4. Analyze thread and memory organization in CUDA and write CUDA programs 5. Apply the pattern of parallel computation in parallel applications	
Text Books: 1. V. Rajaraman, C. Siva Ram Murthy, *Parallel Computers Architecture and Programming*, (2e), Prentice-Hall India, 2016 2. Michael J. Quinn, Parallel Programming in C with MPI and OpenMP, McGraw Hill Edition, 2003 3. David B. Kirk, Wen-mei W. Hwu, Programming Massively Parallel Processors –A Hands-on approach, (3e), Elsevier Inc., 2016 4. Benedict R. Gaster, Lee Howes, David R, Perhaad Mistry, Dana Schaa, "*Heterogeneous Computing with OpenCL*", (1e), Elsevier Inc ,2012.	
Reference Books:	

1. Shane Cook, *CUDA Programming: A developer's guide to parallel computing with GPUs*, Morgan Kaufman Publication, Elsevier, 2013.
2. Jason Sanders, Edward Kandrot, *CUDA By example: An Introduction to General Purpose GPU Programming*, Addison Wesley, 2011.
3. *CUDA C Programming Guide,* nVIDIA, 2012.