Course Type	Course Code	Name of Course	L	T	P	Credit
DE	**NMCD510**	GPU Computing	3	0	0	3

Prerequisite

- C/C++/Python and the basics of programming.

Course Objective

- **Objective**: Understand GPU Architecture, thread organization, Memory organization, Parallel programming with CUDA and OpenACC.

Learning Outcomes

Upon successful completion of this course, students will:

- Skill of developing data-parallel programming for High Performance Computing (HPC)
- Optimize CUDA Application Programmes
- Can apply GPU computing in variuos Parallel patterns and Convolution Neural Networks

Unit No.	Topics to be Covered	Contact Hours	Learning Outcome
1	GPU Architectures: Understanding Parallelism with GPU, device memories and data transfer, kernel functions. Scalable parallel execution: CUDA Thread Organization. Mapping Threads to Multidimensional Data, synchronization and transparent scalability, Resource Assignment,Querying Device Properties. Thread Scheduling and Latency Tolerance	9	To understand GPU and CUDA programming Architectures To understand thread organization, synchronization
2	Memory and data locality: Importance of Memory Access Efficiency, Matrix Multiplication, CUDA Memory Types, Strategy for Reducing Global Memory Traffic,Concepts of tiling, Boundary Checks, Memory as a Limiting Factor to Parallelism Performance considerations: Global Memory Bandwidth, Warps and SIMD Hardware, More on Memory Parallelism Dynamic Partitioning of Resources, Thread Granularity	8	To know the concept of Memories and their Importance as a Limiting Factor to Parallelism To understand Performance concepts

3	Numerical considerations: Floating-Point Format, Representable Numbers, Special Bit Patterns and Precision, Arithmetic Accuracy and Rounding, Algorithm Considerations. Streams and Multi GPU Solutions: Atomic Operations, Single Stream, Multiple Streams, GPU Work Scheduling, Zero-Copy Host Memory, Portable Pinned Memory	8	To learn the basics of Floating-Point Format, Representable Numbers To understand the Arithmetic Accuracy and Rounding concepts To know the Streams and Multi GPU Solutions concepts
4	Parallel patterns: 1D, 2D Parallel and Tiled Convolutions, Convolution Neural Networks, Convolution Layer, Reduction of Convolution Layer to Matrix Multiplication, parallel histogram computation, Use of Atomic Operations, sparse matrix computation, merge sort	8	To learn Parallel patterns and Convolution Neural Networks with CUDA implementation
5	GPU computing with PyCUDA: PyCUDA Module, Matrix-Matrix Multiplication, Kernel Invocation with GPUArray, Evaluating elementwise expressions with PyCUDA, MapReduce Operation, GPU progaming with NumbaPro Parallel programming with OpenACC: The OpenACC Execution Model, OpenACC Directive Format, Comparing OpenACC and CUDA, Interoperability with CUDA and Libraries	9	To learn Parallel Programming with PyCUDA and OpenACC
Total		**42**	

Text Books:

1. David B. Kirk:Programming Massively Parallel Processors: A Hands-on Approac, Wen-mei W. Hwu, Elsevier, 2016
2. Jason Sanders:CUDA by Example: An Introduction to General-Purpose GPU Programming, Edward Kandrot, publisher Addison-Wesley Professional, 2010

Reference Books:

1. John Cheng, Max Grossman, Ty McKercher: Professional CUDA C Programming,John Wiley & Sons, 2014
2. Dr. Brian Tuomanen: Hands-On GPU Programming with Python and CUDA: Explore high-performance parallel computing with CUDA, Packt Publishing Ltd, 2018