**CSE 3202 PARALLEL COMPUTER ARCHITECTURE AND PROGRAMMING [2 1 0 3]**

**Course Objectives:**

* Discuss the architecture of GPU, history of GPU computing and concepts related to parallel programming languages and models.
* Illustrate point-to-point and collective communication primitives in MPI.
* Explain the architecture of OpenCL and writing efficient kernel functions using OpenCL.
* Describe Threads, Memory Organization and other programming concepts related to CUDA.
* Discuss the concepts related to parallel virtual machines.

**Course Outcomes:**

* Tell about the concepts related to parallel programming languages, architecture of GPU and parallel virtual machines.
* Write MPI programs using point-to-point and collective communication primitives.
* Solve parallel programming tasks using OpenCL.
* Analyze thread and memory organization in CUDA and writing kernel programs.

1. GPU ARCHITECTURE

Introduction, GPUs as parallel computers, Architecture of a modern GPU, Need for parallelism, Parallel programming languages and models, History of GPU Computing- Evolution of graphics pipelines, GPU computing.

(Chapter 1.1, 1.2, 1.3, 1.4, 2.1, 2.2 of Text Book1) (4 hrs)

1. 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.

(Chapter 4.1, 4.2, 4.4.1 - 4.4.5, 4.5, 4.6, 6.5of Text Book2) (6 hrs)

1. OpenCL ARCHITECTURE

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 Device Architecture – Superscalar Execution, VLIW, Hardware Multithreading.

(Selected topics from Chapter 2 and 3 of Text Book 3) (6 hrs)

1. OpenCL PROGRAMMING

OpenCL APIs, OpenCL programs for vector-vector addition, Selection Sorting, Merge Sorting, Binary search, String search, Calculation of value of π, Product of 2 matrices, Matrix-vector multiplication, Transpose of a matrix.

(Selected topics from Chapter 2 of Text Book 3)(7 hrs)

5**.** INTRODUCTION TO CUDA

Introduction, Data Parallelism, CUDA Program Structure, Matrix-Matrix Multiplication, Device memories and Data transfer, Kernel functions and Threads, Runtime APIs and Error Handling.

(Chapter 3.1, 3.2, 3.3, 3.4, 3.5 of Textbook1) (6 hrs)

6. CUDA THREADS AND MEMORY ORGANIZATION

Introduction, CUDA Thread Organization, Importance of Memory Access Efficiency, CUDA Device Memory types.

(Chapter 4.1, 5.1, 5.2 of Text Book1) (4 hrs)

7. PARALLEL VIRTUAL MACHINE

Introduction, Starting PVM, Process Management, Communication functions, Comparison of parallel programming models. (Selected topics from Chapter 6 of Text Book 4) (3hrs)

**Text Books:**

1. D. Kirk and W. Hwu , “*Programming Massively Parallel Processors –A Hands-on approach*”, Elsevier Inc., 1st Edition, 2010.
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. V.Rajaraman, C. Siva Ram Murthy, “*Parallel Computers Architecture and Programming*” Prentice-Hall India, 2000.

**References:**

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.