INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal - 500 043, Hyderabad, Telangana

COURSE CONTENT

GPU ARCHITECTURES & COMPUTING								
VI Semester: CSE								
Course Code	Category	Hours / Week			Credits	Maximum Marks		
		L	T	P	C	CIA	SEE	Total
ACSD33	Elective	3	0	0	3	40	60	100
Contact Classes: 48	Tutorial Classes: Nil	Practical Classes: Nil				Total Classes: 48		
Prerequisites: Computer Architecture								

I. COURSE OVERVIEW:

This course provides a comprehensive introduction to GPU computing and parallel programming. It begins with foundational concepts of graphics processors and GPU architectures, followed by detailed exploration of CUDA and memory hierarchies. Synchronization techniques and inter-device communication are covered alongside practical programming strategies. The course also introduces tools for debugging and profiling GPU programs and concludes with advanced topics like dynamic parallelism, multi-GPU systems, and real-world case studies in image processing and deep learning.

II. COURSES OBJECTIVES:

The students will try to learn

I The parallel programming with Graphics Processing Units (GPUs).

II The fundamental principles of GPU architecture and how GPUs differ from traditional CPU-based systems.

III The key components and structure of modern GPU architectures, including CUDA cores, memory hierarchy, and compute units.

IV The parallel computing techniques and concepts, particularly how to exploit the parallel processing power of GPUs for high-performance computing tasks.

V The GPU programming using frameworks such as CUDA and OpenCL, enabling efficient execution of parallel algorithms on GPUs.

III. COURSE OUTCOMES:

At the end of the course, students should be able to:

CO1 Understand and describe the fundamental architecture of GPUs and how they differ from traditional CPU-based systems.

CO2 Identify the key components of modern GPU architectures, including CUDA cores, memory hierarchy, and compute units, and explain their functions.

CO3 Demonstrate the ability to apply parallel computing techniques for high-performance tasks using GPU-based systems.

CO4 Gain proficiency in GPU programming using frameworks like CUDA and OpenCL, and implement parallel algorithms efficiently on GPUs.

CO5 Understand and manage GPU memory systems, including global, shared, and local memory, and optimize memory usage for performance improvement.

CO6 Optimize GPU performance by applying techniques such as task scheduling, load balancing, and parallel execution strategies.

IV. COURSE CONTENT:

MODULE –I: Introduction (9)

History, graphics processors, graphics processing units, GPGPUs, Clock speeds, CPU / GPU comparisons, heterogeneity, accelerators, parallel programming, CUDA OpenCL / Open ACC, Hello world computation kernels, launch parameters, thread hierarchy, warps /wave fronts, thread blocks / workgroups, streaming multiprocessors, 1D / 2D /3D thread mapping, device properties.

MODULE –II: MEMORY (10)

Memory hierarchy, DRAM / global, local / shared, private / local, textures, constant memory, pointers, parameter passing, arrays and dynamic memory, multi-dimensional arrays, memory allocation, memory copying across devices, programs with matrices, performance evaluation with different memories.

MODULE-III: SYNCHRONIZATION (10)

Memory consistency, barriers (local versus global), atomics, memory fence, prefix sum, reduction, programs for concurrent data structures (worklists), linked-lists, synchronization across CPU and GPU.

Functions: device functions, host functions, kernels functions, using libraries (Thrust), develop libraries.

MODULE-IV: SUPPORT AND STREAMS (10)

Debugging GPU programs, profiling, profile tools, performance aspects, asynchronous processing, tasks, task-dependence, overlapped data transfers, default stream, synchronization with streams, events, event-based synchronization - overlapping data transfer and kernel execution, pitfalls.

MODULE-V: ADVANCED TOPICS AND CASE STUDIES (9)

dynamic parallelism, unified virtual memory, multi-GPU processing, peer access, heterogeneous processing Case Studies: image processing, graph algorithms, deep learning.

V. TEXTBOOKS:

1. David Kirk, Wen-meiHwu, “Programming Massively Parallel Processors: A Hands-on Approach”, Morgan Kaufman, 2010.
2. Wen-Mei W. Hwu ,"GPU Computing Gems: Emerald Edition".

VI. REFERENCE BOOKS:

1. CUDA Programming: A Developer's Guide to Parallel Computing with GPUs; Shane Cook; Morgan Kaufman; 2012
2. "CUDA Programming: A Developer's Guide to Parallel Computing with GPUs" by Shane Cook.

VII. ELECTRONICS RESOURCES:

1. https://www.cse.iitk.ac.in/users/swarnendu/courses/autumn2020-cs610/gpu-cuda.pdf
2. https://homepages.laas.fr/adoncesc/FILS/GPU.pdf

VIII. MATERIALS ONLINE

1. Course outline description
2. Tutorial question bank
3. Tech talk topics
4. Open-ended experiments
5. Definitions and terminology
6. Assignments
7. Model question paper – I
8. Model question paper – II
9. Lecture notes
10. PowerPoint presentation
11. E-Learning Readiness Videos (ELRV)