Nirma University
Institute of Technology, School of Technology
M.Tech Computer Science & Engineering / M Tech Computer Science and Engineering (Data Science)
Semester – II

L	T	P	C
3	0	2	4

Course Code	3CS12D304
Course Name	Multicore and GPU Computing

Course Learning Outcomes (CLOs):

At the end of the course, students will be able to

1. comprehend modern multi-core processor micro-architectures and interconnect technologies
2. analyse the memory hierarchy and performance characteristics
3. recognize the need for atomic operations and variety of locking mechanisms
4. explore architecture of general purpose graphics processing units and their common programming models

Syllabus:

Teaching Hours

Unit I **5**

Introduction: Introduction to Advanced Architectures of multi-core processors, GPGPUs, Cell BE, Universality and Multicore Architectures, Introduction to concepts of parallel programming

Unit II **15**

CUDA Programming: Introduction to CUDA architecture for parallel processing, CUDA Parallelism Model, Foundations of Shared Memory, Introduction to CUDA-C, Parallel programming in CUDA-C, Thread Cooperation and Execution Efficiency, Constants memory and events, memory management, CUDA C on multiple GPUs, Hashing and Natural Parallelism, Scheduling and Work Distribution, Atomics, Barriers and Progress, Transactional Memory

Unit III

Performance Tuning: Spin-Locks and Contention, Race detection, Deterministic replay, Global Predicate Detection **05**

Unit IV **05**

Open CL Programming: Introduction to OpenCL, OpenCL Setup, Basic OpenCL, Advanced OpenCL

Unit V **05**

Shared-memory programming: OpenMP: Introduction to OpenMP, Parallel Programming using OpenMP

Unit VI **10**

Case Study: Study of Multicore architectures like Fermi/GeForce / Tesla / Maxwell / Quadro / Kepler / Pascal / Volta

Self-Study:

The self-study contents will be declared at the commencement of semester. Around 10% of the questions will be asked from self-study contents.

Laboratory Work:

Laboratory work will be based on above syllabus with minimum 5 experiments to be incorporated.

Suggested Readings^:

1. Jason Sanders, CUDA by Example: An Introduction to General-Purpose GPU Programming, Addison-Wesley
2. David Kirk and Wen-mei Hwu, Programming Massively Parallel Processors: A Hands-on Approach, Morgan Kaufmann
3. Ryoji Tsuchiyama, Takashi Nakamura, Takuro Iizuka, Akihiro Asahara, Satoshi Miki, The OpenCL Programming, Fixstars Corporation

L=Lecture, T=Tutorial, P=Practical, C=Credit

^this is not an exhaustive list