RK UNIVERSITY

SYLLABUS

Course Title	Computer Unified Device Architecture (Elective - I)
Course Code	CP910
Course Credit	Lecture : 03
	Practical : 01
	Tutorial : 00
	Credits : 04

Course Objective

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

- **Use** modern Parallel computing - GPU architectures.
- **Understand** the evolution from the SIMD (Single Instruction, Multiple Data) architecture to the current architectural features and by discussing the trends for the future.
- **Explore** main Programming Model for Multicore architecture: CUDA.
- **Understand** fundamental ways CUDA exposes its parallelism.
- **Analyse** different types of GPU memory optimization.
- **Apply** different algorithmic Strategies for Optimizing the Parallel Reduction Primitive in CUDA

Detailed Syllabus

Sr. No.	Name of chapter & details	Hours Allotted
	Section – I	
1.	**Introduction to Massively Parallel Computing** Generic Multicore Chip – CPU, Generic Many core Chip – GPU, Heterogeneous Parallel Computing, CPU Vs. GPU, Graphics Processing Unit – GPU, Architecture of a Modern GPU, Compute Capability, CUDA, CUDA Hardware: Memory Model, CUDA Programming Model, Program execution	04
2.	**Introduction to Data Parallelism and CUDA C** Data Parallelism, CUDA Program Structure, A Vector Addition Kernel, Device Global Memory and Data Transfer, Kernel Functions and Threading, CUDA Thread Organization, Mapping Threads to Multidimensional Data,	07

3.	**Data-Parallel Execution Model** Matrix-Matrix Multiplication—A More Complex Kernel, Synchronization and Transparent Scalability, Assigning Resources to Blocks, Querying Device Properties, Thread Scheduling and Latency Tolerance	**04**
4	**CUDA Memories** Importance of Memory Access Efficiency, CUDA Device Memory Types Device Memory, Shared Memory, Page-Locked Host Memory, Portable Memory, Write-Combining Memory, Mapped Memory, Texture memory, Surface Memory	**06**
	Section – II	
5	**Basic Parallel Communication Patterns and Algorithms** parallel communication patterns like map, gather, scatter, stencil Fundamental GPU algorithms: Reduction computation – Sequential reduction, Tree based approach of parallel reduction, Work complexity and step complexity, Shared Memory allocation in CUDA.Thread Synchronization, Grid-Stride Loops, Parallel reduction with shared memory(Version1) and its disadvantages.	**07**
6	**Optimization of parallel reduction** Thread Divergence, Control (Branch) Flow Divergence, Warps, parallel reduction with stride index and non-divergent branch (Version2), Shared memory bank conflicts, Linear addressing, and parallel reduction with linear addressing (Version3). With two loads and first add of the reduction (Version4), Loop unrolling, Parallel reduction with loop unrolling(Version 5), comparison of all versions	**07**
7	**Thrust : Productivity-Oriented Library for CUDA** CUDA Libraries, Thrust – Introduction, Thrust – Provide generic data type, Thrust – Provide generic function, Containers, Iterator, Algorithms.	**03**
8	**New Features in CUDA 6** Unified Memory, XT and drop in libraries, GPU Direct RDMA in MPI, CUDA tools for performance checking – nvproof, visual profiler, MEMCHECK, NSight.	**04**

Instructional Method and Pedagogy:

- Lectures will be conducted with the aid of multi-media projector, blackboard, OHP etc.
- Assignments based on course contents will be given to the students at the end of each unit/topic and will be evaluated at regular interval.

Reference Books

- David B. Kirk, Programming Massively Parallel Processors, Morgen Kauffman
- Online book: CUDA_C_Programming_Guide V5.5, Available at: http://docs.nvidia.com/cuda/index.html
- Jason Sanders and Edward Kandrot, CUDA by Example: An Introduction to General-Purpose GPU Programming, Addision Wesley
- Rob Farber, CUDA Application Design and Development, Morgen Kauffman
- Stephen Keckler, Multicore Processors and Systems, Springer

Additional Resources

- developer.nvidia.com/cuda-downloads
- http://developer.download.nvidia.com/compute/cuda/2_1/toolkit/docs/
- NVIDIA_CUDA_Programming_Guide_2.1.pdf
- http://forums.nvidia.com/index.php?showtopic=181472
- http://gpucoder.livejournal.com/990.html
- http://developer.nvidia.com/gpu-computing-webinars