Approval: 10th Senate Meeting

Course Name: Introduction to Heterogeneous Computing
Course Number: CS508
Credit: 2-0-0-2
Prerequisites: Introductory Programming skill
Students intended for: MS/MTech/PhD/I-PhD/4th year BTech
Elective or Compulsory: Elective
Semester: Odd/Even

Preamble: Heterogeneous computing using central processing units (CPUs) and graphical processing units (GPUs)/DSPs is a new paradigm in the high performance computing (HPC) due to present day computer architectures. It is important to familiarize with such processing units for upcoming computational research. The course is planned to be offered in collaborative manner by a group of faculty from different disciplines.

Course Outline:
The course introduces students to HPC, computing cluster, data parallelism, after a brief review of the programming concepts. Further, parallel computing, especially using GPUs for HPC would be introduced. The course introduces CPU-GPU architectures in brief and various CUDA kernel programming. The course emphasizes on various assignments (hence no practicum lab sessions are included) as application case study from areas such as, image processing, digital signal processing, speech recognition, finite-element engineering, molecular dynamics or computational fluid dynamics. Projects will be assigned to students from their respective research areas either by course instructor or by thesis supervisors. The course also emphasizes on the performance profiling and estimates for the GPU program.

Course Modules:

Module 1: Review of programming concepts
C/C++: Algorithm design, Expressions, Decisive statements, Iteration loops, Functional calls, Recursion, Arrays, Pointers, Addresses, I/O file handling, Error handling
[6]

Module 2: Introduction to HPC
Introduction to uni-processor and multiprocessor architecture, types of parallelism,
Data parallelism, Loop unrolling, communication and synchronization needs,Parallel programs using openMP/ MPI, Introduction to cluster hardware, software, and network
[8]

Module 3: Understanding parallelism with GPUs
CUDA C: GPU architectures, Grids, Blocks and Threads in CUDA, Memory Handling with CUDA, Concepts of tiling and warps for CUDA
[12]

Module 4: Application case studies
Problem statements, designing an algorithm, writing program, verifying the output profiling the performance, optimization of the program
[8]

Text Books:

1. Cook S., CUDA Programming, Morgan Kaufmann. nVIDIA GPU Teaching Kit. Kirk D.B. and Hwu W-M.W. Programming Massively Parallel Processors.