Department of Computer Science and Engineering, Indian Institute of Technology Madras

<table>
<tr><td>Course title</td><td colspan="8"><b>GPU Programming</b></td><td>Course No</td><td colspan="4"><b>CSNEW</b></td></tr>
<tr><td rowspan="2">Department</td><td rowspan="2">CSE</td><td rowspan="2">New Credits</td><td>L</td><td>T</td><td>E</td><td>P</td><td>O</td><td>C</td><td rowspan="2">Old Credits</td><td>L</td><td>T</td><td>P</td><td>C</td></tr>
<tr><td>3</td><td>1</td><td>0</td><td>0</td><td>8</td><td>12</td><td>3</td><td>1</td><td>0</td><td>4</td></tr>
<tr><td>Offered for</td><td colspan="8">BTech / DD / MTech / MS / PhD</td><td>Status</td><td colspan="4">Final</td></tr>
<tr><td>Faculty</td><td colspan="8"><b>Rupesh Nasre.</b></td><td>Type</td><td colspan="4">Theory</td></tr>
<tr><td>Pre-requisite</td><td colspan="8"></td><td>To take effect from</td><td colspan="4">2017</td></tr>
<tr><td>Submission date</td><td colspan="2">Date of approval by DCC</td><td colspan="6">Date of approval by BAC</td><td colspan="5">Date of approval by Senate</td></tr>
<tr><td></td><td colspan="2"></td><td colspan="6"></td><td colspan="5"></td></tr>
</table>

Objectives:
To learn parallel programming with graphics processing units (GPUs)

Outcomes:
Students would learn concepts in parallel programming, implementation of programs on GPUs, debugging and profiling parallel programs.

Course Contents:

Topic (number of lectures + number of tutorials)

Introduction (2 + 1):
- history, graphics processors, graphics processing units, GPGPUs
- clock speeds, CPU / GPU comparisons, heterogeneity
- accelerators, parallel programming, CUDA / OpenCL / OpenACC, Hello World

Computation (3 + 1):
- kernels, launch parameters
- thread hierarchy, warps / wavefronts, thread blocks / workgroups, streaming multiprocessors
- 1D / 2D / 3D thread mapping, device properties, simple programs

Memory (8 + 2):
- 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

Synchronization (6 + 2):
- memory consistency
- barriers (local versus global), atomics, memory fence
- prefix sum, reduction
- programs for concurrent data structures such as worklists, linked-lists
- synchronization across CPU and GPU

Functions (3 + 1):
- device functions, host functions, kernels, functors
- using libraries (such as Thrust), developing libraries

Support (1 + 2):
- debugging GPU programs
- profiling, profile tools, performance aspects

Streams (3 + 1):
- asynchronous processing, tasks, task-dependence
- overlapped data transfers, default stream, synchronization with streams
- events, event-based-synchronization

- overlapping data transfer and kernel execution, pitfalls

Case studies (3 + 2):
- image processing
- graph algorithms
- simulations
- deep learning
- ...

Advanced topics (8 + 2):
- dynamic parallelism
- unified virtual memory
- multi-GPU processing
- peer access
- heterogeneous processing

Course evaluation would involve programming assignments.

Text Books:

Programming Massively Parallel Processors: A Hands-on Approach; David Kirk, Wen-mei Hwu; Morgan Kaufman; 2010 (ISBN: 978-0123814722)

Reference Books:

CUDA Programming: A Developer's Guide to Parallel Computing with GPUs; Shane Cook; Morgan Kaufman; 2012 (ISBN: 978-0124159334)