CSE 5154 HIGH PERFORMANCE COMPUTING SYSTEMS [3 1 0 4]

Course Outcomes

After completing the course, the student will be able to:

CO1: Analyse the structure of parallel computers.

CO2: Develop and analyse the parallel algorithm for a given parallel computer.

CO3: Write MPI programs using point-to-point and collective communication primitives.

CO4: Solve parallel programming tasks using OpenCL.

CO5: Analyze thread and memory organization in CUDA and writing kernel programs.

1. INTRODUCTION TO PARALLEL COMPUTERS:
   Introduction to Parallel Computing, Need for Parallel Computing, Parallel processing, Programmatic levels of parallel processing, Parallel processing mechanisms, Parallel Computer Structures, Parallel Architectural Classification Schemes, Handler's classification, Feng's classification, Applications of parallel processing.
   (1.1.2, 1.2, 1.3, 1.3.1, 1.3.2, 1.3.3, 1.4, 1.5 of Text 1) (5 Hrs)

2. PIPELINING:
   Linear and non-linear pipelining principles, Classification of pipeline processor, Nonlinear pipelining, General Pipelines and Reservation tables, Problems on pipelining.
   (3.1, 3.2 of Text 1) (5 Hrs)

3. SYNCHRONOUS PARALLEL PROCESSING:
   SIMD Computer Organization, Masking and Data routing mechanisms, Inter-PE Communications, SIMD interconnection networks.
   (5.1, 5.2 of Text 1) (4 Hrs)

4. THREAD LEVEL PARALLELISM AND MULTIPROCESSORS:
   Introduction, Processor characteristics for multiprocessing, Centralized Shared Memory Architecture, Performance of Symmetric Shared Memory Multiprocessor, Distributed Shared Memory, Interconnection Networks.
   (7.1, 7.2 of Text 1, 5.1, 5.2, 5.3, 5.4 of Text 2) (5 Hrs)

5. ELEMENTARY PARALLEL ALGORITHMS: Hypercube SIMD model, Shuffle-exchange SIMD model, 2D mesh SIMD model.
(6.1, 6.2 of Text 3) (4 Hrs)

6. MESSAGE PASSING PROGRAMMING
Introduction, Message passing model, MPI basic data types and functions, Point-to-point communication, Blocking and nonblocking communication, Standard send-receive, Synchronous send-receive, Buffered send-receive, Collective communication, Benchmarking parallel performance, MPI error handling functions, Problems on basic mathematical functions, matrix manipulations, sort algorithms.
(4.1, 4.2, 4.4.1 - 4.4.5, 4.5, 4.6, 6.5 of Text 4)
(7 Hrs)

7. OpenCL ARCHITECTURE: OpenCL standard, OpenCL specification, Kernels and openCL execution models, Platform Model, Host/Device Interaction, Execution environment, Contexts, Command Queues, Memory Objects, Program Object and Kernel Object, Program layout, Memory model, Writing Kernels.
(CH 2 of Text 5)
(5 Hrs)

8. Basic OpenCL EXAMPLES: OpenCL APIs, Hello World program in OpenCL, Other OpenCL APIs in detail, Work-item and work-group, OpenCL memory model, Example applications, Convolution.
(CH 4, CH 7, and CH 9 of Text 5)
(7 Hrs)

9. CUDA PROGRAMMING: CUDA programming model, CUDA tools, CUDA Libraries, Comparison of Open CL and CUDA performance.
(CH 3 and CH 4 of Text 6) (6 Hrs)

References:

1. Kai Hwang, Faye A. Briggs, *Computer Architecture and Parallel Processing*, Tata McGraw-Hill India, 2012.
2. John L. Hennessy David A. Patterson *Computer Architecture: A Quantitative Approach (5e),* 2014.

3. Michael J Quinn, *Parallel Computing: Theory and Practice*, *(2e),* Tata McGraw Hill, 2002.
4. Michael J. Quinn, *Parallel Programming in C with MPI and OpenMP*, McGraw Hill, 2003.
5. Benedict R. Gaster, Lee Howes, David R, Perhaad Mistry, Dana Schaa, *Heterogeneous Computing with OpenCL*, Morgan Kaufmann, 2012.
6. David B. Kirk, Wen-mei W. Hwu, *Pragramming Massively Parallel Processors,A Hands-on Approach*, *(2e),*Elsevier, 2012.