Course code	Course title	L	T	P	J	C
CSE5006	MULTICORE ARCHITECTURES	2	0	2	0	3
		Syllabus version				
		V. 1.1				

Course Objectives:

1. To provide knowledge on basics of Multicore architectures and parallel programming models
2. To design and develop parallel programs using parallel computing platforms such as OpenMP, CUDA
3. To apply program optimizations on parallel programs and evaluate the performance using profiling tools

Expected Course Outcome:

After successfully completing the course the student should be able to

CO1. Outline the developments in the evolution of multi-core architectures and parallel programming paradigms

CO2. Comprehend the various programming languages and libraries for parallel computing platforms

CO3. Use of profiling tools to analyse the performance of applications by interpreting the given data

CO4. Compare and contrast the features of parallel programming languages such as OpenMP and CUDA

CO5. Write parallel programs using OpenMP and CUDA

CO6. Evaluate efficiency trade-offs among alternative parallel computing architectures for an efficient parallel Application design

CO7. Analyse performance parameters such as speed-up, efficiency for parallel programs against serial programs

Student Learning Outcomes (SLO):	2,11,14,17

Module:1	Introduction to Multi-Core Architectures	2 hours	SLO: 2

Evolution of multicores through Moor's Law, Comparisons of single core, multi-core, multi-processing and hyper threading

Module:2	Parallel Computers and programming	5 hours	SLO: 2

Threading Concepts, Communication Architectures and Communication Costs, Thread Level Parallelism(TLP), Instruction Level Parallelism(ILP), Comparisons, Cache Hierarchy and Memory-level Parallelism, Cache Coherence, Parallel programming models, Shared Memory and Message Passing, Vectorization

Module:3	OpenMP programming (Open multi-processing)	5 hours	SLO: 2

Introduction to OpenMP, Parallel constructs, Runtime Library routines, Work-sharing constructs, Scheduling clauses, Data environment clauses, atomic, master Nowait Clause, Barrier Construct

Module:4	CUDA Programming(Compute Unified Device Architecture)	6 hours	SLO: 2

Introduction to GPU Computing, CUDA Programming Model, CUDA API, Simple Matrix, Multiplication in CUDA , CUDA Memory Model, Shared Memory Matrix Multiplication, Additional CUDA API Features			
Module:5	**Performance Analysers**	**4 hours**	**SLO: 14**
Trace analyzer and collector (ITAC), VTune Amplifier XE, Energy Efficient Performance, Integrated Performance Primitives (IPP)			
Module:6	**Contemporary tools**	**3 hours**	**SLO: 14**
MKL (Math Kernel Library), Threading Building Blocks, CUDA Tools			
Module:7	**HTC and MTC**	**3 hours**	**SLO: 14**
HTC (High Throughput Computing), MTC (Many Task Computing), Top 500 Super computers in the world, Top 10 Super Computer architectural details, Exploring Linpack			
Module:8	**Contemporary issues:**	**2 hours**	**SLO: 11**
	Total Lecture hours:	**30 hours**	

Text Book(s)	
&	
Reference Books	
1.	Rob Farber, “CUDA Application Design and Development”, Morgan Kaufmann Publishers, 2013
2	Shameem Akhter and Jason Roberts, “Multi-Core Programming”, 1st edition, Intel Press, 2012
3	Robert Oshana, “Multicore Software Development Techniques: Applications, Tips, and Tricks”, Newnes,1 edition, 2015
4	David B. Kirk , Wen-mei W. Hwu, “Programming Massively Parallel Processors: A Hands-on Approach (Applications of GPU Computing Series)”, 1st edition, Morgan Kaufmann, 2010.

List of Challenging Experiments (Indicative)		**SLO: 14,17**
1.	Practice with Open MP	2 hours
2.	**OpenMp Sample Programs** Execution Time estimation Practicing sample programs Development of documentation for observations	2 hours
3.	**Develop a sample program using Execution Environment Routines and write interesting observations by comparing various routines**	2 hours
4.	**Develop a program using following construct and describe scenario for the need of construct** 1. parallel Construct 2. Determining the Number of Threads for a parallel Region 3. Work-sharing Constructs	8 hours

	a. loop construct b. sections construct c. single construct 4. schedule clause a. static b. Dynamic c. guided 5. Data Environment Constructs a. Shared Clause b. Critical Construct c. Reduction Clause 6. Master Construct 7. Nowait clause 8. Barrier Construct 9. Atomic Construct	
5.	**Analysis through any one of profiling tools (ITAC/VTune /EEP/IIP)** 1 Experimental setup 2 Parallelizing given serial program into parallel 3 Analysing parallel programs	6 hours
6	**CUDA programming** 1 Write a CUDA C/C++ program that add two array of elements and store the result in third array 2 How to Reverse Single Block in an Array using CUDA C/C++ 3 CUDA C program for Matrix addition and Multiplication using Shared memory 4 Write CUDA C/C++ program for Vector Addition. Modify your program so, that it can add two vector of arbitrary size	8 hours
	Total Laboratory Hours	28 hours

Recommended by Board of Studies	DD-MM-YYYY		
Approved by Academic Council	No. 46	Date	24.08.2017