Course code	Course title		L	T	P	C
MCSE503L	Computer Architecture and Organization		3	0		3
Pre-requisite		Syllabus version				
		v. xx.xx				

Course Objectives

1. To provide knowledge on the basics of computer architectures and organization that lays the foundation to study high-performance architectures
2. To design and develop parallel programs using parallel computing platforms such as OpenMP, CUDA
3. To evaluate the performance using profiling tools and optimize parallel codes using various optimization techniques

Course Outcomes

1. Outline the developments in the evolution of computer architectures and parallel programming paradigms
2. Comprehend the various programming languages and libraries for parallel computing platforms
3. Use of profiling tools to analyze the performance of applications by interpreting the given data
4. Evaluate efficiency trade-offs among alternative parallel computing architectures for an efficient parallel application design
5. Develop parallel programs using OpenMP and CUDA and analyze performance parameters such as speed-up, and efficiency for parallel programs against serial programs

Module:1	Computer Evolution And Performance	5 hours

Defining Computer Architecture and Organization, Overview of Computer Components, Von Neumann architecture, Harvard Architecture CISC & RISC, Flynn’s Classification of Computers, Moore's Law, Multi-threading, Comparisons of Single Core, Multi Processors, and Multi-Core architectures, Metrics for Performance Measurement

Module:2	Memory Hierarchy	8 hours

Key Characteristics of Memory systems, Memory Hierarchy, Cache Design policies, Cache Performance, Cache Coherence, Snoopy Protocols, Cache coherence protocols, MSI, MESI, MOESI

Module:3	Parallel Computers	8 hours

Instruction Level Parallelism(ILP), Compiler Techniques for ILP & Branch Prediction, Thread Level Parallelism (TLP), Threading Concepts, Shared Memory, Message Passing, Vectorization

Module:4	Multithreaded Programming using OpenMP	7 hours

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

Module:5	Programming for GPU	6 hours

Introduction to GPU Computing, CUDA Concepts, CUDA Programming Model, Program Structure of CUDA & Execution, Methods for operations on Device Memory, Thread Organization, Examples

Module:6	Performance Analyzers	6 hours

Performance Evaluation, performance bottlenecks, Profiling categories; Profiling tools: Trace

analyzer and collector (ITAC), VTune Amplifier XE, Energy Efficient Performance, Integrated Performance Primitives (IPP)		
Module:7	**Energy Efficient Architectures**	**5 hours**
Overview of power issues, CMOS Device-level Power dissipation basics, Sources of energy Consumption, Strategies to save power or Energy, Low power designs, Power management techniques		
Module:8	**Contemporary Issues**	**1 hours**
	Total Lecture hours:	**45 hours**

Text Book(s)	
1.	William Stallings, Computer Organization and Architecture: Designing for Performance, Pearson, 2022, 11th Edition, Pearson
2	Gerassimos Barlas, Multicore and GPU Programming: An Integrated Approach, 2022, 2nd edition, Morgan Kaufmann
Reference Books	
1.	J.L. Hennessy and D.A. Patterson. Computer Architecture: A Quantitative Approach. 5th Edition, 2012, Morgan Kauffmann Publishers.
2.	Shameem Akhter, Jason Roberts, Multi-core Programming: Increasing Performance Through Software Multi-threading, 2010, Intel Press, BPB Publications
Mode of Evaluation: CAT / Written Assignment / Quiz / FAT	

Recommended by Board of Studies	26-07-2022		
Approved by Academic Council	No. xx	Date	DD-MM-YYYY

Course code	**Course title**	**L**	**T**	**P**	**C**
MCSE503P	**Computer Architecture and Organization LAB**	**0**	**0**	**2**	**1**
Pre-requisite		**Syllabus version**			
		v. xx.xx			

Course Objectives

1. To provide knowledge on basics of computer architectures and organization that lays foundation to study high performance architectures
2. To design and develop parallel programs using parallel computing platforms such as OpenMP, CUDA
3. To evaluate the performance using profiling tools and optimize parallel codes using various optimization techniques

Course Outcome

1. Outline the developments in the evolution of computer architectures and parallel programming paradigms
2. Comprehend the various programming languages and libraries for parallel computing platforms
3. Use of profiling tools to analyze the performance of applications by interpreting the given data
4. Evaluate efficiency trade-offs among alternative parallel computing architectures for an efficient parallel Application design.
5. Develop parallel programs using OpenMP and CUDA and analyze performance parameters such as speed-up, efficiency for parallel programs against serial programs

Indicative Experiments

1.	Set-up an environment for OpenMP Programming: Activities: create a Project using Visual Studio, Writing Sample OpenMp Program, Setting up properties, compile & Execute OpenMP program, OpenMP manual study, Creation of Login credential on Intel for Intel Parallel Studio
2.	OpenMP program using following construct and describe scenario for the need of construct Use of Parallel Construct, Determine the Number of processors in a parallel Region, Find the thread ID of each processor
3.	Computation of Execution Time Using OpenMP clock, Using windows clock
4.	OpenMP Program using various Environment Routines to access the processor run-time information and write interesting observations by comparing various routines
5.	OpenMP program using following Worksharing Constructs and describe scenario for the need of construct loop construct, sections construct, single construct
6.	OpenMP program using following schedule clauses and describe scenario for the need of clause Static, Dynamic, Guided
7.	Develop parallel programs for given serial programs and profile the program using Vtune Analysis tool Matrix-Matrix multiplication, Matrix-Vector multiplication
8.	Develop parallel programs for given serial programs and profile the program using Vtune Analysis tool Quicksort, Minimum Spanning Tree

9.	CUDA-platform setup on NVIDIA / Google Colab		
10.	Write a CUDA C/C++ program that add two array of elements and store the result in third array		
11.	Write a CUDA C/C++ program that Reverses Single Block in an Array; CUDA C/C++		
12.	Write a CUDA C program for Matrix addition and Multiplication using Shared memory		
		Total Laboratory Hours	**30 hours**
Text Book(s)			
1.	Gerassimos Barlas, Multicore and GPU Programming: An Integrated Approach, 2022, 2nd edition, Morgan Kaufmann		
Reference Books			
1.	Shameem Akhter, Jason Roberts, Multi-core Programming: Increasing Performance Through Software Multi-threading, 2010, Intel Press, BPB Publications		
Mode of Evaluation: CAT / Mid-Term Lab/ FAT			
Recommended by Board of Studies	26-07-2022		
Approved by Academic Council	No. xx	Date	DD-MM-YYYY