Course Code	Course Title	L	T	P	C
OLMCA607	Computer Architecture	3	1	0	4
Pre-requisite	NIL	Syllabus version			
		1.0			

Course Objectives:

1. To infer the evolution and operation of modern architectures.
2. To apply fundamental techniques to speed-up program execution.
3. To analyze the impact of design principles on computer performance.

Course Outcomes:

1. Understand the organization and performance characteristics of different processor architecture.
2. Interpret techniques to improve processor's ability to exploit parallelism.
3. Identify characteristics and challenges in multicore.
4. Design the organization of cache and virtual memory.
5. Develop parallel programming for computer problems with multicore.

Module:1 Modern Computer Architectures

Introduction, Von Neumann and Harward Architecture, Fundamentals of RISC, CISC, Instruction Level parallelism (ILP)-Concepts and challenges, Instruction Scheduling: Branching with Prediction, Dynamic Scheduling: Hazards and Solutions, Measuring Performance of ILP, Limitations of ILP.

Module:2 Threads and Multiprocessors Architecture

Thread level parallelism, simultaneous multithreading, introduction to multiprocessor architecture, types, Shared Memory Architecture, Distributed Memory Architecture, Limitation.

Module:3 Multicore Architecture

Flynn Classification of Machines, Evolution of multicore, Architecting with multicore: Homogeneous, Heterogeneous cores, Shared resources, shared buses, optimal resource sharing strategies, Performance evaluation of multicore processors.

Module:4 Memory Module Design

Conceptual view of memory cell, memory address map, memory connections to CPU, Key characteristics of Memory, Main Memory Design, cache memories-cache memory management techniques, Types of caches- Lookthrough and look aside, write through, write around, unified, split caches, Cache details: multilevels, cache levels, cache misses, performance issues, mean memory access time, execution time, cache coherence protocols, snoopy, MSI,MESI, MOESI

Module:5 Multi-Threading Concepts

Fundamentals of multithreading programming, concurrency Vs parallelism, threading design concepts for developing an application, correctness concepts: critical region, mutual exclusion, synchronization, race conditions, Multithread Performance: performance concepts: simple speedup, computing speedup, efficiency, granularity, load balance

Module:6 Multicore programming

Introduction to OpenMP, OpenMP directives, parallel constructs, work-sharing constructs, data environment constructs, synchronization constructs, extensive API library for finer control, benchmarking multicore architectures: benchmarking of processors, comparison of processor performance for specific processors.

Module:7	**Multicore power and performance measurement**	
Multicore power and performance measurement using programming constructs (like loop unrolling) and tools/utilities (like gprof from GNU, Linux, Linux time command, Tuning and Analysis Utilities, VTune) using multicore benchmarks.		
	Total Lecture hours:	**120 hours**
Text Book(s)		
1.	John L Hennessey, David A Patterson, Computer Architecture: A Quantitative Approach, 2011, 5th Edition, Morgan Kauffmann.	
Reference Books		
1.	Shameen Akhter, Jason Roberts, Multicore programming, 2006, 1st Edition, Intel press.	
2.	Barbara Chapmen, Gabriele Jost, Ruud van de pas, Using OpenMP: Portable shared memory, parallel programming (scientific and engineering computation). 2008, 1st Edition, MIT Press.	
3.	Vincent P Huering, Harry F Jordan, Computer System Design and Architecture, 2004, 2nd Edition, Pearson.	
4.	David B Kirk, Wen-mei W Hwu, Programming Massively Parallel Processors: A Handson Approach (Application (Application of GPU Computing Series) 2013, 2nd Edition, Morgan Kaufmann	
5.	William Stallings, Computer Organization and Architecture: Designing for Performance, 2019, 11th Edition, Prentice Hall.	
Mode of Evaluation: Quiz / Assignment / Mid-Term, FAT		
Recommended by Board of Studies	01-11-2023	
Approved by Academic Council	No. 72	Date: 13-12-2023