

<b>CS305: Parallel Computer Architecture</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>Computer Architecture</b>
	<b>3</b>	<b>1</b>	<b>0</b>	

**Course Objective:** To introduce fundamentals of parallel, pipelines and superscalar architecture.

<b>S. No.</b>	<b>Course Outcomes (CO)</b>
<b>CO1</b>	Define the fundamental concepts and classification schemes in parallel computing architectures.
<b>CO2</b>	Explain the principles of multi-core and multi-threaded architectures, including their performance issues and optimization techniques.
<b>CO3</b>	Apply program optimization techniques and parallelization strategies in the development of parallel programs.
<b>CO4</b>	Analyze different parallel computer architectures and evaluate their performance, including memory hierarchy and communication latency.
<b>CO5</b>	Evaluate compiler optimization issues and operating system techniques for efficient multiprocessing and parallel program execution.
<b>CO6</b>	Design and implement parallel computing solutions for real-world applications in areas such as digital signal processing and image processing.

<b>S. No</b>	<b>Contents</b>	<b>Contact Hours</b>
<b>UNIT 1</b>	Introduction: Introduction to parallel computing, need for parallel computing, parallel architectural classification schemes, Flynn's , Fang's classification, performance of parallel processors, distributed processing, processor and memory hierarchy, bus, cache & shared memory, introduction to super scalar architectures, quantitative evaluation of performance gain using memory, cache miss/hits.	<b>6</b>
<b>UNIT 2</b>	Multi-core Architectures: Introduction to multi-core architectures, issues involved into writing code for multi-core architectures, development of programs for these architectures, program optimizations techniques, building of some of these techniques in compilers, Open MP and other message passing libraries, threads, mutex etc.	<b>6</b>

<b>UNIT 3</b>	Multi-threaded Architectures: Parallel computers, Instruction level parallelism (ILP) vs. thread level parallelism (TLP), Performance issues: Brief introduction to cache hierarchy and communication latency, Shared memory multiprocessors, General architectures and the problem of cache coherence, Synchronization primitives: Atomic primitives; locks: TTS, ticket, array; barriers: central and tree; performance implications in shared memory programs; Chip multiprocessors: Why CMP (Moore's law, wire delay); shared L2 vs. tiled CMP; core complexity; power/ performance; Snoopy coherence: invalidate vs. update, MSI, MESI, MOESI, MOSI; performance trade-offs; pipelined snoopy bus design; Memory consistency models: SC, PC, TSO, PSO, WO/WC, RC; Chip multiprocessor case studies: Intel Montecito and dual-core, Pentium4, IBM Power4, Sun Niagara	<b>10</b>
	Compiler Optimization: Issues Introduction to optimization, overview of parallelization; Shared memory programming, introduction to Open MP; Dataflow analysis, pointer analysis, alias analysis; Data dependence analysis, solving data dependence equations (integer linear programming problem); Loop optimizations; Memory hierarchy issues in code optimization.	<b>10</b>
<b>UNIT 4</b>	Operating System: Issues Operating System issues for multiprocessing Need for pre-emptive OS; Scheduling Techniques, Usual OS scheduling techniques, Threads, Distributed scheduler, Multiprocessor scheduling, Gang scheduling; Communication between processes, Message boxes, Shared memory; Sharing issues and Synchronization, Sharing memory and other structures, Sharing I/O devices, Distributed Semaphores, monitors, spin-locks, Implementation techniques on multi-cores; Open MP, MPI and case studies	<b>10</b>
<b>UNIT 5</b>	Applications: Case studies from Applications: Digital Signal Processing, Image processing, Speech processing.	<b>6</b>
	<b>Total</b>	<b>48</b>