

CS305: Parallel Computer Architecture	L	T	P	Computer Architecture
	3	1	0	

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

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

S. No	Contents	Contact Hours
UNIT 1	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.	6
UNIT 2	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.	6

UNIT 3	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	10
	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.	10
UNIT 4	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	10
UNIT 5	Applications: Case studies from Applications: Digital Signal Processing, Image processing, Speech processing.	6
	Total	48