MULTI-CORE ARCHITECTURE AND PROGRAMMING
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – VI

Subject Code	15CS666	IA Marks	20
Number of Lecture Hours/Week	3	Exam Marks	80
Total Number of Lecture Hours	40	Exam Hours	03

CREDITS – 03

Course objectives: This course will enable students to

- Explain the recent trends in the field of Computer Architecture and describe performance related parameters
- Illustrate the need for quasi-parallel processing.
- Formulate the problems related to multiprocessing
- Compare different types of multicore architectures

Module – 1	Teaching Hours
Introduction to Multi-core Architecture Motivation for Concurrency in software, Parallel Computing Platforms, Parallel Computing in Microprocessors, Differentiating Multi-core Architectures from Hyper- Threading Technology, Multi-threading on Single-Core versus Multi-Core Platforms Understanding Performance, Amdahl's Law, Growing Returns: Gustafson's Law. **System Overview of Threading** : Defining Threads, System View of Threads, Threading above the Operating System, Threads inside the OS, Threads inside the Hardware, What Happens When a Thread Is Created, Application Programming Models and Threading, Virtual Environment: VMs and Platforms, Runtime Virtualization, System Virtualization.	8 Hours

Module – 2	
Fundamental Concepts of Parallel Programming : Designing for Threads, Task Decomposition, Data Decomposition, Data Flow Decomposition, Implications of Different Decompositions, Challenges You'll Face, Parallel Programming Patterns, A Motivating Problem: Error Diffusion, Analysis of the Error Diffusion Algorithm, An Alternate Approach: Parallel Error Diffusion, Other Alternatives. **Threading and Parallel Programming Constructs:** Synchronization, Critical Sections, Deadlock, Synchronization Primitives, Semaphores, Locks, Condition Variables, Messages, Flow Control- based Concepts, Fence, Barrier, Implementation-dependent Threading Features	8 Hours

Module – 3	
Threading APIs : Threading APIs for Microsoft Windows, Win32/MFC Thread APIs, Threading APIs for Microsoft. NET Framework, Creating Threads, Managing Threads, Thread Pools, Thread Synchronization, POSIX Threads, Creating Threads, Managing Threads, Thread Synchronization, Signaling, Compilation and Linking.	8 Hours

Module – 4	
OpenMP: A Portable Solution for Threading : Challenges in Threading a Loop, Loop-carried Dependence, Data-race Conditions, Managing Shared and Private Data, Loop Scheduling and Portioning, Effective Use of Reductions, Minimizing Threading Overhead, Work-sharing Sections, Performance-oriented Programming, Using Barrier and No wait, Interleaving Single-thread and Multi-thread Execution, Data Copy-in and Copy-out, Protecting Updates of Shared	8 Hours

Variables, Intel Task queuing Extension to OpenMP, OpenMP Library Functions, OpenMP Environment Variables, Compilation, Debugging, performance	
Module – 5	
Solutions to Common Parallel Programming Problems : Too Many Threads, Data Races, Deadlocks, and Live Locks, Deadlock, Heavily Contended Locks, Priority Inversion, Solutions for Heavily Contended Locks, Non-blocking Algorithms, ABA Problem, Cache Line Ping-ponging, Memory Reclamation Problem, Recommendations, Thread-safe Functions and Libraries, Memory Issues, Bandwidth, Working in the Cache, Memory Contention, Cache-related Issues, False Sharing, Memory Consistency, Current IA-32 Architecture, Itanium Architecture, High-level Languages, Avoiding Pipeline Stalls on IA-32,Data Organization for High Performance.	**8 Hours**
Course outcomes: The students should be able to: - Identify the issues involved in multicore architectures - Explain fundamental concepts of parallel programming and its design issues - Solve the issues related to multiprocessing and suggest solutions - Point out the salient features of different multicore architectures and how they exploit parallelism - Illustrate OpenMP and programming concept	
Question paper pattern: The question paper will have TEN questions. There will be TWO questions from each module. Each question will have questions covering all the topics under a module. The students will have to answer FIVE full questions, selecting ONE full question from each module.	
Text Books:	
1. Multicore Programming , Increased Performance through Software Multi-threading by Shameem Akhter and Jason Roberts , Intel Press , 2006	
Reference Books:	
NIL	

H. O. D.
Dept. Of Information Science & Engineering
Alva's Institute of Engg. & Technology
Mijar, MOODBIDRI - 574 225