ICS803 Elective – III Multicore Architecture

Teacher Name:

Ms. Raksha Pandey

Course Structure

L	T	P	
3	1	0	4

Prerequisite:

Course Content:

Unit-I: Multi-core Architectures

Introduction to multi-core architectures, issues involved into writing code for multi-core architectures, Virtual Memory, VM addressing, VA to PA translation, Page fault, TLB- Parallel computers, Instruction level parallelism (ILP) vs. thread level parallelism (TLP), Performance issues, OpenMP and other message passing libraries, threads, mutex etc.

Unit-II: Multi-threaded Architectures

Brief introduction to cache hierarchy - Caches: Addressing a Cache, Cache Hierarchy, States of Cache line, Inclusion policy, TLB access, Memory Op latency, MLP, Memory Wall, 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

Unit-III: Compiler Optimization Issues

Code optimizations: Copy Propagation, dead Code elimination , Loop Optimizations-Loop Unrolling, Induction variable Simplification, Loop Jamming, Loop Unswitching, Techniques to improve detection of parallelism: Scalar Processors, Special locality, Temporal locality, Vector machine, Strip mining, Shared memory model, SIMD architecture, Dopar loop, Dosingle loop.

Unit-IV: Control Flow analysis

Control flow analysis, Flow graph, Loops in Flow graphs, Loop Detection, Approaches to Control Flow Analysis, Reducible Flow Graphs, Node Splitting. Dataflow analysis: Analysis of Structured programs, Reaching definition Analysis, Control Tree based.

Unit-V: Data-Flow Analysis

Data Flow analysis, Interval Analysis, Backward Analysis, Available Expression, Live variable Analysis, Very busy Expression, pointer analysis, alias analysis; Data Dependence Analysis : data Dependence, solving data dependence

equations (integer linear programming problem); Data Dependency graph, Basic Block dependence, Data Dependence in loops, iteration space, iteration Vector, Data dependency in parallel loops, Loop optimizations.

Text and Reference Books:

1. Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman, "Compilers: Principles, Techniques & Tools", 2nd Ed, 2006

2. Shameem Akhter and Jason Roberts, Multi-Core Programming, Intel Press, 2006 3. Randy Allen, Ken Kennedy, "Optimizing Compilers for Modern Architectures: A Dependence-based Approach", Morgan Kaufmann publishers, 2002

Lesson Plan

	Topic	**No. of Lectures**
UNIT –I	Introduction to multi-core architectures, issues involved into writing code for multi-core architectures	2
	Virtual Memory, VM addressing, VA to PA translation, Page fault,	2
	TLBParallel computers, Instruction level parallelism (ILP) vs. thread level parallelism (TLP),Performance issues	2
	OpenMP and other message passing libraries, threads, mutex etc	2
UNIT-II	Brief introduction to cache hierarchy - Caches: Addressing a Cache, Cache Hierarchy	2
	States of Cache line, Inclusion policy, TLB access, Memory Op latency	2
	MLP, Memory Wall, communication latency, Shared memory multiprocessors, General architectures and the problem of cache coherence	2
	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;	3

	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;	3
	Chip multiprocessor case studies: Intel Montecito and dual-core, Pentium4, IBM Power4, Sun Niagara	2
UNIT-III	Code optimizations: Copy Propagation, dead Code elimination , Loop Optimizations-Loop Unrolling, Induction variable Simplification	2
	Loop Jamming, Loop Unswitching, Techniques to improve detection of parallelism: Scalar Processors, Special locality, Temporal locality, Vector machine, Strip mining, Shared memory model, SIMD architecture, Dopar loop, Dosingle loop.	3
UNIT-IV	Control flow analysis, Flow graph, Loops in Flow graphs	2
	Loop Detection, Approaches to Control Flow Analysis, Reducible Flow Graphs, Node Splitting	2
	Dataflow analysis: Analysis of Structured programs, Reaching definition Analysis	2
UNIT-V	Control Tree based Data Flow analysis, Interval Analysis, Backward Analysis, Available Expression,	2
	Live variable Analysis, Very busy Expression, pointer analysis, alias analysis; Data Dependence Analysis : data Dependence, solving data dependence equations (integer linear programming problem)	2
	Data Dependency graph, Basic Block dependence, Data Dependence in loops, iteration space, iteration Vector, Data dependency in parallel loops, Loop optimizations.	3

Assignments

Assignment-1

Q1. Define Single Core CPU, Dual Core CPU and Quad Core CPU.

Q2. What are Vector Processors and GPU (Graphics Processing Unit)?

Q3. Differentiate SIMD and MIMD systems.

Q4. Explain the role of Distributed and Shared memory architectures in memory systems.

Assignment-2

Q1. Differentiate Snooping cache coherence and Directory based cache coherence.

Q2. Give Amdahl's law. Also define the scalability and taking timings.

Q3. What is a serial program? How to parallelize a serial program?

Q4. Why is algorithm complexity important? What is the impact of Data Structures on Performance?

Assignment-3

Q1. What is super-linear scaling. Explain with program.

Q2. Explain cache conflict and capacity. Define Data Races.

Q3. What are:

a. Spin Locks
b. Semaphores
c. Readers-Writers Locks
d. Barriers

Q4. Give the structure of Message Queues. What are other approaches to sharing data between Threads.

Assignment-4

Q1. Explain the Error Checking with function call and with check the number of threads.

Q2. Give brief description of **coherence** and **consistency** with Memory Model.

Q3. Give the brief explanation about:

a. The Parallel Directive (there clauses and restrictions)
b. The Parallel for Directive
c. Caveats
d. The atomic directive
e. Critical directive

Q4. Give a set of library functions and explain their role in OpenMP.

Assignment-5

Q1. Explain Cache-Coherence with example.

Q2. Discuss and describe the pitfalls of parallelism.

Q3. Give the OpenMP Execution Model. Explain with the help of program.

Q4. How can we improve performance through data density and locality?