Department of Electrical and Computer Engineering
Faculty of Engineering and Architecture
American University of Beirut

EECE 422 – Parallel Computer Architecture and Programming

Catalog description:
A course on high-performance computer architectures with emphasis on shared memory and distributed parallel architectures and programming models. Topics include: simultaneous multithreading processors, multicore processors, SIMD processors, UMA, NUMA and COMA shared-memory multiprocessors, distributed multiprocessors, snoopy and directory-based cache coherence protocols, memory consistency models, high performance synchronization methods, speculative lock elision, shared memory programming model, message passing programming model and transactional memory programming model. To consolidate the material presented in class, students work on designing parallel programs using the OpenMP threading environment and MPI message passing programming standard.

Credit hours: 3 credits

Required or elective: Elective for ECE and CCE students
Prerequisites:
By course: EECE 321,
By topic: Computer Organization

Textbook(s) and/or required materials:
David E. Culler and Jaswinder Pal Singh, with Anoop Gupta. Parallel Computer Architecture: A Hardware/Software Approach. Morgan Kaufmann, 1998.

Computer usage:
OpenMP and MPI parallel programming standards

Course Objectives

1. Understand how multithreaded processors work.
2. Understand how SIMD processors work.
3. Understand how multicore and multiprocessor computer systems work.
4. Understand how multiprocessors implement advanced synchronization techniques
5. Learn the identifying characteristics of popular parallel programming models.
6. Learn how to design parallel programs using OpenMP and MPI.

Course Topics

1. Introduction
2. Multithreaded processors
3. SIMD/Vector Processors
4. Multicore and Uniform Memory (UMA) Access processors and Snoopy Cache Coherence
5. Distributed Multiprocessors and Directory Based Cache Coherence

6. Cache Only Memory Architecture (COMA)
7. Memory Consistency models
8. Synchronization with locks in shared memory multiprocessors
9. Speculative Lock Elision and Transactional Memory
10. Parallel programming models
11. Designing Parallel Programs: partitioning, communication and synchronization, data dependences, granularity and load balancing, I/O, performance analysis and tuning
12. Parallel programming with OpenMP
13. Parallel programming with MPI

Course Learning Outcomes

1. Explain how vector/SIMD architectures enhance execution performance.
2. Explain different types of multiprocessor architectures and multiprocessor interconnects.
3. Explain different cache coherence techniques and how they are supported in hardware.
4. Explain different memory consistency models and their role in shared memory programming.
5. Explain how synchronization with locks works in a shared memory multiprocessor system.
6. Explain benefits and hardware support of speculative lock elision and transactional memory.
7. Write parallel programs using OpenMP.
8. Write parallel programs using MPI.

Class/laboratory schedule:

Two 75-minutesd lectures or three 50-minute lectures per week.

Evaluation methods

1.	Programming Assignments	30%
2.	Quizzes	40%
3.	Final Exam	30%

Professional component

Engineering topics:	100%
General education:	0%
Mathematics and basic sciences:	0%

Person(s) who prepared this description and date of preparation

Haitham Akkary, October 2012.

Date of last revision: Feb 3, 2013.