**ECE568: Introduction to Parallel Processing – Spring Semester 2015**

**Professor Ahmed Louri**

**A-Introduction:**

The need to solve ever more complex problems continues to outpace the ability of today's most powerful computers to execute the required programs within reasonable time periods, power budgets and cost. Technology constraints have forced computer designers to bet their entire future on **parallelism (concurrency)** as the only method of improving computing performance. This change is impacting every facet of our society as every computing device from cell phones, PCs, servers, to supercomputers and datacenters will need to confront parallelism**. Parallel processing refers to a class of computing techniques that perform more than one operation at a time. These techniques are dominating the design of modern computers at present time**. By connecting a number of processors together into a single system and having these connected processors cooperate to solve a single problem that exceeds the ability of any one of the processors, it is hoped to dramatically reduce computation time and power consumption for a wide range of complex computing problems.

**B-Course Objectives:**

The objectives of this course are:

(1) To study how parallel computers work and how to analyze the correct designs of parallel architectures, especially within the technological constraints. The course is a comprehensive study of modern parallel computer architectures and parallel processing techniques and their applications from basic concepts to state-of-the-art computer systems. It provides in-depth coverage of fundamentals, design complexity, power, reliability and performance coupled with treatment of parallelism at all levels.

(2) To prepare students for a career in designing the computer systems of the future.

**C-Topics to be covered in this course include:**

**First,** the need for parallel processing and the limitations of uniprocessors are introduced.

**Second**, a substantial overview and basic concepts of parallel processing and their impact on computer architecture are introduced. This will include major parallel processing paradigms such as pipelining, vector processing, instruction, thread, data level parallelism (ILP, TLP, DLP), vector processing, GPUs, GPGPUs, accelerators, simultaneous-multithreading (SMT), multi/many-core processors, shared-memory multiprocessors, distributed multi-computing, and cloud/datacenter computing.

**Third**, we then address the architectural support for parallel processing such as (1) parallel memory organization and design, (2) parallel cache design and cache coherence strategies, (3) shared-memory vs distributed-memory systems, (4) processor/core design, (5) communication networks for parallel computers, (6) emerging technologies for parallel processing: (for memory, interconnect, processing)

**Fourth,** we will study principles of parallel programming design and examples of parallel programs for each category

**Finally**, and if time permits, we will study case studies of modern parallel computers (commercial as well as experimental).

While the course is suitable for students interested in computer engineering, it also provides foundations for students interested in parallel computing, computer performance evaluation, technology advancement and future trends in computing design.

**D-Relevant Books:**

**The subject matter of this course is very diverse and cannot be captured by a single textbook. I will be using several books (listed below)**. However, I would recommend that you get hold of the following book

- 1-"*Parallel Computer Organization and Design*", Michel Dubois, Murali Annavaram, and Per Stenstrom, Cambridge University Press, 2012

- Other additional books are:
- 2- *"Advanced Computer Architecture: Parallelism, Scalability, Programmability"*, by Kai Hwang, McGraw Hill 1993. As you can see this is a bit old but has many of the fundamentals of the field.
- *3-"Parallel Computer Architecture: A Hardware/Software Approach"*, by David Culler J. P. Singh, Morgan Kaufmann, 1999.
- *4- "Scalable Parallel Computing: Technology, Architecture, Programming"* Kai Hwang and Zhiwei Xu, McGraw Hill 1998.
- *5-"Introduction to Parallel Computing," second edition* Ananth Grama Gupta, Karypis, Kumar, Pearson, Adison Wesley, 2003.
- 6-"*Principles and Practices of Interconnection Networks,"* William J. Dally and Brian Towles, Morgan Kaufmann 2003.

Additionally, there are several other books written on various parts of the course that can be considered as optional. Please come see me for more details.

There will also be several handouts from recent technical conferences and journals related to the field. These will be available throughout the semester.

**Instructor:** Dr. Ahmed Louri, **Course days:** Tuesday/Thursday 12:30 - 1:45 pm

**Office Hours**: Tuesday/Thursday 3:30 - 4:30.

**Means of Contact**: louri@ece.arizona.edu, office ECE 456S

**Prerequisites:** Knowledge of computer organization at the undergraduate level is sufficient (courses like ECE 369, ECE 462/562, ECE 372 would be sufficient).

**D-Grading Policy:**

1- Homework (25%): Conceptual "paper and pencil" problems with the main goal of giving students an opportunity to think hard and in depth about the design concepts studied in each sub-topic while testing their ability to think abstractly. There will be 3 – 4 assignments
2- Class Recitations (25%): A recitation is a presentation made by a student to demonstrate knowledge of a subject matter. Students will be assigned recent papers on parallel processing. They will study these papers and present the contents in class and answer questions
3- Two exams (40%) (one midterm exam and one final exam)
4- Class participation (10%).

**E-Tentative Course Outline (Topics listed below may not be covered in this order).**

**1-Introdution to Modern Computer Architectures (This will be a brief review or background material).**

(a) What is computer architecture
(b) Need for parallel processing
(c) Components of a parallel architecture
(d) Performance (specifically parallel computing performance metrics)
(e) Technological challenges

**2-Classes of parallel computers (brief introduction to the models).**

(a) Bus-based symmetric multiprocessors (SMPs)
(b) Array Processors (SIMD architectures: Streaming, GPUs, GPGPUs, etc.)
(c) Shared-memory multiprocessors (UMA, NUMA, CC-NUMA, COMA, etc.)
(d) Message-passing multiprocessors
(e) Distributed systems

**3- Communication in parallel computers: Interconnection Networks**

(a) Basic communication performance criteria
(b) Design space for interconnection networks
(c) Switching strategies
(d) Network topologies: static topologies, dynamic topologies, reconfigurable networks
(e) Routing strategies
(f) Flow control

(g) Network-on-Chips (NoCs)
(h) Technologies for interconnects

**4- Multiprocessor systems**

(a) Symmetric Multiprocessors (SMPs)
(b) Snoopy cache coherence protocols for SMPs
(c) Distributed-shared memory systems (DSMs)
(d) Cache coherence for DSMs: Directory-based protocols
(e) Scalable-shared memory systems
(f) Cache-only shared-memory systems

**5-Chip Multiprocessors (CMPs) (Multicore and Many-core architectures)**

(a) Rationale for CMPs
(b) Core Multi-threading
(c) CMP architectures
(d) Programming models for CMPs

**6-Distributed systems (very brief as there is an entire course on this topic: ECE 677).**

(a) Message-passing programming models
(b) Clusters
(c) Data centers and ware-house scale computing

**7- Parallel program design and parallel programming**

(a) Parallel program design principles (briefly)
(b) Parallel programming languages: MPI, OpenMP, CUDA, etc.
(c) Examples of parallel programming

**8- Case studies of modern commercial parallel computers (it time permits)**

**F- What do you expect to get out of this course?**

1-In-depth understanding of the design and engineering of modern parallel computers.

2- Technology forces impacting future computing

3-Fundamental architectural issues

4-Basic design techniques for processors, memory, interconnects, caches, programs, etc.

5- Understanding of cache coherence protocols, interconnection networks, parallel memory, etc.

6- Understanding of the underlying engineering trade-offs in parallel computing

7- Multicore design

8- Basics of parallel programming design

9- Interaction between machine design and practical parallel programming

10 -Future directions in computing design

**G-Class Interaction:**

- Show initiative and participate in the discussion
- Ask questions during class
- Read class material
- Come to office hours
- Don’t be intimidated, keep in mind that in the learning process EVERY QUESTION IS A VALID QUESTION!
- If you don’t understand ask!
- Be proactive and you will succeed in this course