

750731, Parallel Computer Architecture

3 hours per week, 3 credit hours, prerequisite: **none**

Teaching Method: 30 hours Lectures (2 hours per week), 7 hours Seminars (1 per 2 weeks), 8 hours

Laboratories (1 per 2 weeks)

Aims: The aims of this module are to introduce the main classes of parallel computer architectures and their distinguishing characteristics; examine some examples of particular architectures that were important in the development of parallel computers; examine some examples of recent parallel architectures; introduce some main principles of parallel programming and the effects of the architecture on algorithm design; understand how to write efficient parallel programs.

Learning Outcomes:

On completion of this module, the student should be able to:

- Understand the main classes of parallel architectures and their distinguishing characteristics and relative merits.
- Understand the major characteristics of some examples of the main architectural classes.
- Be aware of various styles of parallel hardware architecture and the structure and operation of their major components.
- Understand the impact of parallel computer architecture on software design and performance.
- Understand the fundamental aspects of parallel processing
- Appreciate the nature of a small number of performance-critical applications.
- Comprehend the distinction between message-passing (process-based) and data-sharing (threadbased) programming models
- Be capable of performance analysis of (competing designs for) hardware components for parallel systems.

- Understand the link between source-code and hardware behaviour, and be capable of analysing the execution performance of small fragments of data-parallel code in terms of various classes of overheads associated with parallel execution.
- Be familiar with performance measures for parallel systems.
- Understand the theoretical limitations of parallel computing such as intractability.
- Design and analyze simple parallel algorithms.
- Write efficient parallel application programs

Textbooks and Supporting Materials:

1- Foster, I. T., *Designing and Building Parallel Programs: Concepts and Tools for Parallel Software Engineering*, (ISBN 0-201-575-949), Addison-Wesley, 1995. (Very good on message-passing

style of programming, but has useful chapters on other aspects of the course)

2- Culler, D. E. and Singh, J.P., with Gupta A., *Parallel Computer Architecture: a hardware/software approach*, (ISBN 1-55860-343-3), Morgan Kaufmann 1999. (Very good book on architecture level)

3- Hockney, R. W. and Jesshope, C.R., *Parallel Computers 2*, (ISBN 0-862-748-124), Adam Hilger, 1988. (A good, albeit old-fashioned reference text for fundamental techniques)

20

4- Barry Wilkinson, Michael Allen. *Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers*, 2nd ed., Pearson/Prentice Hall, 2005. ISBN 0-13-140563.

5- Jessen Havill, *An Introduction to Linux at Denison*, 2003.

6- David E. Culler and Jaswinder Pal Singh, with Anoop Gupta. *Parallel Computer Architecture: A Hardware/Software Approach*, Morgan Kaufmann, 1998. ISBN: 1-55860-343-3.

7- George Almasi and Allan Gottlieb, *Highly-Parallel Computing*, 2nd Edition, Benjamin-Cummings, 1994.

8- Michael J. Quinn. *Parallel Programming in C with MPI and OpenMP*, McGraw Hill, 2004, ISBN 0-

07-282256-2

9- Grama, A. Gupta, G. Karypis and V. Kumar. Introduction to Parallel Computing, 2n edition,

Addison

Wesley, 2002, ISBN 0-201-64865-2.

10- H. El-Rewini, T.G. Lewis, Distributed and Parallel Computing, Manning, 1997, ISBN 0-13-

795592-

8.

11- B. Wilkinson, Computer Architecture Design and Performance, Prentice Hall, 1991

12- Dezso Sima, Terence Fountain, Peter Kacsuk, Advanced Computer Architectures A design

Space

Approach, Addison Wesley, 1997

*** Plus some Research Papers on the topics**

Synopsis:

1- Computer Organization for parallel and distributed computing: Pipeline and vector processors,

Multicomputers and computer networks, multiprocessors: Synchronization, Interprocess

communication;

2- Massively parallel architecture: Associative processors, Array Processors (SIMD), Large-scale

MIMD; Non-von Neumann-type computers: data-flow machines, reduction machines

3- Introduction on Parallel Architectures: Taxonomy of systems, SIMD, MIMD, systolic arrays.

Parallel

programming paradigms. Applications using a multiple processor parallel architecture.

4- SIMD Array Processors: Example architectures: Illiac IV, AMT DAP, Connection Machine, systolic

arrays.

5- Interconnection Networks: Basic characteristics, routing functions, network topologies, dynamic

(switched) networks (cross-bar, bus, Benes networks, shuffle-exchange networks). Example:

Connection Machine.

6- MIMD Shared-Memory Architectures: shared memory multi-processors: Example architectures: BBN Butterfly, DEC AlphaServer 8000 as a symmetric multiprocessor architecture.

7- MIMD Message-Passing Architectures: The Transputer as a building block, complete machines using Transputers, hypercube machines (Cosmic Cube, Intel Hypercube and hypercubes using Transputers).

8- Programming Parallel Machines: Parallel Computation Models, writing parallel programs for abstract machines, the divide-and-conquer paradigm and some example algorithms, the doubling paradigm and example algorithms, programming real machines (SIMD, shared-memory MIMD, and message-passing MIMD)

9- Performance Measures: Granularity, Speed Up, Efficiency, Cost

Assessment: Two 1-hour midterm exams (15% each); Assignments (10%) (two assignments on parallel architectures and possible computations on them); Writing a research paper/Presentation/Final project (20%); 2-hours Final Exam (40%)