جامعة الملك فهد للبترول والمعادن

College of Computer Science and Engineering
Information and Computer Science Department

Syllabus
ICS 233 – Computer Architecture and Assembly Language
Second Semester 2006-2007 (062)

1. Course Description:

Machine organization; Assembly language: addressing, stacks, argument passing, arithmetic operations, decisions, modularization; Input/Output Operations and Interrupts; Memory Hierarchy and Cache memory; Pipeline Design Techniques; Super-scalar architecture; Parallel Architectures.

2. Prerequisite: COE 202, ICS 201

3. Textbook:

a. *Computer Organization and Design: The Hardware / Software Interface*. Third Edition. David A. Patterson John L. Hennessy. Morgan Kaufmann, 2005. ISBN: 1558606041 (Lecture).

b. *MIPS Assembly Language Programming*. First edition. Robert L. Britton; Prentice Hall; ISBN: 0131420445 (Lab).

4. Reference Books:

1. *Computer Architecture: Design and Performance*, Barry Wilkinson, Prentice-Hall, 2nd Edition 1996. ISBN: 0-13-518200-X.
2. *Computer Systems Design and Architecture*, Vincent P. Heuring & Harry F. Jordan. Addison Wesley, 1997. ISBN: 0-8053-4330-X.

5. Course Objective:

a. Expose students to the trade-off analysis in designing various aspects of Computer Architecture, which includes Control Unit Design, Instruction Set Design, Memory Hierarchy, Instruction Level Pipeline, and Multi-processing.

b. To provide students with intermediate level experience in assembly language programming.

6. Course Coverage Outline:

The course is organized into seven modules:

1. Data representation
2. Machine organization

3. MIPS Assembly language programming
4. Memory System Design
5. Input/Output subsystems
6. Functional Organization
7. Introduction to Parallel architectures

7. Learning Outcomes:

Upon successful completion of this course, the student will be able to:

1. Ability to explain how pipelining and parallelism can be applied to the design of scalar and superscalar processors.
2. Ability to describe the principles of memory management.
3. Ability to analyze memory latency, cycle time, bandwidth, and interleaving.
4. Understanding the design of the data path and the control unit of a simple CPU, and the fetching and execution control sequences of simple instructions at the register transfer level.
5. Understanding the design of the control unit and the generation of the control signals using hardwired and micro-programmed implementations.
6. Awareness of various architectural enhancements and instructions sets and their effects on system performance.
7. Ability to analyze, design, implement, debug, and test assembly language programs.

8. Assessment:

- Lab attendance is mandatory.
- Students' work in the lab, the lab quizzes and lab project are kept as the evaluation strategy.
- Grading policy:

Lab work evaluation	7 %
Lab Quizzes (2 × 3.5)	7 %
Project on MIPS Assembly Language	6 %

9. Detailed Lab Coverage:

Week	Lab Topic	Chapters in textbooks
1	Introduction to Computer Systems Components and their Functions	Chap 1 (LAB)
2	Introduction to Assembly Language Programming	Chap 4 (LAB)
3	Algorithm Development in Pseudocode.	Chap 2 (LAB), 3 (LAB), 4 (LAB)

4	Array, Flow control, and SYSCALL instructions	Chap 2 (LAB), 4 (LAB), 5 (LAB)
5	Function and Stack	Chap 5 (LAB), 6 (LAB), 7 (LAB)
6	**Lab Quiz-1** Exception and Interrupts	Chap 8 (LAB), Chap 9 (LAB)
7	Integer Multiplication and Division	
8	Floating Point Instructions	Chap 11 (LAB)
9	**Lab Quiz-2** Designing a register file	
10	Data Path Design	
11	Control Unit Design	
12	Pipelined Data path Design	
13	Pipeline Hazards	Chap 10 (LAB)
14	Cache Memory	
15	**Project Presentation**	

10. <u>Important Notes:</u>

1. Lab attendance is a must. Having ***three*** lab absences or more will earn you a DN grade.
2. No make-up lab quizzes.
3. Cheating in whatever form will not be taken lightly and could lead to damaging consequences.
4. You will get the best of this course if you attend all quizzes and submit the project perfectly.