

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
COLLEGE OF COMPUTER SCIENCES & ENGINEERING

COMPUTER ENGINEERING DEPARTMENT

COE 205 Computer Organization & Assembly Language
Syllabus - Term 101

Catalog Description

Introduction to computer organization. Signed and unsigned number representation, character representation, ASCII codes. Assembly language programming, instruction format and types, memory and I/O instructions, dataflow, arithmetic, and flow control instructions, addressing modes, stack operations, and interrupts. Datapath and control unit design. RTL, microprogramming, and hardwired control. Practice of assembly language programming.

Prerequisite: COE 202 and ICS 102

Instructor Dr. Aiman H. El-Maleh. Room: 22/407-5 Phone: 2811
Email: aimane@kfupm.edu.sa

Office Hours SMW 11:00-12:00, T 1:00-2:00 PM and by appointment

Course Objectives

After successfully completing the course, students will be able to:

1. Describe the basic components of a computer system, its instruction set architecture and its basic fetch-execute cycle operation.
2. Describe how data is represented in a computer and recognize when overflow occurs.
3. Recognize the basics of assembly language programming including addressing modes.
4. Analyze, design, implement, and test assembly language programs.
5. Recognize, analyze, and design the basic components of a simple CPU including datapath and control unit design alternatives.

Course Learning Outcomes

1. Ability to analyze, design, implement, and test assembly language programs.
2. Ability to use tools and skills in analyzing and debugging assembly language programs.
3. Ability to design the datapath and control unit of a simple CPU.
4. Ability to demonstrate self-learning capability.
5. Ability to work in a team.

Text Books & References

- **Kip Irvine: Assembly Language for Intel-Based Computers, 4th edition, 2002.**
- *Introduction to Assembly Language Programming: From 8086 to Pentium Processors*, Sivarama P. Dandamudi, et al., Springer Verlag, 1998. (ISBN: 0387985301).
- *Computer Systems Design and Architecture*, Vincent Heuring, Harry F. Jordan, Miles Murdocca, Addison Wesley 1997. (ISBN 0-8053-4330-X).
- *Assembly Language Programming and Organization of the IBM PC*, Ytha Yu and Charles Marut, McGraw Hill, 1992. (ISBN: 0-07-072692-2).
- *Online Material:* assembly.pc.ccse.kfupm.edu.sa

Grading Policy

Discussions & Reflections	5%
Programming Assignments	10%
Quizzes	10%
Exam I	15% (Th. Oct. 28, 2010)
Exam II	20% (Sat. Dec. 18, 2010)
Laboratory	20%
Final	20%

- Attendance will be taken regularly. For each missed 3 classes, a penalty of 0.5 will be deducted.
- Excuses for officially authorized absences must be presented no later than one week following resumption of class attendance.
- Late assignments will be accepted but you will be penalized 10% per each late day.
- A student caught cheating in any of the assignments will get 0 out of 10%.
- No makeup will be made for missing Quizzes or Exams.

Course Topics

1. **Introduction and Information Representation.** **7 lectures**
Introduction to computer organization. Instruction Set Architecture. Computer Components. Fetch-Execute cycle. Signed number representation ranges. Overflow.
2. **Assembly Language Concepts.** **7 lectures**
Assembly language format. Directives vs. instructions. Constants and variables. I/O. INT 21H. Addressing modes.
3. **8086 Assembly Language Programming.** **19 lectures**
Register set. Memory segmentation. MOV instructions. Arithmetic instructions and flags (ADD, ADC, SUB, SBB, INC, DEC, MUL, IMUL, DIV, IDIV). Compare, Jump and loop (CMP, JMP, Cond. jumps, LOOP). Logic, shift and rotate. Stack operations. Subprograms. Macros. I/O (IN, OUT). String instructions. Interrupts and interrupt processing, INT and IRET.
4. **CPU Design.** **12 lectures**
Register transfer. Data-path design. 1-bus, 2-bus and 3-bus CPU organization. Fetch and execute phases of instruction processing. Performance consideration. Control steps. CPU-Memory interface circuit. Hardwired control unit design. Microprogramming. Horizontal and Vertical microprogramming. Microprogrammed control unit design.