King Fahd University of Petroleum and Minerals
College of Computer Sciences and Engineering
Department of Computer Engineering

COE 301 Computer Organization (3-3-4)
ICS 233 Computer Architecture & Assembly Language (3-3-4)

Instructor: Dr. Marwan Abu-Amara
Office: 22-145
Phone: 1632
E-mail: marwan@kfupm.edu.sa

Term: 162 (2nd term 2016–2017)
Day & Time: UTR 08:00 AM – 08:50 AM
Location: 24-114
Prerequisite: COE 202 and ICS 201
Textbook: David A. Patterson and John L. Hennessy, *Computer Organization and Design: The Hardware /Software Interface*, Fifth Edition, Morgan Kauffmann Publishers, 2013.
Office Hours: UTR 11:00 AM – 11:55 AM (or by appointment)
Web Site: http://faculty.kfupm.edu.sa/COE/marwan

Tentative Grading Policy:

- Prog. Assignments **10%** (Each prog. assignment may carry a different weight)
- Quizzes **10%** (Each quiz may carry a different weight)
- Laboratory **10%**
- Project **10%**
- Major Exam I **20%** (Saturday March 18, 2017, 10:00 AM)
- Major Exam II **20%** (Saturday April 29, 2017, 10:00 AM)
- Final Exam **20%** (Sunday May 28, 2017, 12:30 PM)

Course Learning Outcomes

1. Describe the organization and operation of integer and floating-point arithmetic units. **(COE 301)**
2. Apply the knowledge of mathematics to processor performance analysis. **(COE 301)**
3. Design the datapath and control of a processor. **(COE 301)**
4. Describe the memory hierarchy and caches. **(COE 301)**
1. Analyze, write, and test MIPS assembly language programs. **(ICS 233)**
2. Use software tools for assembly language programming and for CPU design and simulation. **(ICS 233)**

IMPORTANT NOTES:

- All KFUPM regulations and standards will be enforced. Attendance will be checked each class. The KFUPM rule pertaining to a DN grade will be strictly enforced (i.e. **> 9 absences** will result in a DN grade).
- If you are late to the class for more than 5 minutes (i.e. arrive after 08:05 AM), you will **NOT be allowed to enter** the classroom and you will be considered absent for that class.
- Only university approved/certified excuses will be accepted, and should be presented **no later than 1 week** after absence.
- Use of cell phones, smart phones, and tablets during class period and during exams is absolutely **prohibited**.
- Assignments are to be submitted **in class** on the due date during the class period. Late assignments will **NOT be accepted**.
- You have up to the next class period to object to the grade of a assignment, a quiz, or a major exam from the end of the class time in which the graded papers have been distributed back. If for some reason you cannot contact me within this period, send me an email requesting an appointment. The email should be sent before the next class period.
- **NO make-up exams**. ALL assignments and quizzes will be counted towards your grade.
- All exams are **common**.

Weekly Breakdown

Week	Topics
1	• Introduction to computer organization, high-level, assembly, and machine languages. Classes of computers, components of a computer system, technology improvements, chip manufacturing process, programmer's view of a computer system.
2	• Introduction to assembly language programming, instructions, registers, assembly language statements, directives, text, data, and stack segments. Defining data, arrays, and strings. Memory alignment, byte ordering, and symbol table. System calls, console input and output.
3	• Review of unsigned/signed integers, binary addition and subtraction, carry and overflow. • MIPS instruction set architecture, instruction formats, R-type integer arithmetic, logic, and shift instructions, immediate operands, I-type arithmetic and logic instructions, pseudo-instructions.
4	• Control flow, branch and jump instructions, translating if-else statements and logical expressions. Compare instructions, and conditional-move instructions. • Arrays, allocating arrays statically in the data segment and dynamically on the heap, computing the memory addresses of array elements.
5	• Load and store instructions, translating loops, using pointers to traverse arrays, addressing modes, jump and branch limits. • Defining functions (procedures) in assembly language, function call and return instructions. Passing arguments by value and by reference in registers, and the return address register.
6	• The stack segment, allocating and freeing stack frames, leaf versus non-leaf functions, preserving registers across function calls. Allocating and referencing a local array on the stack. Bubble Sort example and its translation into assembly code. Recursive functions, allocating stack frames and translating recursive functions into assembly language.
7	• Integer multiplication, unsigned and signed multiplication, sequential multiplier hardware, faster (tree) hardware multiplier. • Integer division, sequential division hardware, integer multiplication and division instructions in MIPS, and example programs.
8	• Floating point representation, IEEE 754 standard, normalized and de-normalized numbers, zero, infinity, NaN, FP comparison, FP addition, FP multiplication, rounding and accurate arithmetic. MIPS floating-point instructions. Floating-point programs. Example on Matrix Multiplication.
9	• Designing a processor, register transfer level, datapath components, clocking methodology, single-cycle datapath, implementing a register file and multifunction ALU. • Control signals and control unit, ALU control, single-cycle delay analysis and clock cycle.
10	• CPU performance and metrics, CPI, performance equation, MIPS as a metric, Amdahl's law, energy and power consumption, benchmarks and performance of recent processors. • Multi-cycle instruction execution, CPI of a multi-cycle processor, Performance comparison of a single-cycle versus a multi-cycle processor.
11	• Pipelining versus serial execution, MIPS 5-stage pipeline, pipelined datapath, pipelined control, pipeline performance.
12	• Pipeline hazards: structural, data, and control hazards, load delay, hazard detection, stall and forwarding unit, and delayed branching.
13	• Main memory organization and performance, SRAM, DRAM, latency and bandwidth, memory hierarchy, cache memory, locality of reference.
14	• Cache memory organization: direct-mapped, fully-associative, and set-associative caches, handling cache miss, write policy, and replacement policy.
15	• Cache performance, memory stall cycles, and average memory access time. • Review