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: 22-204
Phone: 1632
E-mail: marwan@kfupm.edu.sa
Term: 172 (2nd term 2017–2018)
Day & Time: UTR 10:00 AM – 10:50 AM
Location: 24-153
COE 301 Prerequisite: COE 202 and ICS 102 **ICS 233 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 09:00 AM – 09:50 AM (or by appointment)
Web Site: http://faculty.kfupm.edu.sa/COE/marwan

Tentative Grading Policy:

- Attendance **2%** (– 0.25% for every unexcused absence. 10 absences = DN grade)
- Prog. Assignments **8%** (Each prog. assignment may carry a different weight)
- Quizzes **10%** (Each quiz may carry a different weight)
- Laboratory **10%**
- Project **15%**
- Midterm Exam **25%** (Saturday March 24, 2018 – 10:00 AM)
- Final Exam **30%** (Tuesday May 15, 2018 – 08:00 AM)

Course Learning Outcomes

1. Describe the instruction set architecture of a MIPS processor
2. Analyze, write, and test MIPS assembly programs
3. Describe organization and operation of integer and floating-point arithmetic units
4. Design the datapath and control of a single-cycle (non-pipelined) CPU
5. Design the datapath and control of a pipelined CPU and handle hazards
6. Describe the organization and operation of memory and caches
7. Analyze the performance of processors and caches

IMPORTANT NOTES:

- Only university approved/certified excuses will be accepted, and should be presented **no later than 1 week** after returning back.
- Use of cell phones, smart phones, and tablets during class period and during exams is absolutely **prohibited**.
- Programming assignments are to be submitted on the due date by the deadline time. Late assignments will **NOT be accepted**.
- You have up to the next class period to object to the grade of a programming assignment, a quiz, or an 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 programming assignments and quizzes will be counted towards your grade.
- All exams are **common**.

Course Topics

Week	Topics	Ref.
1	• Introduction to computer organization, high-level, assembly, and machine languages. Components of a computer system.	1.1-1.5
2	• Memory Hierarchy, Fetch-Execute Cycle. Technology Improvements. 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.	1.1-1.5 A.9-A.10
3	• Integer storage sizes, review of 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.	Class notes, 2.1-2.7
4	• MIPS Integer multiply and divide instructions. • Control flow, branch and jump instructions, translating if-else statements and logical expressions. Compare instructions, and conditional-move instructions.	Class notes, p. 195 2.1-2.7
5	• Arrays, allocating arrays statically in the data segment and dynamically on the heap, computing the memory addresses of array elements. • Load and store instructions, translating loops, using pointers to traverse arrays, addressing modes, jump and branch limits.	Class notes, 2.1-2.7 2.10, 2.14 A.9-A.10
6	• 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. • 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, translating recursive functions into assembly language.	2.8
7 & 8	• Floating point representation, IEEE 754 standard, de-normalized numbers, zero, infinity, NaN. • FP comparison, FP addition, FP multiplication, rounding and accurate arithmetic. • MIPS floating-point instructions: load/store, arithmetic, data movement, convert, compare, branch, FP system calls. Floating-point programs. Example on Matrix Multiplication.	3.5
Midterm Exam (Saturday March 24, 2018 – 10:00 AM)		
9 & 10	• 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.	4.1-4.4
11	• CPU performance and metrics, CPI of a multi-cycle processor, performance equation, performance comparison of a single-cycle versus a multi-cycle processor. • Amdahl's law, energy and power consumption, benchmarks and performance of recent processors.	1.6
12 & 13	• Pipelining versus serial execution, timing diagrams, MIPS 5-stage pipeline, pipelined datapath, pipelined control, pipeline performance • Pipeline hazards: structural, data, and control hazards, load delay, hazard detection, stall and forwarding unit, delayed branching, and branch prediction.	4.5-4.8
14 &15	• Main memory organization, SRAM vs DRAM storage cells, DRAM refresh cycles, latency and bandwidth, trends in DRAMs, memory hierarchy, cache memory, locality of reference. • Cache memory organization: direct-mapped, fully-associative, and set-associative caches, handling cache miss, write policy, write buffer, and replacement policy. • Cache performance, memory stall cycles, and average memory access time. Introduction to multi-level caches, multi-level cache performance.	5.1-5.4