# COE 233: Digital Logic and Computer Organization
# Spring 2022 – Term 212

## COMPUTER ENGINEERING DEPARTMENT

## KING FAHD UNIVERSITY OF PETROLEUM & MINERALS


**Instructor:** Dr. Muhamed Mudawar Office: 22/410-2 Phone: 4642
**Office Hours:** UTR 10 AM to 12 noon or by appointment
**Course URL:** https://faculty.kfupm.edu.sa/coe/mudawar/coe233/
**Email:** mudawar@kfupm.edu.sa


### Catalog Description

Number systems. Boolean Algebra and Minimization of Boolean functions. Combinational circuits analysis and design: multiplexers, decoders, adders, and ALU. Sequential circuits analysis: flip-flops, Registers. MIPS instruction set architecture. Assembly language: selection and repetition structures. Single-cycle and pipelined processor design. Memory hierarchy. ROM, RAM, and cache memories. Evaluation of processor and cache performances.

**Prerequisite: ICS 104**

### Course Objectives

1. To offer students basic and intermediate levels of knowledge on digital logic, assembly language, and computer organization.
2. To enable students to design and evaluate logic circuits, pipelined processors, and cache memories.

### Course Learning Outcomes

After completing the course, students should be able to:

1. Design small and medium scale logic circuits.
2. Describe the instruction set architecture of a processor, such as MIPS.
3. Design and implement MIPS Assembly Language programs.
4. Design the data path and control of single cycle and basic pipelined processors.
5. Describe the organization and operation of memory and caches.
6. Analyze and evaluate the performance of processors and caches.

### Reference

David Harris and Sarah Harris, *Digital Design and Computer Architecture*, Second Edition, Morgan Kaufmann, 2013.

### Grading

| | | |
|---|---|---|
| Assignments | 20% | |
| Quizzes | 20% | |
| Midterm Exam | 30% | **Saturday, March 19, at 10 AM** |
| Final Exam | 30% | **To be announced** |

| Week | Topics |
|---|---|
| 1 | • Weighted number systems: decimal, binary, octal, hexadecimal, and base conversions.<br>• Representing fractions.<br>• Binary codes, character codes, ASCII. |
| 2 | • Binary and hexadecimal addition and subtraction<br>• Binary multiplication and bit shifting.<br>• Signed Integers: sign-magnitude, 1's complement, and 2's complement.<br>• Range, carry, overflow, converting subtraction into addition of 2's complement. |
| 3 | • Boolean algebra, truth tables, basic identities, DeMorgan's theorem.<br>• Algebraic manipulation and expression simplification.<br>• From truth table to logic expression: minterms and maxterms.<br>• Logic gates, logic diagrams, standard forms, and two-level gate implementation. |
| 4 | • Additional gates: NAND, NOR, XOR, XNOR, and their properties.<br>• Half-Adder, Full-Adder, and Ripple-Carry Adder.<br>• Decoders, Implementing Functions with Decoders, Multiplexers |
| 5 | • Design examples, designing a simple ALU.<br>• Memory elements: latches, flip-flops, and registers.<br>• ROM and static RAM. |
| 6 | • Introduction to assembly language programming, instructions, registers, integer storage sizes, assembly-language statements, directives, memory segments.<br>• Defining data, arrays, and strings. Memory alignment, byte ordering, and symbol table. System calls, console input and output. |
| 7 | • MIPS instruction formats, R-type integer arithmetic, logic, and shift instructions, immediate operands, I-type arithmetic and logic instructions, pseudo-instructions.<br>• Control flow, branch and jump instructions, translating if-else statements and logical expressions. Compare instructions, and conditional-move instructions. |
| 8 | • Arrays, allocating arrays statically in the data segment and dynamically on the heap, computing the memory addresses of array elements.<br>• Load and store instructions, translating loops, using pointers to traverse arrays, addressing modes, jump and branch limits. |
| 9 | • MIPS Integer multiply and divide instructions, Integer to string conversion and vice-versa.<br>• 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. |
| 10 | • The stack segment, allocating and freeing stack frames, leaf versus non-leaf functions, preserving registers. Bubble Sort example and its translation into assembly code.<br>• Designing the components of a single-cycle processor. |
| 11 | • Assembling a datapath from its components.<br>• Control signals and control unit, ALU control, and PC control. |
| 12 | • CPU performance and metrics, CPI of a multi-cycle processor.<br>• Performance equation, MIPS as a metric, Amdahl's law.<br>• Performance comparison of a single-cycle versus multi-cycle processor. |
| 13 | • Pipelining versus serial execution, timing diagrams<br>• MIPS 5-stage pipeline, pipelined datapath, pipelined control, pipeline performance. |
| 14 | • Pipeline hazards: structural, data, and control hazards, load delay.<br>• Memory organization, SRAM vs DRAM, DRAM refresh cycles, latency and bandwidth. |
| 15 | • Cache memory organization: direct-mapped, fully-associative, and set-associative caches, handling cache miss, write policy, and replacement policy.<br>• Cache performance, hit and miss rates, memory stall cycles, and average memory access time. |