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

Department of Computer Engineering

COE 202 Digital Logic Design (3-0-3)

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

Term: 161 (1st term 2016–2017)
Day & Time: UTR 10:00 AM – 10:50 AM
Location: 24-255
Prerequisite: PHYS 102
Textbook: *Introduction to Logic Design*, Alan B. Marcovitz, Third Edition, McGraw-Hill, 2010.
Office Hours: UTR 11:00 AM – 11:55 AM (or by appointment)
Web Site: http://faculty.kfupm.edu.sa/COE/marwan

Tentative Grading Policy:

- Homeworks **07%** (Each homework may carry a different weight)
- Verilog Assignments **07%** (Each Verilog assignment may carry a different weight)
- Quizzes **06%** (Each quiz may carry a different weight)
- Major Exam I **20%** (Saturday October 22, 2016, 1:00 PM)
- Major Exam II **25%** (Saturday November 26, 2016, 1:00 PM)
- Final Exam **35%** (Wednesday January 11, 2017, 7:00 PM)

Course Learning Outcomes

1. Ability to use math and Boolean algebra in performing computations in various number systems and simplification of Boolean algebraic expressions.
2. Ability to design efficient combinational and sequential logic circuit implementations from functional description of digital systems.
3. Ability to use CAD tools to simulate and verify logic circuits.

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). *Check your university e-mail regularly for warnings regarding your absences.*
- If you are late to the class for more than 5 minutes (i.e. arrive after 10:05 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**.
- Homeworks/Assignments are to be submitted **in class** on the due date during the class period. Late homeworks/assignments will **NOT be accepted**.
- You have up to the next class period to object to the grade of a homework/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 homeworks/assignments and quizzes will be counted towards your grade.
- Final exam is **comprehensive & common**.

Tentative schedule:

Week	Topics
1	• Introduction. • Weighted Number Systems: Decimal, Binary, Octal and Hexadecimal. • Number base conversion (Dec to Bin, Oct, and Hex, General).
2	• Arithmetic in Binary and Hex (addition, subtraction, multiplication). • BCD Codes: Excess-3 & other BCD codes, Character Storage, ASCII Code. Error Detection, Parity Bits. • Binary logic and gates, Truth tables, Boolean Algebra, Basic identities. Principle of duality, DeMorgan's Theorem. • Algebraic manipulation of Boolean expressions.
3	• Algebraic manipulation of Boolean expressions. • Canonical and Standard forms, Minterms, Maxterms, Sum of products & Products of Sums. • 2-Level gate implementation (SOP, POS). • From Truth tables to Boolean Expressions.
4	• Propagation delay. Timing diagrams. • Introduction to Verilog: Verilog Syntax, Definition of a Module, Gate Level Modeling, Using Modelsim simulation tool. Module Instantiation, Propagation Delay, Behavioral Modeling, Boolean Equation-Based Behavioral Models of Combinational Logic, Assign Statement, Propagation Delay & Continuous Assignment, Test Bench Example.
5-6	• K-Map method of simplification: 2, 3 and 4-variable maps. Implicants, Prime Implicants, Essential Prime Implicants. • POS simplification. • Don't care conditions and simplification. • Universal gates (NAND, NOR) • Implementation using NAND and NOR gates: 2-level & Multilevel implementation. • Exclusive-OR (XOR) and Equivalence (XNOR) gates, Odd and Even Functions, Parity generation and checking.
7-8	• Combinational Circuit Design Procedure & Examples. • Iterative combinational circuit design. • Half and Full Adders. • Ripple Carry Adder (RCA) design and Delay analysis of RCA. • Signed Numbers: sign-magnitude, 1's complement, and 2's complement. • Signed Binary Arithmetic. (Addition and Subtraction). • Binary Adder-Subtractor.
9-10	• Decoders 2x4, 3x8, 4x16. Designing large decoders from smaller decoders. Function implementation using decoders. • Encoders: Priority Encoders. • Multiplexers: 2x1, 4x1. Constructing large MUXs from smaller ones. • Function implementation using multiplexers. • MSI Design Examples.
11-13	• Introduction to Verilog: Verilog Operators, Behavioral Description of an Adder, Always block, Procedural Assignment, If Statements, Case Statements, Comparator, Arithmetic & Logic Unit. Multiplexer, Encoder, Priority Encoder, Decoder, Seven Segment Display Decoder. • Sequential Circuits: Latches, Clocked latches: SR and D. Flip-Flops: Master-Slave, D-FF. • Analysis of Sequential Circuits. State table, State diagram. • Mealy vs. Moore machine. • Sequential Circuit Design. Design procedure, State diagrams and state tables. • Asynchronous/Direct Clear and Set Inputs. Setup, Hold, FF propagation delay. Calculation of maximum clock frequency.
14-15	• Verilog modeling of D-Latch, D Flip Flop – Synchronous Set/Reset, D Flip Flop–Asynchronous Set/Reset. Verilog Structural modeling of sequential circuits, Verilog FSM modeling. • Registers, Registers with parallel load. • Shift Registers. Bi-directional shift register. Applications of shift registers. • Synchronous Binary Counters: Up-Down Counters • Counters with Parallel load, enable, synchronous clear and asynchronous clear. • Use of available counters to build counters of different count. • Verilog modeling of: Parallel Load Register, Shift Register, Up-Down Counter. • Combinational & Sequential Circuit Implementation with ROM. • Sequential Circuit Implementation using ROMs.