COE 203 Syllabus

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

Syllabus for COE 203: Digital Logic Laboratory (0-3-1)

General:

Course Title: Digital Logic Laboratory
Course Code: COE 203
Co-requisite(s): COE 202 (Digital Logic Design)

Course Description:

Introduction to information representation, Signals and bits, Logic implementation using discrete logic components (TTL, CMOS). Introduction to Field Programmable Gate Arrays (FPGAs) design flow: design capture (schematic capture), HDL (Verilog) design entry, design verification and test, implementation (including some of its practical aspects), and debugging. Students will use CAD software tools in the lab to design, simulate and implement digital logic circuits on FPGA prototyping board.

Instructors' Information

Name: Masud ul Hasan	**Name:** Saleh AlSaleh
Office: 22/407-1	**Office:** 23/010-4
Phone: 3880	**Phone:** 7035
Email: masud@kfupm.edu.sa	**Email:** salehs@kfupm.edu.sa
Office Hours: M. W. 10:00 – 10:50 AM	**Office Hours:** M. W. 10:00 – 10:50 AM

If office hours are not convenient to you, get an appointment through e-mail.

Grading Policy:

Each Experiment will be graded out of 100 & Project out of 200

75 % Practical Work + 25 % Lab Notebook & Discussion

Textbook(s) and/or other Required Material

Introduction to Logic Design, Alan Marcovitz, 2nd Edition, McGraw-Hill, 2005 and Lab Manual prepared by Computer Engineering Department faculty.

Course Objectives:

The purpose of this lab is to give the student practical experience with the process of design and implementation of digital circuits. It will basically expose the student to two main digital hardware implementation technologies; wire-interconnected discrete components (ICs) and field programmable gate arrays (FPGAs). This also includes getting a hands-on-experience with CAD tools for digital hardware development.

Upon completion of this course students are expected to be able to:

- Design an experiment to test a digital circuit or system and report the resulting implementation complexity (number of gates, wires, inputs, outputs …etc.) and speed (delay, maximum frequency of operation …etc.).
- Build (or simulate) digital circuit or system according to a specified functionality and report the resulting functionality (i.e. behavior of the circuit), complexity (number of gates, FFs, wires, I/Ps, O/Ps …etc.) and performance (delay or maximum frequency).
- Develop computer simulations to correlate or interpret experimental results.
- List and discuss several possible reasons for deviations between predicted and measured results from an experiment, choose the most likely reason and justify the choice, and formulate a method to validate the explanation.
- Keep a **lab notebook** documenting the designs, tests and debugging strategies, etc.

Weekly Breakdown of Lecture Course Material

Week	Topics
1	Lab Introduction, bread boards, FPGA boards, policies, overview of experiments, reporting, team-work, attendance, grading, etc.
2	**(Experiment 1)Introduction** • Boolean Logic and Gates. • Learn logic gates basics (Signal voltages for 0 & 1) • Logic operations: AND, OR, NAND, NOR and NOT and their truth tables. • Familiarize with IC's (IC Pins: inputs, outputs, Vcc & GND, applying inputs and monitoring outputs) • Verify basic gates operation • Using IC Tester and bread boards

3	**(Experiment 2)Integrated Circuits; Electrical Properties and Specifications** • Reading data sheets and extracting required parameters. • Understanding the significance of some major input and output electrical specifications. • Understanding the limitations of driving various loads. • Develop experiments to measure and verify some of these specs
4	**(Experiment 3)Digital Circuit Prototyping with FPGAs: Introduction** • Introduction to FPGAs & Capabilities • Synthesis Flow in FPGAs and Demos for the tools to be used, e.g. schematic capture &Simulation • Half adder experiment on FPGA board
5	**(Experiment 4)Hierarchical Design** • Get more familiar with FPGA and design tool • Learn about symbols and hierarchical design • Build Half-Adder and define it as a symbol • Use the Half-Adder symbols to build a Full-Adder, and define it as a symbol • Build a 4-bit adder using Full-Adder symbols • Model the Half-Adder, Full-Adder, and 4-bit adder using Verilog hardware description language (HDL)
6	**(Experiment 5)Combinational Logic Design** • Design a simple Arithmetic & Logic Unit (ALU) as an example of combinational logic • Use a multiplexor, an Adder, and simple logic gates to implement the Arithmetic Unit • Implement an ALU using Verilog behavioral description • Evaluate the impact of design decisions on the area of the resulting circuit
7	**(Experiment 6)Counters and Registers** • Learn about sequential circuits in general, and how to use counters and registers in particular. • Understand and demonstrate the effect of the bouncing problem on counters. • Use and demonstrate the effect of a Verilog debouncing circuit.
8	**(Experiment 7)Multiplier Design** • Search through the available modules in the XILINX library of elements. • Select the proper XILINX library components/elements of the proper sizes for the design of a simple 4 x 4 multiplier circuit. • Implement and verify the operation of the multiplier circuit. • Learn the use of buses and bus taps. • Learn the use of 7-Segment Display.

9	**(Experiment 8)Clock** • Learn about the clock signal and clock frequency. • Generate slower clock signals from a faster one. • Learn how to use oscilloscopes.
10	**(Experiment 9) Building a Digital Timer** • Learn about and design modulo counters. • Use cascaded modulo counters to build a digital timer. • Generate slower clock signals with exact frequencies.
11	**(Experiment 10)Reaction Timer Part 1 –Generating Random Delay** • Learn how to generate random numbers using a Linear Feedback Shift Register (LFSR). • Learn how to use counters to wait for specific amounts of time before performing an action.
12	**(Experiment 11)Reaction Timer Part 2–Response Time** • Learn how to design a saturating BCD counter. • Learn how to measure user response time. • Finalize the reaction timer experiment.
13	**(Experiment 12)Traffic Light Controller** • States and finite state machines (FSM) • Learn how to write a Verilog model for a given finite state machine • Design and implement a traffic light controller
14 & 15	**Mini Project→ To be decided every term**

Class/Laboratory Schedule

3 lab hours per week.

Computer Usage

Students will use CAD software tools from Xilinx in the lab to design, simulate and implement digital logic circuits on FPGA prototyping boards.

Course Contribution to Meet the Professional Component

This course provides a handling of the design process in computer engineering with a sound academic basis that is integrated with practical applications. It provides a solid understanding of the Design Process, Design Tools, and the right mix of Professional Skills that are critical for career success.

Additional Material Posted for Students on the Web

Description	Link
Lab experiments and supporting material	Available at Blackboard
Files required for debouncing	Available at Blackboard
Files required for using 7-Segment display of FPGA Boards	Available at Blackboard