JOINT INSTITUTE
交大密西根学院

Standard Undergraduate Course Profile

COURSE NUMBER: Ve270	**COURSE TITLE:** Introduction to Logic Design
CREDIT: 4	**PREREQUISITES:** Vg101 or equivalent
TEXTBOOKS/REQUIRED MATERIAL: Frank Vahid, Digital Design 2/e, John Wiley & Sons, 2010. ISBN 9780470531082	**PREPARED BY:** Gang Zheng **DATE OF PREPARATION:** June 10, 2012 **DATE OF DISCIPLINE GROUP APPROVAL:** **DATE OF UC APPROVAL:** Oct. 30, 2013
CATALOG DESCRIPTION: Binary and non-binary systems, Boolean algebra, digital design techniques, logic gates, logic minimization, standard combinational circuits, sequential circuits, flip-flops, synthesis of synchronous sequential circuits, PLAs, ROMs, RAMs, arithmetic circuits, computer-aided design. Laboratory includes design and CAD experiments.	**COURSE TOPICS:** 1. Introduction to logic design, number systems, basic logic gates, truth table 2. Boolean algebra, representation of Boolean functions 3. Combinational logic design process, building blocks, optimizations 4. Latches, Flip-Flops 5. Finite-state machines and controllers, FSM optimizations and tradeoffs 6. Hardware Description Language (Verilog HDL) 7. Programmable Logic Devices 8. Registers and shifters 9. Counters 10. Timing issues 11. Arithmetic components 12. Register-transfer level (RTL) design and examples 13. Memory Components
COURSE STRUCTURE/SCHEDULE: Two 90-minute lectures/week; one 150-minute lab session/week; four or more discussion/recitation sessions	
COURSE OUTCOMES [Student Outcomes in brackets]	After completing Ve270, students should have: 1. Ability to perform simple arithmetic in binary, octal, hexadecimal, BCD number systems 2. Ability to manipulate logic expressions using binary Boolean algebra. 3. Ability to generate the prime implicants of logic functions of 5 or fewer variables using graphical (Karnaugh map) method, and to obtain their minimal two-level implementations with and without don't cares. [1] 4. Ability to analyze and synthesize small multi-level combinational logic circuits containing AND, OR, NOT, NAND, NOR, and XOR gates based on simple delay models. [1] 5. Ability to use basic functional & timing (clocking) properties of latches & flip-flops. 6. Ability to analyze synchronous sequential circuits to extract next-state/output functions [1] 7. Ability to translate a word statement specifying the desired behavior of a simple sequential system into a finite state machine (FSM), to simplify and build the architecture that consists of state register and next state/output logic. [1, 2, 7] 8. An ability to implement simple digital systems using controller and basic datapath components such as registers, memories, counters, multiplexers, ALUs, etc. [1, 2, 7] 9. Basic knowledge of possible issues and restrictions in digital system. [1, 7] 10. Ability to design and test simple digital systems using a hardware description language and CAD tools [1, 6] 11. Knowledge of programmable logic devices and ability of implementing a logic circuit in FPGA. [6, 7] 12. Experience and communication skills to function on a team. [5]
COURSE OBJECTIVES [Course Outcomes in brackets]	1. To teach the fundamental principles in design and implementation of digital logic circuits including combinational circuits, sequential circuits, and finite state machines. [1,2,3,4,5,6,7] 2. To develop skills in top-down design and bottom-up verification for digital components and systems. [8,9,10] 3. To provide hands-on experience with computer aided design tools and programmable logic devices in digital logic design. [10,11] 4. To improve communication skills to effectively function on a team. [12]
ASSESSMENT TOOLS [Course Outcomes in brackets]	Homework [1,2,3,4,5,6,7,8,9] Midterm Exam [1,2,3,4,5,6] Final Exam [6,7,8, 9] Lab experiment [3,4,5,6,7,8,9,10,11,12]

Rev 1: June 2015; Rev 2: July 2020