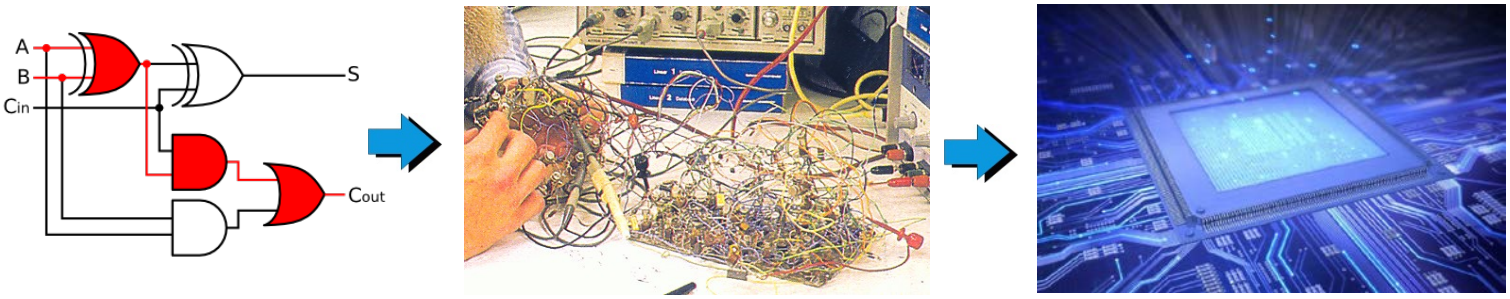
Summer 2015 – Online Course Offering

EELE 261 – Introduction to Logic Circuits (4cr)

Montana State University - Bozeman

Instructor: Dr. Brock J. LaMeres



Do you want to gain 4-credits toward your EE/CpE degree while you're off-campus this summer? Then this fully online course is for you!

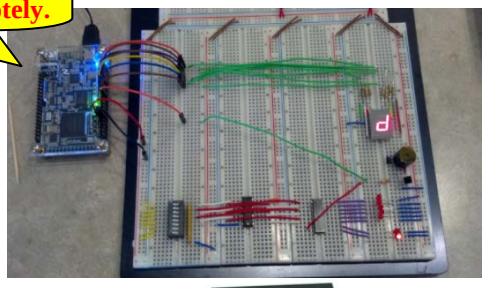
Course Description

This course introduces the concepts of classical digital logic design including number systems, interfacing, Boolean algebra, combinational logic design, and finite state machines. This course also covers Hardware Description Languages for the structural design and simulation of digital systems. Modern digital design of combinational logic and state machines is covered using VHDL and a logic synthesizer. This course contains a laboratory experience where students design and implement logic circuits using discreet parts and programmable logic devices.

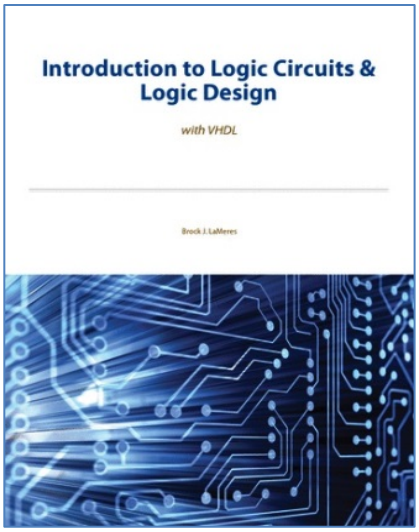
Course Logistics

Prerequisites:	College algebra.
Dates:	12-week summer session, May 18th to August 7th.
Lecture (3cr):	The course will follow the textbook exactly. There will be short lecture videos provided for each section. Homework and exams will be in the form of online multiple choice quizzes or logic simulations that will be uploaded to the course DropBox.
Lab (1cr):	A portable lab kit will be checked out that will allow all of the lab experiments to be performed remotely. Lab demonstrations will be conducted using Google+ Hangout video conferencing. Students will use electronic lab notebooks.
Requirements:	The textbook (available from the MSU bookstore) and a computer with video conferencing capability and that is fast enough to run the Altera Quartus II / ModelSim software tools.

A lab kit will be provided to accomplish all experiments remotely.



The *Analog Discovery* USB oscilloscope will be used to take measurements.



Textbook

Introduction to Logic Circuits & Logic Design with VHDL
by Brock J. LaMeres
John Wiley & Sons Publishing (2015)

Note: This book was written specifically for this course.

Student testimonial about the book:

I like having errors in the book. It makes reading like a treasure hunt and I can focus better when I read.

- student in spring 2015 offering of EELE 261

Course Outline

MODULE 1: INTRODUCTION – ANALOG VS. DIGITAL

MODULE 2: NUMBER SYSTEMS

2.1 Positional Number Systems
2.2 Base Conversion
2.3 Binary Arithmetic
2.4 Unsigned and Signed Numbers

MODULE 3: DIGITAL CIRCUITRY & INTERFACING

3.1 Basic Gates
3.2 Digital Circuit Operation
3.3 Logic Families
3.4 Driving Loads

MODULE 4: COMBINATIONAL LOGIC DESIGN

4.1 Boolean Algebra
4.2 Combinational Logic Analysis
4.3 Combinational Logic Synthesis
4.4 Logic Minimization
4.5 Timing Hazards & Glitches

MODULE 5: VHDL (PART 1)

5.1 History of Hardware Description Languages
5.2 HDL Abstraction
5.3 The Digital Design Flow
5.4 VHDL Constructs
5.5 Modeling Concurrent Functionality in VHDL
5.6 Structural Design Using Components
5.7 Overview of Simulation Test Benches

MODULE 6: MSI LOGIC

6.1 Decoders
6.2 Encoders
6.3 Multiplexers
6.4 Demultiplexers

MODULE 7: SEQUENTIAL LOGIC DESIGN

7.1 Sequential Logic Storage Devices
7.2 Sequential Logic Timing Considerations
7.3 Common Circuits Based on Sequential Storage Devices
7.4 Finite State Machines
7.5 Counters
7.6 Finite State Machine's Reset Condition
7.7 Sequential Logic Analysis

Grading

Homework	30%
Lab Experiments	30%
Module Quizzes	20%
Comprehensive Final Exam	20%

Learning Outcomes (what a student will be able to do after taking this course)

1.1	Describe the fundamental differences between analog and digital systems.
1.2	Describe advantages of digital systems compared to analog systems.
1.3	Identify digital systems in use today.
1.4	Identify analog systems in use today.

2.1	Convert unsigned binary numbers to decimal.
2.2	Convert unsigned octal numbers to decimal.
2.3	Convert unsigned hexadecimal numbers to decimal.
2.4	Convert unsigned decimal numbers to binary.
2.5	Convert unsigned decimal numbers to octal.
2.6	Convert unsigned decimal numbers to hexadecimal.
2.7	Convert unsigned binary numbers to octal.
2.8	Convert unsigned binary numbers to hexadecimal.
2.9	Convert unsigned octal numbers to binary.
2.10	Convert unsigned hexadecimal numbers to binary.
2.11	Convert unsigned octal numbers to hexadecimal.
2.12	Convert unsigned hexadecimal numbers to octal.
2.13	Perform unsigned binary addition by hand.
2.14	Perform unsigned binary subtraction by hand.
2.15	Determine the range of an n-bit two's complement number.
2.16	Convert a signed decimal number to binary using two's complement representation.
2.17	Convert a signed two's complement binary number into decimal.
2.18	Perform signed binary addition by hand using two's complement representation.

3.1	Describe the functional operation of a basic logic gate using truth tables, logic expressions, and logic waveforms.
3.2	Determine the noise margins for a logic family based on the DC operating conditions provided by a data sheet.
3.3	Determine the power supply currents of a driver when sourcing and/or sinking multiple loads.
3.4	Extract key switching characteristics from a logic gate's data sheet.
3.5	Draw the transistor-level schematic for basic CMOS gates.
3.6	Determine the output current of a driver when driving multiple gates from the same logic family.
3.7	Determine the output current of a driver when driving resistive loads.
3.8	Determine the output current of a driver when driving LED loads.
3.9	Implement basic logic gates using discrete parts on a breadboard.
3.10	Measure the DC operating conditions (current & voltage) of a digital circuit.
3.11	Measure the AC characteristics (delay and transition time) of a digital circuit.

4.1	Describe the fundamental principles of Boolean algebra.
4.2	Prove Boolean algebra theorems using *proof by exhaustion.*
4.3	Determine the logic expression for a combinational logic circuit from its logic diagram.
4.4	Determine the truth table for a combinational logic circuit from its logic diagram.
4.5	Determine the delay of a combinational logic circuit from its logic diagram.
4.6	Describe the functional operation of a logic circuit using a truth table.
4.7	Describe the functional operation of a logic circuit using a canonical sum of products (SOP) form.
4.8	Describe the functional operation of a logic circuit using a minterm list.
4.9	Describe the functional operation of a logic circuit using a canonical product of sums (POS) form.
4.10	Describe the functional operation of a logic circuit using a maxterm list.
4.11	Minimize a logic expression using a Kaurnaugh Map.
4.12	Find the minimal logic expression from a Kaurnaugh Map.
4.13	Eliminate timing hazards in a minimal logic expression.
4.14	Exploit "don't cares" to form a minimal logic expression.
4.15	Implement simple combinational logic circuits using discrete parts on a breadboard.
4.16	Verify the functionality of simple combinational logic circuits through laboratory experiment.

5.1	Describe the role of hardware description languages in modern digital design.
5.2	Describe the fundamentals of design abstraction in modern digital design.
5.3	Describe the modern digital design flow based on hardware description languages.
5.4	Describe the fundamental constructs of VHDL.
5.5	Design a VHDL model for a combinational logic circuit using concurrent signal assignments and logical operators.

5.6	Design a VHDL model for a combinational logic circuit using conditional signal assignments.
5.7	Design a VHDL model for a combinational logic circuit using selected signal assignments.
5.8	Design a VHDL model for a combinational logic circuit using a structural design approach.
5.9	Perform logic simulations to verify the accuracy of a VHDL model.

6.1	Design a decoder circuit by hand using the classical digital design approach.
6.2	Design a VHDL model of a decoder circuit using concurrent behavior modeling techniques.
6.3	Design an encoder circuit by hand using the classical digital design approach.
6.4	Design a VHDL model of an encoder circuit using concurrent behavior modeling techniques.
6.5	Design a multiplexer circuit by hand using the classical digital design approach.
6.6	Design a VHDL model of a multiplexer circuit using concurrent behavior modeling techniques.
6.7	Design a demultiplexer circuit by hand using the classical digital design approach.
6.8	Design a VHDL model of a demultiplexer circuit using concurrent behavior modeling techniques.
6.9	Implement MSI combinational logic circuits using discrete parts on a breadboard.
6.10	Verify the functionality of MSI combinational logic circuits through laboratory experiment.
6.11	Implement MSI combinational logic circuits using an HDL and a programmable logic device.
6.12	Interface a programmable logic device to discrete parts on a breadboard.

7.1	Describe how the state encoding approach selected in a finite state machine correlates to the number of storage devices required in the circuit.
7.2	Describe the difference between a Mealy and Moore finite state machine architecture.
7.3	Design a finite state machine using the classical digital design approach from a state diagram.
7.4	Design a finite state machine using the classical digital design approach from a word description.
7.5	Design a VHDL model of a finite state machine using concurrent behavior modeling techniques for both the next state and output logic and a structural modeling technique for the state memory where the storage device is provided as a black box and instantiated as a component.
7.6	Design a binary ripple counter using only D-Flip-Flops.
7.7	Design a counter that is implemented as a finite state machine using the classical digital design approach.
7.8	Design a VHDL model of a counter using concurrent behavior modeling techniques for the next state and output logic and a structural modeling technique for the state memory where the storage device is provided as a black box and instantiated as a component.
7.9	Analyze the logic diagram of a FSM to find its next state logic and output logic expressions.
7.10	Analyze the logic diagram of a FSM to find its state transition table.
7.11	Analyze the logic diagram of a FSM to find its state diagram.
7.12	Analyze the logic diagram of a FSM to find its maximum clock frequency.
7.13	Analyze the logic diagram of a FSM to find its maximum clock frequency.
7.14	Implement simple finite state machines using discrete parts on a breadboard.
7.15	Verify the functionality of simple finite state machines through laboratory experiment.
7.16	Implement simple finite state machines using an HDL with concurrent modeling techniques and a programmable logic device.